

TUNCER CEBECI · JIAN P. SHAO
FASSI KAFYEKE · ERIC LAURENDEAU

Computational Fluid Dynamics for Engineers



T. Cebeci J. P. Shao
F. Kafyeke E. Laurendeau

 *Computational Fluid Dynamics for Engineers*

**HORIZONS
PUBLISHING**

*Long Beach, California
Heidelberg, Germany*

Tuncer Cebeci Jian P. Shao
Fassi Kafyeke Eric Laurendeau

Computational Fluid Dynamics for Engineers

From Panel to Navier–Stokes Methods
with Computer Programs

With 152 Figures, 19 Tables, 84 Problems
and a CD-ROM



Tuncer Cebeci

The Boeing Company
Long Beach, CA 90807-5309, USA
and
810 Rancho Drive
Long Beach, CA 90815, USA
TuncerC@aol.com

Jian P. Shao

The Boeing Company
Huntington Beach, CA 92647, USA
jian.p.shao@boeing.com

Fassi Kafyeke

Advanced Aerodynamics Department
Bombardier Aerospace
400 Côte Vertu Road West
Dorval, Québec, Canada H4S 1Y9
fassi.kafyeke@aero.bombardier.com

Eric Laurendeau

Advanced Aerodynamics Department
Bombardier Aerospace
400 Côte Vertu Road West
Dorval, Québec, Canada H4S 1Y9
eric.laurendeau@aero.bombardier.com

ISBN 0-9766545-0-4 Horizons Publishing Inc., Long Beach
ISBN 3-540-24451-4 Springer Berlin Heidelberg New York

Library of Congress Control Number: 2005923905

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Horizons Publishing Inc., 810 Rancho Drive, Long Beach, CA 90815, USA) except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

© Horizons Publishing Inc., Long Beach, California 2005
Printed in Germany

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Please note: All rights pertaining to the Computer Programs are owned exclusively by the authors and Horizons Publishing Inc. The publisher and the authors accept no legal responsibility for any damage caused by improper use of the programs. Although the programs have been tested with extreme care, errors cannot be excluded.

Typeset in MS Word by the authors. Edited and reformatted by Kurt Mattes, Heidelberg, Germany, using LATEX.

Printing and binding: Strauss GmbH, Mörlenbach, Germany
Cover design: Erich Kirchner, Heidelberg, Germany

Printed on acid-free paper 543210

Preface

History reminds us of ancient examples of fluid dynamics applications such as the Roman baths and aqueducts that fulfilled the requirements of the engineers who built them; of ships of various types with adequate hull designs, and of wind energy systems, built long before the subject of fluid mechanics was formalized by Reynolds, Newton, Euler, Navier, Stokes, Prandtl and others. The twentieth century has witnessed many more examples of applications of fluid dynamics for the use of humanity, all designed without the use of electronic computers. They include prime movers such as internal-combustion engines, gas and steam turbines, flight vehicles, and environmental systems for pollution control and ventilation.

Computational Fluid Dynamics (CFD) deals with the numerical analysis of these phenomena. Despite impressive progress in recent years, CFD remains an imperfect tool in the comparatively mature discipline of fluid dynamics, partly because electronic digital computers have been in widespread use for less than thirty years. The Navier–Stokes equations, which govern the motion of a Newtonian viscous fluid were formulated well over a century ago. The most straightforward method of attacking any fluid dynamics problem is to solve these equations for the appropriate boundary conditions. Analytical solutions are few and trivial and, even with today's supercomputers, numerically exact solution of the complete equations for the three-dimensional, time-dependent motion of turbulent flow is prohibitively expensive except for basic research studies in simple configurations at low Reynolds numbers. Therefore, the “straightforward” approach is still impracticable for engineering purposes.

Considering the successes of the pre-computer age, one might ask whether it is necessary to gain a greater understanding of fluid dynamics and develop new computational techniques, with their associated effort and cost. Textbooks on fluid dynamics reveal two approaches to understanding fluid dynamics processes. The first is to devise useful correlations through a progression from demonstrative experiments to detailed experimental investigations that yield additional

understanding and subsequent improvement of the processes in question. The second is to solve simplified versions of fluid dynamics equations for conservation of mass, momentum and energy for comparatively simple boundary conditions. There is great advantage in combining both approaches when addressing complex fluid dynamics problems, but interaction between these two approaches has been limited until recently by the narrow range of useful solutions that could be obtained by analytic methods or simple numerical computations. It is evident, therefore, that any method for increasing the accuracy of computational methods by solving more complete forms of the conservation equations than has been possible up to now is to be welcomed. The numerical approaches of CFD have, in most cases, proven much more powerful than the closed-form analytical solutions of the past. As an example, the flow through the blade passage of a gas turbine is three-dimensional, and, even if we ignore the problem of modeling the behavior of turbulence, the corresponding equations can only be solved by numerical methods; even the inviscid flow in an axisymmetric engine intake cannot be calculated by purely analytic methods. Thus, without computational fluid dynamics, we cannot calculate detailed flow characteristics essential to improving understanding and supporting the design process.

It should be recognized that both experimental and computational fluid dynamics require resources. The cost of experiments in some cases can be prohibitive as, for example, with extensive flight tests of airplanes, full-scale tests of a gas turbine, or destructive testing of expensive components. In such cases, it may be possible to reduce the number of experimental tests by using CFD, since only a relatively small number of experiments are required to check the accuracy of the numerical results. Of course, the cost of obtaining accurate numerical solutions of differential equations may also be large for a complex flow, but still are usually much less than the cost of the additional experiments that would otherwise be required. In reality, the most cost-effective approach to solving a fluid dynamics problem is likely to be a combination of measurements and calculations. Both are subject to uncertainties, but the combination of these two approaches can result in a more cost-effective and more reliable design than by using only one approach or the other, and thus may be necessary to meet today's more stringent requirements for improved performance and reduced environmental impact, along with technical innovation and economy.

This book is an introduction to computational fluid dynamics with emphasis on the solution of conservation equations for incompressible and compressible flows with two independent variables. From the range of formulations in CFD, such as finite-difference, finite volume, finite element, spectral methods and direct numerical simulation, it concentrates on the first two, which are widely used to solve engineering problems. The restriction to two-dimensional flow and the omission of finite element, spectral methods and direct numerical simulation are imposed to facilitate understanding and to allow the essential material to be

presented in a book of modest size. The discussions, however, are general in this introductory book and apply to a variety of flows, including three-dimensional flows.

The format of the book assures that essential topics are covered in a logical sequence. The Introduction of Chapter 1 presents some examples to demonstrate the use of computational fluid dynamics for solving engineering problems of relevance. Chapter 2 presents the conservation equations; it is comparatively brief since detailed derivations are available elsewhere. The third chapter introduces important properties of turbulent flows, and exact and modeled forms of the turbulence equations with explanations to justify the assumptions of the models.

Chapters 4 and 5 provide an introduction to the numerical methods for solving the model equations for conservation equations which are useful for modeling the behavior of the more complete and complicated parabolic, hyperbolic and elliptic partial-differential equations considered in subsequent chapters. Chapter 4 discusses the numerical methods for the model parabolic and elliptic equations and Chapter 5 the model hyperbolic equations and include many computer programs.

The calculation of solutions for inviscid and boundary-layer equations is addressed in Chapters 6 and 7. Chapter 6 discusses finite-difference and panel methods for solving the Laplace equation and include computer programs for single and multi-element airfoils. Chapter 7 discusses the solution of laminar and turbulent boundary-layer equations for a prescribed external velocity distribution and specified transition location and includes a computer program based on Keller's finite-difference method.

The prediction of the onset of transition from laminar to turbulent flow has traditionally been achieved by correlations which are known to have limited ranges of applicability. The use of the e^n -method, based on the solutions of the stability equations, has been proposed as a more general approach. Chapter 8 describes the solution of the stability equations and provides a computer program for solving the Orr–Sommerfeld equation and computing transition with the e^n -method. It also presents applications of the stability/transition program, together with the computer programs of Chapters 6 and 7, to demonstrate how problems of direct relevance to engineering can be addressed by this approach.

Chapter 9 presents grid generation methods and is followed by Chapters 10 to 12 which describe methods for solving Euler (Chapter 10), incompressible Navier–Stokes (Chapter 11) and compressible Navier–Stokes equations. Again computer programs are included in each chapter and summarized in Appendix B.

A one semester course for advanced undergraduate and first-year graduate students would include a brief reading of Chapter 1 followed by Chapters 2, 4, 5 and 10 which include an extensive number of example problems and associated

computer programs arranged to provide the student a better understanding of the computational tools discussed. Parts of the material in Chapters 3, 6, 7 to 9 and 11 and 12 can be covered in a second semester course, with parts of the material in those chapters serving as useful information/reference.

A list of related and current books and solution manuals, including the one for the present book, published by Horizons and Springer-Verlag, is available on the Horizons Web site,

<http://hometown.aol.com/tuncerc/>

The authors would like to express their appreciation to several people who have given thoughts and time to the development of this book. The first and second authors in particular want to thank Herb Keller of the California Institute of Technology, Jim Whitelaw of Imperial College, and Hsun Chen of the California State University, Long Beach. They also want to thank K. C. Chang for proof reading the manuscript and making many useful suggestions. The third and fourth authors like to thank Bombardier Aerospace for supplying some of the applications cited in the text. Thanks are also due to Kurt Mattes for his excellent typing and Karl Koch for the production of the book.

Finally we would like to thank our wives, Sylvia Cebeci, Jennifer Shaw, Nathalie David and Solange Lusinde, and our children for their understanding and the hours they relinquished to us. Their continuous support and encouragement are greatly appreciated.

Long Beach, April 2005

Tuncer Cebeci

Jian P. Shao

Fassi Kafyeke

Eric Laurendeau

Contents

1. Introduction	1
1.1 Skin-Friction Drag Reduction	2
1.1.1 Laminar Flow Control	3
1.1.2 Calculations for NLF and HLFC Wings	6
1.2 Prediction of the Maximum Lift Coefficient of Multielement Wings	10
1.3 Aircraft Design and Power Plant Integration	19
1.4 Prediction of Aircraft Performance Degradation Due to Icing	23
1.4.1 Prediction of Ice Shapes	26
1.4.2 Prediction of Aerodynamic Performance Characteristics	28
1.5 Aerodynamics of Ground-Based Vehicles	34
1.5.1 Applications of CFD to Automobiles	36
References	39
 2. Conservation Equations	41
2.1 Introduction	41
2.2 Navier–Stokes Equations	42
2.2.1 Navier–Stokes Equations: Differential Form	42
2.2.2 Navier–Stokes Equations: Integral Form	48
2.2.3 Navier–Stokes Equations: Vector-Variable Form	50
2.2.4 Navier–Stokes Equations: Transformed Form	51
2.3 Reynolds-Averaged Navier–Stokes Equations	55
2.4 Reduced Forms of the Navier–Stokes Equations	57
2.4.1 Inviscid Flow	60
2.4.2 Stokes Flow	62
2.4.3 Boundary Layers	62
2.5 Stability Equations	64
2.6 Classification of Conservation Equations	67

2.7	Boundary Conditions	70
References		72
Problems		73
3.	Turbulence Models	81
3.1	Introduction	81
3.2	Zero-Equation Models	83
3.2.1	Cebeci-Smith Model	83
3.2.2	Baldwin-Lomax Model	85
3.3	One-Equation Models	87
3.4	Two-Equation Models	88
3.5	Initial Conditions	90
References		93
4.	Numerical Methods for Model Parabolic and Elliptic Equations	95
4.1	Introduction	95
4.2	Model Equations	96
4.3	Discretization of Derivatives with Finite Differences	98
4.4	Finite-Difference Methods for Parabolic Equations	100
4.4.1	Explicit Methods	100
4.4.2	Implicit Methods: Crank–Nicolson	105
4.4.3	An Implicit Method: Keller’s Box Method	109
4.5	Finite-Difference Methods for Elliptic Equations	113
4.5.1	Direct Methods	115
4.5.2	Iterative Methods	121
4.5.3	Multigrid Method	127
References		132
Problems		132
5.	Numerical Methods for Model Hyperbolic Equations	141
5.1	Introduction	141
5.2	Explicit Methods: Two-Step Lax–Wendroff Method	146
5.3	Explicit Methods: MacCormack Method	148
5.4	Implicit Methods	149
5.5	Upwind Methods	152
5.6	Finite-Volume Methods	157
5.7	Convergence and Stability	165
5.8	Numerical Dissipation and Dispersion: Artificial Viscosity	170
References		173
Problems		174

6. Inviscid Flow Equations for Incompressible Flows	179
6.1 Introduction	179
6.2 Laplace Equation and Its Fundamental Solutions	179
6.3 Finite-Difference Method	182
6.4 Hess-Smith Panel Method	189
6.5 A Panel Program for Airfoils	194
6.5.1 MAIN Program	195
6.5.2 Subroutine COEF	196
6.5.3 Subroutine GAUSS	196
6.5.4 Subroutine VPDIS	196
6.5.5 Subroutine CLCM	196
6.6 Applications of the Panel Method	197
6.6.1 Flowfield and Section Characteristics of a NACA 0012 Airfoil	197
6.6.2 Flow Over a Circular Cylinder	198
6.6.3 Multielement Airfoils	201
Appendix 6A Finite Difference Program for a Circular Cylinder	202
Appendix 6B Panel Program for an Airfoil	203
6B.1 MAIN Program	203
6B.2 Subroutine COEF	203
6B.3 Subroutine VPDIS	203
Appendix 6C Panel Program for Multielement Airfoils	203
6C.1 MAIN Program	203
6C.2 Subroutine COEF	204
6C.3 Subroutine VPDIS	204
6C.4 Subroutine CLCM	204
References	204
Problems	204
7. Boundary-Layer Equations	211
7.1 Introduction	211
7.2 Standard, Inverse and Interaction Problems	212
7.3 Numerical Method for the Standard Problem	216
7.3.1 Numerical Formulation	218
7.3.2 Newton's Method	220
7.4 Computer Program BLP	222
7.4.1 MAIN	222
7.4.2 Subroutine INPUT	222
7.4.3 Subroutine IVPL	225
7.4.4 Subroutine GROWTH	225
7.4.5 Subroutine COEF3	226
7.4.6 Subroutine SOLV3	226

7.4.7	Subroutine OUTPUT	229
7.4.8	Subroutine EDDY	229
7.5	Applications of BLP	229
7.5.1	Similar Laminar Flows	230
7.5.2	Nonsimilar Flows	231
References		237
Problems		237
8.	Stability and Transition	243
8.1	Introduction	243
8.2	Solution of the Orr-Sommerfeld Equation	246
8.2.1	Numerical Formulation	247
8.2.2	Eigenvalue Procedure	249
8.3	e^n -Method	253
8.4	Computer Program STP	256
8.4.1	MAIN	257
8.4.2	Subroutine VELPRO	257
8.4.3	Subroutine CSAVE	258
8.4.4	Subroutine NEWTON	258
8.4.5	Subroutine NEWTONI	258
8.5	Applications of STP	259
8.5.1	Stability Diagrams for Blasius Flow	259
8.5.2	Transition Prediction for Flat Plate Flow	259
8.5.3	Transition Prediction for Airfoil Flow	261
References		261
Problems		262
9.	Grid Generation	263
9.1	Introduction	263
9.2	Basic Concepts in Grid Generation and Mapping	264
9.3	Stretched Grids	267
9.4	Algebraic Methods	268
9.4.1	Algebraic Grid Generation Using Transfinite Interpolation	271
9.5	Differential Equation Methods	277
9.6	Conformal Mapping Methods	282
9.6.1	Parabolic Mapping Function	283
9.6.2	Wind Tunnel Mapping Function	285
9.7	Unstructured Grids	288
9.7.1	Delaunay Triangulation	289
9.7.2	Advancing Front Method	292
References		293

10. Inviscid Compressible Flow	295
10.1 Introduction	295
10.2 Shock Jump Relations	296
10.3 Shock Capturing	299
10.4 The Transonic Small Disturbance (TSD) Equation	301
10.5 Model Problem for the Transonic Small Disturbance Equation: Flow Over a Non-Lifting Airfoil	302
10.5.1 Discretized Equation	303
10.5.2 Solution Procedure and Sample Calculations	304
10.6 Solution of Full-Potential Equation	308
10.7 Boundary Conditions for the Euler Equations	309
10.8 Stability Analysis of the Euler Equations	311
10.9 MacCormack Method for Compressible Euler Equations	312
10.10 Model Problem for the MacCormack Method: Unsteady Shock Tube	313
10.10.1 Initial Conditions	314
10.10.2 Boundary Conditions	314
10.10.3 Solution Procedure and Sample Calculations	314
10.11 Model Problem for the MacCormack Method: Quasi 1-D Nozzle	315
10.11.1 Initial Conditions	316
10.11.2 Boundary Conditions	317
10.11.3 Solution Procedure and Sample Calculations	318
10.12 Beam-Warming Method for Compressible Euler Equations	320
10.13 Model Problem for the Implicit Method: Unsteady Shock Tube	321
10.13.1 Solution Procedure and Sample Calculations	321
10.14 Model Problem for the Implicit Method: Quasi-1D Nozzle	322
10.14.1 Solution Procedure and Sample Calculations	325
References	326
Problems	326
 11. Incompressible Navier–Stokes Equations	327
11.1 Introduction	327
11.2 Analysis of the Incompressible Navier–Stokes Equations	328
11.3 Boundary Conditions	329
11.4 Artificial Compressibility Method: INS2D	331
11.4.1 Discretization of the Artificial Time Derivatives	331
11.4.2 Discretization of the Convective Fluxes	332
11.4.3 Discretization of the Viscous Fluxes	334
11.4.4 System of Discretized Equation	335
11.5 Model Problem: Sudden Expansion Laminar Duct Flow	336
11.5.1 Discretization of the Boundary Conditions	337

11.5.2	Initial Conditions	338
11.5.3	Solution Procedure and Sample Calculations	339
11.6	Model Problem: Laminar and Turbulent Flat Plate Flow	342
11.7	Applications of INS2D	344
	References	350
	Problems	351
12.	Compressible Navier–Stokes Equations	353
12.1	Introduction	353
12.2	Compressible Navier–Stokes Equations	354
12.2.1	Practical Difficulties	354
12.2.2	Boundary Conditions	355
12.3	MacCormack Method	356
12.4	Beam–Warming Method	357
12.5	Finite Volume Method	361
12.6	Model Problem: Sudden Expansion Laminar Duct Flow	365
12.6.1	Initial Conditions	365
12.6.2	Boundary Conditions	365
12.6.3	Solution Procedure and Sample Calculations	367
Appendix 12A	Jacobian Matrices of Convection and Diffusion Terms E , F , E_v and F_v	367
Appendix 12B	Treatment of the Region Close to the Boundaries for Eq. (12.5.4)	370
	References	374
	Problems	375
Appendix A	Computer Programs on the Accompanying CD-ROM	377
Appendix B	Computer Programs Available from the First Author	381
Subject Index		391

1

Introduction

In this chapter we present five examples to demonstrate the application of CFD techniques to solve real engineering problems. These examples are taken from the literature and encompass flows which make use of solutions of inviscid, boundary-layer and Navier–Stokes equations. For some of these flows, the reduced forms of the conservation equations, such as inviscid and boundary-layer equations are more appropriate, and for others more general equations are needed. In this way, the scope of this book is clarified further with additional terminology and fluid-dynamics information.

The first example, discussed in Section 1.1, addresses the application of CFD to reduce the drag of a wing by adjustment of pressure gradient by shaping and by suction through slotted or perforated surfaces. The drag of an aircraft can be reduced in a number of ways to provide increased range, increased speed, decreased size and cost, and decreased fuel usage. The adjustment of pressure gradient by shaping and using laminar boundary-layer control with suction are two powerful and effective ways to reduce drag. This is demonstrated with a calculation method for natural laminar flow (NLF) and hybrid laminar flow control (HLFC) wings.

The second example, discussed in Section 1.2, addresses the calculation of the maximum lift coefficient of a wing which corresponds to the stall speed, which is the minimum speed at which level flight can be maintained. A calculation method is described and used to predict the maximum lift coefficient of a high-lift system; this coefficient plays a crucial role in the takeoff and landing of an aircraft.

Aircraft design was traditionally based on theoretical aerodynamics and wind tunnel testing, with flight-testing used for final validation. CFD emerged in the late 1960's. Its role in aircraft design increased steadily as speed and memory of computers increased. Today CFD is a principal aerodynamic technology for aircraft configuration development, along with wind tunnel testing and flight-testing. State-of-the-art capabilities in each of these technologies are needed to achieve superior performance with reduced risk and low cost.

The third example, discussed in Section 1.3, deals with aircraft design and power plant integration.

The fourth example, discussed in Section 1.4, corresponds to a calculation method for predicting the performance degradation of an aircraft due to icing. A NACA icing research aircraft is chosen to compare the calculated results with measurements. The calculations are first performed by computing the ice shapes that form on the leading edges of the lifting surfaces of the aircraft and are followed by flowfield calculations to predict the loss in lift and increase in drag due to ice.

The fifth example, discussed in Section 1.5 is the application of CFD to ground-based vehicles, in particular to automobile aerodynamics development. The use of CFD in this area has been continuously increasing because the aerodynamic characteristics have a significant influence on the driving stability and fuel consumption on a highway. Since the aerodynamic characteristics of automobiles are closely coupled with their styling, it is impossible to improve them much once styling is fixed. Therefore, it is necessary to consider aerodynamics in the early design stage.

CFD also finds applications in internal flows and has been used to solve real engineering problems such as subsonic, transonic and supersonic inlets, compressors and turbines, as well as combustion chambers and rocket engines. These applications are, however, beyond the scope of this book and the reader is referred to the extensive literature available on these problems.

1.1 Skin-Friction Drag Reduction

There are several techniques for reducing the skin-friction drag of bodies. While the emphasis in this section is on aircraft components, the arguments apply equally to the reduction of skin-friction drag on all forms of transportation, including underwater vehicles. The importance of the subject has been discussed in a number of articles; a book edited by Bushnell and Heiner [1] summarizes the research in this area and the reader is referred to this book for an in-depth review of viscous drag reduction and for discussions of the possible savings which can occur from the reduction of the drag. As an example of the argument in support of the importance of the calculation methods used for reducing skin-friction drag, it is useful to point out that a three-percent reduction in the skin-friction drag of a typical long-range commercial transport, which burns around ten million gallons of fuel per year, at 50 cents per gallon, would yield yearly savings of around \$ 150,000.

There have been many suggestions for reducing the skin-friction drag on aircraft components including extension of regions of laminar flow, relaminarization of turbulent flow and modification to the turbulence characteristics of the near-wall flow. In general, these attempts to control the flow depend on changes

to the wall boundary conditions including variations of longitudinal and transverse surface curvatures, the nature of the surface and heat and mass transfer through the surface. A partial exception is the use of thin airfoils (LEBUs) in the outer region of the boundary layer to break up the large eddy structure of turbulent flow [1].

In this section the discussion is limited to laminar flow control (LFC) and the reader is referred to [1] for a discussion of other techniques for reducing the skin-friction drag. In subsection 1.1.1, a brief description of laminar flow control first by “Adjustment of Pressure Gradient by Shaping,” then by “Suction Through Slotted or Perforated Surfaces” is given. This subsection is followed by a description and application of a calculation method to natural laminar flow (NLF) and hybrid laminar flow control (HLFC) wings (Subsection 1.1.2).

1.1.1 Laminar Flow Control

Adjustment of Pressure Gradient by Shaping

Laminar flow on a two-dimensional or axisymmetric body can be achieved by designing the geometry so that there are extensive regions of favorable pressure gradients. This technique is frequently referred to as natural laminar flow (NLF) control and may be implemented on a wing or a body of revolution by bringing the point of maximum thickness as far aft as possible. Typical airfoil sections designed for this purpose are shown on Fig. 1.1 and the location of the onset of transition, where laminar flow becomes turbulent flow, can be estimated by

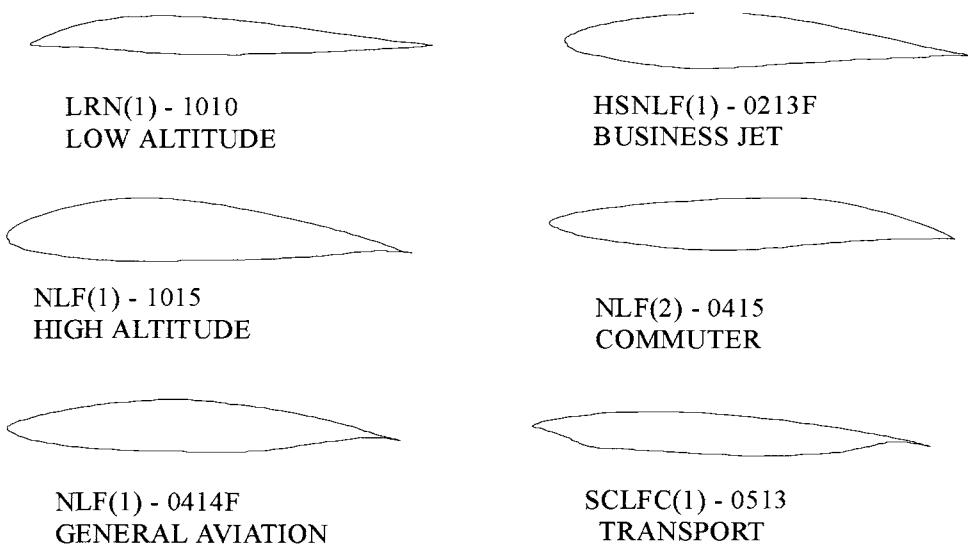


Fig. 1.1. Typical NLF airfoils for a wide range of applications. SCLFC denotes supercritical LFC airfoil.

using the e^n -method discussed in Chapter 8. The success of this technique and of the calculation method also depends on factors besides the pressure gradient including surface roughness, surface waviness, freestream turbulence, and the concentration of a second phase such as rain or solid particles in water, all of which can play a role in triggering transition [2]. The influence of these factors can usually be avoided by careful design, for example by keeping the surface waviness and roughness below the allowable limits.

A number of modern low-speed aircraft make use of extended regions of natural laminar flow on their wings [1] but transonic cruise, and the swept wings required for this configuration, introduce further complications. In particular, flow from the fuselage boundary layer can introduce instabilities which result in turbulent flow along the attachment line of the wing [2], or a favorable pressure gradient on the upper surface can result in a shock wave which interacts with the boundary-layer to cause turbulent flow. The first problem depends on the Reynolds number, sweep angle and curvature of the leading edge and it is possible to shape the leading edge of the wing so that the attachment-line flow is laminar. In this case it is likely that, depending on the sweep angle, the flow may become turbulent away from the attachment line due to the crossflow instability discussed in [2]. In subsection 1.1.2 calculations are presented for a typical NLF wing in incompressible flow to demonstrate the role of sweep angle and crossflow on transition.

Extending the region of natural laminar flow on fuselages in order to reduce the fuselage drag is also important, as indicated by the examples of Fig. 1.2, relevant to transport aircraft [1]. It should be pointed out that the total skin-friction drag of a modern wide-body transport aircraft is about 40% of the total airplane drag, with approximately 3% from nacelles and pylons, 15% from fuselage, 15% from wing, and 8% from empennage. Thus, nacelles and pylons account for about 8% of the total skin-friction drag, while the fuselage, wing and empennage account for 38%, 35% and 20%, respectively. For smaller airplanes, such as the MD-80 and 737, the portion of the total skin-friction drag is usually higher than for wide bodies.

Table 1.1. Drag coefficients for an axisymmetric body with a fineness ratio 6.14 at $\alpha = 0$, $R_L = 40.86 \times 10^6$ [1].

x_{tr}	$C_d \times 10^2$
0.322	2.60
0.15	3.43
0.10	3.62
0.05	3.74

	All-turbulent surfaces	Laminar lifting surfaces
Nacelles and misc.	5.2%	7.6%
Fuselage	48.7%	70.2%
Empennage	14.3%	6.9%
Wing	31.8%	15.3%
Nacelle and others	0.0010	0.0010
Fuselage	0.0092	0.0092
Empennage	0.0027	0.0009
Wing	<u>0.0060</u>	<u>0.0020</u>
Total profile C_D	0.0189	0.0131

Fig. 1.2. Profile drag buildup for all-turbulent transport jet and airplane with laminar lifting surfaces [1].

Table 1.1 shows the reduction in drag coefficient which can be achieved on an axisymmetric body by control of the location of the onset of transition: as an example, a delay of transition by 27% of the body length reduces the drag coefficient by some 30%. As in the case of wings, the onset of transition on fuselages and bodies of revolution can be estimated by an extension of the e^n -method discussed in Chapter 8 from two-dimensional flows to three-dimensional flows discussed in [2, 3].

Suction Through Slotted or Perforated Surfaces

The attainment of laminar flow by adjustment of pressure gradient by shaping becomes increasingly more difficult as the Reynolds number increases because the boundary layer becomes relatively thinner and, as a result, more sensitive to roughness and small disturbances. Thus, there are practical limits to maintaining natural laminar flow at high Reynolds numbers because the effort spent to maintain extremely smooth surfaces may be negated by the increased sensitivity to external factors over which one has little control.

The next technique to maintain laminar flow is the use of active laminar flow control by suction which thins the boundary-layer, generates a fuller velocity profile and leads to increased boundary-layer stability. The use of suction at the leading edge of a wing, through slots or perforated material, can overcome the tendency for the cross-flow velocity to create a turbulent boundary-layer flow beginning at the attachment line [1], see also [2]. The technique is referred to as hybrid laminar-flow control (HLFC) since it combines suction mass transfer with the arrangement of the airfoil (see Fig. 1.3) so as to impose a favorable longitudinal pressure gradient. This type of LFC is applicable to a wide range of small to moderate sized aircraft. The perforated plate makes use of holes of the order of 0.004 inches in diameter with a pitch-to-diameter ratio of around ten

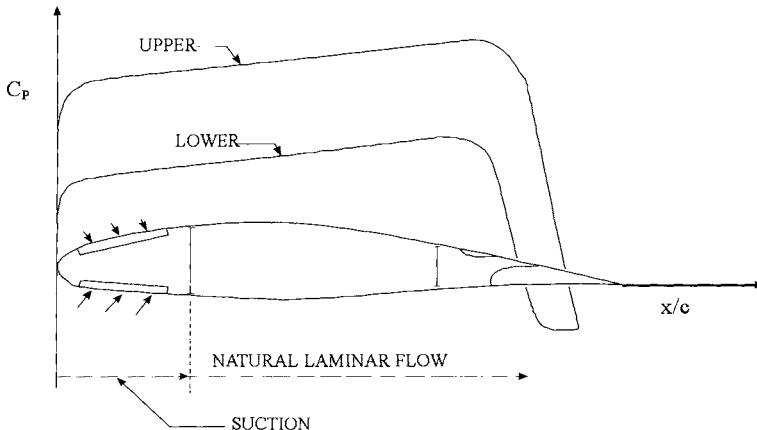


Fig. 1.3. A typical airfoil section for hybrid laminar flow control (HLFC).

and cleaning of the holes can be accomplished by reversing the mass flow while the aircraft is stationary. Extensive wind-tunnel tests have been reported by Pfenninger [1] who made use of vertical slot widths graded from 0.008 to 0.003 inches depending on the thickness of the boundary-layer and a pitch which varied from 3 to 0.6 inches depending on the static pressure. Difficulties were experienced with the effective roughness created by the edges of the slots, but the system was made to operate satisfactorily so that the effects of the cross-flow velocity were removed in that the flow around the leading edge remained laminar. Again, stability (Chapter 8) and boundary-layer (Chapter 7) theories can be used in the design of the HLFC wing, as discussed in the following subsection.

1.1.2 Calculations for NLF and HLFC Wings

A calculation method (Chapter 4 of [1]) based on the solutions of the panel, boundary-layer and stability equations for three dimensional flows can be used to demonstrate the effects of sweep, angle of attack, and suction on transition. A wing with a cross section of the NACA 6-series laminar flow airfoil family developed in the late thirties is chosen for this purpose. Its particular designation is NACA 65-412 where the first digit designates the airfoil series and the second indicates the extent of the favorable pressure gradient in tenths of chord on both upper and lower surfaces at design condition; the third digit gives the design lift coefficient and the last two digits denote the thickness in percent of the chord. The camber line used to generate this airfoil has the NACA designation $a = 1.0$ which means that the additional loading due to camber is uniform along the chord. It also happens that the use of this particular camber line results in an airfoil which has its design lift coefficient at zero angle of attack and all calculations presented here were performed at this angle of attack. The results

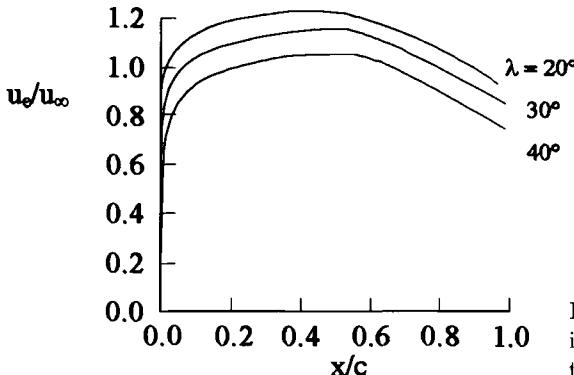


Fig. 1.4. Variation of inviscid velocity distribution with sweep angle for the NACA 65-412 wing.

correspond to a Reynolds number of 10^7 , based on the total freestream velocity V_∞ and chord c , and for several sweep angles ranging from 0° to 50° . The inviscid velocity distribution was computed from the Hess panel method [4, 5] which is an extension of the two-dimensional panel method of Section 6.4 to three-dimensional flows and the boundary-layer calculations were performed by a boundary-layer method for three-dimensional flows which is an extension of the two-dimensional boundary-layer method of Chapter 7 [2, 4]. Transition calculations are performed by using the e^n -method for three-dimensional flows which is an extension of the e^n -method for two-dimensional flows discussed in Chapter 8 [2, 3].

Figure 1.4 shows the inviscid velocity distribution u_e/u_∞ for the upper surface of the wing for $\lambda = 20^\circ$, 30° and 40° and, as can be seen, the flow has a favorable pressure gradient up to around 50-percent chord, followed by an adverse pressure gradient. We expect that the cross-flow instability will be rather weak at lower sweep angles, so that transition will take place in the region where the flow deceleration takes place. With increasing sweep angle, however, crossflow instability [2] will begin to dominate and cause transition to occur in the region of acceleration. The results of Fig. 1.5 for $\lambda = 20^\circ$ confirm this expectation and indicate that amplification factors computed with different frequencies reach values of n as high as 6.75 at $x/c = 0.44$ but not a value of $n = 8$ as required to indicate transition (Chapter 8). Additional calculations show that transition occurs at $x/c = 0.65$ and is not caused by crossflow instability. The results for $\lambda = 40^\circ$, shown in Fig. 1.6, however, indicate that crossflow instability makes its presence felt at this sweep angle, causing transition to occur at $x/c = 0.08$ corresponding to a radian disturbance frequency of 0.03740. The location of the critical frequency is at $x/c = 0.0046$, very close to the attachment line of the wing.

Calculations performed for $\lambda = 30^\circ$, 35° and 50° indicate results similar to those for $\lambda = 40^\circ$ in that the transition location moves closer to the leading edge

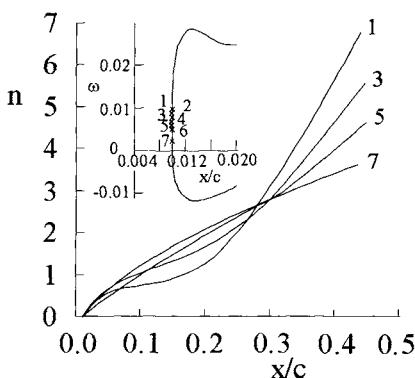


Fig. 1.5. Amplification factors for several frequencies for $\lambda = 20^\circ$. The numbers 1 to 7 show different frequencies used for each amplification calculation (Chapter 8).

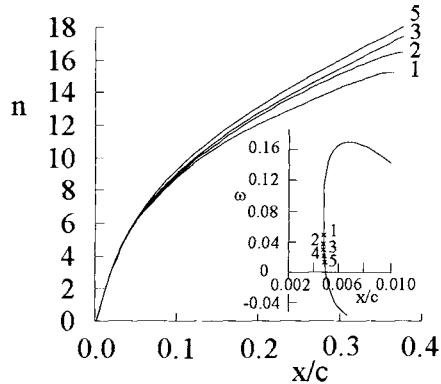


Fig. 1.6. Amplification factors for several frequencies for $\lambda = 40^\circ$.

with increasing sweep angle, occurring at $x/c = 0.22$ for $\lambda = 30^\circ$, at $x/c = 0.12$ for $\lambda = 35^\circ$ and at $x/c = 0.05$ at $\lambda = 50^\circ$.

Figures 1.7 to 1.9 show the calculated amplification factors for the same wing with suction, which is a powerful means of maintaining laminar flow over the whole wing. In practice, however, this is difficult to achieve because of the need for ailerons, flaps and openings for inspection and maintenance. Clearly a suction system adds to the complexity, weight and cost of a design. Increasing suction rates requires larger ducting system and more power so that at some point all available space in the wing may be used up and the higher suction drag will produce diminishing returns. Increased suction also makes the boundary-layer thinner, which in turn reduces the critical height of roughness that will cause transition. If suction is applied through discrete holes or slots and is not

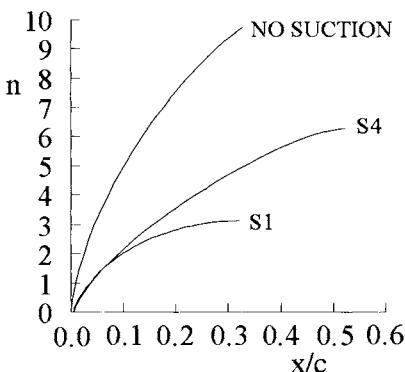


Fig. 1.7. Effect of suction on amplification rates for $\lambda = 30^\circ$.

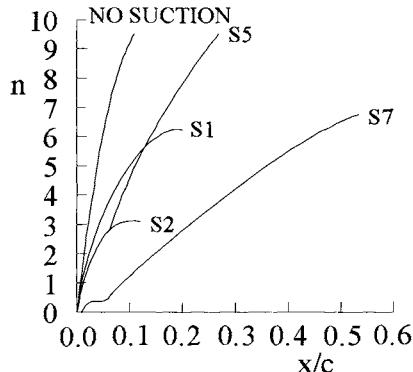


Fig. 1.8. Effect of suction on amplification rates for $\lambda = 40^\circ$.

Table 1.2. Suction rates $\bar{v}_w \equiv v_w/V_\infty$ used in the stability calculations. S1, S2 and S3 are applied to the whole wing while S4 to S8 are applied to the first 5% chord of the wing. S9 is applied to the first 10% of the wing.

	$\bar{v}_w \times 10^4$		$\bar{v}_w \times 10^4$
S1	-3	S5	-5
S2	-5	S6	-7
S3	-7	S7	-10
S4	-3	S8	-12
		S9	-12

distributed over the area, increased suction velocities may cause the suction holes or slots to become critical themselves and act as sources for disturbances. It is important that the suction system be carefully designed by calculating minimum suction rates to maintain laminar flow. In addition, the suction rate distribution must be optimum. A calculation method, such as the one described in Chapter 4 of [1] and [3], is capable of determining the minimum and optimum suction rates for the ducting system. Table 1.2 lists the suction distributions used in the calculations presented here. For simplicity, two types of suction distributions are considered: the first with uniform suction on the whole wing and the second with uniform suction over the front portion of the wing only, e.g. 5% chord from the leading edge.

Figure 1.7 shows the amplification factors for three frequencies: one without suction, and the other two for two types of suction, S1 and S4 for $\lambda = 30^\circ$. As can be seen, a small suction level of $\bar{v}_w = -0.0003$ either over the whole wing, S1, or over the front 5% chord of the wing, S4, is sufficient to maintain laminar flow until separation or transition occurs at $x/c = 0.58$ for S4 and at $x/c = 0.78$ for S1. The calculations for S1 produce a low value of $n = 3$ at $x/c = 0.34$ and indicate that the suction rate is excessive at this sweep angle.

Figure 1.8 shows the results for $\lambda = 40^\circ$ for which case a suction level of $\bar{v}_w = -0.0003$ for S1 yields a maximum value of $n = 6$ at $x/c = 0.20$ and a suction level corresponding to S2 yields a maximum value of $n = 3$ at $x/c = 0.12$. Both cases eliminate transition which occurs at $x/c = 0.08$ without suction, but the latter also eliminates the occurrence of separation while the former delays the separation until $x/c = 0.78$. To avoid excessive suction, two additional cases corresponding to S5 and S7 were considered and it was observed that transition takes place at $x/c = 0.22$ for S5, and the maximum value of n is equal to 6.7 at $x/c = 0.52$ for S7 which shows that the crossflow instabilities can be eliminated in the front portion of the wing. It is interesting to note that the small bump near $x/c = 0.05$ along the curve for S7 shown in Fig. 1.8 is caused by the switch-off of suction at $x/c = 0.05$.

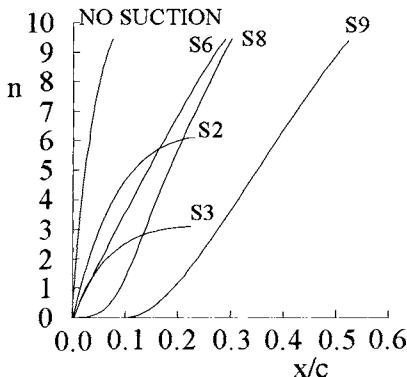


Fig. 1.9. Effect of suction on amplification rates for $\lambda = 50^\circ$.

As expected, it is more difficult to avoid the crossflow instabilities for $\lambda = 50^\circ$ because of the high sweep, and Fig. 1.9 shows that only suction levels corresponding to S2 and S3 can eliminate transition. However, if suction is switched off at 5% chord from the leading edge, transition occurs even if a high suction level of $\bar{v}_w = -0.0012$ is applied. In order to laminarize the flow, it is necessary to extend the range of suction at a suction level of $\bar{v}_w = -0.0012$ for the first 10% chord of the wing, case S9, leading to transition at $x/c = 0.48$ which is 8% upstream of the separation location. Further extensions of the suction area will eliminate transition before separation occurs. From the results corresponding to S8 and S9, it can be seen that the growth of the disturbances can be prevented only in the range over which suction is applied for $\lambda = 50^\circ$. Once the suction is switched off, the disturbances grow with almost constant speed and cause transition to occur downstream, indicating the difficulty of laminarizing the flow on a highly swept-back wing.

1.2 Prediction of the Maximum Lift Coefficient of Multielement Wings

In aircraft design it is very important to determine the maximum lift coefficient as accurately as possible, since this lift coefficient corresponds to the stall speed, which is the minimum speed at which controllable flight can be maintained. Any further increase in angle of incidence will increase flow separation on the wing upper surface, and the increased flow separation results in a loss in lift and a large increase in drag.

The high-lift system of an aircraft plays a crucial role in the takeoff and landing of an aircraft. Without high-lift devices, the maximum lift coefficient, $(C_L)_{\max}$, attainable by a high-aspect-ratio wing is about five times the incidence (in radians) at incidences up to stall. Typical values of $(C_L)_{\max}$ are commonly in the range of 1.0 to 1.5. The addition of high-lift devices such as flaps and

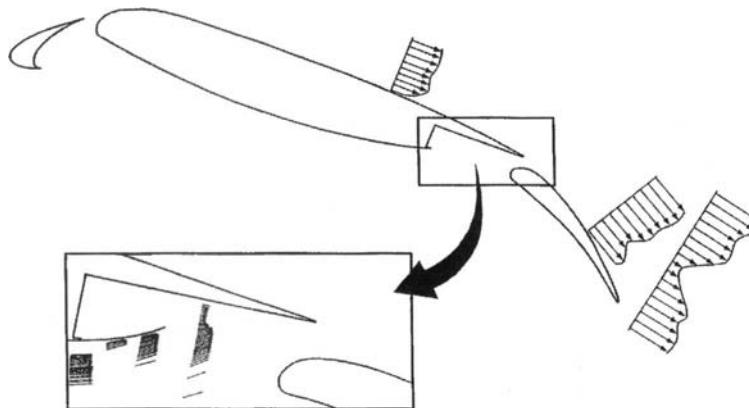


Fig. 1.10. Flow over a typical high-lift system

slats can more than double $(C_L)_{\max}$ with subsequent improvement in takeoff and landing performance. Thus, it is important to predict the performance of high-lift systems that can be designed for high $(C_L)_{\max}$ in landing configuration and high lift-to-drag ratio in take-off configuration. The lower drag also results in lower noise, which is necessary to comply with noise abatement regulations.

Despite the significant advances in CFD, our ability to predict the maximum lift coefficient of multielement wings is still not satisfactory. As shown in Fig. 1.10, the flow about multielement airfoils for high lift is very complex. The main problem is the lack of an accurate turbulence model (Chapter 3) to represent flows with extensive separation. The problem is exacerbated by inaccuracies of numerical solutions of the conservation equations (Chapter 2) at these flow conditions and difficulties in modeling flow near the trailing edge of an airfoil or wing, trailing viscous wakes that may impinge on aft elements, merging boundary-layers, and flow separation.

In this section we describe a useful design method developed by Valarezo and Chin [6]. This method, called “The Pressure Difference Rule”, for predicting the maximum lift coefficient of multielement wings is based on Hess’ panel method which is an extension of the two-dimensional panel method of Section 6.4 to three-dimensional flows. The accuracy of this method, even though the solution is based on the reduced conservation equations and does not include the effects of viscosity, is then demonstrated for the high-lift systems of a transport aircraft as a function of Reynolds number. While this method is appropriate for configuration development, it cannot predict the optimum gap/overhang locations for each of the high-lift wing components; at this time the determination of promising range of locations is performed using two-dimensional CFD methods for multielement airfoils. The final determination of the optimal locations is made in high-lift wind tunnel tests. The ability to predict reliably the optimal

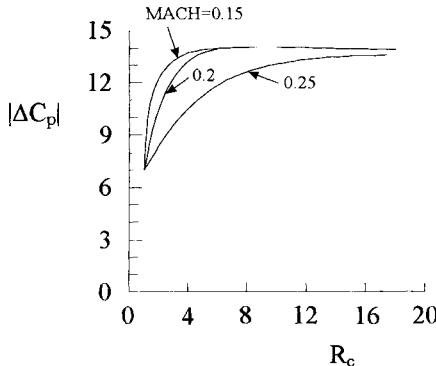


Fig. 1.11. Variation of $|\Delta C_p|$ with chord Reynolds number R_c at maximum lift conditions.

location of flaps and slats is one of the aims of CFD development efforts in high-lift research.

The Pressure Difference Rule of Valarezo and Chin [6] is based on the examination of wind tunnel data which indicates that, at a given Reynolds/Mach number combination, there exists a certain pressure difference ΔC_p between the suction peak of an airfoil (C_p)_{min} and its trailing edge (C_p)_{te} at the maximum lift condition. For the case of a multielement airfoil, the same rule applies to whichever element (leading-edge or main) is critical at maximum lift. Thus, at a given freestream Mach number, there is a “pressure difference” $|\Delta C_p| = |(C_p)_{\min} - (C_p)_{\text{te}}|$ variation with Reynolds number (Fig. 1.11) that indicates when maximum lift is attained. This correlation applies whether or not the airfoil has an auxiliary leading-edge device. Even though the Pressure Difference Rule is based on two-dimensional data, Valarezo and Chin assume the correlation in Fig. 1.11 to be valid also for three-dimensional flows. They determine the maximum lift coefficient of multielement transport wings by the following procedure:

1. Use a panel method to obtain flow solutions at various angles of attack for the desired geometry. While any reliable panel method can be used for this purpose, they use the Hess panel method discussed in detail in [4, 5, 7]. They recommend sufficient surface paneling to ensure adequate definition of the geometry at the leading and trailing edges.
2. For a given freestream Reynolds number and Mach number, construct a pressure difference $|\Delta C_p|$ distribution vs. span based on the wing chord distribution.
3. Determine graphically at what spanwise wing station and wing lift coefficient the solutions obtained from the panel method (Step 1) match the curve constructed in Step 2.

Valarezo and Chin validated this method with RAE experimental data [8] obtained for a high-lift system. The wing had an aspect ratio of 8.35 and wing

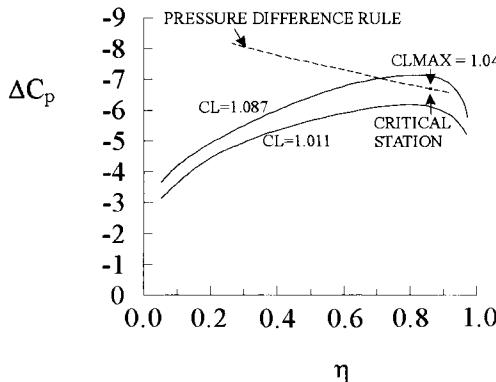


Fig. 1.12. Pressure difference rule prediction of max lift for the RAE wing.

quarter-chord sweep of 28° with a taper ratio of 0.35. The high-lift system included a 16% chord leading-edge slat ($\delta_S = 15^\circ, 20^\circ$ and 25°) and a 34% Fowler flap ($\delta_F = 10^\circ, 25^\circ$ and 40°). The test was conducted transition-free at a Reynolds number of 1.31×10^6 based on the mean wing chord and the nominal Mach number was 0.22. The Pressure Difference Rule was used to predict $(C_L)_{\max}$ for wing configurations corresponding to wing-alone, wing-flap, slat-wing and slat-wing-flap.

Figure 1.12 shows the predicted pressure difference for the wing alone. The results for $C_L = 1.011$ and 1.087 correspond to the panel method solutions at angles of attack of 11.84° and 12.84° , respectively (Step 1). The allowable variation of ΔC_p along the span was obtained from Fig. 1.11 for chord Reynolds numbers of 1.61×10^6 and 1.01×10^6 at spanwise stations η of 0.3 and 0.76, respectively, and for an interpolated $M_\infty = 0.22$, yielding $|\Delta C_p| = 8.2$ at $\eta = 0.30$ and $|\Delta C_p| = 7$ at $\eta = 0.76$. The dashed straight line connecting these two points represents the boundary that predicts when $(C_L)_{\max}$ occurs. According to Fig. 1.12, linear interpolation yields a predicted $(C_L)_{\max}$ of 1.04 and the critical spanwise station is identified at 87% of the span.

The predictions of the Pressure Difference Rule for the RAE wing with different flap deflections are shown in Fig. 1.13 together with the experimental and calculated lift curves. The calculated viscous flow results were obtained by using the interactive boundary-layer method described in detail in [5] and briefly in Chapter 7. The results denoted as semi-empirical were obtained from the inviscid panel method by reducing the nominal flap angle in order to account roughly for the known decambering effect of the boundary-layer and wakes on the aft segments of a multielement wing. Agreement between experiment and prediction is seen to be very good throughout each lift curve up to and including $(C_L)_{\max}$. The effect of flap deflection on maximum lift for the wing-flap configuration is shown in Fig. 1.14, where the method based on the Pressure Difference Rule correctly indicates marginal lift improvements in going from 25° to 40° flaps for this particular wing. The ability to predict this is a key result

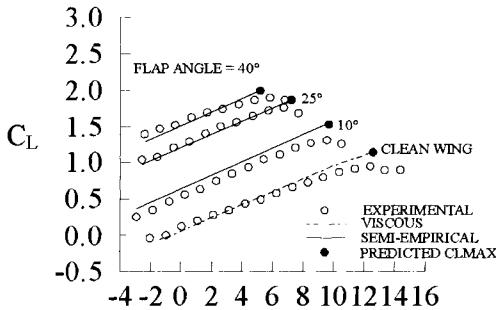


Fig. 1.13. Lift curves for RAE wing.

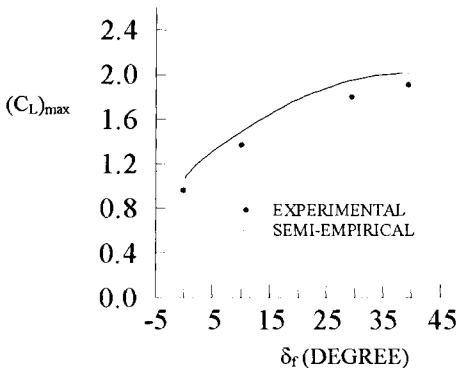


Fig. 1.14. Effect of flap deflection on maximum lift.

since it shows that 40° flaps offer only a minimal improvement to lift, and therefore going from 25° to 40° flaps is not desirable given the substantial drag increase.

Further applications and validations of the Pressure Difference Rule are reported in [6] by Valarezo and Chin for several narrow-body and wide-body transport configurations. Figure 1.16 shows the results for the narrow-body transport of Fig. 1.15. The wing is configured for landing with both leading and trailing-edge devices deployed. The predicted variation of $(C_L)_{max}$ with Reynolds number shown in Fig. 1.16 compares very well with available wind tunnel and flight test results. As can be seen, the variation of $(C_L)_{max}$ with Reynolds number is considerable, and the method based on the Pressure Difference Rule captures it remarkably well.

A particular application of the Pressure Difference Rule to a Regional Jet transport is reported in [9]. Figure 1.17b shows the good correlation obtained between predictions and wind tunnel test results for the cruise configuration of Fig. 1.17a. In [9], a simple method is introduced as an extension of the Pressure Difference Rule that allows the estimation of the maximum lift of an aircraft configuration with leading edge contamination. Aircraft certification regulations stipulate that an aircraft handling characteristics and performance

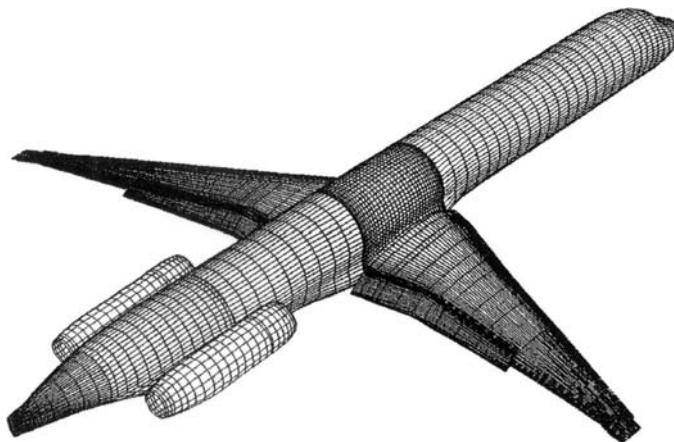


Fig. 1.15. Paneled narrow-body transport.

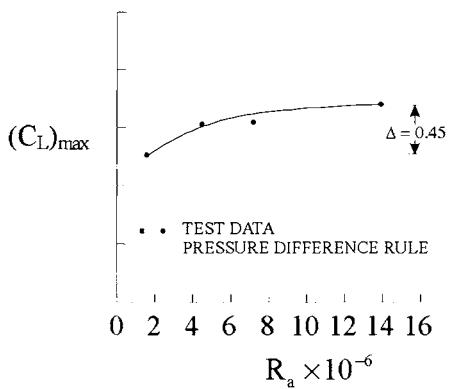


Fig. 1.16. Variation of maximum lift coefficient with Reynolds number based on mean aerodynamic chord, R_a .

must be determined for flight in icing conditions and in roughness conditions. The roughness can be caused by ice, frost, de-icing and anti-icing fluids used prior to take-off, insect contamination, paint and surface irregularities and leading edge damage such as that produced by a hail-storm. Roughness on the wing leading edge affects the stall characteristics of an aircraft and its performance.

The method is based on a combination of the Pressure Difference Rule [6], using a three-dimensional panel method, with results of a two-dimensional interactive viscous-inviscid CFD procedure developed by Cebeci [5] briefly described in Chapter 7. The code is able to predict aerodynamic performance of single and multi element airfoils, including stall, with and without surface roughness, with sweep effects, for steady flows. The code uses a Hess and Smith panel method, which is an extension of the panel method discussed in Section 6.4, to calculate the inviscid flow field with a simple Karman-Tsien compressibility correction formula. A two-dimensional compressible boundary layer code operating in an

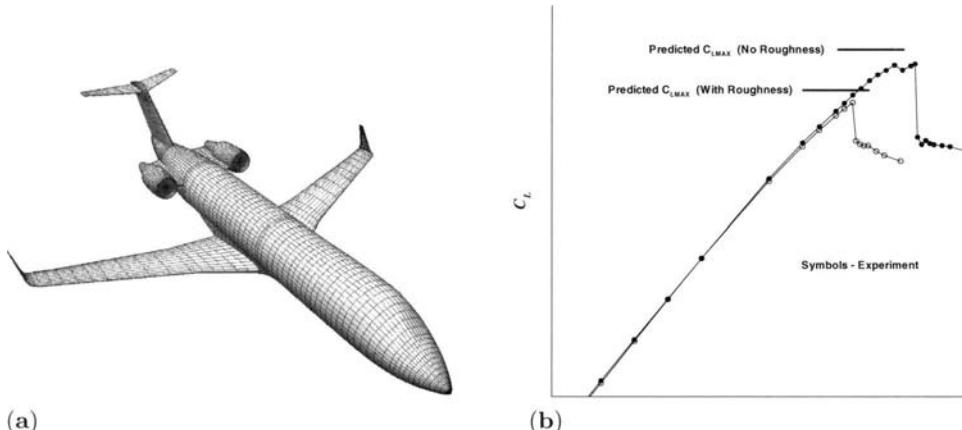


Fig. 1.17. (a) Regional jet cruise configuration panelling. (b) Predicted maximum lift with and without roughness, comparison with experimental data.

inverse mode, is coupled to the panel method. Michel's formula (Chapter 8) is used for transition prediction and the Cebeci-Smith model is used for turbulence modeling with roughness effects. The equivalent sand grain roughness (k_s/c) is the input characterization parameter for the code.

The method is illustrated in Fig. 1.18, using a model of the M100 ONERA wing/body test article [10]. In this application, the VSAERO panel method of Analytical Methods Inc. [9] is used. An initial VSAERO analysis is first conducted to determine the critical spanwise location where the maximum pressure difference occurs. Based on the local chord Reynolds number at that critical section, a two-dimensional (2D) analysis is conducted to determine the incremental effects of roughness on maximum lift. The figure shows the 2D lift curves calculated with and without contamination. This increment is applied to the original limit ΔC_p curve and compared with the original spanwise distributions of ΔC_p to determine the new maximum lift point with contamination. Figure 1.18 shows the limit ΔC_p curves with and without roughness as well as the spanwise distributions of ΔC_p as calculated using VSAERO for several angles of attack. Finally, Fig. 1.18 shows the predicted maximum lift for the configuration with and without roughness. The methodology was validated using the results of the wind tunnel tests carried out on a 1/3 scale model of a regional jet. Tests were conducted at Mach 0.15 and mean chord Reynolds number of 2.72 million, for various levels of wing contamination. Figure 1.17b shows the comparison of predicted and experimentally measured maximum lift coefficients with and without contamination for the cruise configuration. The relative loss in lift due to contamination compares well with experiment, although the absolute levels are slightly over-predicted in this case.

Although Navier-Stokes solvers are now routinely used to analyse full aircraft configurations in cruise conditions, prediction of aircraft high-lift perfor-

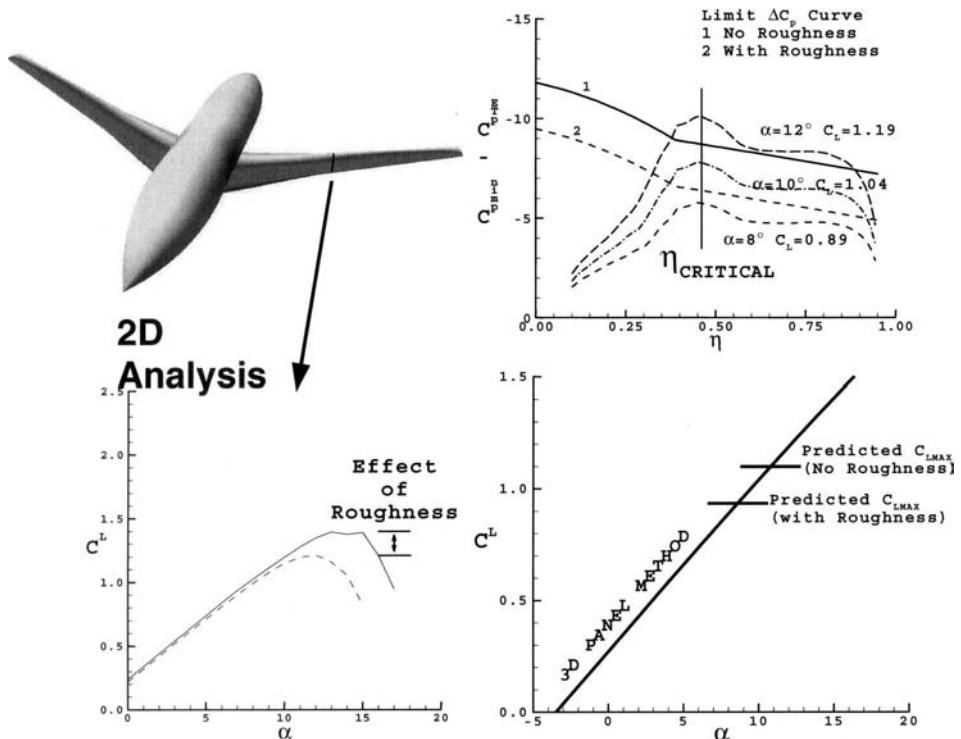


Fig. 1.18. Prediction of roughness effects on wing-body (C_L)_{max} using the pressure difference rule.

mance using these CFD methods is still a challenge. This results from the increased geometric complexity of high-lift configurations with deployed slats and/or flaps and the need to model all the relevant features of a very complex flow. Mesh-generation then becomes a challenging task, even when an unstructured-grid approach is used, and the resulting meshes can be an order of magnitude larger than those needed to accurately predict cruise performance. To model realistic flow around a complete Boeing 777-200 high-lift configuration, Rogers et al. [12] employ 22.4 million grid points using overset grids. The prediction of maximum lift and wing stall constitutes a challenge even for a clean wing configuration, as massive flow separation must be modelled.

An application of a Navier–Stokes method to the investigation of an aircraft maximum lift is reported in [13]. The NSU3D [14] unstructured Navier–Stokes solver is used for the study (Chapter 12). It uses an edge-based, vertex-centred finite-volume scheme for space discretisation and a multi-stage Runge–Kutta technique for time integration with point or line pre-conditioning. An agglomeration multigrid algorithm is implemented for convergence acceleration. Two turbulence models are implemented: the Spalart–Allmaras model (Chapter 3)

and the Wilcox $k-\omega$ model [15, 16]. Both can be used with or without wall functions.

A Challenger aircraft wing/body/nacelle configuration was selected to investigate the ability of NSU3D to predict flows at high angles of attack up to and beyond stall. The geometry modelled represents the wind tunnel model including flap fairings and flow-through nacelles. Even though this is a relatively simple configuration, it presents some meshing difficulties, mainly in the generation of the prism layers required for Navier-Stokes computations. This is due to the presence of such features as narrow gaps in the nacelles where prism layers growing from two facing surfaces can collide if not properly limited. The modelling of the flap fairings also made the generation of the prism layers more complex, since the latter have to wrap around the fairings.

The unstructured mesh (Section 9.7) consists of 209,000 tetrahedra in the field, 6,358,000 prisms around the aircraft surface and 9000 pyramids (to cap incomplete prism layers). The first prism layer is given a thickness of 6×10^{-6} times the wing tip chord to ensure values of y^+ of the order of 1 needed for the application of turbulence models down to the solid surface (Chapter 3). A growth ratio of 1.3 from one layer to the next is imposed. The number of layers varies from 26 on the nacelle core cowl to 35 on the wing, fuselage and wing-body fairing, for a maximum prism layer thickness of 7% of the root chord.

The flow conditions of the wind tunnel data used for comparison are a Mach number of 0.25 and a Reynolds number of 2.2×10^6 , based on the wing mean aerodynamic chord. The stall pattern on this configuration is typical of transonic jets with no slats or leading edge flaps. A leading edge flow separation, due to the bursting of a laminar short bubble, causes a sudden loss of lift at stall.

The relative performance of the Spalart-Allmaras and $k-\omega$ turbulence models in predicting the lift variation with incidence was evaluated on this mesh. Convergence was satisfactory at most angles of incidence: the density residual was reduced by 4 to 5 orders of magnitude at incidences up to 15° . At higher angles of incidence, it did not decrease as much, but the convergence of the lift coefficient was still good. Post-stall isobars and skin-friction lines computed at $\alpha = 14.21^\circ$ using the $k-\omega$ turbulence model are shown in Fig. 1.19a. The predicted lift variation with incidence for the two turbulence models is compared with the experimental data in Fig. 1.19b. These results were obtained with the assumption of fully turbulent flow. At incidences up to 10° , both turbulence models predict lift fairly well. At higher incidences, however, the predicted lift is lower than the experimental data before stall, with the one-equation Spalart-Allmaras model results being worse than those obtained with the two-equation $k-\omega$ model. Both models underpredict the pre-stall lift coefficient, due to an excessive amount of predicted separated flow on the outboard wing. None of the numerical results predicts the sudden drop of lift after stall, but the Spalart-Allmaras predictions show a kink in the lift variation shortly after the experi-

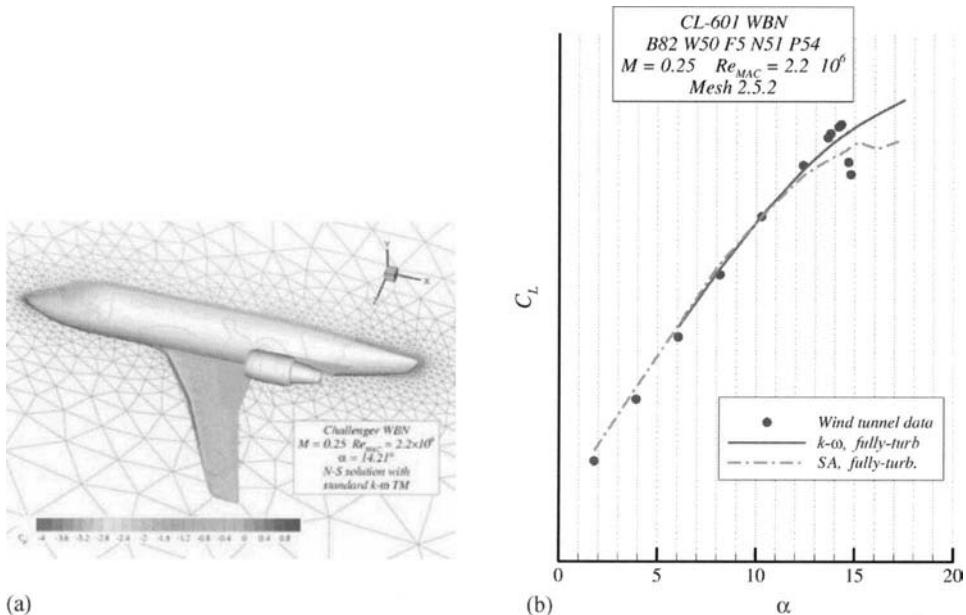


Fig. 1.19. (a) Post-stall isobars and skin-friction lines on a business jet clean-wing configuration at Mach 0.25 and $\alpha = 14.21^\circ$. NSU3D Navier–Stokes solution with a $k-\omega$ turbulence model. (b) Comparison of predicted lift curves with experimental data.

mental stall incidence. Surface flow patterns and pressure distributions indicate that this occurs when the flow separates on the inboard wing. It should be noted that the post-stall flow is highly unsteady. Predicting the post-stall variation of lift with a steady-flow code is therefore questionable. Modelling laminar flow at the leading edge of the wing improves marginally the results but it is fair to conclude that the present models need improvements before they can predict correctly the maximum lift behavior of three-dimensional wings.

1.3 Aircraft Design and Power Plant Integration

Today CFD plays an important role in aircraft design and, together with wind tunnel testing and flight testing, it can help to design an aircraft that has superior performance with reduced risk and low cost. One example of a high performance subsonic jet aircraft is Bombardier's Global Express long-range high-speed business jet [17]. This aircraft, outlined in Fig. 1.20, has a Maximum Take-Off Weight of 95,000 lbs and is powered by two Rolls-Royce BR710 turbofans, each developing 14,750 lbs thrust. The aircraft can fly 8 passengers and 4 crew members over a distance of 6500 NM at Mach 0.80 and 6000 NM at Mach 0.85. The high-speed cruise Mach number is 0.89. The aircraft can



Fig. 1.20. Global Express configuration.

operate on runways of less than 6000 ft, climb to an initial cruise altitude of 43,000 ft and reach a maximum certificated altitude of 51,000 ft.

The combination of speed and fuel requirements of the Global Express, a relatively small aircraft compared to modern jet transports, is a challenge for any aircraft designer. By using advanced CFD methods for design and optimization and wind tunnel testing for verification, it was possible to minimize the drag of the aircraft at high-speed cruise and to arrive at a configuration with good take-off and landing performance. This was achieved by developing an efficient transonic wing, a low drag power-plant installation and an efficient high-lift system with leading edge slats and trailing edge Fowler flaps. The drawing in Fig. 1.20 shows the aerodynamic features of the aircraft that were considered necessary to meet the design requirements. The airplane has a T-tail configuration with two turbofan engines mounted on the aft fuselage to keep the wing free from adverse nacelle/engine interference. In addition the fuselage was tailored in the area of the nacelle and pylon to eliminate drag-producing shocks during cruise at high Mach number. The integration of the power plant required also an optimisation of the pylon shape. The objective was to eliminate undesirable shocks that appeared on the lower surface of the pylon and the nacelle at cruise conditions above Mach 0.8. The aerodynamic configuration was designed and developed in the period between 1991 and 1994 and first flight occurred in 1996. At the time, the validated CFD methods available to the designers were two-dimensional Navier–Stokes solvers and three-dimensional Euler solvers for complete aircraft configurations. The inviscid Euler solvers were coupled with compressible boundary layer codes for lifting surfaces (see Chapter 10 for Euler methods and Chapter 7 for boundary layer methods).

The shaping of the fuselage was first carried out with the aid of the KTRAN Transonic Small Disturbance CFD program [18]. The pylon was not included in

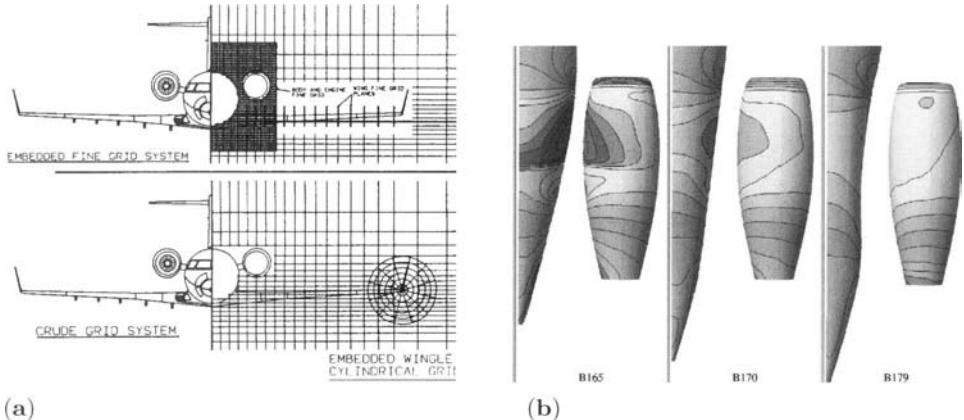


Fig. 1.21. (a) KTRAN rectangular mesh. (b) KTRAN solutions at three stages of Global Express fuselage design (B165, B170, B179), Mach 0.85.

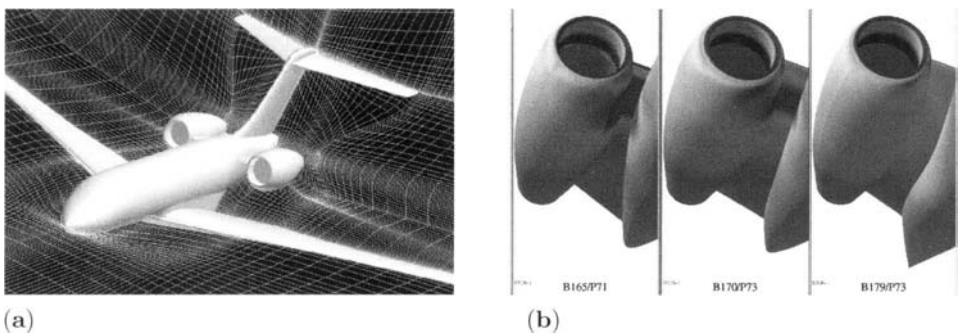


Fig. 1.22. (a) Global Express block-structured Euler mesh. (b) MBTEC Euler solution at three stages of the fuselage and pylon design (B165/P71, B170/P73, B172/P73), Mach 0.85.

the aircraft configuration in these calculations. Since the work required several iterations, KTRAN was ideal for obtaining quick results. Figures 1.21a and 1.21b show the mesh and the results obtained with KTRAN at three different stages of the fuselage design process.

The fuselage shape that was obtained from these calculations was used as input to the MBTEC Euler code [19] to check the flow situation with the addition of the nacelle pylons. Finally the pylons shape and the nacelles position, in terms of incidence and toe-out angles were optimized with the aid of MBTEC. Figure 1.22a shows the multi-block structured mesh generated with the grid generation program MBGRID [20]. Figure 1.22b shows the solution obtained with MBTEC at Mach 0.85 cruise conditions at three different stages of the fuselage and pylon design.

The integration of the pylons and nacelles was verified in a wind tunnel test that was conducted at the Aircraft Research Association (ARA) 8 ft. \times 9 ft.

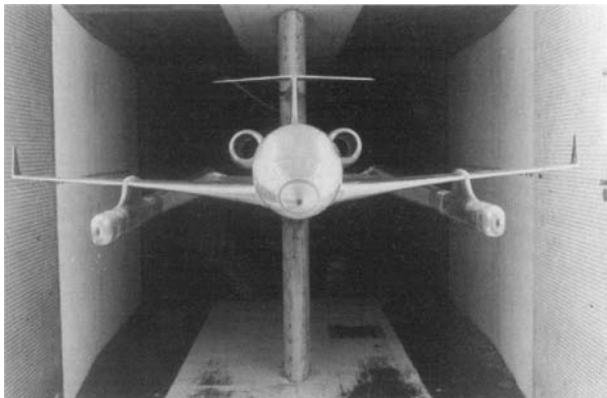


Fig. 1.23. Global Express model on twin sting rig at ARA 8 ft × ft transonic wind tunnel.

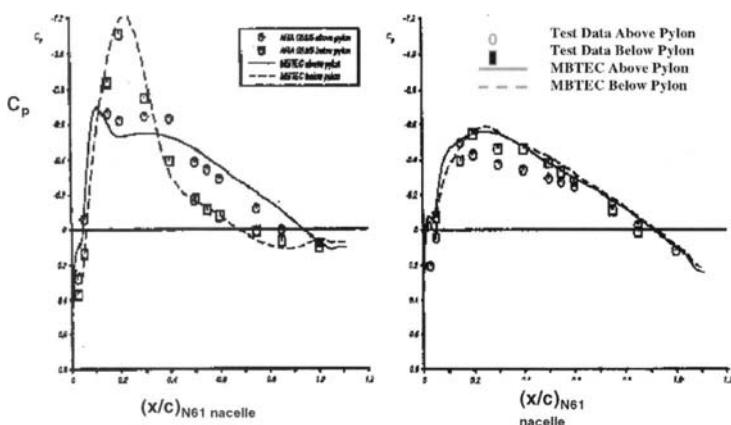


Fig. 1.24. Pylon pressure comparison of Euler solution with wind tunnel test data on the initial (B165/P71) and final configurations (B172/P73).

atmospheric wind tunnel (Bedford, U.K.). The test was carried out using a 7% scale full model mounted on twin stings (Fig. 1.23). The model was built with three interchangeable aft fuselage shapes designated B165, B170 and B172. B165 was the initial configuration. B170 was a configuration with the fuselage diameter reduced aft of the wing trailing edge. B172 was a configuration with the fuselage shaped locally to improve the channel flow at the nacelle location. Two different shapes of pylons were tested (P72 and P73). The nacelles were tested over a matrix of nine combinations of angles of incidence and toe-out. Several nacelle configurations were tested, in order to establish the effect of varying inlet Mass Flow Ratios and the effect of the geometry of the nacelle boat-tail on the fuselage flow. The wing, aft fuselages, nacelles and pylons were pressure tapped. Forces and moment were measured on the twin sting balance.

The test covered a range of Mach number from 0.6 to 0.97 and the results were reduced to a nominal chord Reynolds number of 3.6 million.

The test yielded the following results:

- Large drag improvements were obtained with the re-contoured fuselages at all Mach numbers above Mach 0.70. The largest drag reduction (3% to 5% of total aircraft drag) was obtained with the shaped fuselage B172. The shaped pylon P73 contributed substantially to the weakening of the shock wave in the channel.
- All drag reductions were associated with a lowering of the peak Mach number in the channel between the fuselage and the nacelle, a reduction of the pylon download, and a better control of the diffusion in the aft end of the channel.
- The optimum orientation of the nacelle for drag was found to be precisely the one predicted by using the MBTEC multi-block Euler code for optimal nacelle pressures.

Figure 1.24 shows a comparison of MBTEC predictions with pressures measured on the fuselage above and below the nacelle pylon on the initial (B165/P72) and final (B170/P173) configurations. This comparison shows that the inviscid Euler results (Chapter 10) on the fuselage were a good indicator of the flow field generated on this part of the aircraft.

1.4 Prediction of Aircraft Performance Degradation Due to Icing

Aircraft icing presents a serious hazard for flight at subsonic speeds in visible moisture and at temperatures near or below freezing. Many aircraft have been lost due to ice accumulation. Some twenty accidents where icing was a contributing factor are listed in Fig. 1.25. In the absence of thermal ice protection, ice on wings, control surfaces, and engine intakes can reduce the aerodynamic performance of the aircraft. Therefore, the Federal Aviation Administration (FAA) requires an airplane manufacturer to demonstrate that its aircraft can fly safely in icing conditions as defined by the so-called icing envelopes in the FAA's Federal Airworthiness Regulations (FAR) Part 25, Appendix C [21].

Ideally one would like to prevent ice from accreting anywhere on the airframe, which is unfortunately not always possible. Thus, the analysis of an aircraft's response to an inflight icing encounter plays a key role during the development and certification phase of an aircraft. All icing testing is relatively expensive, however. In today's competitive environment, cost-effective calculation methods must be developed so that the aircraft manufacturer can evaluate the performance of a system for a range of icing conditions and consequently reduce development and certification time and cost. Full-scale icing experiments over a wide range of conditions would be very expensive.

Date	Airline	Location	Aircraft Type	Precipitation and Observations
26 Jan 74	THY	Cuma ovasi, Turkey	F28	Probable cause: frost accretion on the wings
03 Jan 77	JAL	Anchorage	DC-8-62	Fog
04 Jan 77	JAL	Frankfurt	737	Light snow/25-degree wheel reod after liftoff, rime ice observed on wing
27 Nov 78	TWA	Newark	DC-9-10	Blowing rain and snow
20 Dec 78	N40SN	Minneapolis	Learjet	Probable cause: snow and ice on wings
19 Jan 79	N73181	Detroit	Learjet	Probable cause: premature stall caused by accumulation of wing ice
12 Feb 79	Allegheny	Clarksburg	Nord 262	Light snow – frozen snow photographed on empennage after accident
18 Feb 80	Redcote	Boston	Bristol 253	Light snow
13 Jan 82	Air Florida	Wash. D.C.	737	Moderate-to-heavy snowfall
05 Feb 85	Airborne	Philadelphia	DC-9-10	Light freezing rain, ice and snow pellets, fog
12 Dec 85	Arrow Air	Gander	DC-8-63	Light freezing drizzle, snow grains
27 Dec 86	Ozark	Sioux City	DC-9-10	Light freezing drizzle
15 Nov 87	Continental	Denver	DC-9-10	Moderate snow, fog
18 Jan 88	N2814U	New Mexico	Cessna 402	Probable cause: ice/frost removal inadequate
06 Feb 88	N2832J	California	Cessna A188B	Probable cause: ice/frost removal inadequate
23 Dec 88	N5570H	Montana	Piper PA-11	Probable cause: wing ice
10 Mar 89	Air Ontario	Dryden	F28	Heavy snow
25 Nov 89	Korean Air	Kimpo	F28	Dense fog, ice on the wing
16 Feb 91	Ryan	Cleveland	DC-9-10	Light snow
22 Mar 92	USAIR	New York	Fokker F28	Inadequate ice/frost removal
31 Oct 94	American Eagle	Roselawn	ATR72	Ice ridge in front of aileron from supercooled water droplets
09 Jan 97	COMAIR	Detroit	Embraer 120	Icing in holding pattern
07 Apr 99	THY	Turkey	B737-4Q8	Icing after takeoff

Fig. 1.25. Partial list of accidents where icing was a contributing factor.

Heating is required to prevent ice accretion anywhere on the aircraft; the heat keeps the impinging droplets from freezing (running wet) or evaporates all impinging water (anti-icing). Unfortunately, it is not possible to provide sufficient thermal energy to attain complete ice prevention and still remain

economically competitive. Thus, some areas of an aircraft are anti-iced (no ice), some are de-iced (cyclic ice buildup and removal), and some are left unprotected.

In addition to meeting safety requirements, the aircraft industry must meet the challenges of rising operating costs and intense economic competition. The industry, therefore, places heavy emphasis on reducing fuel burn, increasing range, and improving maintainability and reliability. Aircraft ice protection impacts all four of these economic considerations in surprisingly complex ways. For example, ice protection devices must be defined accurately so that high confidence can be placed in the important trade and risk studies. Conservative assumptions can result in excessive predicted ice protection system weight, power, and cost.

For both economic and safety reasons, in 1978 NASA established an icing program at its Lewis Research Center in Cleveland, Ohio. This icing program is guided by three strategic objectives [22]. One is to develop and validate computer codes that will numerically simulate an aircraft's response to an inflight icing encounter. This challenging task requires two steps. The first step is to predict ice accretion on the airframe, which is a very complicated process because of the numerous parameters involved. For example, both the aerodynamic and thermodynamic parameters play an important role in the development of ice accretion at the leading edge of the lifting body. The second step is to predict the aerodynamic performance of the aircraft and its stability and control characteristics when there is some ice on the airframe. For instance, in the case of the flowfield over an iced wing, flow with regions of separation must be computed. The successful development of the desired computer codes offers great advantages:

1. Validated computer codes will substantially reduce developmental and certification testing. This results in reduced time and cost of aircraft development.
2. Numerical simulation, which is an alternative to extensive flight testing, will reduce the high risk of flight testing in icing conditions.
3. Accurate numerical simulations will allow earlier assessment of the effect of ice protection requirements on new aircraft designs.

This section describes the application of a CFD method to predict ice accretion and aircraft performance degradation due to icing, as discussed by Cebeci and Besnard [23]. The results are presented for the NASA research aircraft which is a modified DeHavilland DH-6 Twin-Otter. This aircraft is equipped with electrothermal anti-icers on the propellers, engine inlets, and windshield. Pneumatic de-icer boots are located on the wing outboard of the engine nacelles, on both the horizontal and vertical stabilizers, on the wing struts, and on the rear landing gear struts. The pneumatic de-icers located on the vertical stabilizers, wing struts, and landing gear struts are nonstandard items that

provide additional research capability for measuring component drag through selective deicing. The aircraft is equipped with several standard instruments for measuring icing cloud properties.

Wing leading edge ice shapes are measured in flight with a stereo photography system. Wing section drag is measured with a wake survey probe mounted on the wing behind the region where the stereo photos are taken. A noseboom is used to measure airspeed, angle-of-attack, and sideslip. A flight test system measures flight dynamics along a flight path. The system includes a data acquisition system and an inertial package that contains rate gyros, directional gyros, and servo accelerometers.

Subsection 1.4.1 describes the calculation of ice shapes on the lifting surfaces of this aircraft, and subsection 1.4.2 describes the calculation of aerodynamic performance characteristics of this aircraft, including the icing effects. In both subsections calculated results are presented and compared with experimental data to demonstrate the efficiency and accuracy of each calculation method.

1.4.1 Prediction of Ice Shapes

The prediction of ice shapes requires knowledge of aerodynamics and also of the physics of ice accretion. The nature of the ice depends on meteorological parameters; glaze ice is formed at temperatures slightly below freezing and rime ice is increasingly superimposed as the temperature decreases. Glaze ice is characterized by its buildup to large dimensions, as for example in the familiar "horns" (Fig. 1.26a), and rime ice by its opaque appearance with fine-grained surface, see Fig. 1.26b. The shape of the ice changes with time and is influenced by the nature of the water droplets and ice particles which impinge on the body or are carried past it, depending upon their size and the flow properties which, in turn, depend upon the shape of the body. The resulting impingement, coalescence and accretion depend on the temperatures of the surface and of the discrete and

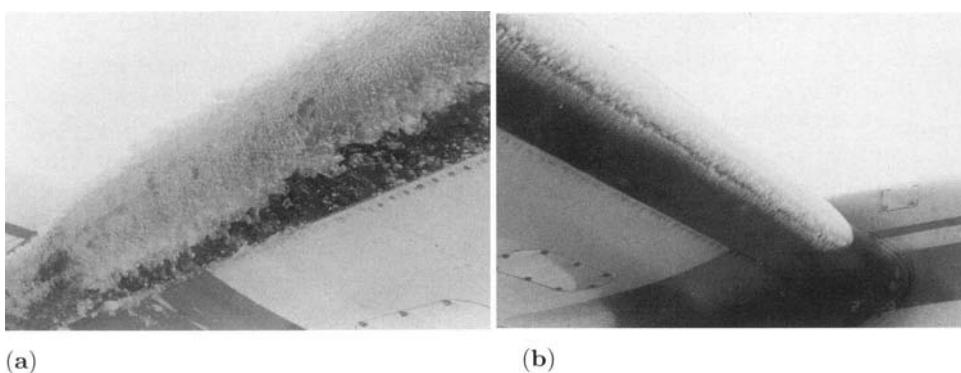


Fig. 1.26. (a) Glaze and (b) rime ice shapes.

continuous phases; the process involves convective heat transfer which is again linked to the shape of the body and the nature of the flow around it. It is evident, therefore, that a mathematical description of the phenomenon of ice formation and its effects on lift and drag requires the solution of time-dependent equations, albeit with comparatively large time scales, with consideration of conservation of mass, momentum and thermal energy and with a model to represent the ice accretion process. In addition, since the flow is turbulent under practically all conditions, the solution of the conservation equations discussed in Chapter 2 requires a closure model with roughness effects. As discussed in Section 8.3, our understanding of turbulent flows on surfaces with roughness and our ability to model them is rather limited, even for geometries less complicated than those with ice. For these reasons, prediction of ice accretion is not just a matter of solving the known conservation equations, as discussed in [23].

A popular and useful computer code for computing ice accretion on single airfoils is the LEWICE code described in [24]. This code has three main modules, as shown in Fig. 1.27. The ice accretion is computed on the airfoil leading edge as a function of time with user specified time intervals. At each given time, the flowfield is determined from a panel code (similar to the one discussed in Section 6.4) so that trajectory and heat transfer calculations can be performed. As ice accretion increases, its shape may become ragged, especially in the case of glaze ice which is characterized by horns, and a rough, irregular surface may develop which leads to higher aerodynamic losses, unlike rime ice. Surface irregularities of the ice shape can lead to multiple stagnation points that increase the difficulty of numerical calculations, including a breakdown of the trajectory calculations.

The automated smoothing procedure of [25] overcomes this difficulty by reducing the amplitude of the surface irregularities without loss of important flow characteristics; this smoothing procedure usually allows the calculations to be performed for greater time intervals than before, without the problems caused by multiple stagnation points.

The flowfield needed to determine the water droplet trajectories is obtained from a panel method similar to the one discussed in Section 6.4. The ice shape is determined from a quasi-steady-state surface heat transfer analysis in which mass and energy equations are solved.

The LEWICE code does a good job of predicting ice shapes on airfoils, especially those corresponding to rime ice (Fig. 1.28a). This is despite the very empirical nature of the expressions used in the heat balance as well as the

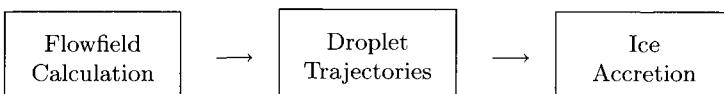


Fig. 1.27. Structure of LEWICE.

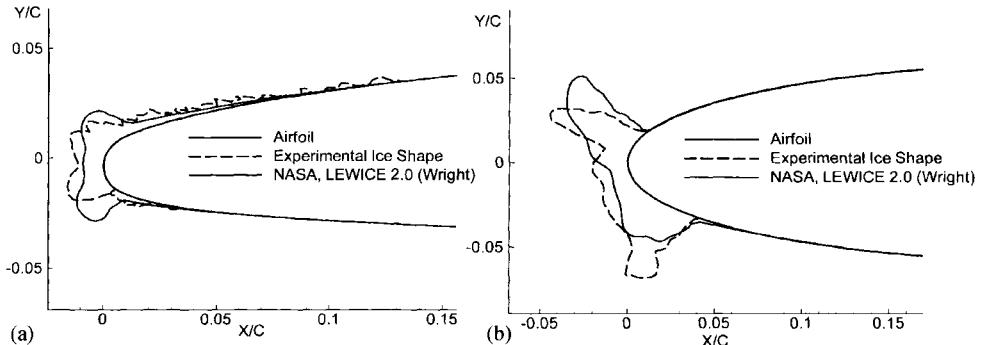


Fig. 1.28. (a) LEWICE prediction of rime ice and (b) LEWICE prediction of glaze ice.

physical model of the ice accretion process. In order to improve our ability to predict ice shapes corresponding to glaze ice (Fig. 1.28b), it is necessary to (a) improve the physical model, and (b) compute the heat balance analysis more accurately. Further details are provided in [5].

1.4.2 Prediction of Aerodynamic Performance Characteristics

The aerodynamic performance characteristics of two- and three-dimensional bodies can be predicted by either using a Navier–Stokes method (see [22] for example) or an interactive boundary-layer method (see Chapter 7, [5]) in which the solutions of inviscid and viscous flow equations are obtained interactively. Both approaches have merit when applied to airfoils, wings, wing-fuselage and high-lift systems. Whereas the Navier–Stokes approach offers generality, it is very computer intensive, requiring considerable run times. Since a viable design method probably will require the evaluation of many flow conditions, cost is a major consideration. Furthermore, if attention is focused on predicting the performance degradation of an aircraft under icing conditions, and consideration is given to the approximations made in formulating the calculation strategy for ice shapes and turbulence modeling of flows with iced shapes, the proper approach becomes clear.

Here the panel method developed by Hess [4, 7], is used to compute the flow-field about the aircraft. Being a panel method and based on the solution of the conservation equations without viscous effects, this method does not produce an accurate prediction of the flowfield about the aircraft, but its accuracy can be improved by incorporating viscous effects with the procedures discussed in [5]. This method is a very useful engineering tool and the workhorse of computational methods in industry for aircraft design (Section 1.2). Performing similar calculations accurately with a Navier–Stokes method is not yet economical.

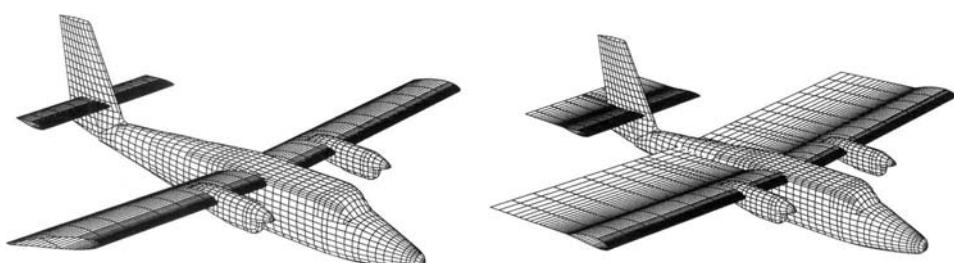
Since an inviscid method cannot predict the viscous drag, and since the viscous effects can reduce the lift of the aircraft and its components, it is common

Table 1.3. Atmospheric Icing Conditions For Twin Otter Tests.

Flight No.	Type of Ice	Pressure (pa)	Speed (ms^{-1})	Duration of encounter (min)	Static temp. (K)	Liquid water content (gm^{-3})	Medium particle diameter (μm)	Unit length roughness parameter
85-17	Rime	88150	59.72	65	261.50	0.22	12.4	5.764
83-11	Rime	84000	71.08	45	262.40	0.29	13.0	7.078
85-24a	Mixed	79600	71.38	15	258.30	0.45	19.5	7.814
85-24b	Mixed	79600	71.94	20	258.70	0.46	15.1	7.813
84-29	Mixed	79600	75.55	49	266.15	0.15	14.6	6.722
84-34	Mixed	82000	70.57	22	266.65	0.58	10.1	12.109
84-27	Glaze	73000	70.57	25	267.95	0.34	15.0	8.708
83-10	Glaze	85500	70.18	26	269.15	0.31	13.0	8.852

to combine viscous effects with an inviscid method, to improve the accuracy of the flowfield calculations (lift) and calculate the viscous drag. Here an interactive boundary-layer method developed for clean and iced airfoils [5] is applied to the lifting surfaces (wing and tail) of the aircraft with a strip-theory approximation [23]. In this method, the inviscid-flow equations are solved for three-dimensional flows by the panel method of Hess [4] and the two-dimensional boundary-layer equations are solved in inverse form with Keller's box method [5].

The icing conditions considered in [23] are given in Table 1.3. The computed ice shapes for the wing correspond to the section where experimental results were available, which was at 69% of the wing semi-span. Computed ice shapes for the tail are shown for 45% of the tail semi-span, though no experimental ice shapes were available for comparison. Since the flowfield is being calculated with a panel method, the Twin Otter was paneled as shown in Fig. 1.29. A total of 11 and 5 lifting strips were taken on the wing and tail, respectively, with 72 and 67 grid points defining each airfoil section.

**Fig. 1.29.** Paneled Twin Otter, (a) without wake, and (b) with wake.

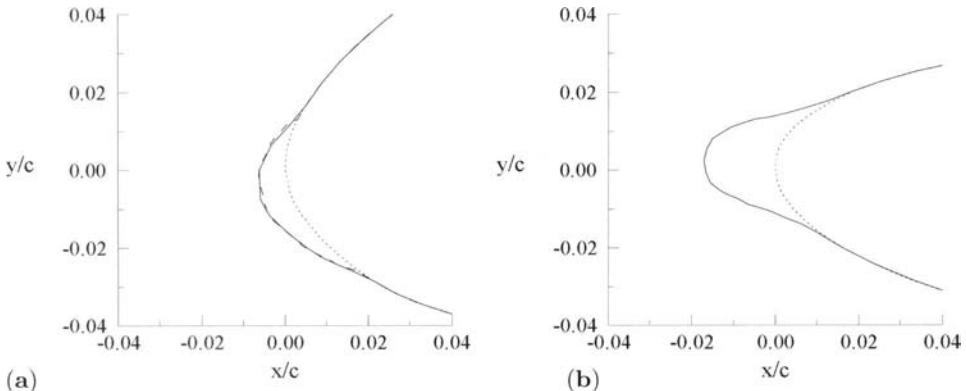


Fig. 1.30. Comparison of computed (solid lines) and experimental (dashed lines) ice shapes for rime ice, flight 85-17: (a) wing, and (b) tail.

Figures 1.30 and 1.31 show a sample of ice shapes reported in [23]. Overall, the computed ice shapes agree well with the experimental data [26–28] in the case of rime ice (flight 85-17). In the case of mixed ice at slightly higher temperatures the calculation method still gives a good prediction of the ice shape (flight 85-24a), although it does not predict the horns observed in flight (flights 85-24b, 84-34). No measured ice shapes were available for comparison under glaze ice conditions. However, it is expected that the measured ice shapes would have exhibited multiple horns, which are difficult to predict.

In [23], studies were first conducted for the “clean” aircraft before studying the effects of ice on the performance degradation of the aircraft for the computed ice shapes on the wing and tail. For this purpose, the Hess panel method and the interactive boundary layer method employing the strip theory approximation were applied to the lifting surfaces of the aircraft at several angles of attack. The

Table 1.4. Aerodynamic Characteristic of the Clean Twin Otter.

Angle of attack (α)	Total aircraft lift coefficient		Section lift coefficient (69% of semi-span)		Section drag coefficient (69% of semi-span)
	Inviscid	Viscous	Inviscid	Viscous	
2°	0.4106	0.3516	0.4664	0.4334	0.00933
4°	0.6158	0.5716	0.6322	0.5910	0.00963
6°	0.8209	0.7615	0.7987	0.7457	0.01000
8°	1.0255	0.9509	0.9636	0.8909	0.01073
10°	1.2292	1.1209	1.1257	1.0238	0.01188
12°	1.4316	1.2405	1.2476	1.1504	0.01307
	1.6318	1.4388	1.4457	1.2719	0.01426

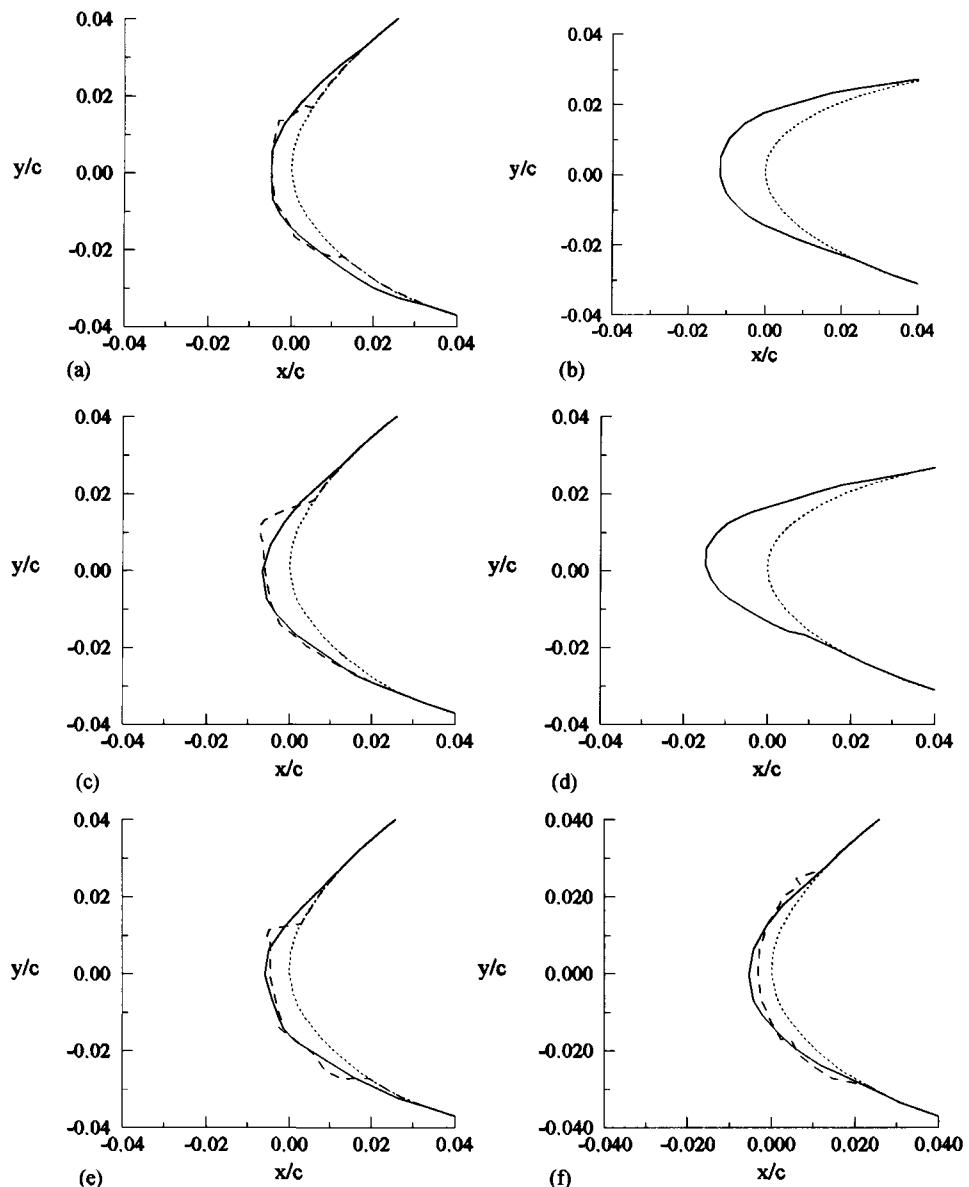


Fig. 1.31. Comparison of computed ice (solid lines) and experimental ice (dashed lines) shapes for mixed ice: (a) wing, flight 85-24a, (b) tail, flight 84-24a, (c) wing, flight 85-24b, (d) tail, flight 85-24b, (e) wing, flight 84-34, and (f) wing, flight 84-29.

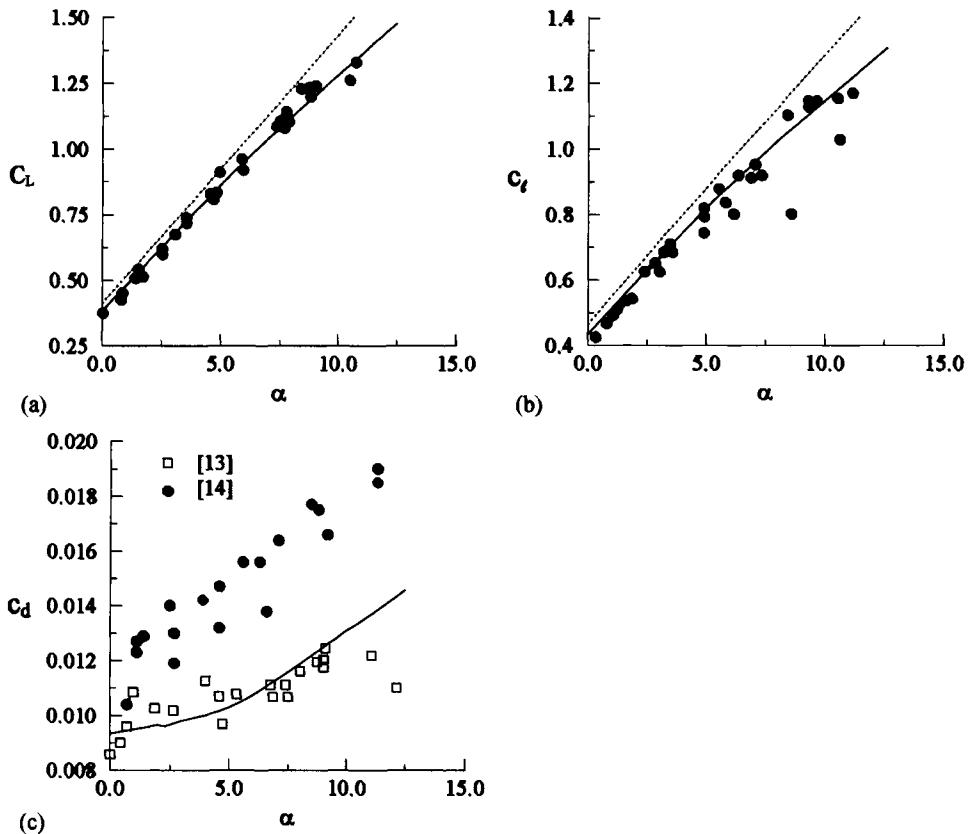


Fig. 1.32. Comparison of calculated (*lines*) and measured (*circles*) (a) total lift coefficient, (b) section lift coefficient, and (c) section drag coefficient. Dotted lines denote inviscid results and continuous lines results with viscous effects.

Reynolds number was specified for cruise at an altitude of 7600 ft and speed of 70 m/s.

Table 1.4 shows the calculated aerodynamic characteristics of the clean aircraft with and without viscous effects, and Fig. 1.32 shows a comparison between the computed and experimental total lift, section lift and drag coefficients of the airplane. It can be seen that the introduction of the viscous effects on the lifting surfaces substantially lowers the lift. As expected, the lift curves are well predicted, but the computed section drag coefficient is too low, partly, at least, because the drag generated by the natural wing roughness (rivets, deicing boots, etc.) is not accounted for. The experimental data related to the lift was obtained from [26] and the drag data from [27]. The drag data from [26] were also reported, although they appear to be inconsistent.

Figure 1.33 shows a comparison between the total airplane lift coefficients with icing conditions corresponding to (1) rime ice of 45 minute accumulation,

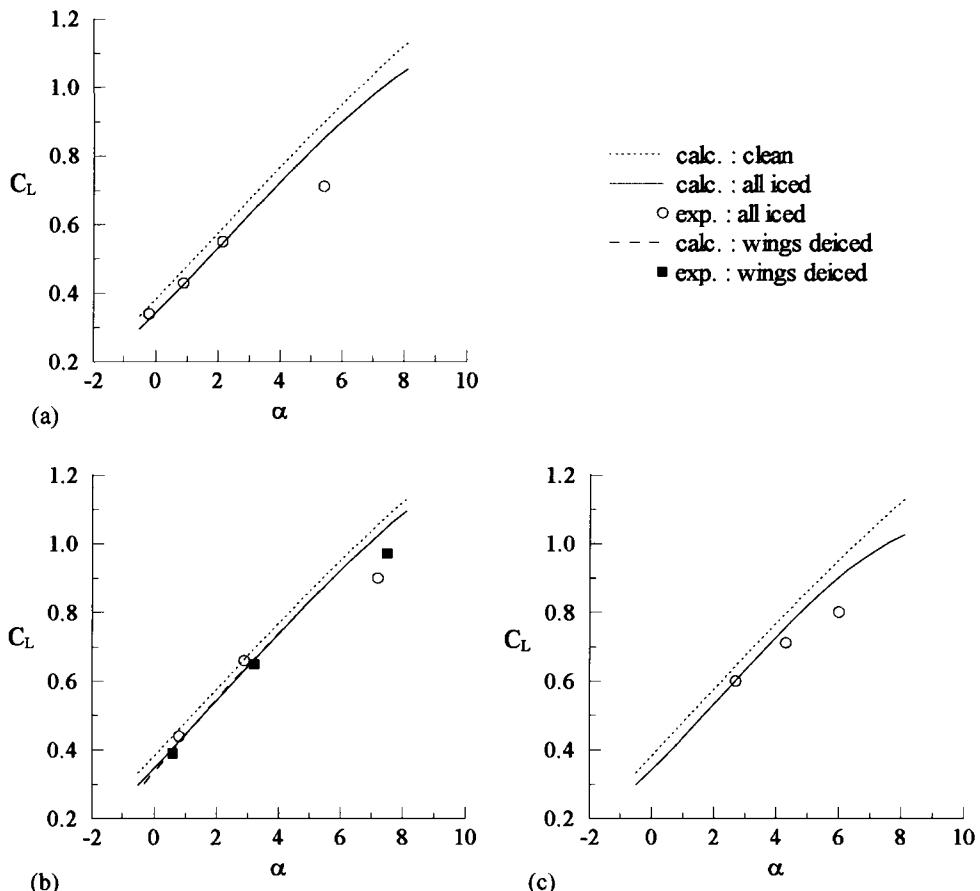


Fig. 1.33. Comparison of calculated and measured airplane lift coefficient for (a) rime ice (flight 83-11), (b) mixed ice (flight 84-34), and (c) glaze ice (flight 84-27).

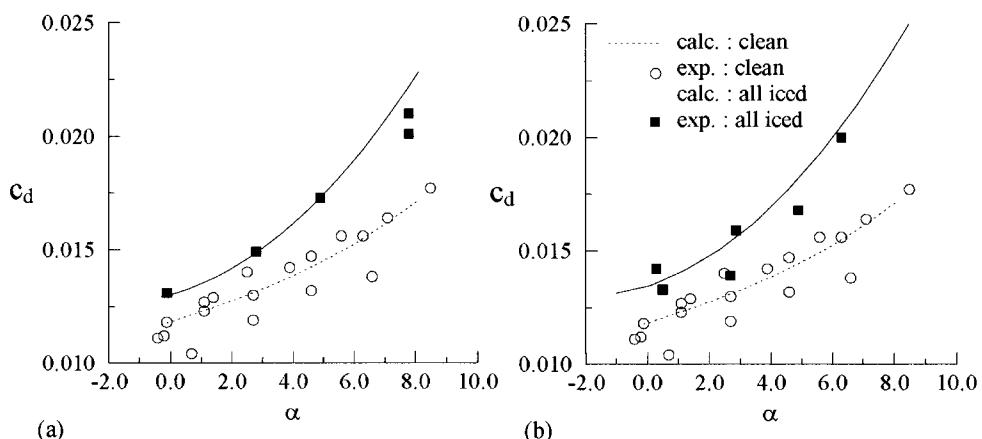


Fig. 1.34. Increase in section drag coefficient due to (a) rime ice (flight 85-17), and (b) mixed ice (flight 85-24a).

flight 83-11, (b) mixed ice of 22 minute accumulation, flight 84-34, and (c) glaze ice of 25 minute accumulation, flight 84-27. The comparisons also include the airplane lift coefficient with the clean wing and tail. The results in Fig. 1.33b also show the calculations for an iced airplane with the wings deiced. In this case (deiced wing), all the ice accumulation between the wing tip and the propeller was removed (deiced) but the wing ice between the propeller and the fuselage was not removed. The calculations for this case were performed for a clean wing between the wing tip and the propeller and for an iced wing between the propeller and fuselage. The tail calculations were performed for an all iced-tail conditions. Whereas the results are good at lower angles of attack, they are not so good at higher angles. This may be due to the ice shape with horns. Figure 1.34 shows a comparison between clean and iced section drag coefficients obtained at 69% wing semi-span. The calculated drag coefficients have been “corrected” such that the clean wing matches the experimental data. Figure 1.34a is for rime ice accumulation of 65 minutes and Fig. 1.34b is for mixed ice accumulation of 15 minutes. Both figures clearly show the drag increase due to ice on the wing.

1.5 Aerodynamics of Ground-Based Vehicles

In recent years, CFD has been increasingly utilized in the automobile industry to reduce the time required to develop new products. As described in detail in [29], the ability of a company to quickly react to the ever-changing needs of the market must be given an even higher priority than simply cutting costs, even though cost minimization is also very important. As a result, all computational methods must satisfy the following two conditions; the first condition is necessary, and the second is sufficient.

1. The method must reproduce the related physics with adequate accuracy, and
2. The method must yield results faster than conducting an experiment.

At this time, however, the advances made in applying CFD to reducing product development time in the automobile industry are somewhat limited, and CFD is still more a subject of research than a practical development tool.

Before some of the applications of CFD to ground-based vehicles are described in the subsection that follows, a brief overview of the aerodynamic problems associated with ground-based vehicles (exemplified by a passenger car) is presented below.

Aerodynamics affects vehicles on the ground in two ways: (1) fuel economy and (2) the stability and controllability of the vehicle. The main difference between airborne and landborne vehicle aerodynamics is that, for an airborne vehicle the oncoming flow is essentially in the axial direction, whereas for a landborne vehicle, the relative wind is not necessarily aligned with the path of

the vehicle. There are, of course, exceptions; this is obvious from the flight of VSTOL-aircraft and helicopters.

The influence of aerodynamic forces depends strongly on the vehicle speed and weight. For example, automobile aerodynamics is not of great concern in traffic on city streets, and a heavy train operator usually ignores the side wind effects. While the aerodynamics of motor vehicles has been investigated for many years, it did not attain prominence until improved roads allowed for greater speeds and fuel crises inspired demand for improved fuel economy. Early attempts to streamline vehicles were mostly based on experience gained from aircraft aerodynamics, and the resulting designs were not always practical or accepted by the general public. As early as the 1920's, it was demonstrated that a drag coefficient of 0.15 was attainable under ideal conditions, which should be compared to the then-prevalent box-design drag coefficient of about 0.8. Improvements have been slow in coming, and the drag coefficient of post-World War II automobiles remained around 0.5 until the fuel crises in the 1970's. As some critics claim, the reduction from 0.8 to around 0.5 was more due to styling than conscious aerodynamic development. Since that time, however, drag coefficients have been reduced to around 0.30 by systematic attention to aerodynamic details, and there is promise for further improvements. Automobile aerodynamics, however, will always be subject to constraints imposed by utility, styling, and public acceptance.

The principal tool used to study automobile aerodynamics has been the wind tunnel. Testing began with small-scale models in aeronautical facilities and has evolved into the use of special full-scale wind tunnels run by the larger automobile manufacturing companies. Testing in a wind tunnel creates its own problems because the boundary layer on the ground plane interferes with the simulation of the actual flow conditions. Several remedies have been proposed such as reflection models, tangential blowing, a moving ground plane, etc., of which the moving ground plane provides the best correlations with road tests. Since flow details underneath the vehicle; and in the wheel well are related to drag, a refined test may include provisions for spinning the wheels, which adds a further complication.

In general, the drag of a typical passenger automobile is essentially pressure drag or is due to local flow separation. For this reason, the shape of the sharp edges from which the flow separates has a definite effect on drag. Reference [29] gives an example of a "detail-optimization" which reduced the drag coefficient from 0.48 to 0.32 without noticeable changes in the appearance of the automobile. Since an automobile is basically a blunt body, flow around its longitudinal edges sets up vortical flow and causes vortex drag that is not necessarily associated with lift or induced drag, although well-rounded shapes resembling half bodies have considerable lift and consequently induced drag. The flow underneath the car has a tendency to diverge to the sides, creating low pressure

and down load on the front and local flow angularity with respect to the front wheels and wheel wells, which results in a high drag unless this flow is damped or diverted. Drag considerations also apply to the internal flow system such as the radiator and air conditioning unit; this will require more attention as the automobiles become more compact.

Under actual driving conditions, one encounters natural wind, that is not usually in the direction of the vehicle motion. This results in an asymmetric flow with respect to the vehicle. Viewed from above, the relative wind angle corresponds to what is known in aerodynamics as the angle of sideslip; this angle results in force components in the direction of the wing and normal to it. Resolution of these components in the car axis system results in a drag component which usually increases with the yaw angle and a side force. The side force manifests itself in a negative yawing moment about the center of gravity of the car, which tends to increase the yaw angle. The cross-flow sensitivity caused by the yawing moment is somewhat reduced by the forward movement of the center of gravity in current front-wheel-drive models. However, as vehicles become lighter to obtain better fuel economy, the problem is bound to reemerge. Simulation of the natural wind profile in a wind tunnel is extremely difficult because the wind speed most likely is not constant and the wind profile is affected by terrain features or buildings.

1.5.1 Applications of CFD to Automobiles

As discussed in [29], there is a large effort underway in applying CFD to road vehicles with different degrees of sophistication. The simplest approach is to use panel methods (Section 6.4, [5]) and calculate the inviscid flow around the body. Even though this approach does not provide flow separation, vortex flow and drag prediction, it can be useful for pin-pointing possible trouble areas such as strong pressure gradients and ground effects on the velocity field. This approach is more suitable to study generic models than to obtain detailed information on a given design. The next degree of sophistication of the CFD approach is to perform inviscid flow calculations with vortex wakes added to the panel method. However, in order to predict flow separation locations and the initial vortex strength, boundary-layer calculations (Chapter 7) must be performed. Provided that the body is relatively smooth, that is, local protuberances and gaps are suppressed, this improved panel-boundary-layer approach shows definite promise [29].

The next degree of sophistication of the CFD approach is to solve the Reynolds-time-averaged Navier–Stokes (RANS) equations. The success of this approach varies; for example, sometimes drag is predicted accurately but the pressure distribution does not agree well with experiments or vice-versa. Some of the discrepancies are due to inadequate meshing which results from efforts to reduce computing time. However, in general they are due to the selection

of turbulence models (Chapter 3, [15, 16]). Much work remains to be done in the calibration and tailoring of turbulence models for vehicle application before results of consistent accuracy can be obtained.

The direct numerical simulation (DNS) is the next approach employed in the application of CFD to vehicle aerodynamics (see Fig. 1.35). The accuracy of the results obtained with this approach is about the accuracy of the results obtained with the RANS approach; that is, drag prediction is within five percent. With almost 10^6 grid points for a half model, DNS has been able to discriminate the effect of several aerodynamic devices (spoilers, flaps) on the drag and lift of a sports car (Fig. 1.36). Grid generation (Chapter 9) is said to require only three days, and CPU-time for a single configuration requires between 10 to 20 hours on a supercomputer.

The other approach used in applying CFD to vehicle aerodynamics is to use zonal methods in which the near field calculations performed with the Navier–Stokes equations are patched to the rest of the flowfield calculations obtained with inviscid flow and boundary-layer equations.

CFD is also useful in calculating internal flows such as in ducts connected to the radiator and air conditioning units. Another interesting area is the ventilation and heat balance of the passenger compartment. The internal flow calculations are not in general performed to the same accuracy required in the drag calculations, but the requirements for these calculations cannot be relaxed too much since the internal flow produces its own drag component.

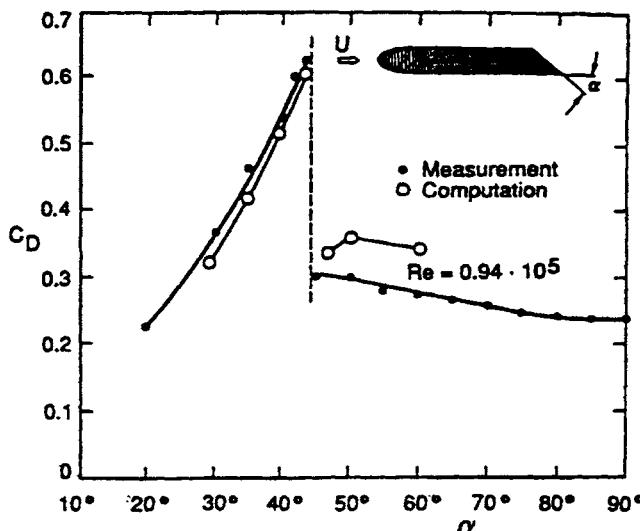


Fig. 1.35. Drag versus rear slant angle α computed with a DNS code and compared to measurements at the same Reynolds number [29].

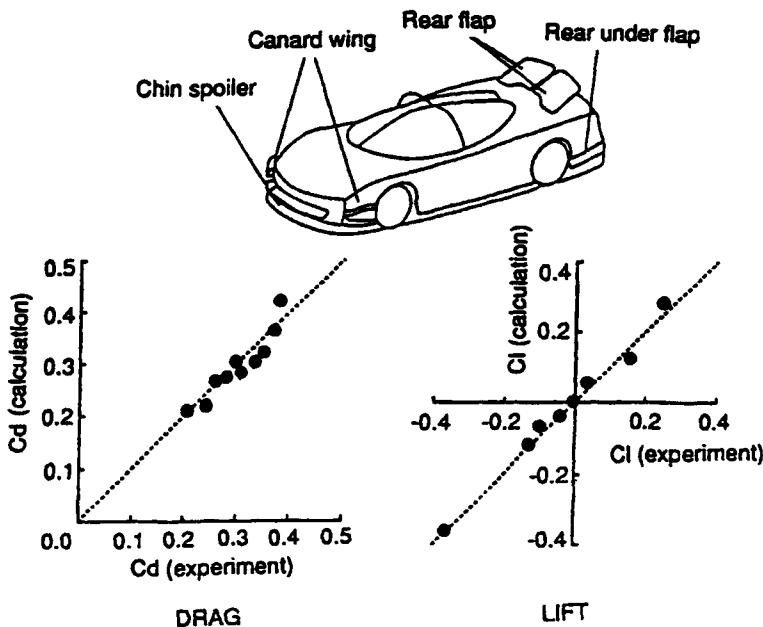


Fig. 1.36. Comparison of computed (DNS) and measured (symbols) drag and lift coefficients for a sports car with various aerodynamic devices [29].

In summary, if one considers that the requirement for accuracy in drag prediction of cars is within 0.5 percent and that the best CFD can produce is within 5 percent, it may appear that CFD has a long way to go in achieving this goal. However, the picture is not as bad as it appears because the other aspects of car aerodynamics do not have to be predicted as closely as drag, which is one of the main factors contributing to fuel consumption and amounts to official 46 percent of the total for a midsize U.S. car design on a highway. The computing times associated with the obtainable accuracy are still excessive in comparison with times required for wind-tunnel testing. It should be pointed out that the previously cited computing time is only to obtain one data point. Looking ahead, it is likely that these computer times will be reduced drastically as massively parallel computers become available. It is likely that the problems with long CPU-time and the limitations in the number of panels will eventually be reduced to a nuisance level. It is also clear that with more computer power becoming available, there is going to be more emphasis placed on improving physical modeling of turbulence, since understanding this process is essential to obtaining more accurate results.

References

- [1] Bushnell, D.M. and Hefner, J.N., *Viscous Drag Reduction*. AIAA "Progress in Aeronautics and Astronautics" Series, 1989.
- [2] Cebeci, T. and Cousteix, J., *Modeling and Computation of Boundary-Layer Flows*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 1998.
- [3] Cebeci, T., *Stability and Transition: Theory and Application*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 2004.
- [4] Hess, J.L., "The Problem of Three-Dimensional Lifting Flow and Its Solution by Means of Surface Singularity Distribution," Computer Methods in Applied Mechanics and Engineering, Vol. 4, pp. 283–319, 1974.
- [5] Cebeci, T., *An Engineering Approach to the Calculation of Aerodynamic Flows*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 1999.
- [6] Valarezo, W.O. and Chin, V.D., "Maximum Lift Prediction for Multielement Wings," AIAA Paper No. 92-0401, Jan 1992.
- [7] Hess, J.L., Friedman, D.M. and Clark, R.W., "Calculation of Compressible Flow About Three-Dimensional Inlets with Auxiliary Inlets, Slats, and Vanes by Means of a Panel Method," McDonnell Douglas Report No. MDC J3789, Jun 1985 (Also, AIAA Paper No. 85-1196 and NASA CR-174975).
- [8] Lovell, D.A., "A Wind-Tunnel Investigation of the Effects of Flap Span and Deflection Angle, Wing Planform and a Body on the High-Lift Performance of a 28° Swept Wing". CP No. 1372, 1977.
- [9] Kafyeke, F., Boyce, F., Kho, C., "Investigation of Airfoil and Wing Performance with Loading Edge Contamination", Proceedings of the CASI 8th Aerodynamics Symposium, Toronto, Canada, April 30th to May 2nd, 2001.
- [10] "Fluid Dynamics Panel Working Group 04 on Experimental Data Base for Computers Program Assessment", AGARD Advisory Report No. 138, July 1984.
- [11] Valarezo, W.O., "Maximum Lift Degradation Due to Wing Upper Surface Contamination." Proceedings of *The First Bombardier International Workshop on Aircraft Icing, Boundary-Layer Stability and Transition*, Montréal, Quebec, Canada, 20th–21st Sept 1993.
- [12] Rogers, S.E., Roth, K., Cao, H.V., Slotnick, J.P.. Whitlock, M., Nash, S.M., Baker, M.D., "Computations of viscous flow for a Boeing 777 aircraft in landing configuration", Journal of Aircraft 48(6), 1060–1068, 2001.
- [13] Langlois, M., Mokhtarian, F., Kafyeke, F., "Navier–Stokes Prediction of Aircraft High-Lift Characteristics", Proceedings of the CASI 9th Aerodynamics Symposium, Montréal, Canada, April 2003.
- [14] Mavriplis, D.J., "Turbulent Flow Calculations using Unstructured and Adaptive Meshes", Int. J. for Numerical Methods in Fluids 13(9), 1131–1152, 1991.
- [15] Cebeci, T., *Analysis of Turbulent Flows*, Elsevier Science, London, 2004.
- [16] Cebeci, T., *Turbulence Models and Their Application*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 2003.
- [17] Kafyeke, F., Mavriplis, F., "CFD for the Aerodynamic Design of Bombardier's Global Express High Performance Jet", AIAA Paper 97-2269, June 1997.
- [18] Kafyeke, F., Piperni, P., Robin, S., "Applications of KTRAN Transonic Small Disturbance Code to the Challenger Business Jet Configuration with Winglets", SAE Paper 881483, October 1988.
- [19] Kafyeke, F., Piperni, P., Mokhtarian, F., Djilali, N., "Development of a Transonic Euler Method for Complete Aircraft Configurations", Proceedings, 3rd Canadian Symposium on Aerodynamics, Nov. 1991.

- [20] Piperni, P., "Multi-Block Grid Generation with CAD-Based Domain Decomposition", Proceedings of the International Conference on Numerical Grid Generation in CFD and Related Fields", 1994, pp. 109–121.
- [21] *Federal Aviation Regulations Parts 25, 27 and 29*. "Airworthiness Standards: Transportation Category Airplanes, Normal Category Rotorcraft, and Transport Category Rotorcraft," Appendix C, Washington, D.C. 20591, Department of Transportation Federal Aviation Admin, 1914.
- [22] Reinmann, J.J., "NASA Aircraft Icing Technoloqy Program," NASA Technical Memorandum 104518, Dec 1991.
- [23] Cebeci, T. and Besnard, E., "Prediction of the Aerodynamic Performance Degradation of an Aircraft in Natural Icing Conditions," AIAA Paper No. 94-0487, Jan 1994.
- [24] Ruff, G.A. and Berkowitz, B.M., "User's Manual for the NASA Lewis Ice Accretion Code (LEWICE)," NASA CR-185129, May 1990.
- [25] Cebeci, T., Chen, H.H. and Alemdaroglu, N., "Fortified LEWICE with Viscous Effects," Journal of Aircraft, Vol. 18, No. 9, pp. 564–571, Sept 1991.
- [26] Ramando, R.J., Mikkelson, K.L., McKnight, R.C., "Performance Degradation of a Typical Twin Engine Commuter Type Aircraft in Measured Natural Icing Conditions," NASA Technical Memorandum 83564, January 1984.
- [27] Mikkelson, K.L., McKnight, R.C., Ramando, R.J., Perkins, P.J., "Icing Flight Research: Aerodynamic Effects of Ice and Ice Shape Documentation with Stereo Photography," NASA Technical Memorandum 86906, AIAA Paper 85-0468, January 1985.
- [28] Mikkelson, K.L., Juhasz, N., Ramando, R.J., McKnight, R.C., Freedman, R., Greishing, J., "In-flight Measurements of Wing Ice Shapes and Wing Section Drag Increase Caused by Natural Icing Conditions," NASA Technical Memorandum 87301, April 1986.
- [29] Hucho, W.F. and Sovran, G., "Aerodynamics of Road Vehicles," Annu. Rev. Fluid Mech., 1993, Vol. 25, pp. 485–537.

2 Conservation Equations

2.1 Introduction

This chapter considers the conservation equations upon which the structure of fluid mechanics has been built. We assume that all significant aspects of an incompressible or compressible flow can be adequately described by the solutions of the conservation equations known as the Navier–Stokes equations. No serious objection to this principle has ever been advanced, and the validity of these equations has been established in so many instances that we may regard as an act of faith and have full confidence in it.

The Navier–Stokes equations are based on the principles of conservation of mass, momentum and energy and are presented in the following section for three-dimensional flows in both differential and integral forms. Detailed derivations can be found in various textbooks such as [1–5], so Section 2.2 does not attempt to provide detailed derivations. This brief presentation emphasizes the equations formulated in both differential and integral forms. Their representation in vector form is also given.

Since most flows are turbulent with fluctuations of pressure, temperature and velocity over a wide range of frequencies, the solution of the Navier–Stokes equations of Section 2.2 presents a formidable challenge which has so far not been met in a wholly satisfactory way and is unlikely to be achieved for the boundary conditions of real engineering flows in the foreseeable future. As a consequence, it is common practice to average the equations so that the equations lose their time dependence. The resulting equations include correlations of fluctuation terms, as discussed in Section 2.3, and these require the assumptions described in Chapter 3. The time-averaged equations, usually called Reynolds Averaged Navier–Stokes (RANS) equations, are approximate representations of flows, and the proper application of the numerical solutions requires that one be familiar with the assumptions and approximations that have been made and their effect on the accuracy of the numerical solutions.

Depending on the flow conditions, it is appropriate and sometimes necessary to use the reduced forms of the Navier–Stokes equations as discussed in Section 2.4. These simplified equations reduce the complexity of solving the full Navier–Stokes equations, provide substantial savings on computer time, and in some situations permit accurate analytical and numerical solutions to the conservation equations.

The prediction of transition from laminar to turbulent flow remains one of the unsolved problems of fluid mechanics. The subject is important in many applications, as discussed in Chapter 1. The only approach that may provide a general prediction method in the near future is the e^n -method based on the solution of the linearized stability equations. In Section 2.5 we discuss the derivation of the stability equations, while discussing their solutions and the e^n -method later in Chapter 8.

The Navier–Stokes equations and their reduced forms are partial-differential equations. Before using numerical methods to solve these equations, one must know whether they are hyperbolic, elliptic or parabolic. In Section 2.6 we discuss the classification of the partial-differential equations, and the general principles one must consider to solve each class of equations subject to the boundary conditions discussed in Section 2.7.

Except for Chapters 10 and 12 in this book, the treatment of the computational fluid dynamics equations are for incompressible flows. For this reason, the derivations and discussions will concentrate mostly on incompressible flows. In some instances, and whenever appropriate and necessary, the conservation equations for compressible flows will also be given in order to pave the way for the discussion of the numerical solution of Euler and Navier–Stokes equations for compressible flows in Chapters 10 and 12.

2.2 Navier–Stokes Equations

The Navier–Stokes equations may be obtained by using infinitesimal or finite control volume approaches, and the governing equations can be expressed in differential or integral forms. In subsection 2.2.1 these equations are presented in differential form, and in subsection 2.2.2 in integral form. For a detailed discussion of the derivation of these equations in either form, the reader is referred to Anderson [4, 5].

2.2.1 Navier–Stokes Equations: Differential Form

The Navier–Stokes equations in differential form can be derived by using an infinitesimal control volume either fixed in space with the fluid moving through it or moving along a streamline with a velocity vector $\underline{V} = (u, v, w)^T$ equal to the flow velocity at each point. Here we follow the second choice, discussed in [2],

and write the Navier–Stokes equations for an incompressible three-dimensional flow as follows:

Continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (2.2.1)$$

x -component of the momentum equation

$$\varrho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) + \varrho f_x \quad (2.2.2)$$

y -component of the momentum equation

$$\varrho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + \left(\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \right) + \varrho f_y \quad (2.2.3)$$

z -component of the momentum equation

$$\varrho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + \left(\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \right) + \varrho f_z \quad (2.2.4)$$

where D/Dt represents the substantial derivative given by

$$\frac{D(\)}{Dt} = \frac{\partial(\)}{\partial t} + u \frac{\partial(\)}{\partial x} + v \frac{\partial(\)}{\partial y} + w \frac{\partial(\)}{\partial z} = \frac{\partial(\)}{\partial t} + \mathbf{V} \cdot \nabla(\) \quad (2.2.5)$$

Equations (2.2.2) to (2.2.4) make use of Newton's second law of motion with their left-hand sides representing mass acceleration per unit volume and their right-hand sides representing the sum of net forces per unit volume acting on the fluid which consists of surface and body forces. Surface forces arise because of molecular stresses in the fluid (such as pressure, p , which is present in a fluid at rest and acts normal to a surface) and viscous stresses which act normal to a surface or tangentially (shear stress). The first term on the right-hand side of Eqs. (2.2.2)–(2.2.4) denotes the net pressure force per unit volume and the minus sign arises because, by definition, a positive pressure acts inward. The

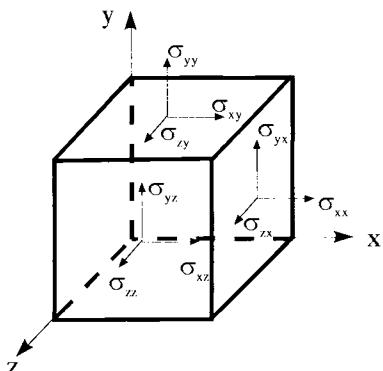


Fig. 2.1. Definitions of viscous stress components applied to the faces of a control volume by the surrounding fluid. Force components are stress components multiplied by areas of corresponding faces.

second, third and fourth terms denote the viscous forces per unit volume, and they arise as a result of the different components of normal and shear stresses shown in Fig. 2.1: the first subscript to the symbol σ represents the direction of the stress and the second the direction of the surface normal. By convention, an outward normal stress acting on the fluid in the control volume is positive, and the shear stresses are taken as positive on the faces furthest from the origin of the coordinates. Thus σ_{xy} acts in the positive x direction on the visible (upper) face perpendicular to the y axis; a corresponding shear stress acts in the *negative* x direction on the invisible lower face perpendicular to the y axis.

Sometimes it is more convenient to write the viscous terms in the momentum equations in tensor notation as

$$\frac{\partial \sigma_{ij}}{\partial x_j} \quad (2.2.6)$$

with $i, j = 1, 2, 3$ for three-dimensional flows: for example, $i = 1, j = 1, 2, 3$ for Eq. (2.2.2). For a constant density “Newtonian” viscous fluid, the normal viscous stresses σ_{ij} ($i = j$) and shear stresses σ_{ij} ($i \neq j$) are obtained from the viscous stress tensor given by

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.2.7)$$

According to Eq. (2.2.7), the normal viscous stress σ_{xx} and the shear stresses σ_{xy} and σ_{xz} in Eq. (2.2.2) are given by

$$\sigma_{xx} = 2\mu \frac{\partial u}{\partial x}, \quad \sigma_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \quad \sigma_{xz} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \quad (2.2.8)$$

with similar expressions for the viscous stress tensor terms in Eqs. (2.2.3) and (2.2.4).

In terms of Eq. (2.2.7), the Navier–Stokes equations can be simplified considerably so that, for example, the x -momentum equation, Eq. (2.2.2) for a Newtonian fluid becomes

$$\frac{Du}{Dt} = -\frac{1}{\varrho} \frac{\partial p}{\partial x} + \nu \nabla^2 u + f_x \quad (2.2.9)$$

with similar expressions for the y - and z -components obtained from Eqs. (2.2.3) and (2.2.4). The resulting equations can be written in vector form as

$$\frac{D\vec{V}}{Dt} = -\frac{1}{\varrho} \nabla p + \nu \nabla^2 \vec{V} + \vec{f} \quad (2.2.10)$$

with ∇^2 denoting the Laplacian operator

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

The equation representing conservation of energy has a form similar to that of the momentum equations (2.2.2) to (2.2.4). It is based on the first law of

thermodynamics and states that the increase at energy in a system (i.e. control volume) is equal to heat added to the system plus the work done on the system. For incompressible flows, the work done on the system is negligible, and the energy equation may be written as

$$\varrho c_p \frac{DT}{Dt} = - \left(\frac{\partial \dot{q}_x}{\partial x} + \frac{\partial \dot{q}_y}{\partial y} + \frac{\partial \dot{q}_z}{\partial z} \right) + \dot{q}_h \quad (2.2.11)$$

where \dot{q} is the conduction heat transfer rate per unit area in the three orthogonal directions. \dot{q}_h is the heat “source” (i.e. radiation, chemical reactions) and c_p is the specific heat. The conduction heat-transfer terms may be written in the form

$$k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right)$$

where k is the constant thermal conductivity.

Compressible Flows

For compressible flows, the Navier–Stokes equations are similar to those given by Eqs. (2.2.1) to (2.2.4) for incompressible flows. Since the fluid properties now also vary with temperature, the continuity and momentum equations are coupled to the energy equation, and the solution of the energy equation provides the temperature distribution in the flowfield. These equations are discussed in some detail in several references, see for example [1, 2], and are summarized below for an unsteady compressible three-dimensional flow.

The continuity equation is

$$\frac{\partial \varrho}{\partial t} + \nabla \cdot (\varrho \vec{V}) = 0 \quad (2.2.12a)$$

For a Cartesian coordinate system, it becomes

$$\frac{\partial \varrho}{\partial t} + \frac{\partial}{\partial x}(\varrho u) + \frac{\partial}{\partial y}(\varrho v) + \frac{\partial}{\partial z}(\varrho w) = 0 \quad (2.2.12b)$$

The momentum equations are identical to those given by Eqs. (2.2.2) to (2.2.4) provided that, with δ_{ij} denoting the Kronecker delta function ($\delta_{ij} = 1$, if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$), the viscous stress tensor σ_{ij} is written as

$$\sigma_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \quad (i, j, k = 1, 2, 3) \quad (2.2.13)$$

The energy equation can be written either in terms of total energy per unit volume, E_t ,

$$E_t = \varrho \left(e + \frac{V^2}{2} \right) \quad (2.2.14)$$

as

$$\frac{\partial E_t}{\partial t} + \nabla \cdot E_t V = \dot{q}_h - \nabla \cdot \dot{q} + \nabla \cdot (\Pi_{ij} \cdot V) + \varrho f \cdot V \quad (2.2.15a)$$

or in terms of internal energy per unit mass, e , as

$$\varrho \frac{De}{Dt} + p(\nabla \cdot V) = \dot{q}_h - \nabla \cdot \dot{q} + \sigma_{ij} \frac{\partial u_i}{\partial x_j} \quad (2.2.15b)$$

In the above equations, Π_{ij} represents the stress tensor given by

$$\Pi_{ij} = -p\delta_{ij} + \sigma_{ij} \quad (2.2.16a)$$

and σ_{ij} ($\partial u_i / \partial x_j$) represents the dissipation function Φ given by

$$\begin{aligned} \Phi = \mu & \left[2 \left(\frac{\partial u}{\partial x} \right)^2 + 2 \left(\frac{\partial v}{\partial y} \right)^2 + 2 \left(\frac{\partial w}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right)^2 \right. \\ & \left. + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right)^2 \right] \end{aligned} \quad (2.2.16b)$$

Finally, we need an equation of state for the fluid to relate p , ϱ and e . The commonest example is the perfect gas law

$$p = (\gamma - 1)c_v \varrho T$$

where

$$e = c_v T, \quad \gamma = c_p / c_v$$

with c_v and c_p representing the specific heats in constant volume and pressure, respectively.

The terms on the left-hand side of the energy equation given by Eq. (2.2.15a) represent the rate of increase of total energy in the control volume (per unit volume) and the rate of total energy lost by convection through the control volume (per unit volume), respectively. The first term on the right-hand side of the equation represents the heat produced per unit volume by external agencies, the second term represents the rate of heat lost by conduction through the control volume (per unit volume), and the third and fourth terms represent the work done on the control volume by the surface forces and body forces, respectively (per unit volume).

In the numerical solution of the conservation equations, it is often preferable to express them in “divergence form” or “conservation form” to avoid numerical difficulties that may arise in some flows, such as flows containing shock waves, when the nondivergence form or the nonconservation form of the equations is used. In conservation form the coefficients of the derivative terms can be constant or variable; if variable, their derivatives do not appear in the equation. For example, the continuity equation (2.2.12a) is in the conservation form. However, if it is written as

$$\frac{\partial \varrho}{\partial t} + u \frac{\partial \varrho}{\partial x} + \varrho \frac{\partial u}{\partial x} + v \frac{\partial \varrho}{\partial y} + \varrho \frac{\partial v}{\partial y} + w \frac{\partial \varrho}{\partial z} + \varrho \frac{\partial w}{\partial z} = 0. \quad (2.2.17)$$

it is in nonconservative form.

The momentum equations in conservation form are

$$\frac{\partial \varrho u}{\partial t} + \frac{\partial}{\partial x}(\varrho u^2 + p - \sigma_{xx}) + \frac{\partial}{\partial y}(\varrho uv - \sigma_{xy}) + \frac{\partial}{\partial z}(\varrho uw - \sigma_{xz}) = \varrho f_x \quad (2.2.18)$$

$$\frac{\partial \varrho v}{\partial t} + \frac{\partial}{\partial y}(\varrho uv - \sigma_{yx}) + \frac{\partial}{\partial x}(\varrho v^2 + p - \sigma_{yy}) + \frac{\partial}{\partial z}(\varrho vw - \sigma_{yz}) = \varrho f_y \quad (2.2.19)$$

$$\frac{\partial \varrho w}{\partial t} + \frac{\partial}{\partial x}(\varrho uw - \sigma_{zx}) + \frac{\partial}{\partial y}(\varrho vw - \sigma_{zy}) + \frac{\partial}{\partial z}(\varrho w^2 + p - \sigma_{zz}) = \varrho f_z \quad (2.2.20)$$

where the components of the viscous stress tensor follow from Eq. (2.2.13),

$$\begin{aligned} \sigma_{xx} &= \frac{2}{3}\mu \left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) \\ \sigma_{yy} &= \frac{2}{3}\mu \left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) \\ \sigma_{zz} &= \frac{2}{3}\mu \left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \end{aligned} \quad (2.2.21)$$

$$\begin{aligned} \sigma_{xy} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) = \sigma_{yx} \\ \sigma_{xz} &= \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) = \sigma_{zx} \\ \sigma_{yz} &= \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) = \sigma_{zy} \end{aligned}$$

Similarly, the energy equation can be written in conservation form as

$$\begin{aligned} \frac{\partial E_t}{\partial t} - \frac{\partial \dot{Q}}{\partial t} - \varrho(f_x u + f_y v + f_z w) \\ + \frac{\partial}{\partial x}(E_t u + pu - u\sigma_{xx} - v\sigma_{xy} - w\sigma_{xz} + \dot{q}_x) \\ + \frac{\partial}{\partial y}(E_t v + pv - u\sigma_{xy} - v\sigma_{yy} - w\sigma_{yz} + \dot{q}_y) \\ + \frac{\partial}{\partial z}(E_t w + pw - u\sigma_{xz} - v\sigma_{yz} - w\sigma_{zz} + \dot{q}_z) = 0 \end{aligned} \quad (2.2.22)$$

2.2.2 Navier–Stokes Equations: Integral Form

The Navier–Stokes equations can also be derived for a finite control volume fixed in space or moving with the fluid. The Navier–Stokes equations derived for a fixed control volume, in either integral or differential form, are expressed in conservation form, while the equations for a moving control volume are necessarily in nonconservation form. As we shall see later, in the numerical solution of the conservation equations, the conservation form is preferable to avoid numerical difficulties that may arise in some flows such as those containing shock waves. The conservation form is also convenient in that the continuity, momentum and energy equations can all be expressed by the same generic equation. A detailed derivation of the Navier–Stokes equations in integral form is given in several references, see for example Arpaci [2] and Anderson [4, 5]. Here we adopt the notation and description in Anderson. For a finite control volume fixed in space, with $d\Omega$ denoting the control volume and dS the control surface, we first write the conservation integral form of the continuity equation for a three-dimensional compressible flow as

$$\frac{\partial}{\partial t} \iiint_{\Omega} \varrho d\Omega + \iint_S \varrho \vec{V} \cdot d\vec{S} = 0 \quad (2.2.23)$$

The first term of this equation denotes the time rate of increase of mass

$$\iiint_{\Omega} \varrho d\Omega$$

inside the control volume, while the second term, with the sign convention that positive mass flow

$$\varrho \vec{V} \cdot d\vec{S}$$

corresponds to outflow and negative to inflow, denotes the net mass flow out of the control volume. Equation (2.2.23) is a specific example of a generic form of the conservation integral equations, which can be written for a general unknown variable U as follows:

$$\frac{\partial}{\partial t} \iiint_{\Omega} U d\Omega + \iint_S \vec{F} \cdot d\vec{S} = \iiint_{\Omega} Q_v d\Omega + \iint_S Q_s dS \quad (2.2.24)$$

As we shall see shortly, when the equations are expressed in integral form, we are concerned with the *flux* of mass, momentum and energy into and out of the volume. Typical examples of fluxes are mass flux, $\varrho \vec{V}$, flux of x -component of momentum, $\varrho u \vec{V}$, flux of y -component of momentum, $\varrho v \vec{V}$, flux of z -component of momentum, $\varrho w \vec{V}$, flux of internal energy, $\varrho e \vec{V}$, and flux of total energy, $\varrho(e + V^2/2) \vec{V}$. In Eq. (2.2.24), U is a quantity which can “accumulate” inside the control volume, \vec{F} is the flux associated with U which serves to increase or

decrease U due to the flow into or out of the control volume across the control surface, Q_v represents any possible sources of U locally inside the volume, and Q_s denotes any possible sources of U on the control surface. For the continuity equation, U is ϱ , \vec{F} is $\varrho\vec{V}$ and $Q_v = Q_s = 0$.

The components of the momentum equation, consistent with the generic integral form of Eq. (2.2.24), are derived and discussed by Anderson [5]. They may be summarized by the following equations:

x -component of the momentum equation

$$\begin{aligned} \frac{\partial}{\partial t} \iiint_{\Omega} (\varrho u) d\Omega + \iint_S (\varrho u \vec{V}) \cdot d\vec{S} \\ = \iint_S (-p \vec{n} \cdot \vec{i} + \sigma_n \vec{n} \cdot \vec{i} + \sigma_s \vec{m} \cdot \vec{i}) dS + \iiint_{\Omega} (\varrho f_x) d\Omega \end{aligned} \quad (2.2.25)$$

y -component of the momentum equation

$$\begin{aligned} \frac{\partial}{\partial t} \iiint_{\Omega} (\varrho v) d\Omega + \iint_S (\varrho v \vec{V}) \cdot d\vec{S} \\ = \iint_S (-p \vec{n} \cdot \vec{j} + \sigma_n \vec{n} \cdot \vec{j} + \sigma_s \vec{m} \cdot \vec{j}) dS + \iiint_{\Omega} (\varrho f_y) d\Omega \end{aligned} \quad (2.2.26)$$

z -component of the momentum equation

$$\begin{aligned} \frac{\partial}{\partial t} \iiint_{\Omega} (\varrho w) d\Omega + \iint_S (\varrho w \vec{V}) \cdot d\vec{S} \\ = \iint_S (-p \vec{n} \cdot \vec{k} + \sigma_n \vec{n} \cdot \vec{k} + \sigma_s \vec{m} \cdot \vec{k}) dS + \iiint_{\Omega} (\varrho f_z) d\Omega \end{aligned} \quad (2.2.27)$$

Here \vec{n} is a unit vector perpendicular to the infinitesimal control surface dS and \vec{m} is a unit vector tangent to the surface and pointing in the direction of the viscous shear stress that acts on the surface.

As with the differential form of the momentum equations, Eqs. (2.2.2) to (2.2.4), the left-hand sides of Eqs. (2.2.25) to (2.2.27) represent momentum flux rates with the first term representing the time rate of change of momentum due to unsteady fluctuations of flow properties inside the control volume and the second term representing the net flow of momentum out of the control volume across the surface S . The right-hand sides of Eqs. (2.2.25) to (2.2.27) represent the sum of the net forces acting on the fluid as it flows through the control volume. The first term represents the components of the surface forces which are composed of pressure and viscous forces, while the second term represents the body force.

The conservation integral form of the energy equation in the form of Eq. (2.2.24) is

$$\begin{aligned} \frac{\partial}{\partial t} \iiint_{\Omega} E_t d\Omega + \iint_S E_t \vec{V} \cdot d\vec{S} &= \iiint_{\Omega} (\varrho \vec{q} + \varrho \vec{f} \cdot \vec{V}) d\Omega \\ &= \iint_S (-\vec{q}_c \cdot \vec{n} - p \vec{n} \cdot \vec{V} + \sigma_n \vec{n} \cdot \vec{V} + \sigma_s \vec{m} \cdot \vec{V}) dS \end{aligned} \quad (2.2.28)$$

Here the vector \vec{q} denotes the volumetric rate of heat addition per unit mass and \vec{q}_c denotes the heat conduction vector.

The above equation is for compressible flow and, as such, it has more terms than the incompressible energy equation given by Eq. (2.2.11). It is, however, the same as the differential form of the energy equation given by Eq. (2.2.15a) for compressible flows. The first term on the left-hand side of Eq. (2.2.28) represents the time rate of change of total energy E_t inside the control volume due to transient variations of flow-field variables, and the second term represents the net rate of change of total energy across the control surface. The terms on the right-hand side of Eq. (2.2.28) correspond to the terms on the right-hand side of Eq. (2.2.24).

2.2.3 Navier–Stokes Equations: Vector-Variable Form

As we shall see in Chapters 4, 5 and 10 to 12, before the application of the numerical methods to the conservation equations, it is convenient to combine the continuity, momentum and energy equations into a compact *vector-variable form*. With l denoting a length scale, the speed of sound, a , denoting the velocity scale, the parameters ϱ , u , v , p , σ_{xx} , σ_{xy} , σ_{yy} , E_t , μ , t and x , y , with ∞ referring to freestream quantities, can be expressed in nondimensional form as

$$\begin{aligned} \tilde{\varrho} &= \frac{\varrho}{\varrho_\infty}, \quad \tilde{u} = \frac{u}{a_\infty}, \quad \tilde{v} = \frac{v}{a_\infty}, \quad \tilde{p} = \frac{p}{\varrho_\infty a_\infty^2}, \\ \tilde{\sigma}_{xx} &= \frac{\sigma_{xx} l}{u_\infty a_\infty^2}, \quad \tilde{\sigma}_{xy} = \frac{\sigma_{xy} l}{u_\infty a_\infty^2}, \quad \tilde{\sigma}_{yy} = \frac{\sigma_{yy} l}{u_\infty a_\infty^2}, \\ \tilde{E}_t &= \frac{E_t}{\varrho_\infty a_\infty^2}, \quad \tilde{\mu} = \frac{\mu}{\mu_\infty}, \quad \tilde{t} = \frac{ta_\infty}{l}, \quad \tilde{x} = \frac{x}{l}, \quad \tilde{y} = \frac{y}{l} \end{aligned} \quad (2.2.29)$$

Then the compressible Navier–Stokes equations in a Cartesian coordinate system, given by Eqs. (2.2.12b), (2.2.18), (2.2.19) and (2.2.15a) can be written in dimensionless form, without body forces or external heat addition for two-dimensional flows, as

$$\frac{\partial \tilde{Q}}{\partial \tilde{t}} + \frac{\partial \tilde{E}}{\partial \tilde{x}} + \frac{\partial \tilde{F}}{\partial \tilde{y}} = \frac{1}{\text{Re}} \left(\frac{\partial \tilde{E}_v}{\partial \tilde{x}} + \frac{\partial \tilde{F}_v}{\partial \tilde{y}} \right) \quad (2.2.30)$$

For simplicity, the \sim will be dropped in dimensionless quantities, and the Reynolds number Re is defined by

$$\text{Re} = \frac{\varrho_\infty l a_\infty}{\mu_\infty} \quad (2.2.31)$$

and the \underline{Q} , \underline{E} , \underline{F} , \underline{E}_v , \underline{F}_v vectors by

$$\underline{Q} = \begin{vmatrix} \varrho \\ \varrho u \\ \varrho v \\ E_t \end{vmatrix}, \quad \underline{E} = \begin{vmatrix} \varrho u \\ \varrho u^2 + p \\ \varrho u v \\ (E_t + p)u \end{vmatrix}, \quad \underline{F} = \begin{vmatrix} \varrho v \\ \varrho u v \\ \varrho v^2 + p \\ (E_t + p)v \end{vmatrix} \quad (2.2.32a)$$

$$\underline{E}_v = \begin{vmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \beta_x \end{vmatrix}, \quad \underline{F}_v = \begin{vmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ \beta_y \end{vmatrix} \quad (2.2.32b)$$

with γ denoting the ratio of specific heats and a the speed of sound, which for ideal gases is given by $a^2 = \gamma p / \varrho$. The viscous stresses are

$$\begin{aligned} \sigma_{xx} &= \frac{2}{3}\mu \left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\ \sigma_{yy} &= \frac{2}{3}\mu \left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right) \\ \sigma_{xy} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \end{aligned} \quad (2.2.33)$$

and we also write

$$\begin{aligned} \beta_x &= u\sigma_{xx} + v\sigma_{xy} + \frac{\mu}{\text{Pr}(\gamma - 1)} \frac{\partial}{\partial x} (a^2) \\ \beta_y &= u\sigma_{xy} + v\sigma_{yy} + \frac{\mu}{\text{Pr}(\gamma - 1)} \frac{\partial}{\partial y} (a^2) \end{aligned}$$

2.2.4 Navier–Stokes Equations: Transformed Form

The Navier–Stokes equations discussed in the previous subsections and expressed for a Cartesian coordinate system are valid for any coordinate system. In many problems it is more convenient to write the equations in general curvilinear coordinates by using a coordinate transformation from the rectangular Cartesian form. To illustrate the procedure, consider a two-dimensional unsteady flow and introduce the generic transformation

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x, y, t) \\ \eta &= \eta(x, y, t) \end{aligned} \quad (2.2.34)$$

For an actual application, the transformation in Eq. (2.2.34) must be given in some analytical or numerical form. Often the transformation is chosen so that

the grid spacing in the curvilinear space is uniform and of unit length, that is $\Delta\eta = 1$, $\Delta\xi = 1$. This produces a computational space ξ, η with a rectangular domain and with a regular uniform mesh so that, as we shall see in Section 6.3, the differencing schemes used in the numerical formulation are simpler. The original Cartesian space is usually referred to as the physical domain.

Using the chain rule of differential calculus, we can write

$$\begin{aligned}\frac{\partial}{\partial t} &= \frac{\partial}{\partial\tau} + \frac{\partial}{\partial\xi}\frac{\partial\xi}{\partial t} + \frac{\partial}{\partial\eta}\frac{\partial\eta}{\partial t} = \frac{\partial}{\partial\tau} + \frac{\partial}{\partial\xi}\xi_t + \frac{\partial}{\partial\eta}\eta_t \\ \frac{\partial}{\partial x} &= \frac{\partial}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial}{\partial\eta}\frac{\partial\eta}{\partial x} = \frac{\partial}{\partial\xi}\xi_x + \frac{\partial}{\partial\eta}\eta_x \\ \frac{\partial}{\partial y} &= \frac{\partial}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial}{\partial\eta}\frac{\partial\eta}{\partial y} = \frac{\partial}{\partial\xi}\xi_y + \frac{\partial}{\partial\eta}\eta_y\end{aligned}\quad (2.2.35)$$

or in compact form as

$$\begin{vmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{vmatrix} = \begin{vmatrix} 1 & \xi_t & \eta_t \\ 0 & \xi_x & \eta_x \\ 0 & \xi_y & \eta_y \end{vmatrix} \begin{vmatrix} \frac{\partial}{\partial\tau} \\ \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \end{vmatrix}\quad (2.2.36)$$

In a similar manner, the second derivatives that appear in the momentum and energy equations can be expressed in transformed variables. They are, however, somewhat more involved than those in Eq. (2.2.35). For example, with the chain rule, it can be shown that

$$\begin{aligned}\frac{\partial^2}{\partial x^2} &= \frac{\partial}{\partial\xi}\xi_{xx} + \frac{\partial}{\partial\eta}\eta_{xx} + \frac{\partial^2}{\partial\xi^2}\xi_x^2 + \frac{\partial^2}{\partial\eta^2}\eta_x^2 + 2\frac{\partial^2}{\partial\eta\partial\xi}\eta_x\xi_x \\ \frac{\partial^2}{\partial y^2} &= \frac{\partial}{\partial\xi}\xi_{yy} + \frac{\partial}{\partial\eta}\eta_{yy} + \frac{\partial^2}{\partial\xi^2}\xi_y^2 + \frac{\partial^2}{\partial\eta^2}\eta_y^2 + 2\frac{\partial^2}{\partial\eta\partial\xi}\eta_y\xi_y \\ \frac{\partial^2}{\partial x\partial y} &= \frac{\partial}{\partial\xi}\xi_{xy} + \frac{\partial}{\partial\eta}\eta_{xy} + \frac{\partial^2}{\partial\xi^2}\xi_x\xi_y + \frac{\partial^2}{\partial\eta^2}\eta_x\eta_y + \frac{\partial^2}{\partial\eta\partial\xi}(\eta_x\xi_y + \xi_x\eta_y)\end{aligned}\quad (2.2.37)$$

In terms of the transformation defined by Eq. (2.2.34), the vector form of the transformed Navier–Stokes equations, Eq. (2.2.30), can be written as

$$\begin{aligned}\frac{\partial Q}{\partial\tau} + \xi_t\frac{\partial Q}{\partial\xi} + \eta_t\frac{\partial Q}{\partial\eta} + \xi_x\frac{\partial E}{\partial\xi} + \eta_x\frac{\partial E}{\partial\eta} + \xi_y\frac{\partial F}{\partial\xi} + \eta_y\frac{\partial F}{\partial\eta} \\ = \frac{1}{Re} \left(\xi_x\frac{\partial E_v}{\partial\xi} + \eta_x\frac{\partial E_v}{\partial\eta} + \xi_y\frac{\partial F_v}{\partial\xi} + \eta_y\frac{\partial F_v}{\partial\eta} \right)\end{aligned}\quad (2.2.38)$$

The coefficients of the derivatives in Eq. (2.2.35) with respect to ξ, η , namely $\xi_t, \xi_x, \xi_y, \eta_t, \eta_x$ and η_y are metric terms which can be obtained from the transformation given by Eq. (2.2.34). If the relations between the independent variables

in physical space and transformed space are given analytically, then the metrics can be obtained in closed form. In general, however, we usually are provided with just the (x, y) coordinates of grid points and numerically generate the metrics using finite-difference quotients.

Reversing the role of the independent variables in the chain rule formulas, Eq. (2.2.36) becomes

$$\begin{aligned}\frac{\partial}{\partial \tau} &= \frac{\partial}{\partial t} + \frac{\partial}{\partial x} x_\tau + \frac{\partial}{\partial y} y_\tau \\ \frac{\partial}{\partial \xi} &= \frac{\partial}{\partial x} x_\xi + \frac{\partial}{\partial y} y_\xi, \quad \frac{\partial}{\partial \eta} = \frac{\partial}{\partial x} x_\eta + \frac{\partial}{\partial y} y_\eta\end{aligned}\quad (2.2.39)$$

which can be written in matrix form

$$\begin{vmatrix} \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{vmatrix} = \begin{vmatrix} 1 & x_\tau & y_\tau \\ 0 & x_\xi & y_\xi \\ 0 & x_\eta & y_\eta \end{vmatrix} \begin{vmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{vmatrix} \quad (2.2.40)$$

Solving Eq. (2.2.40) for the curvilinear derivatives in terms of the Cartesian derivatives yields

$$\begin{vmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{vmatrix} = \frac{1}{J^{-1}} \begin{vmatrix} J^{-1} (x_\eta y_\tau - x_\tau y_\eta) & (x_\tau y_\xi - y_\tau x_\xi) \\ 0 & y_\eta & -y_\xi \\ 0 & -x_\eta & x_\xi \end{vmatrix} \begin{vmatrix} \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{vmatrix} \quad (2.2.41)$$

where J^{-1} is the inverse Jacobian determinant defined by

$$J^{-1} \equiv \frac{\partial(x, y)}{\partial(\xi, \eta)} \equiv \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} = x_\xi y_\eta - x_\eta y_\xi \quad (2.2.42)$$

Evaluating Eq. (2.2.41) for the metric terms by comparing to the matrix of Eq. (2.2.36), we find that

$$\begin{aligned}\xi_t &= (x_\eta y_\tau - x_\tau y_\eta) / J^{-1}, \quad \eta_t = (x_\tau y_\xi - y_\tau x_\xi) / J^{-1} \\ \xi_x &= \frac{y_\eta}{J^{-1}}, \quad \eta_x = -\frac{y_\xi}{J^{-1}}, \quad \xi_y = -\frac{x_\eta}{J^{-1}}, \quad \eta_y = \frac{x_\xi}{J^{-1}},\end{aligned}\quad (2.2.43)$$

We note from Eq. (2.2.43) that if we are given the inverse transformation

$$\begin{aligned} t &= \tau \\ x &= x(\xi, \eta, \tau) \\ y &= y(\xi, \eta, \tau) \end{aligned} \tag{2.2.44}$$

so that ξ , η and τ are now the independent variables, then the metric coefficients in Eqs. (2.2.38) can be obtained from the relations given by Eq. (2.2.43).

At this point, we notice that Eqs. (2.2.38) are in a weak conservation form. That is, even though none of the flow variables (or more appropriately, functions of the flow variables) occur as coefficients in the differential equations, the metrics do. However, as discussed by Pulliam [6], the expressions

$$\begin{aligned} \frac{\partial}{\partial \tau} (J^{-1}) + \frac{\partial}{\partial \xi} \left(\frac{\xi_t}{J} \right) + \frac{\partial}{\partial \eta} \left(\frac{\eta_t}{J} \right), \\ \frac{\partial}{\partial \xi} \left(\frac{\xi_x}{J} \right) + \frac{\partial}{\partial \eta} \left(\frac{\eta_x}{J} \right), \end{aligned}$$

and

$$\frac{\partial}{\partial \xi} \left(\frac{\xi_y}{J} \right) + \frac{\partial}{\partial \eta} \left(\frac{\eta_y}{J} \right)$$

are defined as invariants of the transformation and are analytically equal to zero. Eqs. (2.2.38) can then be expressed in the strong conservation form and written as

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \frac{1}{\text{Re}} \left(\frac{\partial \hat{E}_v}{\partial \xi} + \frac{\partial \hat{F}_v}{\partial \eta} \right) \tag{2.2.45}$$

where

$$\hat{Q} = J^{-1} \begin{vmatrix} \varrho \\ \varrho u \\ \varrho v \\ E_t \end{vmatrix}, \quad \hat{E} = J^{-1} \begin{vmatrix} \varrho U \\ \varrho uU + \xi_x p \\ \varrho vU + \xi_y p \\ U(E_t + p) - \xi_t p \end{vmatrix}, \quad \hat{F} = J^{-1} \begin{vmatrix} \varrho V \\ \varrho uV + \eta_x p \\ \varrho vV + \eta_y p \\ V(E_t + p) - \eta_t p \end{vmatrix} \tag{2.2.46a}$$

with the contravariant velocities U and V defined by

$$U = \xi_t + \xi_x u + \xi_y v, \quad V = \eta_t + \eta_x u + \eta_y v \tag{2.2.46b}$$

The viscous flux terms are

$$\hat{E}_v = J^{-1}(\xi_x \underline{E}_v + \xi_y \underline{F}_v), \quad \hat{F}_v = J^{-1}(\eta_x \underline{E}_v + \eta_y \underline{F}_v) \tag{2.2.46c}$$

with \underline{E}_v and \underline{F}_v given by Eq. (2.2.32b). The stress terms, such as σ_{xx} , σ_{yy} , etc. are also transformed in terms of the ξ and η derivatives where

$$\begin{aligned} \sigma_{xx} &= \frac{\mu}{3} [4(\xi_x u_\xi + \eta_x u_\eta) - 2(\xi_y v_\xi + \eta_y v_\eta)] \\ \sigma_{yy} &= \frac{\mu}{3} [-2(\xi_x u_\xi + \eta_x u_\eta) + 4(\xi_y v_\xi + \eta_y v_\eta)] \\ \sigma_{xy} &= \mu(\xi_y u_\xi + \eta_y u_\eta + \xi_x v_\xi + \eta_x v_\eta) \end{aligned} \tag{2.2.47}$$

$$\begin{aligned}\beta_x &= u\sigma_{xx} + v\sigma_{xy} + \frac{\mu}{\text{Pr}(\gamma - 1)} \left[\xi_x \frac{\partial}{\partial \xi} (a^2) + \eta_x \frac{\partial}{\partial \eta} (a^2) \right] \\ \beta_y &= u\sigma_{xy} + v\sigma_{yy} + \frac{\mu}{\text{Pr}(\gamma - 1)} \left[\xi_y \frac{\partial}{\partial \xi} (a^2) + \eta_y \frac{\partial}{\partial \eta} (a^2) \right]\end{aligned}\quad (2.2.48)$$

2.3 Reynolds-Averaged Navier–Stokes Equations

The Navier–Stokes equations of the previous section also apply to turbulent flows if the values of fluid properties and dependent variables are replaced by their instantaneous values. A direct approach to solving the equations for turbulent flows is to solve them for specific boundary conditions and initial values that include time-dependent quantities. Mean values are needed in most practical cases, so an ensemble of solutions of time-dependent equations is required. Even for the most restricted cases, this approach, referred to as direct numerical simulation (DNS) and discussed in [6], becomes a difficult and extremely expensive computing problem because the unsteady eddy motions of turbulence appear over a wide range of scales. The usual procedure is to average the equations rather than their solutions, as discussed in [2, 3, 7].

In this section and following sections, we shall consider the differential form of the conservation equations and, for simplicity, restrict the discussion to incompressible flows. The treatment of compressible flow equations is similar but rather lengthy. Whenever appropriate, the governing equations for compressible flows will be given without derivation.

In order to obtain the conservation equations for turbulent flows, we replace the instantaneous quantities in the equations by the sum of their mean and fluctuating parts. For example, the instantaneous values of the u -, v - and w -velocities are expressed by the sum of their mean \bar{u} , \bar{v} , \bar{w} and fluctuating parts u' , v' and w' , and the temperature T by \bar{T} and T' , that is,

$$u = \bar{u} + u', \quad v = \bar{v} + v', \quad w = \bar{w} + w', \quad T = \bar{T} + T' \quad (2.3.1)$$

With the help of the continuity equation, (2.2.1), one can now write the left-hand sides of the momentum and energy equations in *conservation form* and introduce the above relations into the continuity, momentum and energy equations. After time averaging and making use of the substantial derivatives given by Eq. (2.2.5), the Reynolds averaged Navier–Stokes (RANS) equations for three-dimensional incompressible flow can be written in the following form:

$$\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} = 0 \quad (2.3.2)$$

$$\varrho \frac{D\bar{u}}{Dt} = -\frac{\partial p}{\partial x} + \mu \nabla^2 \bar{u} + \varrho f_x - \varrho \frac{\partial}{\partial x} (\bar{u}'^2) - \varrho \frac{\partial}{\partial y} (\bar{u}' \bar{v}') - \varrho \frac{\partial}{\partial z} (\bar{u}' \bar{w}') \quad (2.3.3)$$

$$\varrho \frac{D\bar{v}}{Dt} = -\frac{\partial p}{\partial y} + \mu \nabla^2 \bar{v} + \varrho f_y - \varrho \frac{\partial}{\partial x} (\overline{v' u'}) - \varrho \frac{\partial}{\partial y} (\overline{v'^2}) - \varrho \frac{\partial}{\partial z} (\overline{v' w'}) \quad (2.3.4)$$

$$\varrho \frac{D\bar{w}}{Dt} = -\frac{\partial p}{\partial z} + \mu \nabla^2 \bar{w} + \varrho f_z - \varrho \frac{\partial}{\partial x} (\overline{w' u'}) - \varrho \frac{\partial}{\partial y} (\overline{w' v'}) - \varrho \frac{\partial}{\partial z} (\overline{w'^2}) \quad (2.3.5)$$

$$\varrho c_p \frac{D\bar{T}}{Dt} = \dot{q}_w + k \nabla^2 T - \varrho c_p \frac{\partial}{\partial x} \overline{T' u'} - \varrho c_p \frac{\partial}{\partial y} \overline{T' v'} - \varrho c_p \frac{\partial}{\partial z} \overline{T' w'} \quad (2.3.6)$$

It is common to drop the overbars on the basic time variables; this results in a continuity equation identical to that given by Eq. (2.2.1), and the left-hand sides of the momentum and energy equations, Eqs. (2.3.3) to (2.3.6), become identical to the equations for laminar flow. The right-hand sides of the momentum and energy equations also resemble the right-hand sides of Eqs. (2.2.2) to (2.2.4) and (2.2.11) with the addition of the Reynolds normal, shear stress, and heat flux terms: in our previous notation the Reynolds stresses in Eq. (2.3.3) represent the turbulent contributions to σ_{xx} , σ_{xy} and σ_{xz} , respectively. The mean viscous contributions are still given by Eq. (2.2.7) and are based on the mean-velocity components. Equations (2.2.2) to (2.2.4) thus apply to both laminar and turbulent flows, provided that the so-called “stress tensor”, σ_{ij} , including the viscous contributions, is written as

$$\sigma_{ij} = -\varrho \overline{u'_i u'_j} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.3.7a)$$

or

$$\sigma_{ij} = \sigma_{ij}^t + \sigma_{ij}^\ell \quad (2.3.7b)$$

where now σ_{ij}^t denotes the Reynolds stresses so that for three-dimensional flows $\sigma_{xx}^t = -\overline{\varrho u'^2}$, $\sigma_{xy}^t = \sigma_{yx}^t = -\overline{\varrho u' v'}$, $\sigma_{xz}^t = \sigma_{zx}^t = -\overline{\varrho u' w'}$, $\sigma_{yy}^t = -\overline{\varrho v'^2}$, $\sigma_{yz}^t = \sigma_{zy}^t = -\overline{\varrho v' w'}$, $\sigma_{zz}^t = -\overline{\varrho w'^2}$, and σ_{ij}^ℓ is the viscous stress tensor as given by Eq. (2.2.7) for a Newtonian fluid.

We note that, as is the case for the momentum equations, additional terms appear on the right-hand side of the energy equation, (2.3.6). These terms, which are the thermal analogs of the Reynolds-stress gradients in Eqs. (2.3.3) to (2.3.5), are called the turbulent heat-flux gradients. For example, $\varrho c_p \overline{T' v'}$ is the rate of flux of enthalpy in the y -direction per unit area in the (x, z) plane, due to turbulent fluctuations. These terms, together with the Reynolds-stress terms in the momentum equations, introduce additional unknowns into the conservation equations. To proceed further, additional equations for these unknown quantities, or assumptions regarding the relationship between the unknown quantities to the time-mean flow variables, are needed. This is referred to as the “closure” problem in turbulent flows: we shall discuss turbulence modeling in some detail in Chapter 3.

2.4 Reduced Forms of the Navier–Stokes Equations

The conservation equations can be reduced to simpler forms by examining the relative magnitudes of the terms in the equations. In the application of this procedure, known as “order-of-magnitude” analysis, to two-dimensional steady flows, it is common to introduce two length scales L and δ (which are, respectively, parallel and normal to the wall) to assume a typical velocity to be of order u_e , and to estimate the relative magnitudes of inertia, pressure, and viscous and body force terms in the Navier–Stokes equations. For example, if we assume that a typical viscous stress is at the form

$$\tau = \mu \frac{\partial u}{\partial y}$$

then the viscous forces are of order

$$\mu \frac{u_e}{L}$$

per unit area and, since a typical pressure force is of the order

$$\varrho u_e^2$$

per unit area, the ratio of the two forces is

$$\frac{\text{Pressure Force}}{\text{Viscous Force}} = \frac{\varrho u_e^2}{\mu u_e / L} = \frac{\varrho u_e L}{\mu} = \frac{u_e L}{\nu} = R_L \quad (2.4.1)$$

For an incompressible flow with small temperature differences, the Reynolds number, R_L , is the principal parameter for determining the nature of the flow.

Two important nondimensional groups for heat flux, with equivalents for other scalar properties, are the Prandtl number

$$\text{Pr} \equiv \frac{\mu C_p}{k} = \frac{\nu}{k/\varrho C_p} = \frac{\nu}{\alpha} \quad (2.4.2)$$

which represents the ratio of diffusion coefficients, ν and α , and the Grashof number

$$\text{Gr} = \text{Ri} \cdot R_L^2 \quad (2.4.3)$$

which represents the product of the Richardson number Ri [$\equiv (\Delta\varrho/\varrho)(gh/u^2)$] and the square of the Reynolds number.

Using order of magnitude arguments, the conservation equations of the previous section can be simplified by neglecting some of the viscous terms. For example, in some three-dimensional flows the viscous terms $\partial\sigma_{xz}/\partial z$, $\partial\sigma_{yz}/\partial z$ and $\partial\sigma_{zz}/\partial z$ in Eqs. (2.2.2) to (2.2.4) and the heat transfer term $\partial q_z/\partial z$ in Eq. (2.2.22) are omitted, and the resulting form of the Navier–Stokes equations, referred to as “parabolized Navier–Stokes equations,” are solved together with the continuity equation. In other flows the Navier–Stokes equations are

simplified further by retaining only the viscous and heat transfer terms with derivatives in the coordinate direction normal to the body surface y or, for free shear flows, the direction normal to the thin layer. This is referred to as the thin-layer Navier–Stokes approximation and leads to the following equations for three-dimensional flows with the continuity equation remaining unaltered:

$$\varrho u \frac{\partial u}{\partial x} + \varrho v \frac{\partial u}{\partial y} + \varrho w \frac{\partial u}{\partial z} = - \frac{\partial p}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2} - \varrho \frac{\partial}{\partial y} \overline{u'v'} + \varrho f_x \quad (2.4.4)$$

$$\varrho u \frac{\partial v}{\partial x} + \varrho v \frac{\partial v}{\partial y} + \varrho w \frac{\partial v}{\partial z} = - \frac{\partial p}{\partial y} + \mu \frac{\partial^2 v}{\partial y^2} - \varrho \frac{\partial}{\partial y} \overline{v'v'} + \varrho f_y \quad (2.4.5)$$

$$\varrho u \frac{\partial w}{\partial x} + \varrho v \frac{\partial w}{\partial y} + \varrho w \frac{\partial w}{\partial z} = - \frac{\partial p}{\partial z} + \mu \frac{\partial^2 w}{\partial y^2} - \varrho \frac{\partial}{\partial y} \overline{v'w'} + \varrho f_z \quad (2.4.6)$$

Blottner [8] provides a good review of the significance of these equations which, along with additional assumptions, are used in the parabolized Navier–Stokes solution procedure. Note that these are not the boundary-layer equations (subsection 2.4.3): we do *not* neglect $\partial p/\partial y$, for instance. Figure 2.2 shows the hierarchy of simplification of the Navier–Stokes equations.

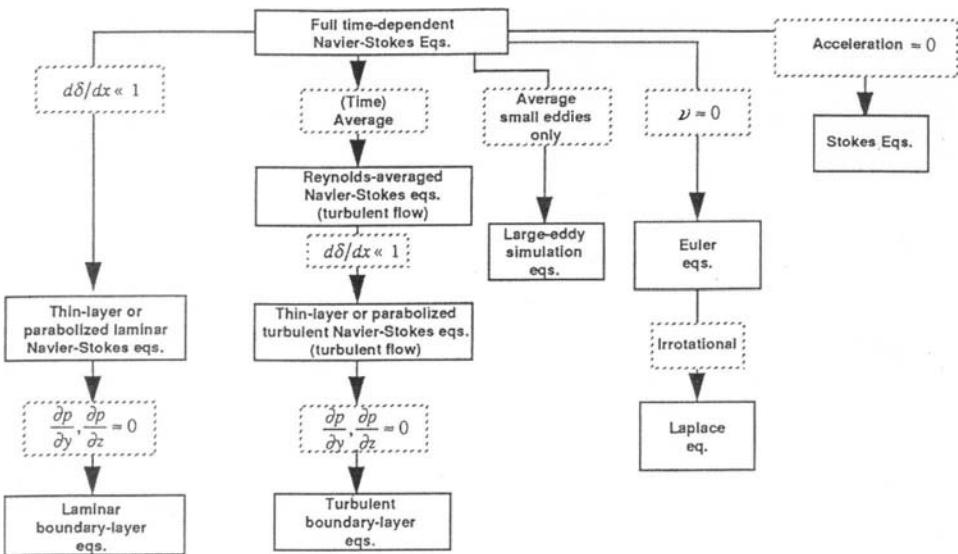


Fig. 2.2. Simplification of the Navier–Stokes equations. Dashed boxes denote simplifying approximations.

For three-dimensional compressible flows these equations, in either differential form or transformed form, are given in several references (see for example [1]). For two-dimensional compressible unsteady flows, the thin-layer Navier–Stokes equations can be obtained from Eqs. (2.2.45). Applying the thin-layer

approximation to Eqs. (2.2.45)–(2.2.47), with η locally normal to the surface and all the viscous terms associated with derivatives neglected, we obtain

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \frac{1}{\text{Re}} \frac{\partial \hat{S}}{\partial \eta} \quad (2.4.7)$$

where

$$\hat{S} = J^{-1} \begin{vmatrix} 0 \\ \eta_x m_1 + \eta_y m_2 \\ \eta_x m_2 + \eta_y m_3 \\ \eta_x (u m_1 + v m_2 + m_4) + \eta_y (u m_2 + v m_3 + m_5) \end{vmatrix} \quad (2.4.8a)$$

with

$$\begin{aligned} m_1 &= \frac{\mu}{3} (4\eta_x u_\eta - 2\eta_y v_\eta) \\ m_2 &= \mu (\eta_y u_\eta + \eta_x v_\eta) \\ m_3 &= \frac{\mu}{3} (-2\eta_x u_\eta + 4\eta_y v_\eta) \\ m_4 &= \frac{\mu}{\text{Pr}(\gamma - 1)} \eta_x \frac{\partial}{\partial \eta} (a^2) \\ m_5 &= \frac{\mu}{\text{Pr}(\gamma - 1)} \eta_y \frac{\partial}{\partial \eta} (a^2) \end{aligned} \quad (2.4.8b)$$

Equation (2.4.7) is strictly for laminar flows. As is shown in Chapter 3, its extension to turbulent flows is usually carried out by using eddy viscosity (ε_m) and turbulent Prandtl number (Pr_t) concepts. This is done by replacing the coefficient of viscosity μ by

$$\mu + \varrho \varepsilon_m \quad (2.4.9)$$

and by replacing the coefficient of thermal conductivity k ($\equiv \mu C_p / \text{Pr}$) with

$$k + \frac{C_p \varepsilon_m \varrho}{\text{Pr}_t} \quad (2.4.10)$$

The relations can easily be incorporated into Eq. (2.4.7) by replacing μ in the definitions of m_1 , m_2 , m_3 by

$$\mu (1 + \varepsilon_m / \nu) \quad (2.4.11)$$

and $[\mu / \text{Pr}(\gamma - 1)]$ in the definitions of m_4 and m_5 by

$$\frac{\mu}{\text{Pr}(\gamma - 1)} \left(1 + \frac{\text{Pr}}{\text{Pr}_t} \frac{\varepsilon_m}{\nu} \right) \quad (2.4.12)$$

Unlike μ and k , however, the parameters ε_m and Pr_t are not properties of the fluid but depend on the flowfield and are related to the velocity and temperature field by empirical formulas.

2.4.1 Inviscid Flow

One simplification of the Navier–Stokes equations assumes all σ -stresses to be locally negligible, which corresponds to inviscid flow. With the neglect of viscous forces, Eq. (2.2.10) becomes

$$\frac{D\vec{V}}{Dt} = -\frac{1}{\varrho}\nabla p + \vec{f} \quad (2.4.13)$$

which is known as the Euler equation. For a steady flow with no body forces, the Euler equation reduces to

$$(\vec{V} \cdot \nabla)\vec{V} = -\frac{\nabla p}{\varrho} \quad (2.4.14)$$

If we take a dot product of the above equation with a differential length of a streamline \vec{ds} , the Euler equations integrate (see Problem 2.4) to give

$$\frac{V^2}{2} + \int \frac{dp}{\varrho} = \text{constant} \quad (2.4.15)$$

where $V^2 = u^2 + v^2 + w^2$. For a steady incompressible flow for which ϱ is constant, the integrated Euler equation, Eq. (2.4.15), becomes

$$p + \frac{1}{2}\varrho V^2 = \text{constant} \quad (2.4.16)$$

which is called the *Bernoulli* equation. For an isentropic compressible flow [$\varrho = (\text{constant})p^{1/\gamma}$], Eq. (2.4.15) can be written as

$$\frac{V^2}{2} + \frac{\gamma}{\gamma - 1} \frac{p}{\varrho} = \text{constant} \quad (2.4.17)$$

which is known as the *compressible Bernoulli equation*. We note that the two forms of the Bernoulli equation are valid only along a *given* streamline since the constants appearing in these equations can vary between streamlines. They can, however, become valid *everywhere in the flowfield* if the flow is *irrotational*, which is defined by zero vorticity

$$\boldsymbol{\omega} = \nabla \times \vec{V} = 0 \quad (2.4.18)$$

This condition implies the existence of a scalar function ϕ , called the velocity potential, defined by

$$\vec{V} = \nabla\phi \quad (2.4.19)$$

In this case the continuity equation, Eq. (2.2.1), can be combined with Eq. (2.4.19) to obtain Laplace's equation

$$\nabla^2\phi = 0 \quad (2.4.20)$$

which provides a good approximation to some real incompressible flows at high Reynolds numbers where the viscous effects are negligible, as is sometimes the case when there is no flow separation on the body, as we shall discuss in Chapter 6.

The velocity potential also exists for compressible flows; in this case, however, in addition to the requirements imposed by Eqs. (2.4.18) and (2.4.19), there is an additional requirement on density in the continuity equation since density is not constant but varies. Using the irrotationality condition, which implies isentropic flow, density can be expressed as a function of temperature alone

$$\varrho = \text{const.} T^{1/(\gamma-1)} \quad (2.4.21)$$

and with the definition of speed of sound $a (\equiv \sqrt{\gamma RT})$, it can be written as

$$\varrho = \text{const.} a^{2/(\gamma-1)} \quad (2.4.22)$$

substituting Eq. (2.4.22) into the continuity equation (2.2.12b), and expanding the resulting equation, the continuity equation becomes

$$a^{\frac{2}{\gamma-1}} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + \frac{u(a^2)^{\frac{2-\gamma}{\gamma-1}}}{\gamma-1} \frac{\partial}{\partial x}(a^2) + \frac{v(a^2)^{\frac{2-\gamma}{\gamma-1}}}{\gamma-1} \frac{\partial}{\partial y}(a^2) = 0 \quad (2.4.23)$$

for a two-dimensional steady flow. Assume adiabatic flow so that total enthalpy is constant.

$$a^2 + \frac{\gamma-1}{2}(u^2 + v^2) = \text{const} \quad (2.4.24)$$

and substitute Eq. (2.4.24) into Eq. (2.4.23). After simplification,

$$a^2 \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) - u \left(u \frac{\partial u}{\partial x} + v \frac{\partial v}{\partial x} \right) - v \left(u \frac{\partial u}{\partial y} + v \frac{\partial v}{\partial y} \right) = 0 \quad (2.4.25)$$

With the definition of velocity potential, Eq. (2.4.19), Eq. (2.4.25) can be written as

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \frac{1}{a^2} \left[\left(\frac{\partial \phi}{\partial x} \right)^2 \frac{\partial^2 \phi}{\partial x^2} + \left(\frac{\partial \phi}{\partial y} \right)^2 \frac{\partial^2 \phi}{\partial y^2} + 2 \frac{\partial \phi}{\partial x} \frac{\partial \phi}{\partial y} \frac{\partial^2 \phi}{\partial x \partial y} \right] \quad (2.4.26a)$$

or as

$$\left(1 - \frac{u^2}{a^2} \right) \frac{\partial^2 \phi}{\partial x^2} + \left(1 - \frac{v^2}{a^2} \right) \frac{\partial^2 \phi}{\partial y^2} - 2 \frac{uv}{a^2} \frac{\partial^2 \phi}{\partial x \partial y} = 0 \quad (2.4.26b)$$

Equation (2.4.26) is known as the *full potential equation* for a two-dimensional steady flow with no body force. Unlike the Laplace equation to which it reduces as $a \rightarrow \infty$ (incompressible flow), it is nonlinear. Assuming that, with $M_\infty = V_\infty/a_\infty$,

$$M_\infty^2 \left(\frac{u}{V_\infty} \right)^2 \ll 1, \quad M_\infty^2 \left(\frac{v}{V_\infty} \right)^2 \ll 1, \quad M_\infty^2 \left(\frac{uv}{V_\infty} \right) \ll 1 \quad (2.4.27a)$$

and

$$\frac{M_\infty^2}{1 - M_\infty^2} \left(\frac{u}{V_\infty} \right) \ll 1, \quad M_\infty^2 \left(\frac{v}{V_\infty} \right) \ll 1 \quad (2.4.27b)$$

it can be expressed in the linearized form (see Problem 2.5)

$$(1 - M_\infty^2) \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (2.4.28)$$

2.4.2 Stokes Flow

A second simplification of the Navier–Stokes equations arises when the inertia terms are small enough to neglect. This situation arises when the Reynolds number R_L is much less than unity, because the velocity u_e is very small, or the scale of the flow L is very small, or the fluid is very viscous. In this case, the resulting equations are known as the Stokes equations, written in tensor notation as

$$\nabla^2 u_i = \frac{1}{\mu} \frac{\partial p}{\partial x_i} \quad (2.4.29)$$

These *linear* equations provide a good approximation to flows with Reynolds numbers less than unity such as in some lubrication problems; for simple boundaries they can be solved analytically, as discussed by Schlichting [9].

2.4.3 Boundary Layers

A third simplification of the Navier–Stokes equations occurs when ($\delta/L \ll 1$ strictly, $d\delta/dx \ll 1$, $d\delta/dz \ll 1$); this includes both laminar and turbulent flows at high Reynolds numbers. In this case, the continuity equation remains the same as Eq. (2.2.1), but the y -component of the momentum equation is eliminated and the pressure p is assumed only to be a function of x and z . For three-dimensional incompressible laminar and turbulent flow, the momentum and energy equations can be written as

$$\varrho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2} - \varrho \frac{\partial}{\partial y} (\bar{u}' v') + \varrho f_x \quad (2.4.30)$$

$$\varrho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + \mu \frac{\partial^2 w}{\partial y^2} - \varrho \frac{\partial}{\partial y} (\bar{w}' v') + \varrho f_z \quad (2.4.31)$$

$$\varrho C_p \frac{DT}{Dt} = \dot{q}_w + k \frac{\partial^2 T}{\partial y^2} - \varrho C_p \frac{\partial}{\partial y} \bar{T}' v' \quad (2.4.32)$$

However, unlike the thin-layer Navier–Stokes equations given by Eqs. (2.4.4) to (2.4.6) and the Stokes equations, Eq. (2.4.29), the pressure is not computed as part of the solution but is specified in the solution procedure. As will be shown in Section 2.6, while the Navier–Stokes equations for incompressible flows are

elliptic, the boundary-layer equations are parabolic with disturbances propagating only downstream and not upstream. This property of the boundary-layer equations significantly reduces the complexity of the solution procedure.

For two-dimensional flows, the boundary-layer equations simplify further and can be written as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.4.33)$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\varrho} \frac{dp}{dx} + \nu \frac{\partial^2 u}{\partial y^2} - \frac{\partial}{\partial y} (\bar{u}'v') \quad (2.4.34)$$

$$u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \frac{\dot{q}_w}{\varrho C_p} + \frac{k}{\varrho C_p} \frac{\partial^2 T}{\partial y^2} - \frac{\partial}{\partial y} (\bar{T}'v') \quad (2.4.35)$$

As will be discussed in detail in Chapter 7, the solutions of the boundary-layer equations can be obtained in their partial-differential equation form with assumptions made for the Reynolds stress and heat flux terms. This approach is called the differential approach, in contrast to the *integral* approach based on the solutions of momentum and energy integral equations, which are ordinary differential equations. These integral equations result from integrating the boundary-layer equations across the shear layer [3, 7] and introducing definitions of boundary-layer parameters. For two-dimensional incompressible flows, the momentum and energy integral equations are, respectively, given by

$$\frac{d\theta}{dx} + \frac{\theta}{u_e} (H + 2) \frac{du_e}{dx} = \frac{c_f}{2} \quad (2.4.36)$$

$$\frac{d\theta_T}{dx} + \frac{\theta_T}{u_e} \frac{du_e}{dx} = \text{St} \quad (2.4.37)$$

Here, θ , δ^* , H , c_f in the momentum integral equation denote momentum thickness, displacement thickness, shape factor and local skin-friction coefficient, respectively, and are defined by

$$\theta = \int_0^\infty \frac{u}{u_e} \left(1 - \frac{u}{u_e} \right) dy \quad (2.4.38a)$$

$$\delta^* = \int_0^\infty \left(1 - \frac{u}{u_e} \right) dy \quad (2.4.38b)$$

$$H = \frac{\delta^*}{\theta} \quad (2.4.38c)$$

$$c_f = \frac{\tau_w}{1/2 \varrho u_e^2} \quad (2.4.38d)$$

In the energy integral equation, the parameter θ_T and Stanton number St are defined by

$$\theta_T = \int_0^\infty \frac{u}{u_e} \frac{T - T_e}{T_w - T_e} dy \quad (2.4.39a)$$

$$St = \frac{\dot{q}_w}{\rho c_p(T_w - T_e) u_e} \quad (2.4.39b)$$

Whereas the integration of boundary-layer equations across the shear layer leads to ordinary differential equations that are much easier to solve than partial-differential equations, their solutions require auxiliary equations. The momentum integral equation has three unknowns, θ , H and c_f , and the energy integral equation has two unknowns, θ_T and St ; their solutions require at least two other equations to solve the momentum equation and one for the energy equation. While there are several useful integral methods developed for two-dimensional flows that require much less computer time than the differential methods, they are limited to simple flows [3, 7]. These limitations are not present in differential methods and, although the partial differential form of the boundary-layer equations is more difficult to solve, its use is preferred in this book and in current engineering practice and the solution procedures are described in some detail in Chapter 7.

2.5 Stability Equations

As will be explained in detail in Chapter 8, a turbulent boundary layer can develop from a laminar boundary layer by the gradual amplification of infinitesimal disturbances within the boundary layer or introduced from the external flow or by imperfections in the body surface. At first these disturbances are so weak that they have practically no influence on the mean flow but they gradually increase until the flow is distorted and the fluctuating velocity and pressure characteristics of turbulent flow appear. Transition can result from the growth of these disturbances in the boundary layer and takes place over a streamwise distance that depends on the boundary conditions.

The behavior of these disturbances within a laminar boundary layer can be studied by small-disturbance theory, as will be discussed briefly in this section and in more detail in Chapter 8. To derive the stability equations for two-dimensional mean flows with two-dimensional disturbances, assume that u , v and p in the two-dimensional form of Eqs. (2.2.1) to (2.2.3) represent the instantaneous components of the flow properties and, as in the discussion of turbulence in Section 2.3, these components are divided into a mean-flow term and a fluctuating term so that the instantaneous velocity components are $\bar{u} + u'$ and $\bar{v} + v'$ and the instantaneous pressure is $\bar{p} + p'$. The mean-flow velocity and pressure terms satisfy the boundary-layer equations for a steady laminar flow given by Eqs. (2.4.33) and (2.4.34) with the Reynolds stress term neglected and with overbars on u , v and p .

Next consider Eqs. (2.2.1) to (2.2.3) with instantaneous velocity components and pressure expressed in terms of their mean and fluctuating components. Since u' , v' and p' are small, their squares and products can be neglected. Noting that

the mean velocity and pressure satisfy the two-dimensional equations of motion, the equations simplify further and can be written as

$$\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} = 0 \quad (2.5.1)$$

$$\frac{\partial u'}{\partial t} + u' \frac{\partial u}{\partial x} + u \frac{\partial u'}{\partial x} + v' \frac{\partial u}{\partial y} + v \frac{\partial u'}{\partial y} = -\frac{1}{\varrho} \frac{\partial p'}{\partial x} + \nu \nabla^2 u' \quad (2.5.2)$$

$$\frac{\partial v'}{\partial t} + u' \frac{\partial v}{\partial x} + u \frac{\partial v'}{\partial x} + v' \frac{\partial v}{\partial y} + v \frac{\partial v'}{\partial y} = -\frac{1}{\varrho} \frac{\partial p'}{\partial y} + \nu \nabla^2 v' \quad (2.5.3)$$

These equations, with the overbars on u and v dropped for convenience, can be simplified further by noting that all velocity fluctuations and their derivatives are of the same order of magnitude and by assuming that the mean flow velocity u is a function of y only so that Eq. (2.4.33) gives $v = 0$, that is,

$$u = u(y), \quad v = 0 \quad (2.5.4)$$

This assumption is known as the *parallel flow approximation*; which, with the introduction of dimensionless quantities defined by

$$\begin{aligned} u'_* &= \frac{u'}{u_0}, & v'_* &= \frac{v'}{u_0}, & u_* &= \frac{u}{u_0}, & p'_* &= \frac{p'}{\varrho u_0^2}, & t_* &= \frac{tu_0}{L} \\ x_* &= \frac{x}{L}, & y_* &= \frac{y}{L}, & R &= \frac{u_0 L}{\nu} \end{aligned} \quad (2.5.5)$$

allows Eqs. (2.5.1) to (2.5.3) to be written as

$$\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} = 0 \quad (2.5.6)$$

$$\frac{\partial u'}{\partial t} + u \frac{\partial u'}{\partial x} + v' \frac{\partial u}{\partial y} = -\frac{\partial p'}{\partial x} + \frac{1}{R} \left(\frac{\partial^2 u'}{\partial x^2} + \frac{\partial^2 u'}{\partial y^2} \right) \quad (2.5.7)$$

$$\frac{\partial v'}{\partial t} + u \frac{\partial v'}{\partial x} = -\frac{\partial p'}{\partial y} + \frac{1}{R} \left(\frac{\partial^2 v'}{\partial x^2} + \frac{\partial^2 v'}{\partial y^2} \right) \quad (2.5.8)$$

For convenience, the subscript $*$ on the dependent and independent variables has been dropped.

Equations (2.5.6) to (2.5.8) form a set of coupled partial-differential equations with solutions that describe how disturbances originate near the surface $y = 0$ and spread out through the boundary layer and beyond as they are convected along the local streamlines. To study the properties of these equations, we apply the standard procedure of stability theory, namely separation of variables. Assume that the small disturbance is a sinusoidal traveling wave and represent a two-dimensional disturbance as

$$q'(x, y, t) = q(y)e^{i(\alpha x - \omega t)} \quad (2.5.9)$$

Here x , y , t are dimensionless quantities defined in Eq. (2.5.5), $q(y)$ is the amplitude function of a typical flow variable $q'(x, y, t)$, α is a dimensionless wave number, and ω is the radian (circular) frequency of the disturbance. The dimensionless forms of α and ω are defined by

$$\alpha = \frac{2\pi L}{\lambda_x} = \alpha^* L, \quad \omega = \frac{\omega^* L}{u_0} \quad (2.5.10)$$

where λ_x denotes the wavelength in the x -direction. In general q , q' , α and ω are *complex*. With q'_r and q'_i denoting the real and imaginary parts of q' , the magnitude of q' is $[(q'_r)^2 + (q'_i)^2]^{1/2}$ and its relative phase angle is $\tan^{-1}(q'_i/q'_r)$. The real part of the exponential term represents a growth of disturbance amplitude in x or t , while the imaginary part, $\exp(i\theta)$, can be rewritten as $\cos\theta + i\sin\theta$, which represents the sinusoidal oscillation in x or t .

The small-disturbance equations given by Eqs. (2.5.6), (2.5.7) and (2.5.8) can also be expressed in other forms. Eliminating pressure and introducing a stream function $\psi(x, y, t)$ such that

$$u' = \frac{\partial\psi}{\partial y}, \quad v' = -\frac{\partial\psi}{\partial x} \quad (2.5.11)$$

we can express the momentum equations (2.5.7) and (2.5.8) as a fourth-order partial differential equation

$$\frac{\partial}{\partial t} \nabla^2 \psi + u \frac{\partial}{\partial x} \nabla^2 \psi - \frac{\partial\psi}{\partial x} \frac{d^2 u}{dy^2} = \frac{1}{R} \nabla^4 \psi \quad (2.5.12)$$

where $\nabla^4 = \nabla^2 \nabla^2$ and $-\nabla^2 \psi$ is the fluctuating z -component of vorticity, $\partial v'/\partial x - \partial u'/\partial y$. Equation (2.5.12) represents the rate of change of fluctuating vorticity following the fluid along a mean streamline.

Taking q' in Eq. (2.5.9) to represent the disturbance stream function ψ with $q(y)$ replaced by $\phi(y)$, and introducing the resulting expression into Eq. (2.5.12), we obtain the following fourth-order ordinary differential equation for the amplitude $\phi(y)$

$$\phi^{iv} - 2\alpha^2 \phi'' + \alpha^4 \phi = iR(\alpha u - \omega)(\phi'' - \alpha^2 \phi) - iR\alpha u'' \phi \quad (2.5.13)$$

where a prime denotes differentiation with respect to y . This equation is known as the *Orr–Sommerfeld equation* and is the fundamental equation for incompressible stability theory.

The solutions of Eq. (2.5.13) correspond to small disturbance waves and are sometimes called Tollmien–Schlichting waves. Despite the major assumptions made to derive this equation, the solutions of Eq. (2.5.13) are encouragingly close to the experimental results. The support to linear stability theory was first provided by the experiments of Schubauer and Skramstad [10] who used a specially designed low turbulence wind tunnel and generated small sinusoidal

disturbances in a boundary layer on a flat plate by means of a vibrating ribbon held parallel to the plate and normal to the freestream. The measured neutral stability curve, critical Reynolds number and amplification rates, discussed in Chapter 8, were found to agree well with the predictions of the linear stability theory.

2.6 Classification of Conservation Equations

To understand some of the basic ideas underlying the classification of conservation equations, we examine the general linear second-order partial-differential equation in two independent variables:

$$Lu \equiv a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + fu \equiv h(x, y) \quad (2.6.1)$$

Rotation of the xy -plane into the $\bar{x}\bar{y}$ -plane by the coordinate transformation,

$$\bar{x} = x \cos \theta + y \sin \theta, \quad \bar{y} = -x \sin \theta + y \cos \theta \quad (2.6.2)$$

allows Eq. (2.6.1) to be transformed to

$$\bar{L}u \equiv \bar{a} \frac{\partial^2 u}{\partial \bar{x}^2} + \bar{c} \frac{\partial^2 u}{\partial \bar{y}^2} + \bar{d} \frac{\partial u}{\partial \bar{x}} + \bar{e} \frac{\partial u}{\partial \bar{y}} + fu = h \quad (2.6.3)$$

This removal of the mixed derivative term by choosing θ so that $\bar{b} = 0$ is similar to the transformation applied to the general quadratic equation; the new coordinate axes are aligned with the principal axes of the conic section. Indeed, we adopt the terminology for the conics to classify the differential equations.

Thus if $\bar{a}\bar{c} > 0$ (i.e., \bar{a} and \bar{c} have the same sign), we call Eq. (2.6.3) an *elliptic* equation; if $\bar{a}\bar{c} < 0$, we call Eq. (2.6.3) a *hyperbolic* equation; and finally if $\bar{a}\bar{c} = 0$ (but both do not vanish), we call Eq. (2.6.3) a *parabolic* equation. As with the conics, we note that $b^2 - 4ac = \bar{b}^2 - 4\bar{a}\bar{c}$ is an invariant of the rotation. Thus we say that Eq. (2.6.1) is elliptic, hyperbolic, or parabolic if $b^2 - 4ac < 0$, > 0 , or $= 0$, respectively.

In the elliptic case we can stretch or compress the coordinates by introducing

$$\bar{\bar{x}} = \frac{\bar{x}}{\sqrt{|\bar{a}|}}, \quad \bar{\bar{y}} = \frac{\bar{y}}{\sqrt{|\bar{c}|}} \quad (2.6.4)$$

and we get finally

$$\bar{\bar{L}}u = \frac{\partial^2 u}{\partial \bar{\bar{x}}^2} + \frac{\partial^2 u}{\partial \bar{\bar{y}}^2} + \bar{\bar{d}} \frac{\partial u}{\partial \bar{\bar{x}}} + \bar{\bar{e}} \frac{\partial u}{\partial \bar{\bar{y}}} + fu = h \quad (2.6.5)$$

The leading term here is the well-known Laplacian in the $\bar{x}\bar{y}$ -plane. So we see that, by a simple rotation and stretching of the coordinate system, any elliptic equation in the plane can be reduced to the Laplacian, plus lower-order terms.

In the hyperbolic case, with $\bar{a} > 0$, $\bar{c} < 0$ and using Eq. (2.6.4), we obtain from Eq. (2.6.3)

$$\bar{\bar{L}}u = \frac{\partial^2 u}{\partial \bar{x}^2} - \frac{\partial^2 u}{\partial \bar{y}^2} + \bar{d}\frac{\partial u}{\partial \bar{x}} + \bar{e}\frac{\partial u}{\partial \bar{y}} + fu = h \quad (2.6.6)$$

Here the leading terms form the wave operator and

$$\partial^2 u / \partial \bar{x}^2 - \partial^2 u / \partial \bar{y}^2 = 0$$

is the wave equation, so that any hyperbolic equation in the plane is analogous to the wave equation, with additional lower-order terms.

In the parabolic case, with $\bar{a} > 0$, $\bar{c} = 0$, $\bar{e} < 0$ and using

$$\bar{\bar{x}} = \frac{\bar{x}}{\sqrt{\bar{a}}}, \quad \bar{\bar{y}} = \frac{\bar{y}}{|\bar{e}|} \quad (2.6.7)$$

rather than Eq. (2.6.4), we rewrite Eq. (2.6.3) as

$$\bar{\bar{L}}u = \frac{\partial^2 u}{\partial \bar{\bar{x}}^2} - \frac{\partial u}{\partial \bar{\bar{y}}} + \bar{d}\frac{\partial u}{\partial \bar{\bar{x}}} + fu = h \quad (2.6.8)$$

The leading terms here form the heat or diffusion operator, that is

$$\partial^2 u / \partial \bar{\bar{x}}^2 - \partial u / \partial \bar{\bar{y}} = 0$$

is the heat-conduction equation with u representing a nondimensional temperature.

One of the basic differences between the various types of partial-differential equations is in their “domains of dependence” and their “regions (or domains) of influence.” Suppose each Eq. (2.6.1) is to be solved in some region R of xy -space with boundary B . Then the domain of dependence of a point P in R consists of all those points on B which are required in order to uniquely determine the solution at the point in question (see Fig. 2.3). Conversely, the “region of influence” of a point P consists of all those points in R at which the solution is altered when a change in the solution at the point P occurs (see Fig. 2.4). In general, elliptic equations have the property that the domain of dependence of any point is a curve or surface completely enclosing the point, as in Fig. 2.3a. In parabolic and hyperbolic equations this is not the case. The extent of the domain of dependence of a point is determined by the intersection of the so-called characteristic curves through that point with the “physical” boundary B . The characteristics also form part of the boundary of the domains of influence, see Fig. 2.4. The total number of characteristics is equal to the number of dependent variables. For a parabolic system they all coincide with a line normal to the timelike direction (Fig. 2.4b), whereas they are distinct for a hyperbolic system. An elliptic equation has no real characteristics. For further details, see Hirsch [11].

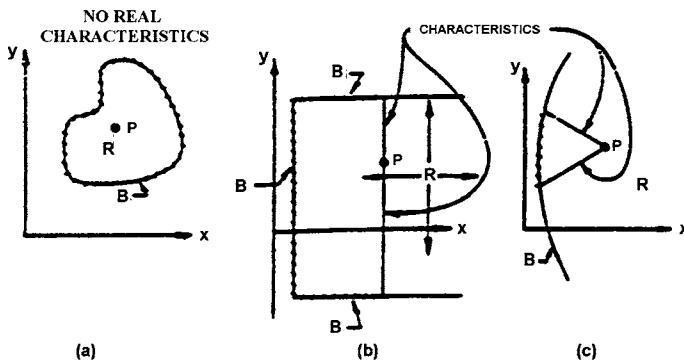


Fig. 2.3. Domains of dependence of the point P for the three classical equations to be solved in R . (a) Elliptic. (b) Parabolic. (c) Hyperbolic. The crosshatching shows the part of B , the boundary of R , that is the domain of dependence for P .

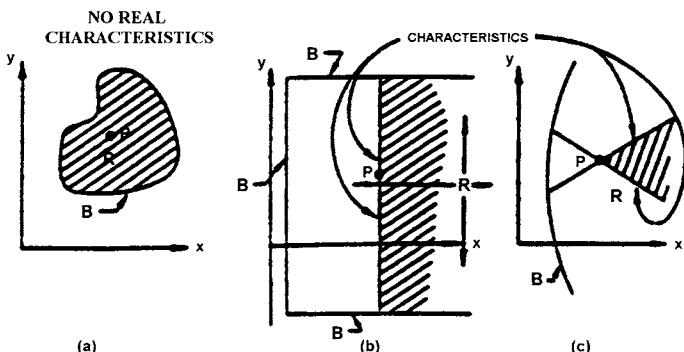


Fig. 2.4. Regions of influence of three classical equations. (a) Elliptic. (b) Parabolic. (c) Hyperbolic. Shading shows regions of influence. Strictly speaking, the boundary B in case (a) is at infinity.

The two-dimensional steady Navier-Stokes equations can be reduced to a fourth-order nonlinear equation that turns out to be elliptic. It does not correspond exactly to the generic elliptic Eq. (2.6.5), but the domain of dependence of any point is as in Fig. 2.3a. The equations for steady inviscid irrotational flow are second-order partial-differential equations. They are elliptic in subsonic flow and hyperbolic in supersonic flow for which the Mach lines are the characteristics. In a partly subsonic, partly supersonic flow these equations are said to be of “mixed type” or of elliptic-hyperbolic type. The boundary-layer equations are said to be parabolic, since the domain of influence is usually like that in Fig. 2.4b, but this is not the case when reverse flow ($u < 0$) is present. Thus, these equations may be of mixed type. The three-dimensional boundary-layer equations have more complicated domains of influence, and their type cannot be easily classified.

2.7 Boundary Conditions

The nature of the equations and their domains of dependence and zones of influence have implications for boundary conditions. Thus, for example, the steady, two-dimensional incompressible form of the momentum equation for laminar flow,

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (2.7.1)$$

is elliptic, and boundary conditions are required on all sides of the solution domain (see Section 4.5). The equation is always solved together with the equation of continuity and the normal momentum equation so that u , v and p or their gradients must be specified for a problem on its boundaries.

In the case of steady, two-dimensional boundary-layer flows, the equation

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{dp}{dx} + \nu \frac{\partial^2 u}{\partial y^2} \quad (2.7.2)$$

is parabolic so that, with the continuity equation and with known pressure, u and v or their gradients are required on three sides of the solution domain. The forms of these equations appropriate to turbulent flows are presumed to have the same requirements. The boundary conditions vary considerably from problem to problem and there are two main types.

In external flows, the shear layers flowing in the x -direction adjoin an effectively “inviscid” freestream extending to $y = \infty$. On the lower side there may be either a solid surface, usually taken as $y = 0$ as in Fig. 2.5a, in which case the viscous region is called a “wall shear layer,” or there may be another inviscid

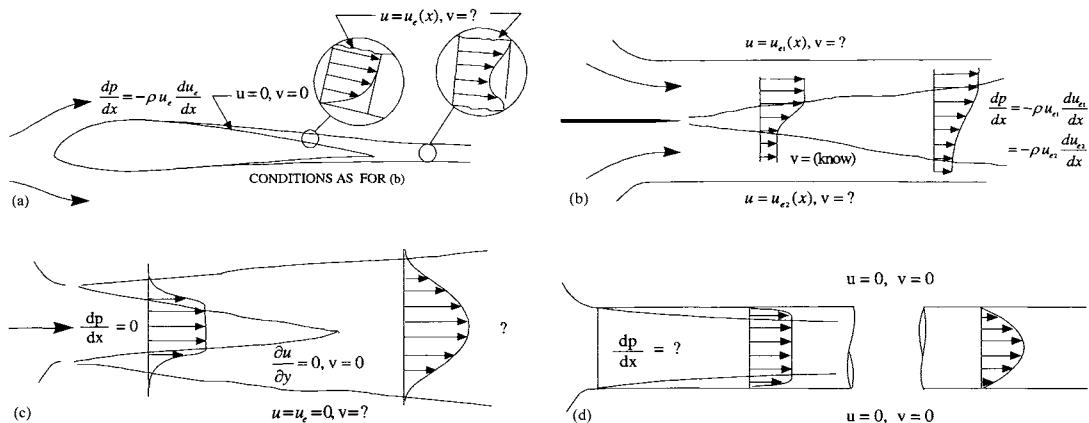


Fig. 2.5. Boundary conditions for shear layers. (a) Boundary layer and wake airfoil. (b) Mixing layer between parallel streams. (c) Merging mixing layers in jet. (d) Merging boundary layers in pipe flow.

stream extending to $y = -\infty$ (Fig. 2.5b), in which case the viscous region is called a “free shear layer.” The variation of the static pressure p with x within the shear layer depends on the shape of the solid body, and, additionally on the displacement effect of the shear layer, which may either be small enough to be neglected or can be accounted for, for example, by the interactive procedures to be described in Chapter 7. In this case there are three boundary conditions for the velocity field that must be specified, two at the wall, $y = 0$, and the other at the boundary-layer edge, $y = \delta$. The conditions at the wall are usually the specification of normal (v) and tangential (u) components of velocity, and that at the edge specifies u as the external velocity $u_e(x)$. They can be summarized as

$$y = 0, \quad u = 0, \quad v = v_w(x) \quad (2.7.3a)$$

$$y \rightarrow \delta, \quad u = u_e(x) \quad (2.7.3b)$$

where δ is sufficiently large so that $\partial u / \partial y$ at the boundary-layer edge is small, say around 10^{-4} . The transpiration velocity, $v_w(x)$, may be either suction or injection. On a nonporous surface it is equal to zero.

For boundary-layer flows, the energy equation, being second order, requires *two boundary conditions*, one at the wall, $y = 0$, and one at the thermal boundary-layer edge, δ_t , which may be smaller or larger than the hydrodynamic boundary-layer thickness, δ . For a Prandtl number, Pr , less than unity, $\delta_t > \delta$, and conversely for Pr greater than unity, $\delta_t < \delta$. The boundary conditions at the wall may correspond to either specified wall temperature, $T_w(x)$, or wall heat flux, $\dot{q}_w(x)$, and with the requirement at the boundary-layer edge, they can be summarized as

$$y = 0, \quad T = T_w(x), \quad \text{or} \quad \left(\frac{\partial T}{\partial y} \right)_w = -\dot{q}_w(x)/k \quad (2.7.4a)$$

$$y = \delta_t, \quad T = T_e \quad (2.7.4b)$$

where T_e is constant for an incompressible flow but is a function of x for a compressible flow with pressure gradient.

In free shear layers (Figs. 2.5b and c), the external velocity must be specified on both edges. The difficulties associated with the v boundary condition in free shear layers are less obvious. If the flow is symmetrical (Fig. 2.5a), no problem arises: the initial symmetrical velocity profile is specified and v is required to be zero on the centerline and $u = u_e$ at one edge. If the flow is not symmetrical (Fig. 2.5b), a *boundary condition for v cannot be found* from consideration of the shear layer and the boundary-layer equations alone. In the real flow the behavior of v outside the shear layer depends on the v -component equation of motion and the continuity equation, applied throughout the flow and not merely in the shear layer. Of course, a similar problem occurs in determining u_e either in the boundary layer or wake of Fig. 2.5a; the latter problem is both more

familiar and less perplexing. In a real external-flow problem, $u_e(x)$ is obtained from the solution of the inviscid flow equations. If the external flow is affected by the displacement effect of the shear layer, iteration between the external-flow and shear-layer calculations is necessary, as discussed briefly in Chapter 7 and in more detail in [12]. The alternative is to solve the Navier–Stokes equations throughout the flowfield.

Internal flows (Fig. 2.5d) consist of shear layer or layers filling part or all of the space between two solid boundaries. In this case the pressure distribution is set predominantly by the displacement effect of the shear layer. It is convenient to distinguish flows in which the shear layers fill the cross section and flows such as that near the entrance to a duct (the left-hand part of Fig. 2.5d) where a region of effectively inviscid flow obeying Bernoulli's equation remains.

It is also convenient to distinguish the “entrance region”, in which the velocities change with x , and the “fully developed” region far downstream in a constant area duct, in which the velocities do not, but of course the pressure continues to decrease with x . As will be shown in Chapter 7, the relevant boundary conditions in each case are similar to those for external flows, except that it is also necessary to satisfy the requirement of constant mass flow between the solid surfaces.

The boundary conditions for hyperbolic equations are rather involved in comparison to those for elliptic or parabolic equations. For this reason, their discussion is postponed to Chapters 4 and 10. For a detailed discussion, the reader is referred to Hirsch [11].

References

- [1] Anderson, D.A., Tannehill, J.C. and Pletcher, R.H.: *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Co., 1984.
- [2] Arpacı, V.S. and Larsen, P.S.: *Convection Heat Transfer*, Prentice Hall, New Jersey, 1984.
- [3] Cebeci, T.: *Convective Heat Transfer*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 2002.
- [4] Anderson, J.: *Aerodynamics*, McGraw Hill, N.Y., 1988.
- [5] Anderson, J.: *Introduction to Computational Fluid Dynamics*. McGraw Hill, N.Y., 1995.
- [6] Pulliam, T.H.: Efficient Solution Methods for the Navier–Stokes Equations, Lecture notes for the Von Karman Institute for Fluid Dynamics Lecture Series: Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings, Brussels, Belgium, Jan 20–24, 1986.
- [7] Cebeci, T.: *Analysis of Turbulent Flows*. Elsevier, London, 2004.
- [8] Blottner, F.G.: Significance of the Thin-Layer Navier–Stokes Approximation. In *Numerical and Physical Aspects of Aerodynamic Flows III*, p. 184 (ed. T. Cebeci), Springer, N.Y., 1996.
- [9] Schlichting, H.: *Boundary-Layer Theory*. McGraw-Hill, N.Y., 1968.

- [10] Schubauer, G. B. and Skramstad: "Laminar boundary-layer oscillations and transition on a flat plate". NACA Tech. Rept. 909, 1947.
- [11] Hirsch, C.: *Numerical Computation of Internal and External Flows*, Volume 1, John Wiley and Sons, N.Y., 1988.
- [12] Cebeci, T.: *An Engineering Approach to the Calculation of Aerodynamic Flows*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 1999.
- [13] Rogers, S.E. and Kwak, D., "An Upwind Differencing Scheme for the Time Accurate Incompressible Navier-Stokes Equations," *AIAA J.*, Vol. 28, No. 2, pp. 253–262, 1990.

Problems

2-1. Multiplying the x -component of the momentum equation, Eq. (2.2.2). by u , and using the definition of the substantial derivative, one obtains

$$\frac{D}{Dt} \left(\frac{1}{2} u^2 \right) = \frac{u}{\varrho} \frac{\partial p}{\partial x} + \frac{u}{\varrho} \left[\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right] + u f_x \quad (\text{P2.1.1})$$

Adding the corresponding equations for $D/Dt(1/2v^2)$ and $D/Dt(1/2w^2)$, which can be most easily derived by changing the variables in Eq. (P2.1.1) in cyclic order, show that the resulting expression with V denoting the total velocity can be written as

$$\begin{aligned} \frac{D}{Dt} \left(\frac{1}{2} V^2 \right) &= -\frac{1}{\varrho} \left(u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} + w \frac{\partial p}{\partial z} \right) + \frac{u}{\varrho} \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) \\ &\quad + \frac{v}{\varrho} \left(\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \right) + \frac{w}{\varrho} \left(\frac{\partial \sigma_{zy}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \right) \\ &\quad + u f_x + v f_y + w f_z \end{aligned} \quad (\text{P2.1.2})$$

Equation (P2.1.2) is known as the kinetic energy equation. Its left-hand side represents the rate of increase of kinetic energy per unit mass of the fluid as the fluid moves along a streamline. The terms on the right-hand side, which can be written in tensor notation as

$$\frac{u_i}{\varrho} \left(-\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \varrho f_i \right)$$

represent, respectively, the rates at which work is done on a unit mass of the fluid by the pressure, by the viscous stresses, and by the body force per unit mass, f , to produce kinetic energy of the mean motion.

2-2. Noting that for incompressible flows the continuity and momentum equations in tensor notation can be written as

$$\frac{\partial u_j}{\partial x_j} = 0 \quad (\text{P2.2.1})$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\varrho} \frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + f_i \quad (\text{P2.2.2})$$

show that the kinetic energy equation, (P2.1.2) can also be written as

$$\frac{D}{Dt} \left(\varrho \frac{u_i u_i}{2} \right) = -u_i \frac{\partial p}{\partial x_i} + u_i \frac{\partial \sigma_{ik}}{\partial x_k} - \varrho f_i \quad (\text{P2.2.3})$$

Hint: Take the scalar product of u_j and the momentum equation for u_i , Eq. (P2.2.2), with its left-hand side in conservation form

$$u_j \left[\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_k} (u_i u_k) \right] = -\frac{1}{\varrho} \frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ik}}{\partial x_k} + f_i$$

together with the scalar product of u_i and the momentum equation for u_j . Add the resulting two expressions to get

$$\begin{aligned} \varrho \frac{\partial}{\partial t} (u_i u_j) + \varrho \frac{\partial}{\partial x_k} (u_i u_j u_k) &= -u_j \frac{\partial p}{\partial x_i} - u_i \frac{\partial p}{\partial x_j} + u_j \frac{\partial \sigma_{ik}}{\partial x_k} \\ &\quad + u_i \frac{\partial \sigma_{jk}}{\partial x_k} + u_j f_i + u_i f_j \end{aligned} \quad (\text{P2.2.4})$$

and then set $i = j$.

2-3. Noting the definition of σ_{ij} in the form given by Eq. (2.3.7) and following the procedure of Problem 2.2, show that the mean kinetic energy equation of the mean motion can be written as

$$\frac{D}{Dt} \left(\varrho \frac{u_i u_i}{2} \right) = -u_i \frac{\partial p}{\partial x_i} + u_i \frac{\partial \sigma_{ik}^\ell}{\partial x_k} - u_i \frac{\partial}{\partial x_k} (\varrho \overline{u'_i u'_k}) + \varrho f_i \quad (\text{P2.3.1})$$

which is identical to Eq. (P2.2.3) if we assume that σ_{ij} represents both viscous and turbulent stresses. The third term in Eq. (P2.3.1) can be written as

$$-\frac{\partial}{\partial x_k} (u_i \varrho \overline{u'_i u'_k}) + \varrho \overline{u'_i u'_k} \frac{\partial u_i}{\partial x_k} \quad (\text{P2.3.2})$$

The first term in this equation represents the spatial transport of mean kinetic energy by the turbulent fluctuations; it is sometimes called the “gain from energy flux” or “the divergence of the energy flux transmitted by the working of the mean flow against the Reynolds stress.” The second term represents the “loss to turbulence” or the production of turbulent energy from the mean flow energy.

2-4. Defining a differential length of a streamline by \vec{ds} , which for a Cartesian coordinate system is

$$\vec{ds} = dx i + dy j + dz k \quad (\text{P2.4.1})$$

and taking the dot product of Eq. (2.4.15) with (P2.4.1) and noting that \vec{V} has the same direction as \vec{ds} , show that, with $V^2 = u^2 + v^2 + w^2$, the left-hand side of the resulting expression can be written as

$$(\vec{V} \cdot \nabla) \vec{V} \cdot \vec{ds} = V \cdot \frac{\partial \vec{V}}{\partial s} ds = V dV = d \left(\frac{V^2}{2} \right) \quad (\text{P2.4.2})$$

and Eq. (2.4.16) follows.

2-5. Derive Eq. (2.4.28).

Hint: Start with Eq. (2.4.26b). Assuming u, v to be small relative to V_∞ , write

$$\frac{\partial \phi}{\partial x} = V_\infty + u, \quad \frac{\partial \phi}{\partial y} = v \quad (\text{P2.5.1})$$

Substitute Eq. (P2.5.1) into Eq. (2.4.24)

$$a^2 + \frac{\gamma - 1}{2} [(V_\infty + u)^2 + v^2] = a_\infty^2 + \frac{\gamma - 1}{2} V_\infty^2 \quad (\text{P2.5.2})$$

expand a_∞^2/a^2 by the binomial theorem

$$\frac{a_\infty^2}{a^2} = 1 + \frac{\gamma - 1}{2} M_\infty^2 \left(2 \frac{u}{V_\infty} + \frac{u^2 + v^2}{V_\infty^2} \right) \quad (\text{P2.5.3})$$

and use the assumptions in Eqs. (2.4.27).

2-6. Starting from the full-potential equation, Eq. (2.4.26), derive the transonic small disturbance (TSD) equation written in non-conservative form,

$$\left[1 - M_\infty^2 - (\gamma + 1) \frac{u}{V_\infty} M_\infty^2 \right] \phi_{xx} + \phi_{yy} = 0 \quad (\text{P2.6.1})$$

and, in conservative form,

$$\left[(1 - M_\infty^2) \phi_x - \frac{(\gamma + 1)}{2} \frac{M_\infty^2}{V_\infty} \phi_x^2 \right]_x + \phi_{yy} = 0 \quad (\text{P2.6.2})$$

Hint: Use Eqs. (P2.5.1) and (P2.5.3), with the following small disturbance approximations:

$$\frac{u}{V_\infty} \ll 1, \quad \frac{v}{V_\infty} \ll 1$$

2-7. (a) With the definitions in Eq. (2.5.5), show that Eqs. (2.5.1) to (2.5.3) can be written in the form given by Eqs. (2.5.6) to (2.5.8).

(b) Show that with q' in Eq. (2.5.9) corresponding to u', v' and p' , the continuity and momentum equations given by Eqs. (2.5.6) to (2.5.8) can be written as

$$i\alpha u' + \frac{dv'}{dy} = 0 \quad (\text{P2.7.1})$$

$$\frac{\partial^2 u'}{\partial y^2} - \alpha^2 u' = iR \left[(\alpha u - \omega) u' - iv' \frac{du}{dy} + \alpha p' \right] \quad (\text{P2.7.2})$$

$$\frac{\partial^2 v'}{\partial y^2} - \alpha^2 v' = iR \left[(\alpha u - \omega) v' - i \frac{dp'}{dy} \right] \quad (\text{P2.7.3})$$

(c) For wall boundary-layer flows, Eqs. (2.5.6) to (2.5.8) or (P2.7.1) to (P2.7.3) are subject to the following boundary conditions at $y = 0$

$$u' = v' = 0 \quad (\text{P2.7.4})$$

Show that u' , v' and p' behave exponentially as $y \rightarrow \infty$ with a typical asymptotic representation

$$u' \sim A_1 e^{-\xi_1 y} + A_2 e^{-\xi_2 y} \quad (\text{P2.7.5})$$

where A_1 and A_2 are constants and ξ_1 and ξ_2 are defined by

$$\xi_1^2 = \alpha^2 \quad (\text{P2.7.6a})$$

$$\xi_2^2 = \alpha^2 + iR[\alpha u_e - \omega] \quad (\text{P2.7.6b})$$

with the restriction that their real parts are not negative.

2-8. The equations in Problem 2.7 can also be expressed in other forms. Show that if we represent the amplitude functions of the perturbation quantities u' , v' and p' by f , ϕ and Π , respectively, and introduce them into Eqs. (P2.7.1) to (P2.7.3), we get

$$\frac{d\phi}{dy} = -i\alpha f \quad (\text{P2.8.1})$$

$$\alpha f \left[i(\alpha u - \omega) + \frac{\alpha^2}{R} \right] + \alpha \frac{du}{dy} \phi + i\Pi \alpha^2 - \frac{\alpha}{R} \frac{d^2 f}{dy^2} = 0 \quad (\text{P2.8.2})$$

$$\frac{d\phi}{dy} = -i(\alpha u - \omega) \phi - \frac{\alpha^2}{R} \phi + \frac{1}{R} \frac{d^2 \phi}{dy^2} \quad (\text{P2.8.3})$$

2-9. Show that the equations in Problem 2.7 can also be expressed as a fourth-order differential equation in ϕ in the form given by Eq. (2.5.13).

2-10. The boundary conditions for the Orr–Sommerfeld equation follow from the relations given by Eqs. (P2.7.4) and (P2.7.5). At the wall, it follows from Eq. (P2.7.1) that

$$\bar{y} = 0, \quad \phi = \phi' = 0 \quad (\text{P2.10.1})$$

At the edge of the boundary layer, Eq. (2.5.13) reduces to

$$\phi^{iv} - (\xi_1^2 + \xi_2^2)\phi'' + \xi_1^2\xi_2^2\phi = 0 \quad (\text{P2.10.2})$$

(a) In order that the disturbances in the boundary layer decay near the edge of the boundary layer so that ϕ and $\phi' \rightarrow 0$ as $y \rightarrow \infty$, show that the solutions of Eq. (P2.10.2) must be of the form given by Eq. (P2.7.5).

(b) Show that, with $D \equiv d/dy$, the “edge” boundary conditions on ϕ can be written as

$$(D^2 - \xi_1^2)\phi + (\xi_1 + \xi_2)(D + \xi_1)\phi = 0 \quad (\text{P2.10.3})$$

$$(D + \xi_2)(D^2 - \xi_1^2)\phi = 0 \quad (\text{P2.10.4})$$

2-11. Noting that the characteristic equation for Eq. (P2.10.1) is

$$m^4 - (\xi_1^2 + \xi_2^2)m^2 + \xi_1^2\xi_2^2 = 0$$

verify Eq. (P2.7.5).

2-12. Classify the following equations into hyperbolic, elliptic or parabolic type:

$$(a) u_{xx} + 2u_{xy} + u_{yy} - (1 + xy)u = 0$$

$$(b) u_{yy} + (1 + x^2)u_x - u_y + u = 0$$

$$(c) 5u_{xx} - 3u_{yy} + u_x + u = \sin xy$$

2-13. Determine the regions where the Tricomi's equation

$$u_{xx} + xu_{yy} = 0$$

is of elliptic, parabolic or hyperbolic type.

2-14. For what values of x and y are each of the following equations hyperbolic, parabolic, elliptic?

$$(a) (y + 1)u_{xx} + 2xu_{xy} + u_{yy} = x + y$$

$$(b) xu_{xx} - yu_{xy} + xu_{yy} + u_x = 0$$

$$(c) (y + 1)u_{xx} + 2xu_{xy} + yu_{yy} = u$$

$$(d) (1 - y)u_{xx} + 2(1 - x)u_{xy} + (1 + y)u_{yy} + yu_x = \sin x$$

2-15. Suppose that u , u_x and u_y are known at every point on a curve T in the xy -plane, where u is the solution of the equation

$$A(x, y)u_{xx} + B(x, y)u_{xy} + C(x, y)u_{yy} = 0$$

Show that if the slope y' of T satisfies the auxiliary equation

$$A(x, y)(y')^2 - B(x, y)y' + C(x, y) = 0$$

then the values of the second derivative u_{xx} , u_{xy} and u_{yy} cannot be determined at the points of T .

2-16. Show that in Eq. (2.2.32a) pressure is related to the conservative flow variables, Q , by the equation of state g.

$$p = (\gamma - 1) \left[E_t - \frac{1}{2} \varrho(u^2 + v^2) \right] \quad (\text{P2.16.1})$$

2-17. In the numerical solution of the incompressible Navier–Stokes equations by the pseudocompressibility method [13] a time derivative of pressure is added to the continuity equation so that the two-dimensional version of Eq. (2.2.1) becomes

$$\frac{\partial p}{\partial t} + \beta \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (\text{P2.17.1})$$

where β is known as the pseudocompressibility constant. Here t represents pseudo-time and is not related to physical time.

Show that with the continuity equation given by Eq. (P2.17.1), the incompressible Navier–Stokes equations in Cartesian coordinates for two-dimensional flows can be written in the following form

$$\frac{\partial D}{\partial t} + \frac{\partial}{\partial x}(\tilde{E} - \tilde{E}_v) + \frac{\partial}{\partial y}(\tilde{F} - \tilde{F}_v) = 0 \quad (\text{P2.17.2})$$

where

$$D = \begin{vmatrix} p \\ u \\ v \end{vmatrix}, \quad \tilde{E} = \begin{vmatrix} \beta u \\ u^2 + p \\ uv \end{vmatrix}, \quad \tilde{F} = \begin{vmatrix} \beta v \\ uv \\ v^2 + p \end{vmatrix} \quad (\text{P2.17.3})$$

$$\tilde{E}_v = \begin{vmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \end{vmatrix}, \quad \tilde{F}_v = \begin{vmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \end{vmatrix} \quad (\text{P2.17.4})$$

2-18. Using the transformation given by Eq. (2.2.34), show that the Laplace equation expressed in the physical plane (x, y)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

can be written in the form

$$\begin{aligned} & \frac{\partial^2 u}{\partial \xi^2}(\xi_x^2 + \xi_y^2) + \frac{\partial^2 u}{\partial \eta^2}(\eta_x^2 + \eta_y^2) + 2 \frac{\partial^2 u}{\partial \xi \partial \eta}(\eta_x \xi_x + \eta_y \xi_y) \\ & + \frac{\partial u}{\partial \xi}(\xi_{xx} + \xi_{yy}) + \frac{\partial u}{\partial \eta}(\eta_{xx} + \eta_{yy}) = 0 \end{aligned} \quad (\text{P2.18.1})$$

in the computational plane (ξ, η) .

2-19. Using the transformation

$$\begin{aligned}\xi &= \xi(r, \theta) \\ \eta &= \eta(r, \theta)\end{aligned}\quad (\text{P2.19.1})$$

Show that the Laplace equation in polar coordinates

$$\nabla^2 u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} \quad (\text{P2.19.2})$$

can be expressed in the following form

$$\begin{aligned}\frac{\partial^2 u}{\partial \xi^2} \left(\xi_r^2 + \frac{\xi_\theta^2}{r^2} \right) + \frac{\partial^2 u}{\partial \eta^2} \left(\eta_r^2 + \frac{\eta_\theta^2}{r^2} \right) + 2 \frac{\partial^2 u}{\partial \xi \partial \eta} \left(\xi_r \eta_r + \frac{\xi_\theta \eta_\theta}{r^2} \right) \\ + \frac{\partial u}{\partial \xi} \left(\xi_{rr} + \frac{\xi_r}{r} + \frac{\xi_{\theta\theta}}{r^2} \right) + \frac{\partial u}{\partial \eta} \left(\eta_{rr} + \frac{\eta_r}{r} + \frac{\eta_{\theta\theta}}{r^2} \right) = 0\end{aligned}\quad (\text{P2.19.3})$$

in the computational plane.

2-20. Determine the metric coefficients in Eq. (P2.19.3) for ξ and η defined by

$$\begin{aligned}\xi &= \theta \\ \eta &= \frac{1}{r}\end{aligned}\quad (\text{P2.20.1})$$

and show that the Laplace equation, Eq. (P2.19.3) can be written as

$$\eta^2 \frac{\partial^2 u}{\partial \eta^2} + \eta \frac{\partial u}{\partial \eta} + \frac{\partial^2 u}{\partial \xi^2} = 0 \quad (\text{P2.20.2})$$

2-21. For a perfect gas, show that

$$p = (\gamma - 1) \varrho \left(E_t - \frac{V^2}{2} \right) \quad (\text{P2.21.1})$$

Hint: Note that

$$c_v = \frac{R}{\gamma - 1}, \quad e = c_v T$$

2-22. Show that, with mass transfer (ϱv_w), the momentum integral integration, Eq. (2.4.36), can be written as

$$\frac{d}{dx} (u_e^2 \theta) + u_e v_w + \delta^* u_e \frac{du_e}{dx} = \frac{\tau_w}{\varrho} \quad (\text{P2.22.1})$$

3 Turbulence Models

3.1 Introduction

The Reynolds-averaged equations of Section 2.3 and their reduced forms in Section 2.4 cannot be solved without information about the various correlation terms that make up the stress tensor, and the same is true for the energy equation. It is well known that these terms, which represent turbulent diffusion, are much larger than those corresponding to laminar diffusion except in the immediate vicinity of a wall, and in turbulent wall boundary layers, wakes, jets and more complex flows, these turbulent diffusion terms are of similar magnitude to the convective terms.

This chapter presents a brief description of various models to address the closure problem of turbulence modeling. The subject has been studied extensively in the past three decades and useful reviews have been provided in many journal articles and in several books, see for example, Wilcox [1], Cebeci [2, 3], Durbin and Reif [4]. For a detailed description of turbulence models and their accuracy, the reader is referred to these books.

Early approaches to turbulence modeling include the mixing length, ℓ , assumptions of Prandtl [5] and eddy-viscosity, ε_m , assumptions of Boussinesq [6] for wall boundary layers and jets. Kolmogorov [7] and Rotta [8] proposed models based on partial-differential equations but, in the absence of digital computers, could not solve them. The early models provided a foundation which is still in use today. For example, the concepts of a mixing length,

$$-\varrho\overline{u'v'} = \varrho\ell^2 \left(\frac{\partial u}{\partial y} \right)^2 \quad (3.1.1)$$

and eddy-viscosity expressions of the form

$$-\varrho\overline{u'v'} = \varrho\varepsilon_m \frac{\partial u}{\partial y} \quad (3.1.2)$$

have been used in the vast majority of publications concerned with turbulence

modeling and are still used for many flows. Note that these are merely *definitions* of ℓ and ε_m : the assumption is that they will vary more slowly or more simply than the shear stress and therefore be easier to correlate empirically. Their names imply an analogy between turbulent motion and the molecular motion that leads to viscous stresses, but this is now recognized as completely misleading.

The specification of ε_m or ℓ may be made in terms of algebraic equations or in terms of a combination of algebraic and differential equations and this has given rise to terminology involving the number of differential equations. Thus, the closures may be described in terms of zero, one and two differential equations. For a two-dimensional boundary-layer, the zero-equation approach usually treats a turbulent boundary layer as a composite layer with separate expressions for ε_m or ℓ in each region. The Cebeci-Smith (CS) model discussed in detail in [2, 3] and briefly in Section 3.2 is a typical example for this approach.

In the one-differential-equation approach the eddy viscosity is written, with c_μ denoting a constant, as

$$\varepsilon_m = c_\mu k^{1/2} \ell \quad (3.1.3)$$

with k obtained from a differential equation which represents the transport of turbulence energy and ℓ from an algebraic formula. The Spalart-Allmaras (SA) model discussed in detail in [1–3, 9] and briefly in Section 3.3 is a good, useful model that uses this approach.

In the two differential equation approach, the eddy viscosity is written as

$$\varepsilon_m = \frac{c_\mu k^2}{\varepsilon} \quad (3.1.4)$$

with k and ε obtained from differential equations which represent the transport of turbulence energy and its rate of dissipation. While one-equation models have found little favor except for the SA model, and where transport of turbulence characteristics is important as in strong adverse gradients or in separated flows, two equations have found extensive use. Various forms of two-equation models have been proposed and details have been given, for example in [1–3]. Three popular models that are based on this approach are the $k-\varepsilon$ model discussed briefly in Section 3.3 and in detail in [1–3] and the $k-\omega$ and SST models discussed in [1–3].

The Reynolds shear stress can also be modelled by using the Reynolds transport equation as described for example in [1–3]. A popular model is due to Launder, Reece and Rodi [10]. However, in general, the stress-transport models often give poor predictions of complex flows [11]. Also increased numerical difficulties (complexity of programming, expense of calculations, occasional instability) cause these models not to be used in most engineering problems. For this reason, stress-transport models are not discussed here. For a detailed description of these models, the reader is referred to [1].

3.2 Zero-Equation Models

The zero-equation, often referred to as algebraic eddy viscosity and/or mixing length models, are used to model the Reynolds shear stress term in the momentum equations. For a two-dimensional incompressible laminar and turbulent flow, the continuity and momentum equations (2.4.33) and (2.4.34), respectively, are solved subject to given initial conditions and boundary conditions with $-\bar{u}'\bar{v}'$ represented by Eq. (3.1.2). For a limited range of flows, it is possible to specify the turbulent viscosity of Eq. (3.1.2) as the mixing length of Eq. (3.1.1) in closed algebraic expressions. Examples include:

$\ell/\delta = 0.09$	for a plane jet
$= 0.075$	for a round jet
$= 0.16$	for a plane wake
$= 0.41y/\delta$	for the near wall region of a boundary layer
$= 0.09$	for the outer region of a boundary layer

The restricted range of the correlations should be noted, together with the simplicity of the approach. No differential equations are solved for turbulence quantities, and turbulence transport is, consequently, not represented.

3.2.1 Cebeci-Smith Model

For external flows, a popular zero-equation model is the Cebeci-Smith (CS) model. This model assumes that a turbulent boundary layer can be represented in terms of eddy viscosities for inner and outer layers. The corresponding functions are empirical and based on limited ranges of experimental data; the range of data is, however, extensive so that the full algebraic formulation is the most general available at present.

In the inner and outer regions of a boundary layer on a smooth surface, with or without mass transfer, the eddy viscosity ε_m is written as [2, 3]

Inner region: $0 \leq y \leq y_c$

$$(\varepsilon_m)_i = l^2 \left| \frac{\partial u}{\partial y} \right| \gamma_{\text{tr}} \quad (3.2.1)$$

Here the mixing length l is given by

$$l = \kappa y \left[1 - \exp \left(-\frac{y}{A} \right) \right] \quad (3.2.2a)$$

where $\kappa = 0.40$ and A is a damping-length constant, which may be represented by

$$A = 26 \frac{\nu}{N} u_\tau^{-1}, \quad N = (1 - 11.8p^+)^{1/2} \quad p^+ = \frac{\nu u_e}{u_\tau^3} \frac{du_e}{dx}, \quad (3.2.2b)$$

Outer region: $y_c \leq y \leq \delta$

$$(\varepsilon_m)_0 = \alpha u_e \delta^* \gamma_{\text{tr}} \gamma \quad (3.2.3)$$

Here δ^* denotes the displacement thickness

$$\delta^* = \int_0^\delta \left(1 - \frac{u}{u_e} \right) dy \quad (3.2.4a)$$

and γ accounts for the intermittency of the outer region and is represented by

$$\gamma = \frac{1}{2} \left[1 - \text{erf} \frac{[y - Y]}{\sqrt{2} \sigma} \right] \quad (3.2.4b)$$

where Y and σ are general intermittency parameters, with Y denoting the value of y for which $\gamma = 0.5$, and σ the standard deviation. The dimensionless intermittency parameters Y/δ^* and σ/δ^* are expressed as functions of H as shown in Fig. 3.1a. The variation of the ratio of boundary-layer thickness δ to δ^* with H is shown in Fig. 3.1b. The parameter α is calculated from

$$\alpha = \frac{0.0168}{\left[1 - \beta \left(\frac{\partial u}{\partial x} / \frac{\partial u}{\partial y} \right)_m \right]^{1.5}} \quad (3.2.5)$$

Here subscript m denotes the location where turbulent shear is maximum. The parameter β is given by

$$\beta = \begin{cases} \frac{6}{1 + 2R_t(2 - R_t)} & R_t \leq 1.0 \\ \frac{1 + R_t}{R_t} & R_t \geq 1.0 \end{cases} \quad (3.2.6)$$

Here R_t denotes the ratio of wall shear to maximum Reynolds shear stress [2, 3],

$$R_t = \frac{\tau_\omega}{(-\varrho(u'v')_m)} \quad (3.2.7)$$

In Eqs. (3.2.1) γ_{tr} is an intermittency factor which represents the streamwise region from the onset of transition to turbulent flow. It is defined by the following expression

$$\gamma_{\text{tr}} = 1 - \exp \left[-G(x - x_{\text{tr}}) \int_{x_{\text{tr}}}^x \frac{dx}{u_e} \right] \quad (3.2.8)$$

where x_{tr} is the location of the start of the transition and the factor G is given empirically by

$$G = \frac{3}{C^2} \frac{u_e^3}{\nu^2} R_{x_{\text{tr}}}^{-1.34} \quad (3.2.9a)$$

with $R_{x_{\text{tr}}}$ denoting Reynolds number, $R_{x_{\text{tr}}} = (u_e x / \nu)_{\text{tr}}$, and C a constant with a recommended value of 60 for high Reynolds flows. For lower Reynolds number, C is given by [2, 3]

$$C^2 = 213(\log R_{x_{\text{tr}}} - 4.7323) \quad (3.2.9b)$$

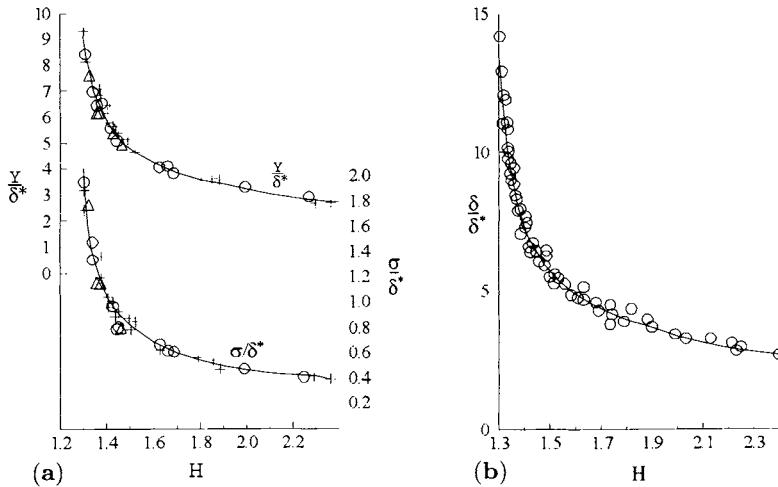


Fig. 3.1. Variation of Y/δ^* , σ/δ^* and δ/δ^* with H according to the data of Fiedler and Head [12].

3.2.2 Baldwin–Lomax Model

Due to its simplicity and its good success in external boundary-layer flows, the CS model has also been used extensively in the solution of the Reynolds-averaged Navier–Stokes equations for turbulent flows. For the inner region, Baldwin and Lomax [13] use the expressions given by Eqs. (3.2.1) and (3.2.2). In the outer region, since the length scale δ^* is not well defined in the Navier–Stokes calculations due to the lack of precise definition of boundary-layer thickness, they use alternative expressions of the form

$$(\varepsilon_m)_o = \alpha c_1 \gamma y_{\max} F_{\max} \quad (3.2.10a)$$

or

$$(\varepsilon_m)_o = ac_1 \gamma c_2 u_{\text{diff}}^2 \frac{y_{\max}}{F_{\max}} \quad (3.2.10b)$$

with $c_1 = 1.6$ and $c_2 = 0.25$. The quantities F_{\max} and y_{\max} are determined from the function

$$F = y \left(\frac{\partial u}{\partial y} \right) [1 - e^{-y/A}] \quad (3.2.11)$$

with F_{\max} corresponding to the maximum value of F that occurs in a velocity profile and y_{\max} denoting the y -location of F_{\max} . u_{diff} is the difference between maximum and minimum velocities in the profile

$$u_{\text{diff}} = u_{\max} - u_{\min} \quad (3.2.12)$$

where u_{\min} is taken to be zero except in wakes.

In Navier–Stokes calculations, Baldwin and Lomax replace the absolute value of the velocity gradient $\partial u / \partial y$ in Eqs. (3.2.1) and (3.2.11) by the absolute value of the vorticity $|\omega|$,

$$|\omega| = \left| \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right| \quad (3.2.13a)$$

and the intermittency factory γ in Eq. (3.2.4b) is written as

$$\gamma = \left[1 + 5.5 \left(\frac{c_3 y}{y_{\max}} \right)^6 \right]^{-1} \quad (3.2.13b)$$

with $c_3 = 0.3$. The studies conducted by Stock and Haase [14] clearly demonstrate that the modified algebraic eddy viscosity formulation of Baldwin and Lomax is not a true representation of the CS model since their incorporation of the length scale in the outer eddy viscosity formula is not appropriate for flows with strong pressure gradients.

Stock and Haase proposed a length scale based on the properties of the mean velocity profile calculated by a Navier–Stokes method. They recommend computing the boundary-layer thickness δ from

$$\delta = 1.936 y_{\max} \quad (3.2.14)$$

where y_{\max} is the distance from the wall for which $y|\partial u / \partial y|$ or F in Eq. (3.2.11) has its maximum. With δ known, u_e in the outer eddy viscosity formula, Eq. (3.2.3), is the u at $y = \delta$, and γ is computed from

$$\gamma = \left[1 + 5.5 \left(\frac{y}{y_0} \right)^6 \right]^{-1} \quad (3.2.15)$$

based on Klebanoff's measurements on a flat plate flow and not from Eq. (3.2.13b). The displacement thickness δ^* for attached flows is computed from its definition, Eq. (3.2.4a), and, for separated flows from

$$\delta^* = \int_{y_{u=0}}^{\delta} \left(1 - \frac{u}{u_e} \right) dy \quad (3.2.16)$$

either integrating the velocity profile from $y = 0$, or $y = y_{u=0}$ to δ , or using the Coles velocity profile, see Eq. (3.5.6). The results obtained with this modification to the length scale in the outer CS eddy viscosity formula improve the predictions of the CS model in Navier–Stokes methods as discussed in Stock and Haase [14].

A proposal which led to Eq. (3.2.14) was also made by Johnson [15]. He recommended that the boundary-layer thickness δ is calculated from

$$\delta = 1.2 y_{1/2} \quad (3.2.17)$$

where

$$y_{1/2} = y \quad \text{at} \quad \frac{F}{F_{\max}} = 0.5. \quad (3.2.18)$$

For a discussion regarding other length scales, the reader is referred to [1, 3].

3.3 One-Equation Models

Of the several methods that fall in this group, we only consider the method due to Spalart and Allmaras [2, 9]. This method employs a single transport equation for eddy viscosity, is popular for wall boundary-layer and free-shear flows and is used in both boundary-layer and Navier–Stokes methods. Its defining equations are as follows.

$$\varepsilon_m = \tilde{\nu}_t f_{v_1} \quad (3.3.1)$$

$$\begin{aligned} \frac{D\tilde{\nu}_t}{Dt} &= c_{b_1} [1 - f_{t_2}] \tilde{S}\tilde{\nu}_t - (c_{w_1} f_w - \frac{c_{b_1}}{\kappa^2} f_{t_2}) \left(\frac{\tilde{\nu}_t}{d} \right)^2 + \frac{1}{\sigma} \frac{\partial}{\partial x_k} \left[(\nu + \tilde{\nu}_t) \frac{\partial \tilde{\nu}_t}{\partial x_k} \right] \\ &+ \frac{c_{b_2}}{\sigma} \frac{\partial \tilde{\nu}_t}{\partial x_k} \frac{\partial \tilde{\nu}_t}{\partial x_k} \end{aligned} \quad (3.3.2)$$

Here

$$c_{b_1} = 0.1355, \quad c_{b_2} = 0.622, \quad c_{\nu_1} = 7.1, \quad \sigma = \frac{2}{3} \quad (3.3.3a)$$

$$c_{w_1} = \frac{c_{b_1}}{\kappa^2} + \frac{(1 + c_{b_2})}{\sigma}, \quad c_{w_2} = 0.3, \quad c_{w_3} = 2, \quad \kappa = 0.41 \quad (3.3.3b)$$

$$f_{\nu_1} = \frac{\chi^3}{\chi^3 + c_{\nu_1}^3}, \quad f_{\nu_2} = 1 - \frac{\chi}{1 + \chi f_{\nu_1}}, \quad f_w = g \left[\frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6} \right]^{1/6} \quad (3.3.3c)$$

$$\chi = \frac{\tilde{\nu}_t}{\nu}, \quad g = r + c_{w_2}(r^6 - r), \quad r = \frac{\tilde{\nu}_t}{\tilde{S}\kappa^2 d^2} \quad (3.3.3d)$$

$$\tilde{S} = S + \frac{\tilde{\nu}_t}{\kappa^2 d^2} f_{\nu_2}, \quad S = \sqrt{2\Omega_{ij}\Omega_{ij}} \quad (3.3.3e)$$

$$f_{t_2} = c_{t_3} e^{-c_{t_4}\chi^2}, \quad c_{t_3} = 1.1, \quad c_{t_4} = 2 \quad (3.3.3f)$$

where d is the distance to the closest wall and S is the magnitude of the vorticity, $\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$.

The wall boundary condition is $\tilde{\nu}_t = 0$. In the freestream and as initial condition 0 is best, and values below $\frac{\nu}{10}$ are acceptable [9].

For boundary-layer flows, Eq. (3.3.2) can be written as

$$\begin{aligned} u \frac{\partial \tilde{\nu}_t}{\partial x} + v \frac{\partial \tilde{\nu}_t}{\partial y} &= c_{b_1} (1 - f_{t_2}) \tilde{S}\tilde{\nu}_t + \frac{1}{\sigma} \left\{ \frac{\partial}{\partial y} \left[(\nu + \tilde{\nu}_t) \frac{\partial \tilde{\nu}_t}{\partial y} \right] + c_{b_2} \left(\frac{\partial \tilde{\nu}_t}{\partial y} \right)^2 \right\} \\ &- \left(c_{w_1} f_w - \frac{c_{b_1}}{\kappa^2} f_{t_2} \right) \left(\frac{\tilde{\nu}_t}{d} \right)^2 \end{aligned} \quad (3.3.4)$$

where

$$\tilde{S} = \left| \frac{\partial u}{\partial y} \right| + \frac{\tilde{\nu}_t}{\kappa^2 d^2} f_{\nu_2} \quad (3.3.5)$$

For a detailed discussion of this model and its accuracy compared to other turbulence models, the reader is referred to [1–3].

3.4 Two-Equation Models

There are several two-equation models. Three of the more popular, accurate and widely used models are the $k-\varepsilon$ model of Jones and Launder [16], the $k-\omega$ model of Wilcox [1] and the SST model of Menter which blends the $k-\varepsilon$ model in the outer region and $k-\omega$ model in the near wall region [17]. All three models can be used for a range of flow problems with good accuracy as discussed in [1–3]. Here we discuss the $k-\varepsilon$ model which is the most popular and widely used two-equation eddy viscosity model. For a discussion of the other two models, the reader is referred to [1–3].

In this model, ε_m is given by Eq. (3.1.4). The kinetic energy k and rate of dissipation ε are obtained from the *turbulence kinetic energy* equation written as

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_k} \left[\left(\nu + \frac{\varepsilon_m}{\sigma_k} \right) \frac{\partial k}{\partial x_k} \right] + \varepsilon_m \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} - \varepsilon \quad (3.4.1)$$

and the dissipation equation

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_k} \left[\left(\nu + \frac{\varepsilon_m}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_k} \right] + c_{\varepsilon_1} \frac{\varepsilon}{k} \varepsilon_m \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} - c_{\varepsilon_2} \frac{\varepsilon^2}{k} \quad (3.4.2)$$

For boundary-layer flows at high Reynolds number, Eqs. (3.4.1) and (3.4.2) become

$$u \frac{\partial k}{\partial x} + v \frac{\partial k}{\partial y} = \frac{\partial}{\partial y} \left(\frac{\varepsilon_m}{\sigma_k} \frac{\partial k}{\partial y} \right) + \varepsilon_m \left(\frac{\partial u}{\partial y} \right)^2 - \varepsilon \quad (3.4.3)$$

$$u \frac{\partial \varepsilon}{\partial x} + v \frac{\partial \varepsilon}{\partial y} = \frac{\partial}{\partial y} \left(\frac{\varepsilon_m}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial y} \right) + c_{\varepsilon_1} \frac{\varepsilon}{k} \varepsilon_m \left(\frac{\partial u}{\partial y} \right)^2 - c_{\varepsilon_2} \frac{\varepsilon^2}{k} \quad (3.4.4)$$

The parameters c_μ , c_{ε_1} , c_{ε_2} , σ_k and σ_ε are given by

$$c_\mu = 0.09, \quad c_{\varepsilon_1} = 1.44, \quad c_{\varepsilon_2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3 \quad (3.4.5)$$

The set of equations comprising conservation of mass and momentum, Eqs. (2.2.2) to (2.2.4) together with Eqs. (3.4.1) and (3.4.2), or with boundary-layer approximations, Eqs. (2.3.33), (2.3.34), (3.4.3) and (3.4.5) for two-dimensional flows together with the boundary conditions discussed below, represent a closed set in which the equations for mean momentum, turbulence energy, and dissipation rate have the same form and can generally be solved by the same numerical method.

The above equations given by Eqs. (3.4.1) to (3.4.5) apply only to free shear flows. For wall boundary-layer flows, they require modifications to account for the presence of the wall. Without wall functions, it is necessary to replace the true boundary conditions at $y = 0$ by new “boundary conditions” defined at some distance y_0 outside the viscous sublayer to avoid integrating the equations through the region of large y gradients near the surface as discussed in [2, 3].

There are several approaches that can be used to model the turbulence kinetic energy and rate of dissipation equations near the wall region. For example, in one approach, wall functions are introduced into Eqs. (3.4.3) and (3.4.4) so that the model equations are applicable throughout the whole layer. Launder and Sharma [18] modify Eq. (3.1.4)

$$\nu_t = c_\mu f_\mu \frac{k^2}{\varepsilon} \quad (3.4.6)$$

and rewrite Eqs. (3.4.3) and (3.4.4) as

$$u \frac{\partial k}{\partial x} + v \frac{\partial k}{\partial y} = \frac{\partial}{\partial y} \left(\frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial y} \right) + \nu_t \left(\frac{\partial u}{\partial y} \right)^2 - \varepsilon - 2\nu \left(\frac{\partial k^{1/2}}{\partial y} \right)^2 \quad (3.4.7)$$

$$u \frac{\partial \varepsilon}{\partial x} + v \frac{\partial \varepsilon}{\partial y} = \frac{\partial}{\partial y} \left(\frac{\nu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial y} \right) + c_{\varepsilon_1} \frac{\varepsilon}{k} \nu_t \left(\frac{\partial u}{\partial y} \right)^2 - f_2 c_{\varepsilon_2} \frac{\varepsilon^2}{k} + 2\nu \nu_t \left(\frac{\partial^2 u}{\partial y^2} \right)^2 \quad (3.4.8)$$

where

$$f_\mu = \exp \left[-\frac{3.4}{(1 + R_t/50)^2} \right], \quad R_t = \frac{k^2}{\nu \varepsilon} \quad (3.4.9a)$$

$$f_2 = 1 - 0.3 \exp(-R_t^2) \quad (3.4.9b)$$

The boundary conditions are

$$y = 0, \quad k = \varepsilon = 0 \quad (3.4.10a)$$

$$y \rightarrow \delta, \quad k \rightarrow k_e, \quad \varepsilon \rightarrow \varepsilon_e \quad (3.4.10b)$$

To avoid numerical problems, k_e and ε_e should not be zero. In addition, k_e and ε_e can not be prescribed arbitrarily because their development is governed by the transport equations (3.4.7) and (3.4.8) written at the boundary-layer edge,

$$u_e \frac{dk_e}{dx} = -\varepsilon_e \quad (3.4.11a)$$

$$u_e \frac{d\varepsilon_e}{dx} = -c_{\varepsilon_2} \frac{\varepsilon_e^2}{k_e} \quad (3.4.11b)$$

The above equations can be integrated with respect to x with initial conditions corresponding to k_{e0} and ε_{e0} at x_0 . The solution provides the evolutions of $k(x)$ and $\varepsilon(x)$ as boundary conditions for the k - and ε -equations.

3.5 Initial Conditions

Initial conditions needed in the solution of the transport equations for turbulent flow are often obtained from empirical expressions when experimental data are not available. The choice of these expressions can influence predictions of the turbulence models and in some cases can even lead to the breakdown of calculations. This section presents useful and convenient expressions to use for external boundary-layer flows. Before considering them, however, it is useful to review the composite nature of a wall turbulent boundary layer. As was discussed in [2, 3], for example, the behavior of turbulent flow in the inner region is different from that in the outer region of the flow. Experiments and dimensional analysis show that the length and velocity scales in the inner region, roughly $y/\delta < 0.2$, are ν/u_τ and u_τ , where u_τ denotes the friction velocity defined by $(\tau_w/\rho)^{1/2}$. The velocity profile in the inner region of a smooth wall in the absence of surface roughness and mass transfer can be expressed in the form

$$u^+ = \frac{u}{u_\tau} = f_1(y^+) \quad (3.5.1)$$

which is known as the “law of the wall”. For $y > 30\nu/u_\tau$ approximately, but $y/\delta < 0.20$ approximately, Eq. (3.5.1) can be written as

$$u^+ = \frac{1}{\kappa} \ln y^+ + c \quad (3.5.2)$$

where c is a constant found experimentally to be about 5.0–5.2.

Several expressions have been developed to express the relationship in Eq. (3.5.1) from $y = 0$ to the limit of the validity of the logarithmic law. One convenient and useful expression due to Van Driest [19] is

$$u^+ = \int_0^{y^+} \frac{2}{1 + \sqrt{1 + 4a^2}} dy^+ \quad (3.5.3)$$

where

$$a = \kappa y^+ [1 - \exp(-y^+/A^+)] \quad (3.5.4)$$

Figure 3.2 shows the resulting profile from Eq. (3.5.3). Note that for large y^+ , it reduces to that given by Eq. (3.5.2). The exponential factor is often called the “damping function.”

In the outer region of a boundary layer in zero pressure gradient, the only length and velocity scales are δ and u_τ , and the velocity profile can be expressed as

$$\frac{u_e - u}{u_\tau} = f_1\left(\frac{y}{\delta}\right) \quad (3.5.5)$$

which is known as the velocity-defect law (see Fig. 3.3), valid for $y u_\tau / \nu > 30$. For zero pressure gradient flows, the function f_1 has been found to be independent of Reynolds number (except at Reynolds numbers $R_\theta \equiv u_e \theta / \nu < 5000$,

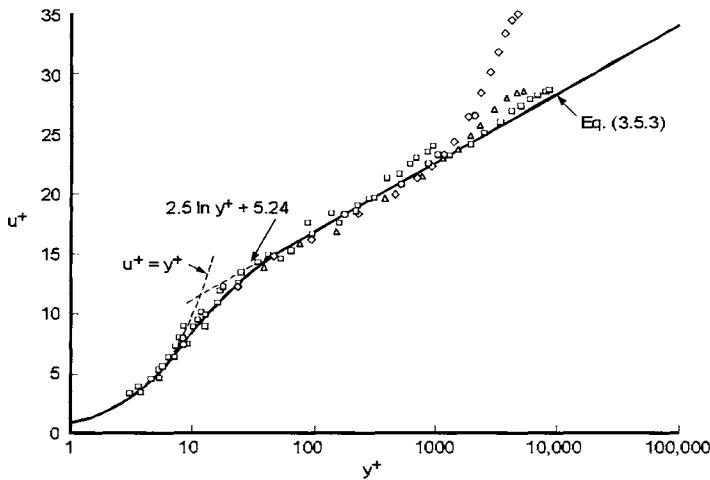


Fig. 3.2. Dimensionless velocity distribution in the inner layer.

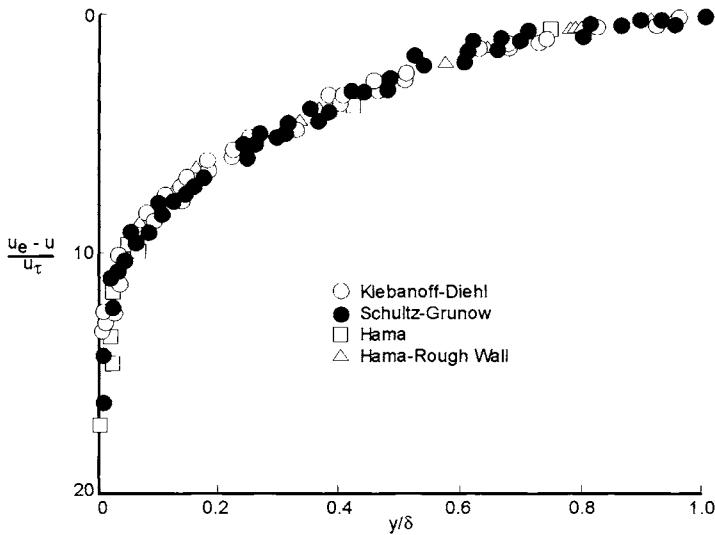


Fig. 3.3. Universal plot of turbulent velocity profiles in zero pressure gradient.

approximately) and, most significantly, independent of surface roughness and mass transfer.

The whole velocity profile in a turbulent boundary layer can be described by the formula proposed by Coles [20]

$$u^+ = f_1(y^+) + \frac{\Pi(x)}{\kappa} w(\eta) \quad (3.5.6)$$

where $\eta = y/\delta$, and the function $w(\eta)$ can be approximated by

$$w(\eta) = 1 - \cos(\pi\eta) \quad (3.5.7)$$

Here Π is a free parameter, depending on flow history in a way that is usually unknown, but constant and equal to about 0.55 in zero pressure gradient for $R_\theta > 5000$, so that Eq. (3.5.6) defines the function F_1 in Eq. (3.5.5).

Equation (3.5.6) gives $\partial u / \partial y$ nonzero at $y = \delta$. To remedy the difficulty, a number of expressions have been proposed. A convenient one proposed by Granville [21] uses a modification of Eq. (3.5.6) written as

$$\frac{u}{u_\tau} = \frac{1}{\kappa} [\ln y^+ + c + \Pi[(1 - \cos \pi\eta) + (\eta^2 - \eta^3)]] \quad (3.5.8)$$

From Eq. (3.5.8) and from the definitions of δ^* and θ it can be shown that

$$\frac{\delta^*}{\delta} = \int_0^1 \frac{u_e - u}{u_\tau} \frac{u_\tau}{u_e} d\eta = \frac{u_\tau}{\kappa u_e} \left(\frac{11}{12} + \Pi \right), \quad (3.5.9a)$$

$$\begin{aligned} \frac{\theta}{\delta} = \int_0^1 \frac{u}{u_e} \left(1 - \frac{u}{u_e} \right) d\eta &= \frac{u_\tau}{\kappa u_e} \left(\frac{11}{12} + \Pi \right) - \left(\frac{u_\tau}{\kappa u_e} \right)^2 \left\{ 2 + 2\Pi \left[1 + \frac{1}{\pi} \text{Si}(\pi) \right] \right. \\ &\quad \left. + 1.5\Pi^2 + \frac{1}{105} - \frac{7}{72} - 0.12925\Pi \right\} \end{aligned} \quad (3.5.9b)$$

From Eq. (3.5.9b), taking $\text{Si}(\pi) = 1.8519$, write

$$\frac{R_\theta}{R_\delta} = \frac{u_\tau}{\kappa u_e} \left(\frac{11}{12} + \Pi \right) - \left(\frac{u_\tau}{\kappa u_e} \right)^2 (1.9123016 + 3.05603\Pi + 1.5\Pi^2)$$

Evaluating Eq. (3.5.8) at $\eta = 1$,

$$\frac{u_e}{u_\tau} = \frac{1}{\kappa} \left[\ln \left(\frac{\delta u_e}{\nu} \frac{u_\tau}{u_e} \right) + c + 2\Pi \right] \quad (3.5.10)$$

For given values of c_f and R_θ , Eqs. (3.5.9b) and (3.5.10) can be solved for δ and Π so that the streamwise profile u can be obtained from Eqs. (3.5.8) and (3.5.9a) in the region $y^+ > 30$, with $f_1(y^+)$ given by Eq. (3.5.2).

The expression given by Eq. (3.5.6) can be extended in several ways. One method uses Eq. (3.5.3) and adds the wake component

$$\frac{\Pi(x)}{\kappa} w(\eta)$$

to it. Another method uses an expression due to Thompson [3]

$$u^+ = \begin{cases} y^+, & y^+ \leq 4 \end{cases} \quad (3.5.11a)$$

$$c_1 + c_2 \ln y^+ + c_3 (\ln y^+)^2 + c_4 (\ln y^+)^3, \quad 4 < y^+ < 50 \quad (3.5.11b)$$

where $c_1 = 1.0828$, $c_2 = -0.414$, $c_3 = 2.2661$, $c_4 = -0.324$.

Some models also require the initial shear stress profile τ . To generate this, one can use an expression given in [3] modified for the pressure gradient effect:

$$\frac{\tau}{\tau_w} = 1 - \frac{2}{c_f} \frac{1}{u_e^2} \left[u_e \frac{du_e}{dx} y + u \int_0^y \frac{\partial u}{\partial x} dy - 2 \int_0^y u \frac{\partial u}{\partial y} dy \right] \quad (3.5.12)$$

Following [3], assume

$$u = u_e g(\eta), \quad \delta/\theta = \text{const.} \quad (3.5.13)$$

and from the momentum integral equation, Eq. (2.4.36), we write

$$u_e^2 \frac{d\theta}{dx} = \frac{\tau_w}{\varrho} - (2\theta + \delta^*) u_e \frac{du_e}{dx} \quad (3.5.14)$$

Using Eqs. (3.5.13) and (3.5.14), it can be shown that Eq. (3.5.12) can be written as

$$\begin{aligned} \frac{\tau}{\tau_w} &= 1 + \frac{\delta}{\theta} \left(\int_0^\eta g^2 d\eta - g \int_0^\eta g d\eta \right) \\ &\quad - \frac{2}{c_f} \frac{\delta}{u_e} \frac{du_e}{dx} \left[\eta - g \int_0^\eta g d\eta + \frac{\delta^*}{\theta} \left(\int_0^\eta g^2 d\eta - g \int_0^\eta g d\eta \right) \right] \end{aligned} \quad (3.5.15)$$

Once the streamwise velocity profile u and shear stress τ are computed by the expressions discussed above, the dissipation rate ε can be computed from the assumption

$$\frac{\tau}{\varrho} \frac{du}{dy} = \varepsilon \quad (3.5.16)$$

which is accurate in the log-law region and adequate in the rest of a boundary layer.

References

- [1] Wilcox, D. C.: *Turbulence Modeling for CFD*. DCW Industries, Inc., 5354 Palm Drive, La Cañada, Calif., 1998.
- [2] Cebeci, T.: *Turbulence Models and Their Application*. Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 2003.
- [3] Cebeci, T.: *Analysis of Turbulent Flows*. Elsevier, London, 2003.
- [4] Durbin, P. A. and Petterson Reif, B. A.: Statistical Theory and Modeling for Turbulent Flows. John Wiley and Sons, New York, 2001.
- [5] Prandtl, L.: Über ein neues Formelsystem für die ausgebildete Turbulenz, Nachrichten von der Akad. der Wissenschaft in Göttingen, 1945.
- [6] Boussinesq, J.: Théorie de l'écoulement tourbillant, *Mém. prés. Acad. Sc.* **XXIII**, 46, 1877.
- [7] Kolmogorov, A.N.: Equations of Turbulent Motion of an Incompressible Fluid, *Izv. Akad. Nauk. SSR Ser. Phys.* **6**, 56, 1942. (English translation, Imperial College, Mech. Eng. Dept. Report ON/6, 1968).

- [8] Rotta, J.C.: Turbulent Boundary Layers in Incompressible Flow, *Prog. Aero. Sci.* **2**, 1, 1962.
- [9] Spalart, P. R. and Allmaras, S. R.: A One-Equation Turbulence Model for Aerodynamics Flows. AIAA Paper 92-0439, 1992.
- [10] Launder, B. E., Reece, G. J. and Rodi, W.: Progress in the Development of a Reynolds Stress Turbulence Closure. *J. Fluid Mech.* **20**, 3, 1975.
- [11] Bradshaw, P.: The Best Turbulence Models for Engineers. Modeling of Complex Flows (M. D. Salas, J. N. Hefner and L. Sakell, eds.) Kluwer, Dordrecht, 1999.
- [12] Fiedler, H. and Head, M. R.: Intermittency Measurements in the Turbulent Boundary Layer, *J. Fluid Mech.* **25**, 719–735, 1986.
- [13] Baldwin, B. S. and Lomax, H.: Thin Layer Approximation of Algebraic Model for Separated Turbulent Flows. AIAA paper No. 78-257, 1978.
- [14] Stock, H. W. and Haase, W.: Determination of Length Scales in Algebraic Turbulence Models for Navier-Stokes Methods. *AIAA J.*, **27**, No. 1, 5–14, 1989.
- [15] Johnson, D. A.: Nonequilibrium Algebraic Turbulence Modeling Considerations for Transonic Airfoils and Wings. AIAA paper No. 92-0026, 1992.
- [16] Jones, W. P. and Launder, B. E.: The Prediction of Laminarization with a Two-Equation Model of Turbulence. *Int. J. Heat and Mass Transfer* **15**, 301–314, 1972.
- [17] Menter, F. R.: Two-Equation Eddy Viscosity Turbulence Models for Engineering Applications. *AIAA J.* **32**, 1299–1310, 1994.
- [18] Launder, B. E. and Sharma, B. I.: Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc. *Letters in Heat and Mass Transfer* **1**, 131–138, 1974.
- [19] Van Driest, E. R.: On Turbulent Flow Near a Wall. *J. Aeronaut. Sci.*, **23**, 1956.
- [20] Coles, D.: The Law of the wake in the turbulent Boundary Layer, *J. Fluid Mech.* **1**, 191, 1956.
- [21] Granville, P. S.: A Modified Law of the Wake for Turbulent Shear Layers, *J. Fluids Engineering*, Sept 1976.

4 Numerical Methods for Model Parabolic and Elliptic Equations

4.1 Introduction

The equations of Chapters 2 and 3 are of little value unless they can be solved. Since known analytical methods are limited to some special flows such as fully developed laminar flows in ducts, numerical methods are required. These methods are invariably implemented on digital computers, so this chapter provides an introduction to computational fluid dynamics and to the numerical methods to be discussed in the subsequent chapters.

The numerical solution of the Navier–Stokes and Euler equations can be obtained by using finite-difference and finite-volume methods. Their numerical solution can also be obtained by finite-element methods. However, these methods are mainly used in computational structural mechanics and are yet to gain popularity in computational fluid dynamics. Thus, this book considers only finite-difference and finite-volume methods. The finite-difference methods are mainly used with the differential form of the conservation equations, and the finite-volume methods are used in conjunction with the integral form of the conservation equations. For reduced forms of the Navier–Stokes equations, such as the boundary-layer equations and the Orr–Sommerfeld equation which involve the numerical solution of ordinary and partial-differential equations, only finite-difference methods are considered. The finite-volume methods will be considered in Chapter 12 in connection with the numerical solution of the Navier–Stokes and Euler equations together with the finite-difference methods.

Model equations for the conservation equations are considered in Section 4.2. These equations are useful to “model” the behavior of the more complete and complicated partial-differential equations considered in subsequent chapters. In this chapter they are used to describe the numerical solution of parabolic and elliptic equations and in the following chapter the numerical solution of hyperbolic equations.

The numerical solution of the parabolic and elliptic equations with finite-difference methods are discussed in Sects. 4.4 and 4.5, respectively, following a brief description of the discretization of derivatives with finite-differences in Section 4.3. The methods in Section 4.4 and Chapter 5 can be either implicit and explicit in contrast to the numerical methods for elliptic equations in Section 4.5 which are always implicit.

4.2 Model Equations

A general idea about the model equations used in computational fluid dynamics can be obtained by considering the simplified form of the conservation equations for one- and two-dimensional flows. For example, for *one-dimensional* flows neglecting the body force in Eq. (2.2.18) and heat transfer term q_h in Eq. (2.2.15b), Eqs. (2.2.12b), (2.2.18) and (2.2.15b) can be written as

$$\frac{\partial \varrho}{\partial t} + \frac{\partial}{\partial x}(\varrho u) = 0 \quad (4.2.1)$$

$$\frac{\partial}{\partial t}(\varrho u) + \frac{\partial}{\partial x}(\varrho u^2 + p) = \frac{\partial \sigma_{xx}}{\partial x} = \frac{4}{3} \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) \quad (4.2.2)$$

$$\varrho \left(\frac{\partial e}{\partial t} + u \frac{\partial e}{\partial x} \right) = -p \frac{\partial u}{\partial x} + \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \sigma_{xx} \frac{\partial u}{\partial x} \quad (4.2.3)$$

If the terms containing the velocity u in Eq. (4.2.3) are neglected by assuming that either the fluid is at rest or the flow velocity is small, then the energy equation for a constant thermal conductivity k and specific heat C_v can be written as

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (4.2.4)$$

This equation is known as the one-dimensional *unsteady heat conduction* equation and is a typical parabolic equation in time in the (x, t) space.

If the velocity u in the continuity equation (4.2.1) is assumed constant, then this equation takes the typical form of a *linear convection* equation

$$\frac{\partial \varrho}{\partial t} + u \frac{\partial \varrho}{\partial x} = 0 \quad (4.2.5)$$

describing the transport of mass by a flow velocity of u . It is a first-order hyperbolic equation.

If the velocity is not negligible but the flow is *incompressible*, it follows from the continuity equation that u is constant. The energy equation (4.2.3) becomes

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (4.2.6)$$

This equation is still parabolic in time in the (x, t) space and is called the *convection-diffusion* equation due to the simultaneous presence of convection by the velocity u and diffusion through the diffusivity coefficient α .

If the effect of the pressure gradient on the momentum equation is neglected, Eq. (4.2.2) becomes

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (4.2.7)$$

which has the structure of a convection-diffusion equation but is nonlinear. It is known as Burger's equation, and it contains the full nonlinearity of the one-dimensional flow equations. For an inviscid flow, it becomes

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 \quad (4.2.8)$$

and is known as the “inviscid” Burger’s equation. Unlike Eq. (4.2.7), which is parabolic, this equation is hyperbolic. It is also nonlinear, as is Eq. (4.2.7).

The discussion on constructing model equations for simplified one-dimensional flows can also be extended to simplified two-dimensional flows. For this purpose, for two-dimensional inviscid flows, the Laplace equation given by Eq. (2.4.20)

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (4.2.9)$$

which is a typical elliptic equation, describes an isotropic diffusion in the (x, y) space.

A similar equation results from Eq. (2.2.11), which is a parabolic in time in the (x, y, t) space. Assuming that the temperature field is in a medium at rest, then in the presence of the source term \dot{q}_h , Eq. (2.2.11) reduces to the Poisson equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = -\frac{1}{k} \dot{q}_h \quad (4.2.10a)$$

An equation similar to this is obtained from Eq. (2.2.9) for a fully developed laminar channel flow in a rectangular duct. Noting that Du/Dt is zero, and neglecting the body force,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{1}{\mu} \frac{dp}{dx} \quad (4.2.10b)$$

If the temperature in Eq. (4.2.10a) is time dependent, and the medium is again at rest, then the unsteady heat conduction equation in two dimensions and with source term becomes

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + \dot{q}_h \quad (4.2.11)$$

This equation is parabolic in time in the (x, y, t) space. Its solutions at large time values should approach the solutions of Eq. (4.2.10a).

4.3 Discretization of Derivatives with Finite Differences

Before the finite-difference methods for parabolic, hyperbolic and elliptic equations are described, it is useful to discuss the discretization of derivatives (either ordinary or partial) with finite differences. For this purpose, consider a function w which is single-valued, finite and continuous functions of ζ . Using Taylor's theorem and with primes denoting differentiation with respect to ζ , we can write

$$w(\zeta + r) = w(\zeta) + rw'(\zeta) + \frac{1}{2}r^2w''(\zeta) + \frac{1}{6}r^3w'''(\zeta) + \dots \quad (4.3.1a)$$

and

$$w(\zeta - r) = w(\zeta) - rw'(\zeta) + \frac{1}{2}r^2w''(\zeta) - \frac{1}{6}r^3w'''(\zeta) + \dots \quad (4.3.1b)$$

Adding both equations,

$$w(\zeta + r) + w(\zeta - r) = 2w(\zeta) + r^2w''(\zeta)$$

provided the fourth- and higher-order terms are neglected. Thus,

$$w''(\zeta) = \frac{1}{r^2}[w(\zeta + r) - 2w(\zeta) + w(\zeta - r)] \quad (4.3.2)$$

with an error of order r^2 , $O(r^2)$.

If Eq. (4.3.1b) is subtracted from Eq. (4.3.1a),

$$w'(\zeta) = \frac{1}{2r}[w(\zeta + r) - w(\zeta - r)] \quad (4.3.3)$$

Equation (4.2.3) approximates the slope of the tangent at P by the slope of the chord AB (see Fig. 4.1) and is called a central-difference approximation. The slope of the tangent can also be approximated by either the slope of the chord PB , giving the *forward-difference* formula

$$w'(\zeta) = \frac{1}{r}[w(\zeta + r) - w(\zeta)] \quad (4.3.4)$$

or the slope of the chord AP , giving the *backward-difference* formula

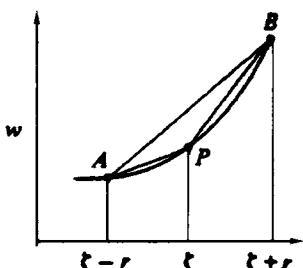


Fig. 4.1. Notation for approximation of derivatives.

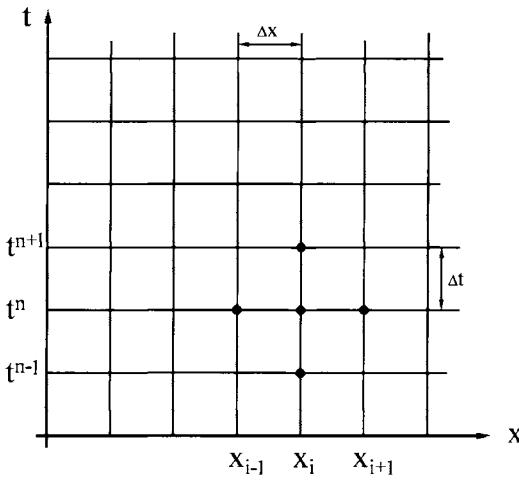


Fig. 4.2. Finite-difference grid notation.

$$w'(\zeta) = \frac{1}{r}[w(\zeta) - w(\zeta - r)] \quad (4.3.5)$$

Note that while Eq. (4.3.3) has an error of $O(r^2)$, both Eqs. (4.3.4) and (4.3.5) have errors of $O(r)$.

Finite-difference methods require a discrete set of points, covering the flow-field. These points are called a net, grid, lattice or mesh. To demonstrate the finite-difference notation for the above derivatives, let us consider Fig. 4.2, which shows a set of uniform net points on the (x, t) plane, that is,

$$\begin{aligned} t_0 &= 0, & t_n &= t_{n-1} + \Delta t, & n &= 1, 2, \dots, N \\ x_0 &= 0, & x_i &= x_{i-1} + \Delta x, & i &= 1, 2, \dots, I \end{aligned} \quad (4.3.6)$$

The quantities Δt and Δx are called the net spacings or mesh widths. A quantity corresponding to each point of the net T_i^n , is sought to approximate $T_i^n \equiv T(t_n, x_i)$, the exact solution at the corresponding point. With this notation, using central-difference, forward-difference and backward-difference formulas, the difference approximation of $\partial T / \partial t$ follows from Eqs. (4.3.3) to (4.3.5) and may be written, respectively, as

$$\frac{1}{2\Delta t}(T_i^{n+1} - T_i^{n-1}) \quad (\text{central difference}) \quad (4.3.7)$$

$$\frac{\partial T}{\partial t}(t_n, x_i) = \frac{1}{\Delta t}(T_i^{n+1} - T_i^n) \quad (\text{forward difference}) \quad (4.3.8)$$

$$\frac{1}{\Delta t}(T_i^n - T_i^{n-1}) \quad (\text{backward difference}) \quad (4.3.9)$$

Similarly, the difference approximation of $\partial^2 T / \partial x^2$ follows from Eq. (4.3.2) and may be written as

$$\frac{\partial^2 T}{\partial x^2}(t_n, x_i) = \frac{1}{\Delta x^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (4.3.10)$$

4.4 Finite-Difference Methods for Parabolic Equations

The model equation used here to illustrate numerical methods for solving parabolic partial-differential equations is the one-dimensional unsteady heat conduction equation given by Eq. (4.2.4), which also serves as a model equation for the boundary-layer equations, to be discussed in detail in Chapter 7.

Equation (4.2.4) requires boundary and initial conditions. For simplicity, as shown in Fig. 4.3, assume that the boundary conditions at $x = 0$ and $x = L$ are given by

$$x = 0, \quad T = T_1(t); \quad x = L, \quad T = T_2(t) \quad (4.4.1)$$

and the initial conditions by

$$t = 0, \quad T = T_0(x) \quad (4.4.2)$$

The solution of Eq. (4.2.4) may be obtained by using either an explicit or an implicit method. In an explicit method, the value of T at the next time step, t^{n+1} , is expressed in terms of T at the previous time step, t^n , and the corresponding equation is solved explicitly at each grid point. In an implicit method, T at the next time step is expressed in terms of its neighboring points at t^{n+1} and the known quantities at t^n , and its solution for all grid points on the time step, t^{n+1} , is obtained simultaneously.

4.4.1 Explicit Methods

In the solution of Eq. (4.2.4) by an explicit method, $\partial T / \partial t$ may be represented by the forward difference formula, Eq. (4.3.8), and $\partial^2 T / \partial x^2$ by Eq. (4.3.10), centering at the *net point* (t_n, x_i) (see Fig. 4.4), that is,

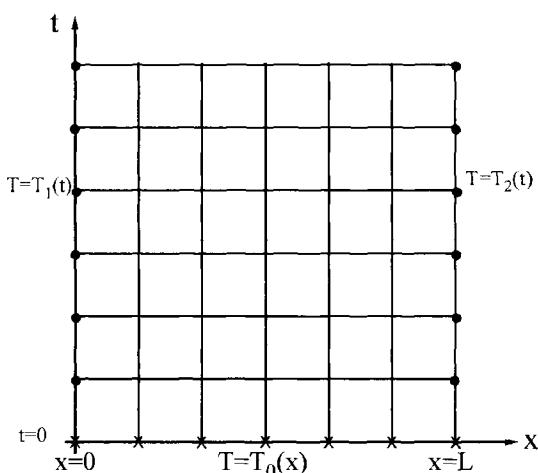


Fig. 4.3. Initial and boundary conditions of Eq. (4.2.4) in the (x, t) plane. Symbols \times denote values known from initial conditions and symbols \bullet denote values known from boundary conditions.

$$\frac{1}{\Delta t}(T_i^{n+1} - T_i^n) = \alpha \frac{1}{\Delta x^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (4.4.3a)$$

or

$$T_i^{n+1} = T_i^n + \alpha \frac{\Delta t}{\Delta x^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n), \quad i = 1, 2, \dots, I-1 \quad (4.4.3b)$$

From Eq. (4.4.3b) it is seen that, by this explicit formulation, the value of T_i^{n+1} is expressed in terms of previous time values that are known, and the equation allows the value of T to be obtained for $i = 1, 2, \dots, I-1$. The values of T at $i = 0$ and I are known from the boundary conditions. The numerical error inherent in this scheme can be shown to be of order $\Delta t + \Delta x^2$ and, as a result, the time step Δt must be kept small to ensure acceptable accuracy. In addition, although explicit formulations are computationally simple, they can lead to numerical instabilities unless the time step is also small. As is shown in Section 5.7, in order to avoid the growth of errors in the operations for solving Eq. (4.4.3), $\frac{\alpha \Delta t}{\Delta x^2}$ must be $< 1/2$.

Example 4.1. Solve Eq. (4.2.4) subject to the following boundary and initial conditions

$$x = 0, \quad T = 0; \quad x = 1, \quad T = 0$$

$$t = 0 \quad T = \begin{cases} 2x & 0 \leq x \leq \frac{1}{2} \\ 2(1-x) & \frac{1}{2} \leq x < 1 \end{cases}$$

by the above explicit method for values of $t = 0.005, 0.01, 0.02, 0.10$ with $\alpha = 1$ for three different spacings in t ,

$$(a) \Delta t = \frac{1}{1000}, \quad (b) \Delta t = \frac{5}{1000}, \quad (c) \Delta t = \frac{1}{100}$$

Compare your solutions with the analytical solution

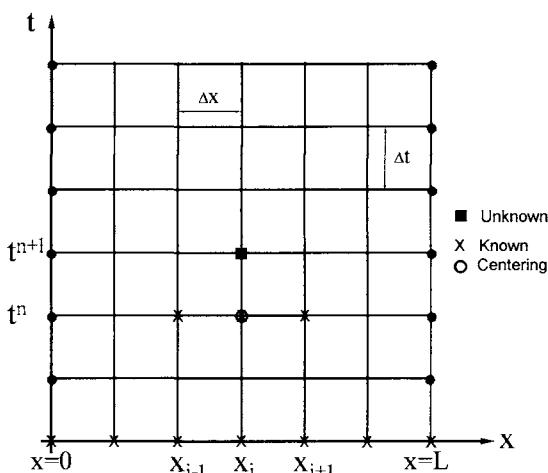


Fig. 4.4. Finite-difference grid for an explicit method.

$$T = \frac{8}{\pi^2} \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2} e^{-(2n+1)^2 \pi^2 t} \sin((2n+1)\pi x) \quad (\text{E4.4.1})$$

at $x = 0.3$ for $t = 0, 0.005, 0.010, 0.020, 0.100$ and determine the percentage error in each case. Discuss the accuracy of the results, the behavior of the solutions and the importance of the ratio $\Delta t/(\Delta x)^2$ in each case.

Solution.

(a) With $\Delta t = 0.001$, take $\Delta x = 0.10$, so $r = \Delta t/\Delta x^2 = 0.10$.

Table E4.1 presents a comparison of finite difference solution (FDS) at $x = 0.30$ obtained with the computer program using the explicit method (see Appendix A, Chapter 4, Example E4.1) with analytical solution (AS). As can be seen, FDS is reasonably accurate. The percentage error is the difference of the solution expressed as a percentage of the analytical solution of Eq. (E4.4.1).

Table E4.1. Comparison of FDS and AS at $x = 0.30$ for $r = 0.10$

t	FDS	AS	Diff	%Error
.000	.6000	.6004	-.0004	-.0006
.005	.5971	.5966	.0005	.0008
.010	.5822	.5799	.0023	.0040
.020	.5373	.5334	.0039	.0073
.100	.2472	.2444	.0028	.0116

The comparison at $x = 0.5$ is not quite so good because of the discontinuity in the initial value of $\partial u/\partial x$, from +2 to -2, at this point. Inspection of Table E4.2 shows, however, that the effect of this discontinuity dies away as t increases.

Table E4.2. Comparison of FDS and AS at $x = 0.5$ for $r = 0.10$

t	FDS	AS	Diff	%Error
.000	1.0000	.9904	.0096	.0097
.005	.8597	.8404	.0193	.0230
.010	.7867	.7743	.0124	.0160
.020	.6891	.6808	.0083	.0122
.100	.3056	.3021	.0035	.0116

(b) With $\Delta t = \frac{5}{1000}$, take $\Delta x = \frac{1}{10}$, so $r = 0.5$.

A comparison of results in Table E4.3 indicates that the FDS is not as good an approximation to Eq. (E4.4.1) as the previous one; nevertheless it would be adequate for most engineering problems.

(c) With $\Delta t = \frac{1}{100}$, take $\Delta x = \frac{1}{10}$, so $r = 1$.

The results presented in Table E4.4 at several x -locations at different t -intervals show that the finite-difference solutions are not acceptable.

These three cases clearly indicate that the value of r is important, and as will be discussed later in Section 5.7, this explicit method is valid only when $0 < r \leq \frac{1}{2}$.

Table E4.3. Comparison of FDS and AS at $x = 0.30$ for $r = 0.5$

t	FDS	AS	Diff	%Error
.000	.6000	.6004	-.0004	-.0006
.005	.6000	.5966	.0034	.0057
.010	.6000	.5799	.0201	.0347
.020	.5500	.5334	.0166	.0312
.100	.2484	.2444	.0040	.0165

Table E4.4. FDS at several y -locations for $r = 1$

t	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6
.0000	.200	.400	.600	.800	1.000	.800
.0100	.200	.400	.600	.800	.600	.800
.0200	.200	.400	.600	.400	1.000	.400
.0300	.200	.400	.200	1.200	-.200	1.200
.0400	.200	.000	1.400	-1.200	2.600	-1.200

In some problems the boundary conditions are expressed in terms of derivatives rather than in terms of T , as in Eq. (4.4.1). In those cases the solution of Eq. (4.2.4) by an explicit or implicit method requires additional work. To illustrate this, consider Example 4.1 with the boundary conditions of the form

$$i = 0, \quad \frac{\partial T}{\partial x} = T \quad (4.4.4a)$$

$$i = I, \quad \frac{\partial T}{\partial x} = -T \quad (4.4.4b)$$

They may be expressed in terms of central differences, Eq. (4.3.7), or by a forward difference, Eq. (4.3.8). With the choice of central differences, Eq. (4.4.4a) can be written as

$$\frac{T_1^n - T_{-1}^n}{2\Delta x} = T_0^n \quad (4.4.5)$$

At $i = 0$, Eq. (4.4.3b) becomes

$$T_0^{n+1} = T_0^n + \alpha \frac{\Delta t}{(\Delta x)^2} (T_1^n - 2T_0^n + T_{-1}^n) \quad (4.4.6)$$

Eliminating T_{-1}^n between Eqs. (4.4.5) and (4.4.6) gives

$$T_0^{n+1} = T_0^n + 2\alpha \frac{\Delta t}{(\Delta x)^2} [T_1^n - (1 + \Delta x)T_0^n] \quad (4.4.7)$$

At $i = I$, Eq. (4.4.3b) becomes

$$T_I^{n+1} = T_I^n + \alpha \frac{\Delta t}{(\Delta x)^2} (T_{I+1}^n - 2T_I^n + T_{I-1}^n)$$

In terms of central differences, the boundary condition at $i = I$ can be written as

$$\frac{T_{I+1}^n - T_{I-1}^n}{2\Delta x} = -T_I^n \quad (4.4.8)$$

so that, similar to Eq. (4.4.7). Eq. (4.4.8) can be written as

$$T_I^{n+1} = T_I^n + 2\alpha \frac{\Delta t}{(\Delta x)^2} [T_{I-1}^n - (1 + \Delta x)T_I^n] \quad (4.4.9)$$

This result could have been deduced from the corresponding equation at $x = 0$ because of symmetry with respect to $x = 1/2$.

Example 4.2. Solve Eq. (4.2.4) subject to the following boundary and initial conditions:

$$x = 0, \quad \frac{\partial T}{\partial x}; \quad x = 1, \quad \frac{\partial T}{\partial x} = -T$$

$$t = 0, \quad T = 1, \quad 0 \leq x \leq 1$$

using

- (a) an explicit method and employing central differences for the boundary conditions,
- (b) an explicit method and employing a forward difference for the boundary condition at $x = 0$.

Compare the numerical results obtained in each case with the analytical solution given by

$$T = 4 \sum_{n=1}^{\infty} \left[\frac{\sec \alpha_n}{(3 + 4\alpha_n^2)} e^{-4\alpha_n^2 t} \cos 2\alpha_n \left(x - \frac{1}{2} \right) \right] \quad 0 < x < 1 \quad (\text{E4.2.1})$$

where α_n are the positive roots of $\alpha \tan \alpha = \frac{1}{2}$. Take $\alpha = 1$, $\Delta t = 0.001$ and $\Delta x = 0.1$.

Solution.

Table E4.5 presents a comparison of FDS of the explicit method with AS when central differences are employed on the boundary conditions. Table E4.6 shows a similar comparison when forward differences are employed on the boundary conditions.

(a)

Table E4.5. Comparison of FDS and AS at $x = 0.30$.

t	FDS	AS	Diff	%Error
0.00	1	1.0026	-0.0026	-0.0026
0.01	0.9984	0.9984	0.0001	0.0001
0.05	0.9475	0.9467	0.0008	0.0008
0.10	0.8719	0.8713	0.0006	0.0007
0.50	0.4401	0.4403	-0.0002	-0.0005
1.00	0.1871	0.1875	-0.0004	-0.0021

(b)

Table E4.6. Comparison of FDS and AS at $x = 0.30$.

t	FDS	AS	Diff	%Error
0.00	1	1.0026	-0.0026	-0.0026
0.01	0.996	0.9984	-0.0024	-0.0024
0.05	0.9287	0.9467	-0.018	-0.019
0.10	0.8458	0.8713	-0.0255	-0.0292
0.50	0.403	0.4403	-0.0374	-0.0848
0.80	0.2313	0.2638	-0.0326	-0.1235
1.00	0.1597	0.1875	-0.0278	-0.1483

4.4.2 Implicit Methods: Crank–Nicolson

In contrast, implicit methods are unconditionally stable and allow significantly larger time steps, with corresponding economy, as long as accuracy is maintained. For a parabolic partial differential, two popular implicit finite-difference methods are due to Crank–Nicolson [1] and Keller [2]. Keller’s method is discussed in subsection 4.4.3 for Eq. (4.2.4) and later in Chapters 7 and 8 for boundary-layer and stability equations.

In this subsection, we describe the Crank–Nicolson method for Eq. (4.2.4). This method uses the finite-difference grid shown in Fig. 4.5, where $\partial^2 T / \partial x^2$ at $(t^{n+1/2}, x_i)$ is expressed by the average of the previous and current time values at t^n and t^{n+1} , respectively,

$$\frac{\partial^2 T}{\partial x^2}(t^{n+1/2}, x_i) = \frac{\alpha}{2} \left[\left(\frac{\partial^2 T}{\partial x^2} \right)_i^{n+1} + \left(\frac{\partial^2 T}{\partial x^2} \right)_i^n \right] \quad (4.4.10a)$$

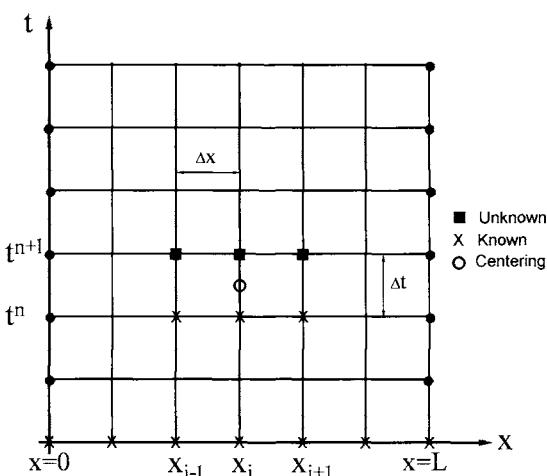


Fig. 4.5. Finite-difference grid for the Crank–Nicolson method. Note that while Δx is uniform, Δt can be nonuniform.

and $\partial T / \partial t$ is expressed by the central-difference formula of Eq. (4.3.7),

$$\frac{\partial T}{\partial t}(t^{n+1/2}, x_i) = \frac{T_i^{n+1} - T_i^n}{\Delta t} \quad (4.4.10b)$$

Before considering the solution of Eq. (4.2.4) using this finite-difference approximation, note that a more general finite-difference approximation to this equation is

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \left[\theta \left(\frac{\partial^2 T}{\partial x^2} \right)_i^{n+1} + (1 - \theta) \left(\frac{\partial^2 T}{\partial x^2} \right)_i^n \right] \quad (4.4.11)$$

where in practice $0 \leq \theta \leq 1$. $\theta = 0$ gives the explicit scheme, $\theta = 1/2$ the Crank–Nicolson method, and $\theta = 1$ a fully implicit backward time-difference method. The time differencing in Eq. (4.4.11) is known as the trapezoidal differencing scheme as well as Crank–Nicolson differencing. The equations are unconditionally stable and convergent for $1/2 \leq \theta \leq 1$, but for $0 \leq \theta < 1/2$

$$\frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2(1 - 2\theta)} \quad (4.4.12)$$

For $\theta = 1/2$, Eq. (4.2.4) can be written as

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

or as

$$a_i T_{i-1}^{n+1} + b_i T_i^{n+1} + c_i T_{i+1}^{n+1} = r_i, \quad 1 \leq i \leq I - 1 \quad (4.4.13)$$

where

$$a_i = 1, \quad b_i = -(2 + \lambda), \quad c_i = 1$$

$$r_i = -\lambda T_i^n - (T_{i+1}^n - 2T_i^n + T_{i-1}^n), \quad 1 \leq i \leq I - 1, \quad (4.4.14)$$

$$\lambda = \frac{2}{\alpha} \frac{\Delta x^2}{\Delta t}$$

Note that r_i contains only values of T^n , not T^{n+1} . At $i = 0$ and I , the boundary conditions T_0^{n+1} and T_I^{n+1} are given by Eq. (4.4.1). As a result, for $i = 1$ and $i = I - 1$, Eq. (4.4.13) can be written as

$$b_1 T_1^{n+1} + c_1 T_2^{n+1} = r_1 - a_1 T_0^{n+1} \equiv r_1^* \quad \text{for } i = 1 \quad (4.4.15a)$$

and

$$a_{I-1} T_{I-2}^{n+1} + b_{I-1} T_{I-1}^{n+1} = r_{I-1} - c_{I-1} T_I^{n+1} \equiv r_{I-1}^* \quad \text{for } i = I - 1 \quad (4.4.15b)$$

With the boundary conditions given by Eq. (4.4.4), again use Eqs. (4.4.5) and (4.4.8). For $i = 0$, it follows from Eq. (4.4.5) that

$$T_{-1}^n = T_1^n - 2(\Delta x)T_0^n \quad (4.4.16a)$$

and

$$T_{-1}^{n+1} = T_1^{n+1} - 2(\Delta x)T_0^{n+1} \quad (4.4.16b)$$

The last two equations permit T_{-1}^n and T_{-1}^{n+1} to be eliminated from the equation obtained by setting $i = 0$ in Eq. (4.4.13); the resulting equation can be written in a form similar to Eq. (4.4.15a),

$$(b_0 - a_0 2\Delta x)T_0^{n+1} + (a_0 + c_0)T_1^{n+1} = r_0 \quad (4.4.16c)$$

The boundary condition at $i = I$ can be dealt in the same way, although in this problem it is easier to make use of the symmetry with respect to $x = 1/2$.

The errors in this carefully centered Crank–Nicolson scheme are of order $\Delta x^2 + \Delta t^2$, but Δt need not be related to Δx for stability purposes. The scheme is unconditionally stable, and second-order accuracy may be achieved with uniform x spacing. On the other hand, the unknown value of T is expressed in terms of I other values of T with two of them known; as a consequence, the computational arithmetic is more extensive than for an explicit method. However, the solution of Eqs. (4.4.13) to (4.4.15) can be obtained easily as in the case when the boundary conditions correspond to those given by Eq. (4.4.4).

In vector notation, Eqs. (4.4.13) and (4.4.15) may be written as

$$A \underline{T} = \underline{r} \quad (4.4.17)$$

where the $(I - 1)$ -dimensional vectors are

$$\underline{T} \equiv \begin{vmatrix} T_1 \\ T_2 \\ \vdots \\ T_{I-2} \\ T_{I-1} \end{vmatrix} \quad \underline{r} \equiv \begin{vmatrix} r_1^* \\ r_2 \\ \vdots \\ r_{I-2} \\ r_{I-1}^* \end{vmatrix} \quad (4.4.18)$$

and the $(I - 1)$ -order matrix with nonzero elements on only three diagonals (called the *tridiagonal matrix*) is

$$A = \begin{vmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ 0 & & a_{I-2} & b_{I-2} & c_{I-2} \\ & & & a_{I-1} & b_{I-1} \end{vmatrix} \quad (4.4.19)$$

Then the solution of Eq. (4.4.19) can be obtained by the Thomas algorithm, which has two sweeps. In the so-called forward sweep, we compute

$$\beta_1 = b_1, \quad s_1 = r_1^*$$

Table 4.1. FORTRAN listing of subroutine THOMAS.

```

SUBROUTINE THOMAS(II,A,B,C,T,R)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(1),B(1),C(1),T(1),R(1)
DIMENSION BETA(201),S(201)
REAL*8 M(201)
BETA(1) = B(1)
S(1) = R(1)
DO I = 2,II
    M(I) = A(I)/BETA(I-1)
    BETA(I) = B(I)-M(I)*C(I-1)
    S(I) = R(I)-M(I)*S(I-1)
ENDDO
T(II) = S(II)/BETA(II)
DO I = II-1,1,-1
    T(I) = (S(I)-C(I)*T(I+1))/BETA(I)
ENDDO
RETURN
END

```

$$m_i = \frac{a_i}{\beta_{i-1}}$$

$$\beta_i = b_i - m_i c_{i-1}, \quad i = 2, 3, \dots, I-1$$

$$s_i = r_i - m_i s_{i-1} \quad (4.4.20)$$

In the backward sweep, we compute

$$T_{I-1}^{n+1} = \frac{s_{I-1}}{\beta_{I-1}}, \quad T_i^{n+1} = \frac{s_i - c_i T_{i+1}^{n+1}}{\beta_i}, \quad i = I-2, I-3, \dots, 1 \quad (4.4.21)$$

Table 4.1 gives the FORTRAN listing of the Thomas algorithm.

Example 4.3. Repeat Example 4.1 using the Crank–Nicolson method. Compare your results with those obtained with the explicit method.

Solution.

A listing of the computer program using the Crank–Nicolson method is given in Appendix A, Chapter 4, Example E4.3. The results are presented in Tables E4.7 and E4.8. Although this method is valid for all finite values of r , a large value will yield an inaccurate approximation for $\partial u / \partial t$.

The results in Table E4.7 are obtained for $\Delta x = 0.10$ and $\Delta t = 1/100$. The solutions are also compared with the analytical solutions. As can be seen, the numerical solution is clearly a good one.

Table E4.8 shows the solutions at various values of x . A glance at Table E4.4 shows that in this example the accuracy of this implicit method over the time-range taken is about the same as for the explicit method which uses ten times as many time steps.

Table E4.7. Comparison of FDS and AS at $x = 0.5$

t	FDS	AS	Diff	%Error
.000	1.0000	.9904	.0096	.0097
.010	.7692	.7743	-.0051	-.0066
.020	.6929	.6808	.0121	.0178
.030	.6175	.6091	.0083	.0137
.040	.5596	.5488	.0109	.0198
.050	.5078	.4959	.0119	.0240
.060	.4621	.4488	.0133	.0296
.070	.4207	.4064	.0143	.0351
.080	.3832	.3681	.0151	.0409
.090	.3491	.3335	.0156	.0468
.100	.3181	.3021	.0159	.0527

Table E4.8. FDS at several x -locations for $r = 1$

t	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6
.0000	.200	.400	.600	.800	1.000	.800
.0100	.213	.399	.584	.738	.769	.738
.0200	.210	.388	.543	.647	.693	.647
.0300	.200	.362	.496	.587	.617	.587
.0400	.186	.334	.454	.532	.560	.532
.0500	.171	.306	.414	.484	.508	.484
.0600	.156	.280	.377	.440	.462	.440
.0700	.143	.255	.344	.401	.421	.401
.0800	.130	.232	.313	.365	.383	.365
.0900	.119	.212	.286	.333	.349	.333
.1000	.108	.193	.260	.303	.318	.303

4.4.3 An Implicit Method: Keller's Box Method

Keller's method, often referred to as the box method has several very desirable features that make it appropriate for the solution of all parabolic partial differential equations. The main features of this method are

1. Only slightly more arithmetic to solve than the Crank–Nicolson method.
2. Second-order accuracy with arbitrary (nonuniform) t and x spacings.
3. Allows very rapid t variations.
4. Allows easy programming of the solution of large numbers of coupled equations

The solution of an equation by this method can be obtained by the following four steps:

1. Reduce the equation or equations to a first-order system.
2. Write difference equations using central differences.
3. Linearize the resulting algebraic equations (if they are nonlinear), and write them in matrix-vector form.
4. Solve the linear system by the block-tridiagonal-elimination method.

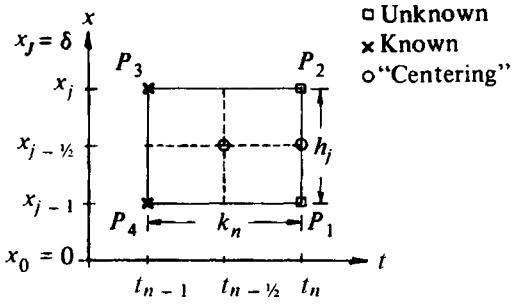


Fig. 4.6. Finite-difference grid for the box method. Note that both h and k can be nonuniform. Here $t_{n-1/2} \equiv 1/2(t_n + t_{n-1})$ and $x_{j-1/2} \equiv 1/2(x_j + x_{j-1})$.

To solve Eq. (4.2.4) by this method, we first express it in terms of a system of two first-order equations by letting

$$\frac{\partial T}{\partial x} = T' = p \quad (4.4.22a)$$

and by writing Eq. (4.2.4) as

$$\frac{\partial p}{\partial x} = p' = \frac{1}{\alpha} \frac{\partial T}{\partial t}. \quad (4.4.22b)$$

Here the primes denote differentiation with respect to x . The finite-difference form of the *ordinary* differential equation (4.4.22a) is written for the midpoint $(t_n, x_{j-1/2})$ of the segment P_1P_2 shown in Fig. 4.6, and the finite-difference form of the *partial* differential equation (4.4.22b) is written for the midpoint $(t_{n-1/2}, x_{j-1/2})$ of the rectangle $P_1P_2P_3P_4$. This gives

$$\frac{T_j^n - T_{j-1}^n}{h_j} = \frac{p_j^n + p_{j-1}^n}{2} = p_{j-1/2}^n, \quad (4.4.23a)$$

$$\frac{1}{2} \left(\frac{p_j^n - p_{j-1}^n}{h_j} + \frac{p_j^{n-1} - p_{j-1}^{n-1}}{h_j} \right) = \frac{1}{\alpha} \frac{T_{j-1/2}^n - T_{j-1/2}^{n-1}}{k_n} \quad (4.4.23b)$$

Rearranging both expressions we can write them in the form

$$T_j^n - T_{j-1}^n - \frac{h_j}{2}(p_j^n + p_{j-1}^n) = 0, \quad (4.4.24a)$$

$$(s_1)_j p_j^n + (s_2)_j p_{j-1}^n + (s_3)_j (T_j^n + T_{j-1}^n) = R_{j-1/2}^{n-1}. \quad (4.4.24b)$$

Here

$$(s_1)_j = 1, \quad (s_2)_j = -1, \quad (s_3)_j = -\lambda_j, \quad (4.4.25a)$$

$$R_{j-1/2}^{n-1} = -2\lambda_j T_{j-1/2}^{n-1} + p_{j-1}^{n-1} - p_j^{n-1}, \quad (4.4.25b)$$

$$\lambda_j = \frac{1}{\alpha} \frac{h_j}{k_n}. \quad (4.4.25c)$$

Equations (4.4.24) are imposed for $j = 1, 2, \dots, J - 1$. At $j = 0$ and J , we have

$$T_0 = T_w, \quad T_J = T_e, \quad (4.4.26)$$

respectively.

Since Eqs. (4.4.23) are linear, as are the corresponding boundary conditions (b.c.) given by Eq. (4.4.26), the system may be written in matrix-vector form as shown below.

$$\begin{array}{cccccc} & T_0 & P_0 & T_j & P_j & T_J & P_J \\ \text{b.c.} & \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 \\ -1 & \frac{-h_1}{2} & 1 & \frac{-h_1}{2} \\ \hline (s_3)_j & (s_2)_j & (s_3)_j & (s_1)_j & 0 & 0 \\ 0 & 0 & -1 & \frac{-h_{j+1}}{2} & 1 & \frac{-h_{j+1}}{2} \\ \hline (s_3)_J & (s_2)_J & (s_3)_J & (s_2)_J & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right] \\ \text{Eq. (4.4.24a)} & & & & & & \\ \text{Eq. (4.4.24b)} & & & & & & \\ \text{b.c.} & & & & & & \\ & \times \left[\begin{array}{c} \left(\begin{array}{c} T_0 \\ p_0 \end{array} \right) \\ \left(\begin{array}{c} T_j \\ p_j \end{array} \right) \\ \left(\begin{array}{c} T_J \\ p_j \end{array} \right) \end{array} \right] = \left[\begin{array}{c} \left(\begin{array}{c} (r_1)_0 \\ (r_2)_0 \end{array} \right) \\ \left(\begin{array}{c} (r_1)_j \\ (r_2)_j \end{array} \right) \\ \left(\begin{array}{c} (r_1)_J \\ (r_2)_J \end{array} \right) \end{array} \right] \end{array} \quad (4.4.27)$$

Here

$$\begin{aligned} (r_1)_0 &= T_w, \quad (r_1)_j = R_{j-1/2}^{n-1}, \quad 1 \leq j \leq J, \\ (r_2)_j &= 0, \quad 0 \leq j \leq J - 1, \quad (r_2)_J = T_e. \end{aligned} \quad (4.4.28)$$

The system of equations given by Eq. (4.4.27) can be rewritten as

$$\mathbb{A}\boldsymbol{\delta} = \mathbf{r}, \quad (4.4.29)$$

where

$$\mathbb{A} = \begin{bmatrix} A_0 & C_0 & & & & \\ B_1 & A_1 & C_1 & & & \\ \cdot & \cdot & \cdot & & & \\ & B_j & A_j & C_j & & \\ & & \cdot & \cdot & & \\ & & & B_{J-1} & A_{J-1} & C_{J-1} \\ & & & & B_J & A_J \end{bmatrix}, \quad \boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_0 \\ \boldsymbol{\delta}_1 \\ \vdots \\ \boldsymbol{\delta}_j \\ \vdots \\ \boldsymbol{\delta}_J \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_j \\ \vdots \\ \mathbf{r}_J \end{bmatrix}, \quad (4.4.30)$$

$$\boldsymbol{\delta}_j = \begin{bmatrix} T_j \\ p_j \end{bmatrix}, \quad \boldsymbol{r}_j = \begin{bmatrix} (r_1)_j \\ (r_2)_j \end{bmatrix}, \quad (4.4.31a)$$

and \mathbf{A}_j , \mathbf{B}_j , \mathbf{C}_j are 2×2 matrices defined as follows

$$\mathbf{A}_0 \equiv \begin{bmatrix} 1 & 0 \\ -1 & \frac{-h_1}{2} \end{bmatrix}, \quad \mathbf{A}_j \equiv \begin{bmatrix} (s_3)_j & (s_1)_j \\ -1 & \frac{-h_{j+1}}{2} \end{bmatrix}, \quad 1 \leq j \leq J-1$$

$$\mathbf{A}_J \equiv \begin{bmatrix} (s_3)_J & (s_1)_J \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_j \equiv \begin{bmatrix} (s_3)_j & (s_2)_j \\ 0 & 0 \end{bmatrix}, \quad 1 \leq j \leq J,$$

$$\mathbf{C}_j \equiv \begin{bmatrix} 0 & 0 \\ 1 & \frac{-h_{j+1}}{2} \end{bmatrix}, \quad 0 \leq j \leq J-1. \quad (4.4.31b)$$

Note that, as in the Crank–Nicolson method, the implicit nature of the method has again generated a tridiagonal matrix, but the entries are 2×2 blocks rather than scalars.

The solution of Eq. (4.4.29) by the block-elimination method consists of two sweeps. In the *forward sweep* we compute Γ_j , Δ_j , and \mathbf{w}_j from the recursion formulas given by

$$\Delta_0 = \mathbf{A}_0, \quad (4.4.32a)$$

$$\Gamma_j \Delta_{j-1} = \mathbf{B}_j, \quad (4.4.32b)$$

$$\Delta_j = \mathbf{A}_j - \Gamma_j \mathbf{C}_{j-1}, \quad 1 \leq j \leq J \quad (4.4.32c)$$

$$\mathbf{w}_0 = \boldsymbol{r}_0, \quad (4.4.33a)$$

$$\mathbf{w}_j = \boldsymbol{r}_j - \Gamma_j \mathbf{w}_{j-1}, \quad 1 \leq j \leq J. \quad (4.4.33b)$$

Here Γ_j has the same structure as B_j , that is,

$$\Gamma_j \equiv \begin{bmatrix} (\gamma_{11})_j & (\gamma_{12})_j \\ 0 & 0 \end{bmatrix}$$

and although the second row of Δ_j , has the same structure as the second row of \mathbf{A}_j ,

$$\Delta_j \equiv \begin{bmatrix} (\alpha_{11})_j & (\alpha_{12})_j \\ -1 & \frac{-h_{j+1}}{2} \end{bmatrix},$$

for generality we write it as

$$\Delta_j \equiv \begin{bmatrix} (\alpha_{11})_j & (\alpha_{12})_j \\ (\alpha_{21})_j & (\alpha_{22})_j \end{bmatrix}$$

In the *backward sweep*, δ_j is computed from the following recursion formulas:

$$\Delta_J \delta_J = \mathbf{w}_J, \quad (4.4.34a)$$

$$\Delta_j \delta_j = \mathbf{w}_j - \mathbf{C}_j \delta_{j+1}, \quad j = J-1, J-2, \dots, 0. \quad (4.4.34b)$$

Example 4.4. Repeat Example 4.1 using Keller's box method. Compare your results with those obtained with the Crank–Nicolson method.

Solution. The solution of Example 4.4 with Keller's box method follows the procedure described in subsection 4.4.3. Essentially after generating the finite-difference grid, initial profiles at $x = 0$, we define $(s_1)_j$ to $(s_3)_j$ in Eq. (4.4.25a) together with $(r_1)_j$ and $(r_2)_j$. Then we use SOLV2 to solve the linear system. Note that in this case wall and edge temperatures are specified.

A comparison of numerical results obtained with the Box method together with those obtained with the Crank–Nicolson method and the analytical method is given in Table E4.9. A listing of the computer program using SOLV2 is given in Appendix A, Example 4.4. As can be seen, the predictions of the box method are in very good agreement with the other two.

Table E4.9. Comparison of the box method results with CN and AS at $x = 0.30$. $\Delta t = 0.0100$, $\Delta x = 0.100$

t	CN	Box	AS
.0000	.600	.6000	.6004
.0100	.584	.5875	.5799
.0200	.543	.5377	.5334
.0300	.496	.4888	.4857
.0400	.454	.4426	.4411
.0500	.414	.4006	.4000
.0600	.377	.3624	.3626
.0700	.344	.3277	.3286
.0800	.313	.2965	.2977
.0900	.286	.2680	.2698
.1000	.260	.2426	.2444

4.5 Finite-Difference Methods for Elliptic Equations

To describe the numerical solution of elliptic equations by finite-difference methods, consider the Poisson equation in the form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad (4.5.1)$$

and its solution for a rectangular region (Fig. 4.7) defined by $0 \leq x \leq a$, $0 \leq y \leq b$, with uniform net spacings Δx and Δy and with

$$x_0 = 0, \quad x_i = i\Delta x, \quad i = 1, 2, \dots, I, I+1 \quad (4.5.2a)$$

$$y_0 = 0, \quad y_j = j\Delta y, \quad j = 1, 2, \dots, J, J+1 \quad (4.5.2b)$$

subject to the boundary conditions specified at four sides of the rectangle

$$i = (0, I+1) \quad 0 \leq j \leq J+1 \quad (4.5.3a)$$

$$j = (0, J+1) \quad 0 \leq i \leq I+1 \quad (4.5.3b)$$

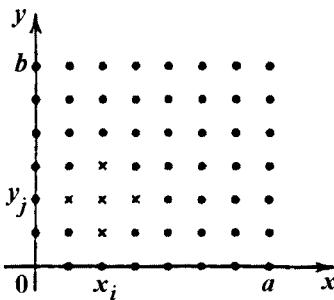


Fig. 4.7. Net points for Laplace difference equation.

Replacing each second derivative in Eq. (4.5.1) by a centered second difference quotient, Eq. (4.3.10), at (i, j) (Fig. 4.7), we get

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = f_{i,j} \quad (4.5.4a)$$

or

$$u_{i,j} - \theta_x(u_{i+1,j} + u_{i-1,j}) - \theta_y(u_{i,j+1} + u_{i,j-1}) = -\delta^2 f_{i,j} \equiv F_{i,j} \quad (4.5.4b)$$

$$1 \leq i \leq I, \quad 1 \leq j \leq J$$

where

$$\delta^2 = \frac{(\Delta x)^2(\Delta y)^2}{2[(\Delta x)^2 + (\Delta y)^2]}, \quad \theta_x = \frac{(\Delta y)^2}{2[(\Delta x)^2 + (\Delta y)^2]}, \quad \theta_y = \frac{(\Delta x)^2}{2[(\Delta x)^2 + (\Delta y)^2]} \quad (4.5.5)$$

The linear equation, (4.5.4), yields a system of IJ algebraic equations with IJ unknowns. Its solution for the $(I+2)(J+2)$ values of u requires $2(I+J)+4$ values from the boundary conditions which can be obtained by using either direct or iterative methods discussed in subsections 4.5.1 and 4.5.2, respectively.

4.5.1 Direct Methods

The linear equations given by Eq. (4.5.4) have a block tridiagonal structure and can be written in vector-matrix form given by Eq. (4.4.29). However, in this case, due to slightly different notation, we write Eq. (4.4.29) again,

$$\mathbb{A}\mathbf{U} = \mathbf{F} \quad (4.5.6)$$

Here \mathbb{A} denotes the coefficient matrix same as that defined in Eq. (4.4.30), but with different indices,

$$\mathbb{A} = \begin{vmatrix} A_1 & C_1 & & & \\ B_2 & A_2 & C_2 & & \\ & \ddots & \ddots & & \\ & & B_j & A_j & C_j & \\ & & & \ddots & \ddots & \\ & & & & B_{J-1} & A_{J-1} & C_{J-1} \\ & & & & & B_J & A_J & \end{vmatrix} \quad (4.5.7)$$

and with A_j , B_j and C_j denoting I -dimensional matrices and I_I denoting the identity matrix of order I

$$A_j = \begin{vmatrix} 1 & -\theta_x & & & \\ -\theta_x & 1 & -\theta_x & & \\ & -\theta_x & 1 & -\theta_x & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \\ & & & & -\theta_x & 1 & -\theta_x \\ & & & & & -\theta_x & 1 \end{vmatrix} \quad (4.5.8a)$$

$$B_j = C_j = -\theta_y I_I \quad (4.5.8b)$$

In Eq. (4.5.6), \mathbf{U} and \mathbf{F} are 3-dimensional compound vectors (i.e., vectors whose components are I -dimensional vectors) and are defined by

$$\mathbf{U} = \begin{vmatrix} u_1 \\ u_2 \\ \vdots \\ u_j \\ \vdots \\ u_J \end{vmatrix}, \quad \underline{u}_j = \begin{vmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{i,j} \\ \vdots \\ u_{I,j} \end{vmatrix}, \quad \mathbf{F} = \begin{vmatrix} F_1 \\ F_2 \\ \vdots \\ F_j \\ \vdots \\ F_J \end{vmatrix}, \quad (4.5.9)$$

where

$$\begin{aligned} F_1 &= f_1 + \theta_x w_1 + \theta_y u_0 \\ F_j &= f_j + \theta_x w_j \quad 2 \leq j \leq J-1 \\ F_J &= f_J + \theta_x w_J + \theta_y \underline{u}_{J+1} \end{aligned} \quad (4.5.10)$$

$$\underline{f}_j = \begin{vmatrix} F_{1,j} \\ F_{2,j} \\ \vdots \\ F_{I,j} \end{vmatrix}, \quad \underline{w}_j = \begin{vmatrix} u_{0,j} \\ 0 \\ \vdots \\ 0 \\ u_{I+1,j} \end{vmatrix}, \quad \underline{u}_0 = \begin{vmatrix} u_{1,0} \\ u_{2,0} \\ \vdots \\ u_{I,0} \end{vmatrix}, \quad \underline{u}_{J+1} = \begin{vmatrix} u_{1,J+1} \\ u_{2,J+1} \\ \vdots \\ u_{I,J+1} \end{vmatrix} \quad (4.5.11)$$

In Eq. (4.5.11) \underline{f}_j comes from the right-hand side of Eq. (4.5.4b); once the function $f(x, y)$ in the Poisson equation is given, \underline{f}_j is known. The column vector \underline{w}_j represents the boundary conditions at $i = 0$ and $i = I + 1$, and \underline{u}_0 and \underline{u}_{J+1} represent the boundary conditions at $j = 0$ and $j = J + 1$, respectively. Note that $w_{i,j} = 0$ for $2 \leq i \leq I - 1$, $w_{1,j} = u_{0,j}$, and $w_{I,j} = u_{I+1,j}$. Of course, all of the $u_{i,j}$ which enter into Eq. (4.5.10) are known quantities determined from the boundary conditions.

In some problems all of the boundary conditions may not be given in terms of u , but are given some in terms of its derivatives. In those cases, the structure of the A_j , B_j , C_j matrices in the coefficient matrix \mathbb{A} can change. To illustrate, consider Problem 4.5 with the boundary conditions are of the form

$$i = 0, \quad \frac{\partial u}{\partial x} = 0; \quad i = I + 1, \quad u = 0 \quad (4.5.12a)$$

$$j = 0, \quad \frac{\partial u}{\partial y} = 0; \quad j = J + 1, \quad u = 0 \quad (4.5.12b)$$

The boundary conditions at $i = 0$ and $j = 0$ may be approximated to first order by the forward difference formula (4.3.9) by

$$u_0 - u_1 = 0 \quad (4.5.13)$$

or to second order, requiring

$$u(\zeta) = a_0 + a_1\zeta + a_2\zeta^2$$

to satisfy $u(0) = u_0$, $u(\Delta\zeta) = u_1$ and $u(2\Delta\zeta) = u_2$, and then setting $du(0)/d\zeta = a_1 = 0$. This procedure yields

$$u_0 - \frac{4}{3}u_1 + \frac{1}{3}u_2 = 0 \quad (4.5.14)$$

The choice given by Eq. (4.5.14) allows the boundary conditions at $i = 0$ and $j = 0$ to be written in the form

$$i = 0, \quad u_{0,j} - \frac{4}{3}u_{1,j} + \frac{1}{3}u_{2,j} = 0 \quad 1 \leq j \leq J \quad (4.5.15a)$$

$$j = 0, \quad u_{i,0} - \frac{4}{3}u_{i,1} + \frac{1}{3}u_{i,2} = 0 \quad 1 \leq i \leq I \quad (4.5.15b)$$

For $i = 1$, Eq. (4.5.4b) becomes

$$u_{1,j} - \theta_x(u_{2,j} + u_{0,j}) - \theta_y(u_{1,j+1} + u_{1,j-1}) = -\delta^2 f_{1,j} = F_{1,j}$$

and, with Eq. (4.5.15a), it can be written as

$$u_{1,j} - \theta_x(u_{2,j} + \frac{4}{3}u_{1,j} - \frac{1}{3}u_{2,j}) - \theta_y(u_{1,j+1} + u_{1,j-1}) = F_{1,j}$$

or as

$$\left(1 - \frac{4}{3}\theta_x\right)u_{1,j} - \frac{2}{3}\theta_x u_{2,j} - \theta_y(u_{1,j+1} + u_{1,j-1}) = F_{1,j}, \quad 2 \leq j \leq J-1 \quad (4.5.16a)$$

For $i = I$, Eq. (4.5.4b) becomes

$$u_{I,j} - \theta_x(u_{I+1,j} + u_{I-1,j}) - \theta_y(u_{I,j+1} + u_{I,j-1}) = F_{I,j}$$

and with the boundary condition at $i = I + 1$, that is,

$$u_{I+1,j} = 0 \quad (4.5.15c)$$

Eq. (4.5.4b) at $i = I$ can be written as

$$u_{I,j} - \theta_x u_{I-1,j} - \theta_y(u_{I,j+1} + u_{I,j-1}) = F_{I,j} \quad 2 \leq j \leq J-1 \quad (4.5.16b)$$

For $j = 1$, Eq. (4.5.4b) becomes

$$u_{i,1} - \theta_x(u_{i+1,1} + u_{i-1,1}) - \theta_y(u_{i,2} + u_{i,0}) = F_{i,1}$$

and, with Eq. (4.5.15b), can be written as

$$u_{i,1} - \theta_x(u_{i+1,1} + u_{i-1,1}) - \theta_y(u_{i,2} + \frac{4}{3}u_{i,1} - \frac{1}{3}u_{i,2}) = F_{i,1}$$

or as

$$\left(1 - \frac{4}{3}\theta_y\right)u_{i,1} - \theta_x(u_{i+1,1} + u_{i-1,1}) - \frac{2}{3}\theta_y u_{i,2} = F_{i,1}, \quad 2 \leq i \leq I-1 \quad (4.5.16c)$$

For $j = J$, Eq. (4.5.4b) becomes

$$u_{i,J} - \theta_x(u_{i+1,J} + u_{i-1,J}) - \theta_y(u_{i,J+1} + u_{i,J-1}) = F_{i,J}$$

and with the boundary condition at $j = J + 1$, that is,

$$u_{i,J+1} = 0 \quad (4.5.15d)$$

Eq. (4.5.4b) at $j = J$ can be written as

$$u_{i,J} - \theta_x(u_{i+1,J} + u_{i-1,J}) - \theta_y(u_{i,J-1}) = F_{i,J}, \quad 2 \leq i \leq I-1 \quad (4.5.16d)$$

At $i = j = 1$, Eq. (4.5.4b), with the relations given by Eqs. (4.5.15a,b), becomes

$$\left(1 - \frac{4}{3}\theta_x - \frac{4}{3}\theta_y\right)u_{1,1} - \frac{2}{3}\theta_x u_{2,1} - \frac{2}{3}\theta_y u_{1,2} = F_{1,1} \quad (4.5.17a)$$

At $i = 1, j = J$, Eq. (4.5.4b), with the relations given by Eqs. (4.5.15a,d) becomes

$$\left(1 - \frac{4}{3}\theta_x\right)u_{1,J} - \frac{2}{3}\theta_x u_{2,J} - \theta_y u_{1,J-1} = F_{1,J} \quad (4.5.17b)$$

At $i = I, j = 1$, Eq. (4.5.4b), with the relations given by Eqs. (4.5.15b,c) becomes

$$\left(1 - \frac{4}{3}\theta_y\right)u_{I,1} - \theta_x u_{I-1,1} - \frac{2}{3}\theta_y u_{I,2} = F_{I,1} \quad (4.5.17c)$$

At $i = I, j = J$, Eq. (4.5.4b), with the relations given by Eqs. (4.5.15c,d) becomes

$$u_{I,J} - \theta_x u_{I-1,J} - \theta_y u_{I,J-1} = F_{I,J} \quad (4.5.17d)$$

The matrices A_j , B_j and C_j in the coefficient matrix \mathbb{A} become

$$A_1 = \begin{vmatrix} a_1^* & -\frac{2}{3}\theta_x & & & \\ -\theta_x & a_2^* & -\theta_x & & \\ & -\theta_x & a_2^* & -\theta_x & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & -\theta_x & a_2^* & -\theta_x \\ & & & & -\theta_x & a_2^* \end{vmatrix} \quad (4.5.18a)$$

$$A_j = \begin{vmatrix} a_3^* & -\frac{2}{3}\theta_x & & & \\ -\theta_x & 1 & -\theta_x & & \\ & -\theta_x & 1 & -\theta_x & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & -\theta_x & 1 & -\theta_x \\ & & & & -\theta_x & 1 \end{vmatrix} \quad 2 \leq j \leq J-1 \quad (4.5.18b)$$

$$A_J = \begin{vmatrix} a_3^* & -\frac{2}{3}\theta_x & & & \\ -\theta_x & 1 & -\theta_x & & \\ & -\theta_x & 1 & -\theta_x & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & -\theta_x & 1 & -\theta_x \\ & & & & -\theta_x & 1 \end{vmatrix} \quad (4.5.18c)$$

$$B_j = -\theta_y I_I \quad 2 \leq j \leq J \quad (4.5.18d)$$

$$C_1 = -\frac{2}{3}\theta_y I_I \quad (4.5.18e)$$

$$C_j = -\theta_y I_I \quad 2 \leq j \leq J-1 \quad (4.5.18f)$$

where

$$a_1^* = 1 - \frac{4}{3}\theta_x - \frac{4}{3}\theta_y, \quad a_2^* = 1 - \frac{4}{3}\theta_y, \quad a_3^* = 1 - \frac{4}{3}\theta_x \quad (4.5.19)$$

Whether the boundary conditions are given in terms of u or its derivatives, the solution of Eq. (4.5.6) can be obtained by the block-elimination method as discussed in subsection 4.4.3 or Gauss' elimination method discussed below. Since the coefficient matrix \mathbb{A} has large blocks of zero elements, it is more efficient to solve Eq. (4.5.6) with the block-elimination method than with the Gaussian elimination method. As we shall see shortly, however, it is still necessary to make partial use of the Gaussian elimination in the block elimination for the solution of elliptic equations.

The block elimination method for this problem is identical to the one discussed before except for the difference of the indices in the coefficient matrix, Eq. (4.4.30). So for convenience the two steps in this method are repeated below.

In the first step of the forward sweep, Γ_j and Δ_j are computed from

$$\Delta_1 = A_1 \quad (4.5.20a)$$

$$\Gamma_j \Delta_{j-1} = B_j \quad j = 2, 3, \dots, J \quad (4.5.20b)$$

$$\Delta_j = A_j - \Gamma_j C_{j-1} \quad j = 2, 3, \dots, J \quad (4.5.20c)$$

In the second part of the forward sweep, the w_j are computed from

$$w_1 = F_1 \quad (4.5.21a)$$

$$w_j = F_j - \Gamma_j w_{j-1} \quad 2 \leq j \leq J \quad (4.5.21b)$$

In the backward sweep, the u_j are computed from

$$\Delta_J u_J = w_J \quad (4.5.22a)$$

$$\Delta_j u_j = w_j - C_j u_{j+1} \quad j = J-1, J-2, \dots, 1 \quad (4.5.22b)$$

In the application of the block elimination method to solve the Laplace difference equations, the Δ_j matrix in Eqs. (4.5.20b) and (4.5.22) is a full matrix of order I and is not a tridiagonal matrix except for $J = 1$. Thus, the inversion of Δ_j is not a trivial task. On the other hand, in the application of this method to solve the difference equations for boundary layers (Chapter 7), the order of Δ_j matrix is generally small. For this reason, the inversion of the Δ_j matrix is relatively simple.

To solve Eqs. (4.5.20b) and (4.5.22), we use the Gaussian elimination method and write both equations in the form

$$Ax = b \quad (4.5.23)$$

Here $A \equiv [a_{ij}]$ is a square matrix of order n , that is,

$$A \equiv \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ a_{i1} & a_{i2} & a_{ij} & \cdots & a_{in} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & a_{nj} & \cdots & a_{nn} \end{vmatrix} \quad (4.5.24)$$

and $\underline{x} = (x_1, \dots, x_i, \dots, x_n)^T$ and $\underline{b} = (b_1, \dots, b_i, \dots, b_n)^T$ with T denoting the transpose. According to the Gaussian elimination method, the elements of \underline{x} are given by

$$x_i = \frac{1}{a_{ii}^{(i-1)}} \left[b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j \right], \quad i = n, \dots, 1 \quad (4.5.25)$$

where

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)}, \quad \begin{matrix} k = 1, \dots, n-1 \\ j = k+1, \dots, n \\ i = k+1, \dots, n \end{matrix} \quad a_{ij}^{(0)} = a_{ij} \quad (4.5.26a)$$

$$b_i^{(k)} = b_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)}, \quad \begin{matrix} k = 1, \dots, n-1 \\ i = k+1, \dots, n \end{matrix} \quad b_i^{(0)} = b_i \quad (4.5.26b)$$

Table 4.2 gives the FORTRAN listing based on the Gaussian elimination. Thus, the block-elimination method together with the Gaussian elimination method

Table 4.2. FORTRAN Listing of Subroutine GAUSS

```

SUBROUTINE GAUSS (N,M,A,B)
DIMENSION A(100,100),B(100,100)
DO 100 K = 1,N-1
  KP      = K + 1
  DO 100 I = KP,N
    R      = A(I,K)/A(K,K)
    DO 200 J = KP,N
      A(I,J) = A(I,J) - R*A(K,J)
    DO 100 J = 1,M
      B(I,J) = B(I,J) - R*B(K,J)
    DO 300 K = 1,M
      B(N,K) = B(N,X)/A(N,N)
    DO 300 I = N-1,1,-1
      IP     = I + 1
      DO 400 J = IP,N
        B(I,K) = B(I,K) - A(I,J)*B(J,K)
    300 B(I,K) = B(I,K)/A(I,I)
    RETURN
  END

```

can be used to solve Eq. (4.5.6). A listing of a subroutine for this purpose is given in Table 4.3 for A_j , B_j and C_j matrices given by Eqs. (4.5.8a,b).

To use the subroutine in Table 4.3, the number of grid points in the x and y directions must be specified by I ($\equiv II$) and J ($\equiv JJ$), respectively, the coefficients θ_x ($\equiv TX$), θ_y ($\equiv TY$) in Eq. (4.5.4a), and the compound vector \mathbf{F} ($\equiv F$) on the right-hand side of Eq (4.5.6). The compound vector \mathbf{F} is obtained from Eq (4.5.10) once the forcing function [$f(x, y)$ in Eqs. (4.5.1)] is defined and the boundary conditions on the four sides of the rectangle are given.

Example 4.5. Compute the temperature distribution in a square region of sides unity subject to the following boundary conditions

$$T(x, 0) = T(x, 1) = 0, \quad T(0, y) = \sin \pi y \quad \text{and} \quad T(1, y) = e^\pi \sin \pi y$$

by solving the heat condition equation,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad 0 \leq x, y \leq 1$$

Compare your solutions with the analytical solution at $x = 0.2, 0.5$ and 0.9 .

$$T(x, y) = e^{\pi x} \sin \pi y$$

Take $\Delta x = \Delta y = 1/10$.

Solution. Table E4.10 presents a comparison between the numerical and analytical results at $x = 0.2, 0.5$ and 0.9 as a function of y . Appendix A contains the computer program.

Table E4.10. Comparison of FDS and AS

y	x=0.2		x=0.5		x=0.9	
	FDS	AS	FDS	AS	FDS	AS
0.1	0.58693	0.57924	1.504	1.48652	5.23614	5.22301
0.2	1.1164	1.10178	2.86078	2.62753	9.95973	9.93476
0.3	1.53659	1.51647	3.93753	3.89176	13.70839	13.67403
0.4	1.80637	1.78271	4.62884	4.57504	6.11517	16.07478
0.5	1.89933	1.87446	4.86705	4.81048	16.9445	16.90203
0.6	1.80637	1.78271	4.62884	4.57504	16.11517	16.07478
0.7	1.53659	1.51647	3.93753	3.89176	13.70838	13.67403
0.8	1.1164	1.10178	2.86078	2.82753	9.95972	9.93476
0.9	0.58693	0.57924	1.504	1.48652	5.23614	5.22301

4.5.2 Iterative Methods

Iterative solutions which may be based on *point* or *block* iterations are more popular than the direct methods used to solve the Laplace difference equations. Again the large number of zero elements in the coefficient matrix A greatly reduces the computational effort required in each iteration. However, care must

Table 4.3. FORTRAN Listing of Subroutine POISSON

```

SUBROUTINE POISSON(II,JJ,TX,TY,F,U)
PARAMETER (NMAX = 100)
DIMENSION A(NMAX,NMAX),B(NMAX,NMAX),C(NMAX,NMAX),
&          D(NMAX,NMAX),E(NMAX,NMAX)
DIMENSION DELTA(NMAX,NMAX,NMAX),DELT(T(NMAX,NMAX),BM(NMAX,NMAX),
DIMENSION DM(NMAX,NMAX),F(NMAX,NMAX),W(NMAX,NMAX),UT(NMAX),
&          U(NMAX,NMAX)
C ELEMENTS OF DIAGONAL VECTORS IN THE BLOCK MATRICES A,B AND C
DO 5 I=1,II
DO 5 J=1,JJ
  A(I,J) = -TX
  B(I,J) = 1.
  C(I,J) = -TX
  D(I,J) = -TY
  E(I,J) = -TY
5 CONTINUE
C ELEMENTS OF W-VECTOR, SEE EQ. (4.5.21a)
DO 10 K=1,II
  W(K,1) = F(K,1)
C ELEMENTS OF DELTA-MATRIX, SEE EQ. (4.5.20a)
DO 10 L=1,II
  DELTA(K,L,1) = 0.
10 CONTINUE
  DELTA(1,1,1) = B(1,1)
DO 20 K=2,II
  DELTA(K,K,1) = B(K,1)
  DELTA(K,K-1,1) = A(K,1)
  DELTA(K-1,K,1) = C(K-1,1)
20 CONTINUE
C FORWARD SWEEP
DO 30 J=2,JJ
  DO 40 K=1,II
    DO 40 L=1,II
      DELTT(K,L) = DELTA(L,K,J-1)
      BM(K, L) = 0.
40 CONTINUE
  DO 50 K=1,II
    BM(K,K) = D(K,J)
50 CONTINUE
C SEE EQ. (4.4.20B)
CALL GAUSS (II,II,DELT,BM)
  DO 60 K=1,II
  DO 60 L=1,II
    DELTA(K,L,J) = 0.
60 CONTINUE
  DELTA(1,1,J) = B(1,J)
  DO 70 K=2,II
    DELTA(K,K,J) = B(K,J)
    DELTA(K,K-1,J) = A(K,J)
    DELTA(K-1,K,J) = C(K-1,J)
70 CONTINUE
  DO 30 K=1,II
    W(K,J) = F(K,J)

```

```

DO 30 L=1,II
C ELEMENTS OF DELTA MATRIX, SEE EQ. (4.4.20C)
  DELTA(K,L,J) = DELTA(K,L,J)-BM(L,K)*E(L,J-1)
C ELEMENTS OF W-VECTOR, SEE EQ. (4.4.21B)
  W(K,J) = W(K,J) -BM(L,K) *W(L,J-1)
30 CONTINUE

C BACKWARD SWEEP
DO 80 K=1,II
  UT(K) = W(K,JJ)
  DO 80 L=1,II
    DM(K,L) = DELTA(K,L,JJ)
80 CONTINUE
C SEE EQ. (4.4.22A)
CALL GAUSS(II,1,DM,UT)
C SEE EQ. (4.4.22B) ", "
DO 90 J=JJ-1,1,-1
  DO 100 K=1,II
    U(K, J+1) = UT(K)
    UT(K) = W(K,J)-E(K,J)*U(K,J+1)
    DO 100 L=1, II
      DM(K,L) = DELTA(K,L,J)
100  CONTINUE
  CALL GAUSS(II,1,DM,UT)
90 CONTINUE
  DO 110 K=1,II
    U(K,1) = UT(K)
110  CONTINUE
  RETURN
END

```

be taken to ensure that sufficient accuracy is obtained in a “reasonable” number of iterations.

A useful point-iteration method is the *Gauss-Seidel* or *successive iteration* method; the Laplace difference equations are written as

$$u_{i,j}^{(\nu+1)} = \theta_x \left(u_{i-1,j}^{(\nu+1)} + u_{i+1,j}^{(\nu)} \right) + \theta_y \left(u_{i,j-1}^{(\nu+1)} + u_{i,j+1}^{(\nu)} \right) - \delta^2 f_{i,j} \quad (4.5.27)$$

$$1 \leq i \leq I, \quad 1 \leq j \leq J$$

In this scheme the evaluation of the new iterates is not completely arbitrary. First compute $u_{1,1}^{(\nu+1)}$ and then, in order, the other elements on the coordinate lines with $j = 1$. Next, $u_{1,2}^{(\nu+1)}$ is determined, etc. By slight changes in the scheme, the calculations could start at either of the other three corners of the rectangle (Fig. 4.7). However, all of these methods have the same rate of convergence.

The rate of convergence for the Gauss-Seidel scheme is

$$R_{GS} = 2\delta^2\pi^2 \left(\frac{1}{a^2} + \frac{1}{b^2} \right) + O(\delta^4) \quad (4.5.28)$$

The convergence rate of the Gauss–Seidel method may be improved by introducing an appropriate *acceleration parameter*. That is, set

$$V_{i,j}^{(\nu+1)} = \theta_x(u_{i-1,j}^{(\nu+1)} + u_{i+1,j}^{(\nu)}) + \theta_y(u_{i,j-1}^{(\nu+1)} + u_{i,j+1}^{(\nu)}) - \delta^2 f_{i,j} \quad (4.5.29)$$

and then, at the point (x_i, y_j) , take

$$\begin{aligned} u_{i,j}^{(\nu+1)} &= \omega V_{i,j}^{(\nu+1)} + (1 - \omega) u_{i,j}^{(\nu)} \\ &= u_{i,j}^{(\nu)} + \omega(V_{i,j}^{(\nu+1)} - u_{i,j}^{(\nu)}) \end{aligned} \quad (4.5.30)$$

Here ω is the acceleration parameter to be determined. Note that for $\omega = 1$ this scheme reduces to that in Eq. (4.5.27), i.e. to the ordinary Gauss–Seidel method. The order in which the components of the new iterates are to be computed is just as in the previous successive scheme.

As discussed by Isaacson and Keller [3], the iterations in this scheme converge if the acceleration parameter ω lies in the interval $0 < \omega < 2$, and its optimal value, ω_{opt} , is

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \eta^2}} \quad (4.5.31a)$$

where $\eta = \cos \pi \Delta x$ for $\Delta x = \Delta y$ and the rate of convergence is

$$R_{\text{AGS}} = 2\delta\pi\sqrt{2(1/a^2 + 1/b^2)} + O(\delta^2) \quad (4.5.31b)$$

By comparing Eq. (4.5.31b) with (4.5.28), it is seen that the power of δ in the rate of convergence for the optimal accelerated Gauss–Seidel (AGS) method is lower than the power of δ appearing in the ordinary Gauss–Seidel method. For $\omega > 1$, the accelerated Gauss–Seidel is called successive over-relaxation (SOR) and for $\omega < 1$, the method is called under-relaxation.

In the *block iteration* method, unlike the point iteration method in which the next iterate is determined at each grid point (see Fig. 4.8a) the iterations are performed at each column or row, as indicated in (b) and (c) of Fig. 4.8. In either choice (column or row), one must solve a tridiagonal matrix with the Thomas algorithm discussed in the previous subsection. To examine this in more detail, consider the choice of column iterations; in this case, at each value of i , Eq. (4.5.4b), can be written as

$$\theta_y(u_{i,j+1} + u_{i,j-1}) - u_{i,j} = \delta^2 f_{i,j} - \theta_x(u_{i+1,j} + u_{i-1,j}) \quad (4.5.32)$$

and is solved for all values of j subject to the boundary conditions. Eq. (4.5.32) is of the form given by Eq. (4.5.13) with coefficients a_j , b_j , c_j and r_j given by

$$\begin{aligned} a_j &= \theta_y, & b_j &= -1, & c_j &= \theta_y \\ r_j &= \delta^2 f_{i,j} - \theta_x(u_{i+1,j} + u_{i-1,j}) \end{aligned} \quad (4.5.33)$$

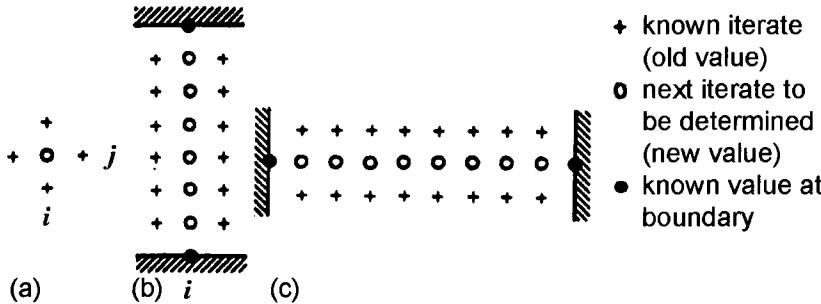


Fig. 4.8. Finite-difference grid and some iterative methods for solving elliptic problems.
(a) Point iteration, (b) block iteration by column, (c) block iteration by row.

Another alternative to solving Eq. (4.5.4) is the *alternating direction implicit* (ADI) method, first introduced for time-dependent problems by Peaceman and Rachford [4]. There are several versions of this method, but in principle they all alternatively solve rows and columns implicitly in a block-iteration scheme.

To solve Eq. (4.5.1) with the ADI scheme, let $\beta = \Delta x / \Delta y$ and write Eq. (4.5.4a) as

$$-[u_{i-1,j} - 2u_{i,j} + u_{i+1,j}] - \beta^2[u_{i,j-1} - 2u_{i,j} + u_{i,j+1}] = -(\Delta x)^2 f_{i,j} \quad (4.5.34)$$

On “solving” for each bracketed term, this equation can be written identically as

$$\begin{aligned} & -u_{i-1,j} + 2u_{i,j} - u_{i+1,j} + \omega u_{i,j} \\ & = \omega u_{i,j} + \beta^2(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) - (\Delta x)^2 f_{i,j} \end{aligned} \quad (4.5.35a)$$

and

$$\begin{aligned} & -\beta^2 u_{i-1,j} + 2\beta^2 u_{i,j} - \beta^2 u_{i+1,j} + \omega u_{i,j} \\ & = \omega u_{i,j} + (u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) - (\Delta x)^2 f_{i,j} \end{aligned} \quad (4.5.35b)$$

where ω is a relaxation parameter. The Peaceman-Rachford ADI iteration is then defined by the equations which have a tridiagonal structure,

$$\begin{aligned} & -u_{i-1,j}^{n+1/2} + (\omega + 2)u_{i,j}^{n+1/2} - u_{i+1,j}^{n+1/2} \\ & = \omega u_{i,j}^n + \beta^2(u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n) - (\Delta x)^2 f_{i,j} \end{aligned} \quad (4.5.36a)$$

and

$$\begin{aligned} & -\beta^2 u_{i,j-1}^{n+1} + (\omega + 2\beta^2)u_{i,j}^{n+1} - \beta^2 u_{i,j+1}^{n+1} \\ & = \omega u_{i,j}^{n+1/2} + (u_{i-1,j}^{n+1/2} - 2u_{i,j}^{n+1/2} + u_{i+1,j}^{n+1/2}) - (\Delta x)^2 f_{i,j} \end{aligned} \quad (4.5.36b)$$

where the n -th iterative values $u_{i,j}^n$ are assumed to be calculated at all grid points from an arbitrary initial approximation $u_{i,j}^{(0)}$ and known boundary values. A new

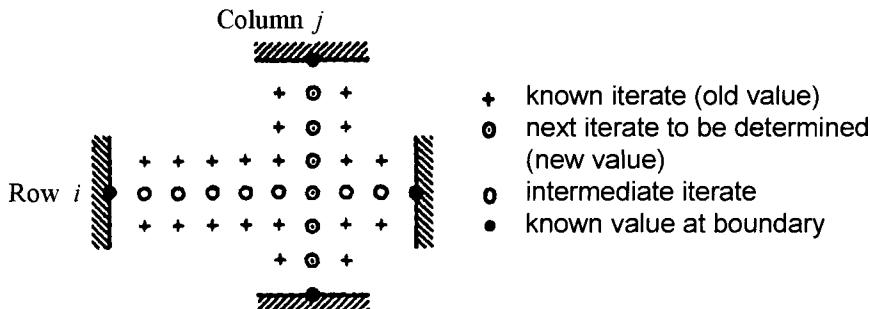


Fig. 4.9. ADI iteration by column and row.

iterate $u_{i,j}^{n+1}$ is determined from a known iterate $u_{i,j}^n$ in two steps. In step 1, the intermediate iterate $u_{i,j}^{n+1/2}$ is determined from Eq. (4.5.36a) for all values of i at column j , and in step 2 the new iterate $u_{i,j}^{n+1}$ is determined from Eq. (4.5.36b) for all values of j at row i , Fig. 4.9. The acceleration parameter ω in Eqs. (4.5.36) is kept constant while sweeping the whole net in the two steps representing one iteration. It was shown by Isaacson and Keller [3] that this scheme converges for any choice of $\omega > 0$. The trick in the proper use of the ADI procedure is *not* to use a single acceleration parameter ω as above, but rather to use a sequence of them, say $\omega_1, \omega_2, \dots, \omega_m$ applied periodically (or cyclically). That is, the calculations in Eq. (4.5.36) are to be carried out m times (using each ω_i for a complete double sweep of the net) in order to compute $u_{i,j}^{n+1}$ from $u_{i,j}^n$. Because the equations are solved alternatively by sets of rows and sets of columns, the method may be considered as a line method with alternating directions.

Example 4.6. Repeat Example 4.5 with SOR and ADI methods. Compare your results with SOR and with those obtained with GS. Take $h = \Delta x = \Delta y = 1/10$. Use Eq. (4.5.31a) with $\omega = 2/(1 + \sin \pi h)$ for SOR and $\omega = 1$ for ADI. Determine the max residual

$$\max_{i,j} |Au - F|_2 \leq \varepsilon$$

for values of $\varepsilon = 10^{-n}$, $n = 1$ to 6. Determine also the max error between the numerical and analytical solutions, that is

$$\max_{i,j} |u_{FDS} - u_{AS}|$$

Solution.

Table E4.11 presents the solutions as a function of iterations for each method.

Table E4.11. Number of iterations, max residual and max error as a function of SOR, GS and ADI schemes.

Tolerance	SOR			Gauss-Seidel			ADI		
	Iter	MaxError	MaxResidual	Iter	MaxError	MaxResidual	Iter	MaxError	MaxResidual
1.00E-07	36	6.68E-02	7.47E-08	154	6.68E-02	9.68E-08	19	6.68E-02	9.94E-08
1.00E-06	31	6.68E-02	6.62E-07	131	6.68E-02	9.73E-07	17	6.68E-02	6.82E-07
1.00E-05	27	6.68E-02	8.53E-06	108	6.67E-02	9.79E-06	15	6.68E-02	4.74E-06
1.00E-04	23	6.68E-02	5.98E-05	85	6.55E-02	9.85E-05	12	6.68E-02	7.15E-05
1.00E-03	20	6.63E-02	7.12E-04	62	5.31E-02	9.90E-04	10	6.71E-02	3.41E-04
1.00E-02	17	6.40E-02	9.30E-03	39	1.44E-01	9.94E-03	7	7.02E-02	7.56E-03
1.00E-01	11	3.98E-01	8.23E-02	17	1.66E+00	9.03E-02	5	9.32E-02	3.22E-02

4.5.3 Multigrid Method

In subsection 4.5.1, we have described Gauss–Seidel, SOR and ADI iterative methods for solving linear and nonlinear elliptic partial differential equations, PDEs, and discussed their convergence rates. The multigrid method discussed by Brandt [5] can be directly applied to solve nonlinear PDEs. It has optimal convergence rate and works well for linear and nonlinear PDEs and for one, two, or three dimensions.

In most iterative methods, the errors corresponding to high frequencies can be quickly reduced in several iterations. However, the reduction of the errors corresponding to low frequencies is slow*. As a result, it is prudent to perform the calculations on a coarse grid Ω_{2h} rather than on a given grid Ω_h only, (Fig. 4.10) so that most low frequencies in Ω_h become high frequencies in Ω_{2h} . In this way, the error corresponding to low frequency can be quickly reduced on Ω_{2h} .

Before we present a brief description of the multigrid method, we discuss transfer operators between the fine and coarse grids. For simplicity, let us consider multilevel structural grids $\Omega_{2h}, \Omega_{4h}, \Omega_{8h}, \dots$. Transferring a discrete function u^{2h} from the coarse grid Ω_{2h} to the fine grid Ω_h is generally called *interpolation* or *prolongation* and is denoted by I_{2h}^h .

$$u^h = I_{2h}^h u^{2h} \quad (4.5.37)$$

Many interpolation methods can be used for this purpose and the linear interpolation method described below is the simplest of these methods and is quite efficient in most cases.

* Problems 4-11 to 4-15 demonstrate the rate of error reductions for a simple equation with a simple method called the weighted Jacobi iteration method.

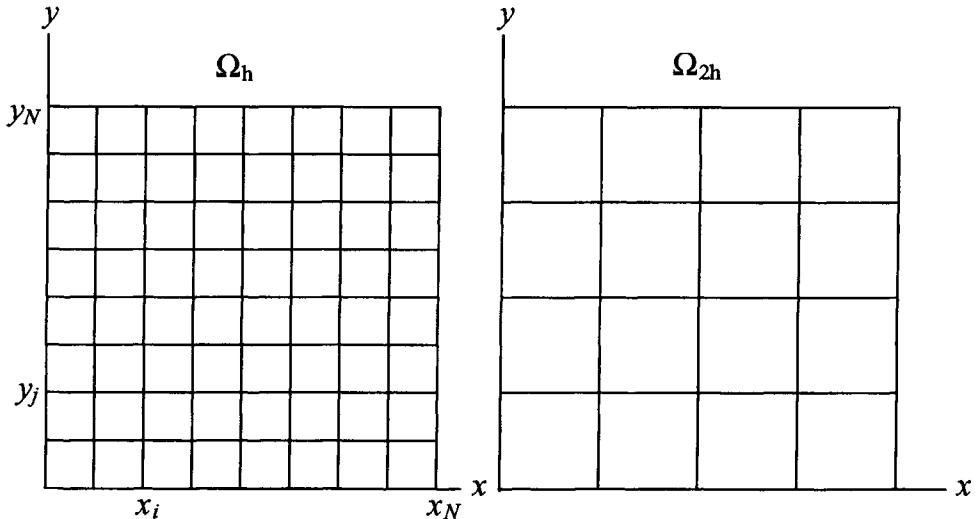


Fig. 4.10. Fine grid and coarse grid.

$$\begin{aligned}
 u_{2i,2j}^h &= u_{i,j}^{2h} \\
 u_{2i+1,2j}^h &= \frac{1}{2}(u_{i,j}^{2h} + u_{i+1,j}^{2h}) \\
 u_{2i,2j+1}^h &= \frac{1}{2}(u_{i,j}^{2h} + u_{i,j+1}^{2h}) \\
 u_{2i+1,2j+1}^h &= \frac{1}{4}(u_{i,j}^{2h} + u_{i+1,j}^{2h} + u_{i,j+1}^{2h} + u_{i+1,j+1}^{2h})
 \end{aligned} \tag{4.5.38}$$

The inter-grid transfers from fine grid Ω_h to coarse grid Ω_{2h} are generally called *restriction* operators and are denoted by I_h^{2h} .

$$u^{2h} = I_h^{2h} u^h \tag{4.5.39}$$

The simplest restriction operator called *injection* and defined by

$$u_{i,j}^{2h} = u_{2i,2j}^h \quad 1 \leq i, j \leq \frac{N}{2} - 1 \tag{4.5.40}$$

allows the values on coarse grid to obtain their values directly from the corresponding fine grid points.

An alternate *restriction* operator is called *full weighting*; it is defined by

$$\begin{aligned}
 u_{i,j}^{2h} &= \frac{1}{16}[u_{2i-1,2j-1}^h + u_{2i-1,2j+1}^h + u_{2i+1,2j-1}^h + u_{2i+1,2j+1}^h \\
 &\quad + 2(u_{2i,2j-1}^h + u_{2i,2j+1}^h + u_{2i-1,2j}^h + u_{2i+1,2j}^h) \\
 &\quad + 4u_{2i,2j}^h] \quad 1 \leq i, j \leq \frac{N}{2} - 1
 \end{aligned} \tag{4.5.41}$$

and has the important property

$$I_{2h}^h = c(I_h^{2h})^T$$

where c is constant. The above interpolation and restriction operators can be extended to general coarse and fine grids, including unstructured grids.

The basic idea of the Multigrid method for

$$\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h \quad (4.5.42)$$

has the following steps.

1. Relax n_1 times on $\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h$ on Ω_h with initial guess \mathbf{U}^h
2. Compute

$$\mathbf{r}^{2h} = I_h^{2h}(\mathbf{F}^h - \mathbf{A}_h \mathbf{U}^h) \quad (4.5.43a)$$

Solve

$$\mathbf{A}_{2h} u^{2h} = \mathbf{r}^{2h} \text{ on } \Omega_{2h} \quad (4.5.43b)$$

3. Correct fine grid approximation:

$$\mathbf{U}^h = \mathbf{U}^h + I_{2h}^h u^{2h} \quad (4.5.44)$$

4. Relax n_2 times on $\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h$ on Ω_h with initial guess \mathbf{U}^h

The step 2 for coarse grid can be carried out in several ways. Three popular procedures, called V-cycle, W-cycle and Full Multigrid (FMV), are shown in Figs. 4.11, 4.12 and 4.13. In the V-cycle, the calculations begin at A with the solutions obtained in step 1 carried out on a fine grid Ω_h . In B the calculations are performed on the coarse grid Ω_{2h} to determine the residual r^{2h} so that u^{2h} on Ω_{2h} can be determined from Eq. (4.5.43b). This procedure in B is repeated for the same number iterations in step 1 at A. In C, the procedure in B is repeated on grid Ω_{4h} . In D, on the coarsest grid, Eq. (4.5.43b) is solved with a direct method.

In E, F and G, the corrections are made to the solutions obtained in D, according to Eq. (4.5.44), for example, in E, the correction is

$$\mathbf{U}^{4h} = \mathbf{U}^{4h} + I_{8h}^{4h} u^{8h}$$

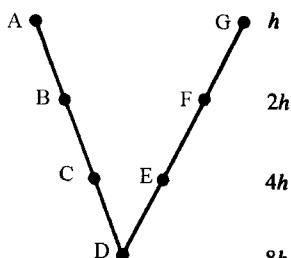


Fig. 4.11. V-cycle on four level grids.

Once \mathbf{U}^{4h} is computed according to the above equation, then step 4 is carried out on Ω_{4h} . Similarly in F and G the corrections are

$$\mathbf{U}^{2h} = \mathbf{U}^{2h} + I_{4h}^{2h} u^{4h}$$

and

$$\mathbf{U}^h = \mathbf{U}^h + I_{2h}^h u^{2h}.$$

Once a V-cycle is completed, the whole procedure, called the multigrid V-cycle method (MV) and designated as

$$\mathbf{U}^h \leftarrow \text{MV}^h(\mathbf{U}^h, \mathbf{F}^h),$$

is repeated until convergence. An algorithm that performs the above tasks is given below.

V-Cycle Method (MV)

$$\mathbf{U}^h \leftarrow \text{MV}^h(\mathbf{U}^h, \mathbf{F}^h)$$

1. Relax n_1 times on $\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h$ on Ω_h with initial guess \mathbf{U}^h
2. If Ω_h = coarsest grid, then go to 4.
Else $\mathbf{F}^{2h} = I_h^{2h}(\mathbf{F}^h - \mathbf{A}_h \mathbf{U}^h)$
 $\mathbf{U}^{2h} = 0$
 $\mathbf{U}^{2h} \leftarrow \text{MV}^{2h}(\mathbf{U}^{2h}, \mathbf{F}^{2h})$
3. Correct $\mathbf{U}^h = \mathbf{U}^h + I_{2h}^h \mathbf{U}^{2h}$
4. Relax n_2 times on $\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h$ on Ω_h with initial guess \mathbf{U}^h

A second popular procedure, called the μ -Cycle method (M μ), is shown below.

μ -Cycle Method (M μ)

$$\mathbf{U}^h \leftarrow \text{M}\mu^h(\mathbf{U}^h, \mathbf{F}^h)$$

1. Relax n_1 times on $\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h$ on Ω_h with initial guess \mathbf{U}^h
2. If Ω_h = coarsest grid, then go to 4.
Else $\mathbf{F}^{2h} = I_h^{2h}(\mathbf{F}^h - \mathbf{A}_h \mathbf{U}^h)$
 $\mathbf{U}^{2h} = 0$
 $\mathbf{U}^{2h} \leftarrow \text{M}\mu^{2h}(\mathbf{U}^{2h}, \mathbf{F}^{2h}) \quad \mu \text{ times.}$
3. Correct $\mathbf{U}^h = \mathbf{U}^h + I_{2h}^h \mathbf{U}^{2h}$
4. Relax n_2 times on $\mathbf{A}_h \mathbf{U}^h = \mathbf{F}^h$ on Ω_h with initial guess \mathbf{U}^h

The μ -Cycle method becomes the V-Cycle method for $\mu = 1$. If $\mu = 2$, the μ -Cycle method is the W-Cycle method (see Fig. 4.12). In practice, only $\mu = 1$ and $\mu = 2$ are used.

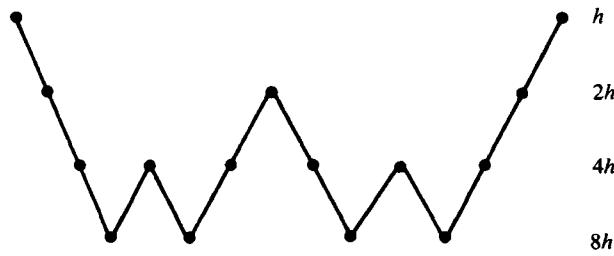


Fig. 4.12. W-cycle on four level grids.

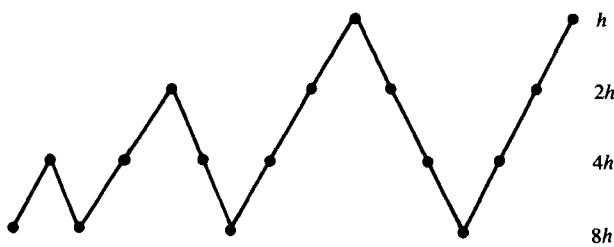


Fig. 4.13. FMV-cycle on four level grids.

Another popular procedure is the so-called *full multigrid V-cycle* (FMV) method described below. In this procedure, the calculations begin on the coarsest grid which provides a good initial guess by interpolation for the next fine grid. On this fine grid, the V-Cycle method is applied (see Fig. 4.13).

Full multigrid V-cycle (FMV) Method.

$$\mathbf{U}^h \leftarrow \text{FMV}^h(\mathbf{U}^h, \mathbf{F}^h)$$

1. Initialize $\mathbf{U}^h, \mathbf{U}^{2h}, \dots; \mathbf{F}^h, \mathbf{F}^{2h}, \dots$

Solve or relax on coarsest grid

\vdots

$$\mathbf{U}^{4h} = \mathbf{U}^{4h} + I_{8h}^{4h} \mathbf{U}^{8h}$$

$$\mathbf{U}^{4h} \leftarrow \text{MV}^{4h}(\mathbf{U}^{4h}, \mathbf{F}^{4h})$$

$$\mathbf{U}^{2h} = \mathbf{U}^{2h} + I_{4h}^{2h} \mathbf{U}^{4h}$$

$$\mathbf{U}^{2h} \leftarrow \text{MV}^{2h}(\mathbf{U}^{2h}, \mathbf{F}^{2h})$$

$$\mathbf{U}^h = \mathbf{U}^h + I_{2h}^h \mathbf{U}^{2h}$$

$$\mathbf{U}^h \leftarrow \text{MV}^h(\mathbf{U}^h, \mathbf{F}^h)$$

On the finest grid, then use the following procedure

1. If Ω_h = coarsest grid, then go to 3.

Else $\mathbf{F}^{2h} = I_h^{2h}(\mathbf{F}^h - \mathbf{A}_h \mathbf{U}^h)$

$$\mathbf{U}^{2h} = 0$$

$$\mathbf{U}^{2h} \leftarrow \text{FMV}^{2h}(\mathbf{U}^{2h}, \mathbf{F}^{2h})$$

2. Correct $\mathbf{U}^h = \mathbf{U}^h + I_{2h}^h \mathbf{U}^{2h}$

3. $\mathbf{U}^h \leftarrow \text{MV}^h(\mathbf{U}^h, \mathbf{F}^h)$ n_2 times

Example 4.7. Repeat Example 4.5 with the MV method by using two grids. Compare your results with SOR, GS, ADI and MV for a convergence criterion of

$$\max_{i,j} |Au - F| \leq \varepsilon = 10^{-7}$$

Assume that $h = \Delta x = \Delta y = 1/256$.

h	Iter	SOR		Gauss-Seidel		ADI		MV	
		MaxError	Iter	MaxError	Iter	MaxError	Iter	MaxError	Iter
1/128	467	4.12E-4	16711	1.89E-4	498	4.12E-4	41	3.40E-4	
1/256	939	1.03E-4	57584	1.24E-3	1218	1.03E-4	41	5.07E-5	
1/512	1875	2.57E-5	193398	5.29E-3	2748	2.65E-5	41	1.27E-4	

Note from these results that, unlike the other iterative methods, the convergence rate of the multigrid methods is independent of the mesh size (here 41 cycles are used for all h).

References

- [1] Crank, J. and Nicholson, P., “A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type,” *Proceedings of the Cambridge Philosophical Society*, Vol. 43, pp. 50–67, January 1947.
- [2] Keller, H. B., “A New Difference Scheme for Parabolic Problems,” *Numerical Solution of Partial Differential Equations*, Vol. II, ed. J. Bramble, Academic, New York, 1970.
- [3] Isaacson, E. and Keller, H. B., *Analysis of Numerical Methods*, John Wiley and Sons, New York, 1966.
- [4] Peaceman, D. W. and Rachford, H. H., The numerical solution of parabolic and elliptic differential equations, *SIAM Journal*, vol. 3, p. 28–41, 1955.
- [5] Brandt, A., “Guide to multigrid development,” In *Multigrid Methods*, Lecture Notes in Mathematics, Vol. 960, Springer, New York, 1982.

Problems

4-1. Show that the following forward and backward difference formulas have second order accuracy (see Section 4.3).

Forward-difference:

$$(u_x)_i = \frac{-3u_i + 4u_{i+1} - u_{i+2}}{2\Delta x} + \frac{\Delta x^2}{3} \frac{\partial^3 u}{\partial x^3} \quad (\text{P4.1.1})$$

$$(u_{xx})_i = \frac{1}{\Delta x^2} (2u_i - 5u_{i+1} + 4u_{i+2} - u_{i+3}) + \frac{11}{12} \Delta x^2 \frac{\partial^4 u}{\partial x^4} \quad (\text{P4.1.2})$$

Backward-difference:

$$(u_x)_i = \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} + \frac{\Delta x^2}{3} \frac{\partial^3 u}{\partial x^3} \quad (\text{P4.1.3})$$

$$(u_{xx})_i = \frac{1}{\Delta x^2} (2u_i - 5u_{i-1} + 4u_{i-2} - u_{i-3}) - \frac{11}{12} \Delta x^2 \frac{\partial^4 u}{\partial x^4} \quad (\text{P4.1.4})$$

4-2. Show that u_{xxx} at i can be approximated by the following forward, backward and central difference formulas.

Forward difference:

$$(u_{xxx})_i = \frac{1}{\Delta x^3} (u_{i+3} - 3u_{i+2} + 3u_{i+1} - u_i) - \frac{\Delta x^3}{2} \frac{\partial^4 u}{\partial x^4} \quad (\text{P4.2.1})$$

or

$$\begin{aligned} (u_{xxx})_i &= \frac{1}{2\Delta x^3} (-3u_{i+4} + 14u_{i+3} - 24u_{i+2} + 18u_{i+1} - 5u_i) \\ &\quad + \frac{21}{12} \Delta x^2 \frac{\partial^5 u}{\partial x^5} \end{aligned} \quad (\text{P4.2.2})$$

Backward difference:

$$(u_{xxx})_i = \frac{1}{\Delta x^3} (u_i - 3u_{i-1} + 3u_{i-2} - u_{i-3}) + \frac{\Delta x}{2} \frac{\partial^4 u}{\partial x^4} \quad (\text{P4.2.3})$$

or

$$\begin{aligned} (u_{xxx})_i &= \frac{1}{2\Delta x^3} (5u_i - 18u_{i-1} + 24u_{i-2} - 14u_{i-3} + 3u_{i-4}) \\ &\quad - \frac{21}{12} \Delta x^2 \frac{\partial^5 u}{\partial x^5} \end{aligned} \quad (\text{P4.2.4})$$

Central difference:

$$(u_{xxx})_i = \frac{1}{2\Delta x^3} (u_{i+2} - 2u_{i+1} + 2u_{i-1} - u_{i-2}) - \frac{1}{4} \Delta x^2 \frac{\partial^5 u}{\partial x^5} \quad (\text{P4.2.5})$$

or

$$\begin{aligned} (u_{xxx})_i &= \frac{1}{8\Delta x^3} (-u_{i+3} + 8u_{i+2} - 13u_{i+1} + 13u_{i-1} - 8u_{i-2} + u_{i-3}) \\ &\quad + \frac{7}{120} \Delta x^4 \frac{\partial^7 u}{\partial x^7} \end{aligned} \quad (\text{P4.2.6})$$

4-3. Show that u_{xxxx} at i can be approximated by the following forward, backward and central difference formulas.

Forward difference

$$\begin{aligned} \left(\frac{\partial^4 u}{\partial x^4} \right)_i &= (u_{xxxx})_i \\ &= \frac{1}{\Delta x^4} (u_{i+4} - 4u_{i+3} + 6u_{i+2} - 4u_{i+1} + u_i) - 2\Delta x \frac{\partial^5 u}{\partial x^5} \end{aligned} \quad (\text{P4.3.1})$$

Backward difference

$$\begin{aligned} \left(\frac{\partial^4 u}{\partial x^4} \right)_i &= (u_{xxxx})_i \\ &= \frac{1}{\Delta x^4} (u_i - 4u_{i-1} + 6u_{i-2} - 4u_{i-3} + u_{i-4}) + 2\Delta x \frac{\partial^5 u}{\partial x^5} \quad (\text{P4.3.2}) \end{aligned}$$

Central difference

$$\begin{aligned} \left(\frac{\partial^4 u}{\partial x^4} \right)_i &= (u_{xxxx})_i \\ &= \frac{1}{\Delta x^4} (u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}) - \frac{\Delta x^2}{2} \frac{\partial^6 u}{\partial x^6} \quad (\text{P4.3.3}) \end{aligned}$$

4-4. Solve Eq. (4.2.4) subject to the following boundary and initial conditions:

$$x = 0, \quad T = 0; \quad x = 1, \quad T = 0$$

$$t = 0, \quad T = 100 \sin \pi x, \quad 0 \leq x \leq 1$$

with the explicit method discussed in subsection 4.4.1 for values of $t = 0.005, 0.01, 0.02, 0.10$, with $\alpha = 0.2$, for three different spacings in t .

$$(a) \Delta t = \frac{1}{1000} \quad (b) \Delta t = \frac{5}{1000} \quad (c) \Delta t = \frac{1}{100}$$

Compare your results with the analytical solution

$$T(t, x) = 100e^{-\alpha\pi^2 t} \sin \pi x$$

at $x = 0.3$ and determine the percentage error in each case. Discuss the accuracy of the results, the behavior of the solutions and the importance of the ratio $\Delta t / (\Delta x)^2$ in each case.

4-5. Repeat Problem 4.4 using the Crank–Nicolson method. Compare your results with those obtained with the explicit method.

4-6. Repeat Problem 4.4 with Keller’s box method and compare your solutions with those obtained with the Crank–Nicolson method.

4-7. Solve Eq. (4.2.4) subject to the following boundary and initial conditions:

$$\begin{aligned} x = 0, \quad \frac{\partial T}{\partial x} &= -\pi T; \quad x = 1, \quad \frac{\partial T}{\partial x} = -\pi T \\ t = 0, \quad T &= \sin \pi(x - 1/4), \quad 0 \leq x \leq 1 \end{aligned}$$

using

- (a) an explicit method and employing central differences for the boundary conditions.
- (b) an explicit method and employing forward differences for the boundary conditions at $x = 0$
- (c) the Crank–Nicolson method with central differences for the boundary conditions.

Compare the numerical results obtained in each case with the analytical solution given by

$$T(t, x) = e^{-\alpha\pi^2 t} \sin \pi(x - 1/4)$$

Take $\alpha = 1$, $\Delta t = 0.0025$ and $\Delta x = 0.02$.

4-8. Repeat Problem 4.6 with Keller's box method and compare your solutions with those obtained in Problem 4.6.

4-9. Compute the temperatures at the grid points indicated by dots in Fig. P41 by solving the heat conduction equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

for a square region of side L with boundary conditions shown in the figure. Take $\Delta x = \delta y = L/100$. Use the direct method discussed in subsection 4.5.1.

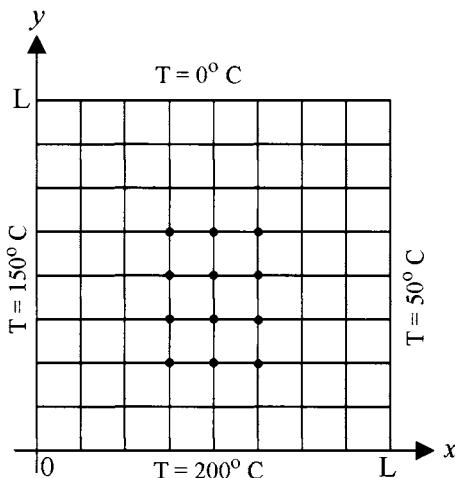


Fig. P4.1.

4-10. Repeat Problem 4.9 with the iterative methods (a) SOR, and (b) ADI methods discussed in subsection 4.5.2.

4-11. The high and low frequency errors can be identified by using the s-called modal analysis. To demonstrate this, consider the central difference approximations to,

$$-u''(x) = f(x) \quad 0 < x < 1 \quad (\text{P4.11.1})$$

subject to the boundary conditions

$$u(0) = u(1) = 0 \quad (\text{P4.11.2})$$

(a) show that Eq. (P4.11.1) can be written as

$$-u_{i-1} + 2u_i - u_{i+1} = \Delta x^2 f_i \quad 1 \leq i \leq N - 1 \quad (\text{P4.11.3})$$

or in matrix vector form as

$$\mathbf{A}\mathbf{u} = \mathbf{f} \quad (\text{P4.11.4})$$

Here the tridiagonal matrix \mathbf{A} is defined by

$$\mathbf{A} = \mathbf{D} - (\mathbf{L} + \mathbf{U}) = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \quad (\text{P4.11.5})$$

where \mathbf{D} is the diagonal of \mathbf{A} and $-\mathbf{L}$ and $-\mathbf{U}$ are the strictly lower and upper triangular parts of \mathbf{A} , respectively.

(b) Show that the eigenvalues of \mathbf{A} are

$$\lambda_k(\mathbf{A}) = 4 \sin^2 \frac{k\pi}{2N} \quad 1 \leq k \leq N - 1 \quad (\text{P4.11.6a})$$

and that the corresponding eigenvectors are

$$\mathbf{w}_k = \begin{bmatrix} \sin \frac{k\pi}{N} \\ \vdots \\ \sin \frac{ki\pi}{N} \\ \vdots \\ \sin \frac{k(N-1)\pi}{N} \end{bmatrix} = \begin{bmatrix} w_{k,1} \\ \vdots \\ w_{k,i} \\ \vdots \\ w_{k,N-1} \end{bmatrix} \quad 1 \leq k \leq N - 1, \quad (\text{P4.11.6b})$$

and that the relation between eigenvalues λ_k and eigenvector \mathbf{w}_k is

$$\mathbf{A}\mathbf{w}_k = \lambda_k \mathbf{w}_k \quad (\text{P4.11.6c})$$

The parameter k denotes the wave number. The eigenvectors with wave numbers in the range $1 \leq k < N/2$ are called low-frequency mode. The eigenvectors with $N/2 \leq k \leq N - 1$ are called high-frequency mode (See Fig. P4.2).

Hint: Show that Eq. (P4.11.6) is equivalent to

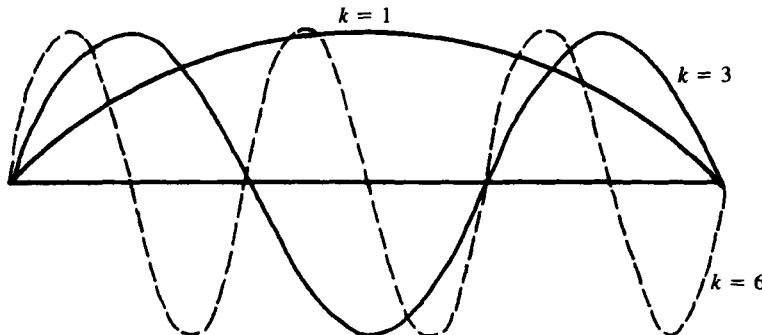


Fig. P4.2. The eigenvectors $\mathbf{w}_{k,i} = \sin(\frac{ik\pi}{N})$, $0 \leq i \leq N$ with wave numbers $k = 1, 3, 6$.

$$-w_{k,i-1} + 2w_{k,i} - w_{k,i+1} = \lambda_k w_{k,i} \quad 0 \leq i \leq N \quad (\text{P4.11.7})$$

4-12. There are several iterative methods that can be used to solve Eq. (P4.11.3). A simple one is the Jacobi iteration method, in which for an initial guess $u_i^{(0)}$, new values of $u_i^{(n+1)}$ are obtained from the previous values of $u_i^{(n)}$ according to

$$u_i^{(n+1)} = \frac{1}{2}(u_{i+1}^{(n)} + u_{i-1}^{(n)} + \Delta x^2 f_i) \quad 1 \leq i \leq N-1 \quad (\text{P4.12.1})$$

An important modification to Eq. (P4.12.1) results by introducing a weighted factor ω . First the intermediate values are calculated as in (P4.12.1)

$$u_i^* = \frac{1}{2}(u_{i+1}^{(n)} + u_{i-1}^{(n)} + \Delta x^2 f_i) \quad 1 \leq i \leq N-1 \quad (\text{P4.12.2a})$$

and to obtain $u_i^{(n+1)}$ that is,

$$u_i^{(n+1)} = u_i^{(n)} + \omega(u_i^* - u_{i-1}^{(n)}) \quad 1 \leq i \leq N-1 \quad (\text{P4.12.2b})$$

which generates an entire family of iterations called the *weighted Jacobi iteration* method. These iteration sweeps are continued until convergence. Note that for $\omega = 1$, the original Jacobi iteration is recovered

(a) Show that the Jacobi iteration method can be expressed as

$$\mathbf{D}\mathbf{u}^{(n+1)} = (\mathbf{L} + \mathbf{U})\mathbf{u}^{(n)} + \mathbf{f} \quad (\text{P4.12.3})$$

or

$$\mathbf{u}^{(n+1)} = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{A})\mathbf{u}^{(n)} + \mathbf{D}^{-1}\mathbf{f} \quad (\text{P4.12.4})$$

where I denotes the identity matrix,

$$\mathbf{I} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & & 1 \end{bmatrix}$$

\mathbf{P}_J , is called the Jacobi iteration matrix

$$\mathbf{P}_J = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}) = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}) \quad (\text{P4.12.5})$$

(b) Show that the weighted Jacobi iteration method can be written as

$$\mathbf{u}^{(n+1)} = [(1 - \omega)\mathbf{I} - \omega\mathbf{P}_J]\mathbf{u}^{(n)} + \omega\mathbf{D}^{-1}\mathbf{f} \quad (\text{P4.12.6})$$

and the weighted Jacobi iteration matrix is

$$\mathbf{P}_\omega = [(1 - \omega)\mathbf{I} - \omega\mathbf{P}_J] = \mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A} \quad (\text{P4.12.7})$$

4-13. The convergence rate of the weighted Jacobi iteration method can be determined from the eigenvalues $\lambda_k(\mathbf{P}_\omega)$ of \mathbf{P}_ω . Show that

$$\lambda_k(\mathbf{P}_\omega) = 1 - \frac{\omega}{2}\lambda_k(\mathbf{A}) = 1 - 2\omega \sin^2 \frac{k\pi}{2N} \quad 1 \leq k \leq N - 1 \quad (\text{P4.13.1})$$

and the corresponding eigenvectors are the same as those given by Eq. (P4.11.6b) and

$$\mathbf{P}_\omega \mathbf{w}_k = \lambda_k(\mathbf{P}_\omega) \mathbf{w}_k \quad (\text{P4.13.2})$$

4-14. The relation between the errors

$$\mathbf{e}^{(n)} = \mathbf{u} - \mathbf{u}^{(n)} \quad (\text{P4.14.1})$$

in each iteration can be examined by

(a) Show that

$$\mathbf{u} = (\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A})\mathbf{u} + \omega\mathbf{D}^{-1}\mathbf{f} \quad (\text{P4.14.2})$$

where \mathbf{u} denotes the exact solution of Eq. (P4.11.4).

(b) Show that

$$\mathbf{e}^{(n+1)} = (\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A})\mathbf{e}^{(n)} = \mathbf{P}_\omega \mathbf{e}^{(n)} \quad (\text{P4.14.3})$$

4-15. The initial error, denoted by,

$$\mathbf{e}^{(0)} = \mathbf{u} - \mathbf{u}^{(0)}$$

can be represented by the eigenvectors of matrix \mathbf{P}_ω in the form

$$\mathbf{e}^{(0)} = \sum_{k=1}^{N-1} c_k^0 \mathbf{w}_k \quad (\text{P4.15.1})$$

(a) By using Eqs. (P4.12.4-5) and (P4.15.1), show that the error at the nth iteration can be expressed as

$$\mathbf{e}^{(n)} = \sum_{k=1}^{N-1} \lambda_k^n(\mathbf{P}_\omega) c_k^0 \mathbf{w}_k = \sum_{k=1}^{N-1} \left[1 - 2\omega \sin^2 \left(\frac{k\pi}{2N} \right) \right]^n c_k^0 \mathbf{w}_k \quad (\text{P4.15.2})$$

and thus high and low frequency errors can be identified.

(b) Show that for $0 < \omega \leq 1$

$$\lambda_k(\mathbf{P}_\omega) < 1 \quad 1 \leq k \leq N-1$$

(c) Show that the weighted Jacobi iteration method converges with the rate

$$\max_{1 \leq k \leq N-1} |\lambda_k(\mathbf{P}_\omega)| = \lambda_1(\mathbf{P}_\omega) = 1 - 2\omega \sin^2 \left(\frac{\pi \Delta x}{2} \right) \approx 1 - \omega \frac{\pi^2 \Delta x^2}{2} \quad (\text{P4.15.3})$$

which implies that the eigenvalue associated with the lowest frequency $\lambda_1(\mathbf{P}_\omega)$ will always be close to 1 for any value ω .

(d) Show that, by requiring

$$\lambda_{N/2}(\mathbf{P}_\omega) = -\lambda_N(\mathbf{P}_\omega), \quad (\text{P4.15.4})$$

the optimal value of $\omega = 2/3$. This means that $|\lambda_k(\mathbf{P}_\omega)| < 1/3$ for all high frequency wave numbers $N/2 \leq k \leq N-1$.

Note $\omega = 2/3$ is independent of grid size Δx , and the errors corresponding to high frequency wave number are reduced much faster than those corresponding to low-frequency numbers.

4-16. Develop a 4-level multigrid program to solve

$$-u''(x) = \pi^2 \sin \pi x \quad 0 < x < 1$$

subject to the boundary conditions

$$u(0) = u(1) = 0.$$

Also show that the convergence rate is independent of grid size.

5

Numerical Methods for Model Hyperbolic Equations

5.1 Introduction

The model equation used to describe the finite-difference methods for hyperbolic equations is the linear convection equation, Eq. (4.2.5), written as

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (5.1.1)$$

with $\partial u / \partial x$ as the flux term. This equation can also be regarded as the model equation for the one-dimensional vector form of the nonlinear Euler equations given by Eqs. (2.2.30) and (2.2.45) with $R_e \rightarrow \infty$ if c is taken to be a function of u . For example, for one-dimensional flow with the neglect of viscous forces, Eq. (2.2.30) can be written as

$$\frac{\partial \tilde{Q}}{\partial \tilde{t}} + \frac{\partial \tilde{E}}{\partial \tilde{x}} = 0 \quad (5.1.2)$$

where

$$\tilde{Q} = \begin{vmatrix} \varrho \\ \varrho u \\ E_t \end{vmatrix} \equiv \begin{vmatrix} q_1 \\ q_2 \\ q_3 \end{vmatrix} \quad (5.1.3a)$$

$$\tilde{E} = \begin{vmatrix} \varrho u \\ \varrho u^2 + p \\ (E_t + p)u \end{vmatrix} \equiv \begin{vmatrix} e_1 \\ e_2 \\ e_3 \end{vmatrix} \quad (5.1.3b)$$

Equation (5.1.2) can also be written in the form

$$\frac{\partial \tilde{Q}}{\partial \tilde{t}} + A(\tilde{Q}) \frac{\partial \tilde{Q}}{\partial \tilde{x}} = 0 \quad (5.1.4)$$

with

$$A \equiv \frac{\partial \underline{E}}{\partial \underline{Q}} \quad (5.1.5a)$$

The term A is called the Jacobian matrix of the flux vector \underline{E} . For \underline{Q} and \underline{E} defined by Eqs. (5.1.3), A is given by

$$A = \begin{vmatrix} \frac{\partial e_1}{\partial q_1} & \frac{\partial e_1}{\partial q_2} & \frac{\partial e_1}{\partial q_3} \\ \frac{\partial e_2}{\partial q_1} & \frac{\partial e_2}{\partial q_2} & \frac{\partial e_2}{\partial q_3} \\ \frac{\partial e_3}{\partial q_1} & \frac{\partial e_3}{\partial q_2} & \frac{\partial e_3}{\partial q_3} \end{vmatrix} \quad (5.1.5b)$$

or (see Problem 5.5)

$$A = \begin{vmatrix} 0 & 1 & 0 \\ (\gamma - 3)\frac{u^2}{2} & (3 - \gamma)u & \gamma - 1 \\ (\gamma - 1)u^3 - \frac{\gamma u E_t}{\varrho} & -\frac{3}{2}(\gamma - 1)u^2 + \frac{\gamma E_t}{\varrho} & \gamma u \end{vmatrix} \quad (5.1.5c)$$

for a perfect gas.

In the solution of hyperbolic equations (as well as parabolic and elliptic equations) it is important to determine the direction and velocities of the propagation of information in the flowfield so that the numerical scheme is consistent with the physics of the flow. A general method for accomplishing this objective is to examine the eigenvalues of the Jacobian matrix. For the one-dimensional Euler equation given by Eq. (5.1.4), these can be obtained from

$$|A - \lambda I| = 0 \quad (5.1.6)$$

Here I is the identity matrix and λ is, by definition, an eigenvalue of the matrix A . With A given by Eq. (5.1.5c), the determinant in Eq. (5.1.6) can be expressed as a cubic equation in terms of the unknown λ ,

$$\begin{aligned} -\lambda\{[(3 - \gamma) - \lambda](\gamma u - \lambda) - (\gamma - 1)[-\frac{3}{2}(\gamma - 1)u^2 + \gamma E_t/\varrho]\} \\ - \{(\gamma - 3)\frac{u^2}{2}(\gamma u - \lambda) - (\gamma - 1)[(\gamma - 1)u^3 - \gamma u E_t/\varrho]\} = 0 \end{aligned} \quad (5.1.7)$$

The three solutions for λ , with c denoting the speed of sound,

$$c = [(1 - \gamma)\gamma(\frac{u^2}{2} - E_t/\varrho)]^{1/2} = [\gamma P/\varrho]^{1/2}$$

are given by

$$\lambda_1 = u \quad (5.1.8a)$$

$$\lambda_2 = u + c \quad (5.1.8b)$$

$$\lambda_3 = u - c \quad (5.1.8c)$$

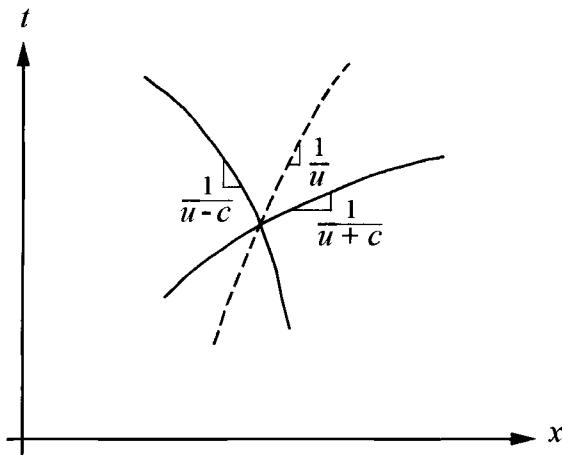


Fig. 5.1. Characteristic lines for one-dimensional unsteady flow.

These eigenvalues give the shapes of the characteristic lines in the x - t plane, as shown in Fig. 5.1. At a given point in the (x, t) plane, there are three characteristic lines with shapes

$$\frac{dt}{dx} = \frac{1}{\lambda_1} = \frac{1}{u}, \quad \frac{1}{\lambda_2} = \frac{1}{u+c}, \quad \frac{1}{\lambda_3} = \frac{1}{u-c}, \quad (5.1.9)$$

respectively. The eigenvalue λ_1 indicates that information is propagated by a fluid element moving at velocity u ; the eigenvalues λ_2 and λ_3 indicate that information is propagated to the right and left, respectively, along the x -axis at the local speed of sound relative to the moving fluid element. In Fig. 5.1, the curve with shape equal to $1/u$ is called a particle path and the curves with shapes $1/(u+c)$ and $1/(u-c)$ are right- and left-running Mach waves.

The characteristic lines play a significant role in the development of numerical methods since information concerning a flowfield travels along the characteristic curves (Section 2.6). Since the eigenvalues of the Jacobian matrix give the shapes of the characteristics and their values represent the velocity and direction of propagation of information, the solution procedure should be consistent with the velocity and direction which information propagates throughout the flowfield.

In flows where the flow variables are smooth and continuous, it is generally satisfactory to use central-difference schemes to solve hyperbolic equations. When the flow variables are not smooth, for example, when there are discontinuities in the flowfield such as shocks, the central-difference schemes are not satisfactory; the solutions exhibit oscillations which may be considerable in the vicinity of the discontinuity and can lead to unacceptable results. Upwing difference schemes are devised to overcome the shortcomings of the central schemes and, thus to obtain solutions of hyperbolic equations.

The characteristics of the system given by Eq. (5.1.4) can be determined from the procedure given in [1]. This requires that the eigenvalues of the Jacobian matrix A , Eq. (5.1.5), and given by Eq. (5.1.8) are determined with three solutions for λ from the differential equations of the characteristics. The compatibility equations can then be obtained by writing Eq. (5.1.4) along the characteristics with the left eigenvectors, $[L^i]$ determined from

$$[L^i]^T [A - \lambda_i I] = 0 \quad (5.1.10a)$$

or

$$[l_1^i \ l_2^i \ l_3^i] [A - \lambda_i I] = 0; \quad i = 1, 2, 3 \quad (5.1.10b)$$

If we let

$$X^{-1} = [L^1 \ L^2 \ L^3]^T \quad (5.1.11)$$

and multiply Eq. (5.1.4) with X^{-1} , we obtain

$$X^{-1} \left[\frac{\partial Q}{\partial t} + A \frac{\partial Q}{\partial x} \right] = 0 \quad (5.1.12)$$

and noting that

$$X^{-1} A = \Lambda X^{-1} \quad (5.1.13)$$

where Λ represents the eigenvalues of the Jacobian matrix A , that is,

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \quad (5.1.14)$$

we can also write Eq. (5.1.13) as

$$X^{-1} A X = \Lambda \quad (5.1.15)$$

Here X represents the right eigenvector matrix of A .

Assuming that the Jacobian matrix A is diagonalizable with real eigenvalues, the compatibility equations can be obtained for the case where A is either constant or nonlinear. In the former case, the left eigenvectors of X^{-1} are automatically constant and Eq. (5.1.12) becomes

$$\left[\frac{\partial X^{-1} Q}{\partial t} + \frac{\partial X^{-1} A X X^{-1} Q}{\partial x} \right] = 0 \quad (5.1.16a)$$

or

$$\left[\frac{\partial X^{-1} Q}{\partial t} + \Lambda \frac{\partial X^{-1} Q}{\partial x} \right] = 0 \quad (5.1.16b)$$

When A is nonlinear, we first determine a vector \tilde{Q} such that

$$\frac{\partial Q}{\partial \tilde{Q}} = X \quad (5.1.17)$$

and using the chain rule, we can write Eq. (5.1.4)

$$\frac{\partial Q}{\partial \tilde{Q}} \frac{\partial \tilde{Q}}{\partial t} + A \frac{\partial Q}{\partial \tilde{Q}} \frac{\partial \tilde{Q}}{\partial x} = 0$$

or

$$X \frac{\partial \tilde{Q}}{\partial t} + AX \frac{\partial \tilde{Q}}{\partial x} = 0 \quad (5.1.18)$$

Multiplying both sides of Eq. (5.1.18) with X^{-1} and using Eq. (5.1.15), we obtain

$$\frac{\partial \tilde{Q}}{\partial t} + A \frac{\partial \tilde{Q}}{\partial x} = 0 \quad (5.1.19)$$

which is similar to Eq. (5.1.1) with three uncoupled, independent equations.

Example 5.1. Assuming that

$$\frac{\partial}{\partial t} \begin{bmatrix} p \\ u \end{bmatrix} + A \frac{\partial}{\partial x} \begin{bmatrix} p \\ u \end{bmatrix} = 0 \quad (\text{E5.1.1a})$$

where

$$A = \begin{bmatrix} 0 & \beta \\ 1 & 2u \end{bmatrix} \quad (\text{E5.1.1b})$$

find (a) the eigenvalues $\lambda_{1,2}$, (b) left eigenvectors $L^{1,2}$ and (c) the right eigenvector matrix X of A

Solution:

(a) Find the eigenvalues $\lambda_{1,2}$ from Eq. (5.1.6), that is,

$$|A - \lambda I| = 0 \quad (\text{E5.1.2})$$

or

$$\lambda^2 - 2u\lambda - \beta = 0$$

Solve the above equation to get

$$\lambda_{1,2} = u \pm \sqrt{u^2 + \beta} \quad (\text{E5.1.3})$$

(b) Calculate the left eigenvectors $L^{1,2}$ with Eq. (5.1.10b), that is,

$$[l_1^i \ l_2^i] [A - \lambda_i I] = 0 \quad (\text{E5.1.4a})$$

From the definition of A and identity matrix I , Eq. (E5.1.4a) can be written as

$$\begin{aligned} -\lambda_i l_1^i + l_2^i &= 0 \\ \beta l_1^i + (2u - \lambda_i) l_2^i &= 0 \end{aligned} \quad (\text{E5.1.4b})$$

and the above equations can be solved to provide

$$l_1^i = 1; \quad l_2^i = \lambda_i \quad (\text{E5.1.5})$$

(c) From Eq. (5.1.11) we can write

$$X^{-1} = [L^1 L^2]^T = \begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix}^T = \begin{bmatrix} 1 & u+c \\ 1 & u-c \end{bmatrix} \quad (\text{E5.1.6})$$

where

$$c = \sqrt{u^2 + \beta}.$$

The right eigenvector matrix X of A is

$$X = (X^{-1})^{-1} = \frac{-1}{2c} \begin{bmatrix} u-c & -u-c \\ -1 & 1 \end{bmatrix} \quad (\text{E5.1.7})$$

and

$$X^{-1}AX = A = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} = \begin{bmatrix} u+c & \\ & u-c \end{bmatrix} \quad (\text{E5.1.8})$$

In this chapter we address the numerical solution of the scalar equation (5.1.1) or the vector equation (5.1.4) with explicit methods in Sections 5.2 and 5.3, implicit methods in Section 5.4 and upwind methods in Section 5.5. Finite volume methods are discussed in Section 5.6 and the convergence and stability of the finite-difference methods in Section 5.7.

5.2 Explicit Methods: Two-Step Lax–Wendroff Method

There are a number of explicit methods developed for hyperbolic equations and an excellent and detailed description of them is given by Hirsch [1]. Among these methods, the *two-step Lax–Wendroff* is the most important of them, due to its uniqueness for linear equations and its important role as the guideline for the development of many schemes to improve some of its deficiencies and its extension to nonlinear equations.

The two-step Lax–Wendroff method is second-order accurate and consists of two steps. (A one-step method employs only data at $t = t^n$ to obtain an approximation to $u(x, t)$ at $t = t^{n+1}$.) In step 1, the values of $u(x, t)$ are computed at $t = t^{n+1/2}$, by discretizing the flux term by central differences at $(x_{i+1/2}, t^n)$ and the time term by forward difference at $(x_{i+1/2}, t^{n+1/2})$, that is,

$$\frac{u_{i+1/2}^{n+1/2} - u_{i+1/2}^n}{\Delta t/2} + c \frac{u_{i+1}^n - u_i^n}{\Delta x} = 0 \quad (5.2.1a)$$

or

$$u_{i+1/2}^{n+1/2} = u_{i+1/2}^n - \frac{\sigma}{2}(u_{i+1}^n - u_i^n) \quad (5.2.1b)$$

where

$$u_{i+1/2}^n = \frac{1}{2}(u_i^n + u_{i+1}^n)$$

The parameter σ is the Courant number, also called the CFL (Courant–Friedrichs–Lewy) number

$$\sigma = c \frac{\Delta t}{\Delta x} \quad (5.2.2)$$

discussed in more detail in Section 5.7.

With the values of u known at $n + 1/2$ time steps for all $i + 1/2$ values, in step 2, Eq. (5.2.2) is then written as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_{i+1/2}^{n+1/2} - u_{i-1/2}^{n+1/2}}{\Delta x} = 0 \quad (5.2.3a)$$

so that u_i^{n+1} can be calculated from

$$u_i^{n+1} = u_i^n - \sigma(u_{i+1/2}^{n+1/2} - u_{i-1/2}^{n+1/2}) \quad (5.2.3b)$$

subject to initial and boundary conditions.

We shall postpone the discussion of initial and boundary conditions for the vector equation (5.1.4) to Chapter 7 and examine only the boundary conditions of the scalar equation (5.1.1) with the initial condition

$$t = 0, \quad u = f(x) \quad (5.2.4)$$

in the region $a \leq x \leq b$.

As discussed by Kreiss [2], for a well-posed problem one must specify an analytical boundary condition at the right boundary ($x = b$) if c is negative or at the left boundary ($x = a$) if c is positive. Hence, in addition to Eq. (5.1.1) and the initial data, Eq. (5.2.4), the analytical boundary conditions must be specified as either

$$x = b, \quad u = g_1(t), \quad c < 0 \quad (5.2.5a)$$

or

$$x = a, \quad u = g_2(t), \quad c > 0 \quad (5.2.5b)$$

If $\partial u / \partial x$ in Eq. (5.1.1) is replaced by a central finite-difference approximation, Eq. (4.3.7), one needs a *numerical boundary condition* at $x = b$ (called outflow boundary) if $c > 0$ or at $x = a$ (called inflow boundary) if $c < 0$. Therefore, a procedure is needed to specify the numerical boundary condition. While there are several approaches for implementing the numerical boundary conditions, see Hirsch [1], approaches based on extrapolation techniques are popular due to their simplicity and are used in Example problems 5.2 and 5.3.

Example 5.2. Use the two-step Lax–Wendroff method to solve Eq. (5.1.1) subject to the following initial and boundary conditions

$$t = 0, \quad u = \begin{cases} \sin 2\pi x & 0 \leq x \leq 1 \\ 0 & 1 \leq x \leq 5 \end{cases} \quad (E5.2.1)$$

$$x = 0, \quad u = 0$$

at $t = 4$ for $\Delta x = 0.01$ and, $\Delta t = 0.001, 0.01$, and 0.02 , $c = 1$. Compare your solution with

$$u = \begin{cases} \sin 2\pi(x - t) & t \leq x \leq t + 1 \\ 0 & \text{otherwise} \end{cases} \quad (E5.2.2)$$

and determine the percentage error. Use second-order extrapolation on the numerical boundary condition

$$u_I^n = 2u_{I-1}^n - u_{I-2}^n \quad (\text{E5.2.3})$$

at $x = 5$.

Solution:

The computer program for this problem is given in Appendix A. Table E5.1 allows a comparison of the numerical and analytical solutions (Max Error) as a function of Courant number σ . As discussed in Section 5.7, Eq. (5.7.20), stability requires $\sigma \leq 1$.

Table E5.1.

$\Delta x = 0.01$	$\sigma = 0.1$	$\sigma = 1$	$\sigma = 2$
Max Error ($t = 4$)	0.0995775	0.0002478	Divergence

5.3 Explicit Methods: MacCormack Method

The MacCormack method is a two-step predictor-corrector scheme that is a variation of the two-step Lax–Wendroff scheme and is identical to the one-step Lax–Wendroff scheme in the linear case. The MacCormack method removes the requirement of computing unknowns at the grid points $i + 1/2$ and $i - 1/2$, and because of this feature, it is very useful for solving the nonlinear unsteady Euler flow equations. Predictor values are defined at (t^{n+1}, x_i) by \bar{u}_i^{n+1} ($\equiv \bar{u}_i$) with a forward difference for the flux term, followed by a corrector step with a backward difference for the flux term. When applied to Eq. (5.1.1), this explicit predictor-corrector method becomes

$$\bar{u}_i = u_i^n - \sigma(u_{i+1}^n - u_i^n) \quad (5.3.1a)$$

$$u_i^{n+1} = \frac{1}{2}(u_i^n + \bar{u}_i^n) - \frac{\sigma}{2}(\bar{u}_i - \bar{u}_{i-1}) \quad (5.3.1b)$$

This method can be written more explicitly in a predictor-corrector sequence where the symmetry between the two steps is more apparent.

$$\bar{u}_i = u_i^n - \sigma(u_{i+1}^n - u_i^n) \quad (5.3.2a)$$

$$\bar{u}_i = u_i^n - \sigma(\bar{u}_i - \bar{u}_{i-1}) \quad (5.3.2b)$$

Updating gives

$$u_i^{n+1} = \frac{1}{2}(\bar{u}_i + \bar{u}_i) \quad (5.3.2c)$$

The above differencing can be reversed, and in some problems such as moving discontinuities it is advantageous to do so. In that case, write

$$\text{Predictor} \quad \bar{u}_i = u_i^n - \sigma(u_i^n - u_{i-1}^n) \quad (5.3.3a)$$

$$\text{Corrector} \quad \bar{u}_i = u_i^n - \sigma(\bar{u}_{i+1} - \bar{u}_i) \quad (5.3.3b)$$

$$\text{Updating} \quad u_i^{n+1} = \frac{1}{2}(\bar{u}_i + \bar{u}_i) \quad (5.3.3c)$$

Example 5.3. Repeat Example 5.2 using the MacCormack method.

Solution:

Table E5.2.

$\Delta x = 0.01$	$\sigma = 0.1$	$\sigma = 1$	$\sigma = 2$
Max Error ($t = 4$)	0.0995605	0.0000018	Divergence

5.4 Implicit Methods

The implicit methods for the hyperbolic flow equation also use central, second-order differences to discretize the spatial flux terms but use a *separate* time integration. Schemes with these properties have been applied by Briley and McDonald [3] and extensively developed by Beam and Warming [4] in conjunction with implicit linear multistep time integration methods and by Jameson et al. [5] with the fourth-order multistage Runge–Kutta time integration scheme. Both of these approaches are discussed in some detail in Chapter 12; here in this introductory exposure, the discussion is restricted to the implicit linear multistep time integration approach.

In the application of linear multistep methods (LMM) to Euler and Navier–Stokes equations, it is seldom necessary to consider more than two-step methods with three time levels. As discussed by Hirsch [1], increasing the number of time intervals can put severe restrictions on the allowable space variables and mesh points. A general two-step method with three time levels applied to the one-dimensional *scalar* form of the time-dependent Euler equation, (5.1.2), is

$$(1+\xi)\tilde{Q}^{n+1} - (1+2\xi)\tilde{Q}^n + \xi\tilde{Q}^{n-1} = -\Delta t \left[\theta \frac{\partial \tilde{E}^{n+1}}{\partial x} + (1-\theta+\phi) \frac{\partial \tilde{E}^n}{\partial x} - \phi \frac{\partial \tilde{E}^{n-1}}{\partial x} \right] \quad (5.4.1)$$

For second-order accuracy in time, the parameters (ξ, θ, ϕ) are related by

$$\phi = \xi - \theta + \frac{1}{2} \quad (5.4.2a)$$

and if, in addition,

$$\xi = 2\theta - \frac{5}{6} \quad (5.4.2b)$$

the method is third-order accurate. Several well-known methods are special cases of the general two-step method given by Eq. (5.4.1): they are summarized in Table 5.1. For further details, see Hirsch [1].

A particular family of schemes, extensively applied, is those with $\phi = 0$. Equation (5.4.1) then becomes

$$(1 + \xi)\tilde{Q}^{n+1} - (1 + 2\xi)\tilde{Q}^n + \xi\tilde{Q}^{n-1} = -\Delta t \left[\theta \frac{\partial \tilde{E}^{n+1}}{\partial x} + (1 - \theta) \frac{\partial \tilde{E}^n}{\partial x} \right] \quad (5.4.3)$$

Table 5.1. Partial list of one- and two-step methods according to Eq. (5.4.1).

θ	ξ	ϕ	Scheme	Accuracy in Time
0	0	0	Euler explicit scheme	$O(\Delta t)$
$\frac{1}{2}$	0	0	One-step implicit trapezoidal scheme	$O(\Delta t^2)$
1	0	0	Euler implicit scheme	$O(\Delta t)$
$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	Two-step implicit trapezoidal scheme	$O(\Delta t^2)$
0	$-\frac{1}{2}$	0	Explicit leapfrog scheme	$O(\Delta t^2)$

which can also be written as

$$\Delta \underline{Q}^n = -\frac{\Delta t}{(1 + \xi)} \theta \frac{\partial}{\partial x} (\Delta \underline{E}^n) - \frac{\Delta t}{(1 + \xi)} \frac{\partial \underline{\tilde{E}}^n}{\partial x} + \frac{\xi}{(1 + \xi)} \Delta \underline{Q}^{n-1} \quad (5.4.4)$$

where

$$\Delta \underline{Q}^n = \underline{Q}^{n+1} - \underline{Q}^n \quad (5.4.5a)$$

$$\Delta \underline{E}^n = \underline{E}^{n+1} - \underline{E}^n \quad (5.4.5b)$$

For $\xi = 0$, we obtain the two-level, one-step scheme, namely the generalized trapezoidal method discussed in subsection 4.4.2.

$$\Delta \underline{Q}^n = -\Delta t \theta \frac{\partial}{\partial x} (\Delta \underline{E}^n) - \Delta t \frac{\partial \underline{\tilde{E}}^n}{\partial x} \quad (5.4.6)$$

which reduces to the Crank–Nicolson method for $\theta = 1/2$.

An essential aspect of the implicit methods is connected to the linearization process of the flux derivative $\partial \underline{E}^{n+1}/\partial x$, which is almost always carried out by using the linearization scheme first introduced by Briley and McDonald [3]: the fluxes at time level $(n + 1)$ are obtained from

$$\begin{aligned} \underline{E}^{n+1} &= \underline{E}^n + \Delta t \left(\frac{\partial \underline{E}}{\partial t} \right)^n + O(\Delta t^2) \\ &= \underline{E}^n + \Delta t \left(A \frac{\partial \underline{Q}}{\partial t} \right)^n + O(\Delta t^2) \\ &= \underline{E}^n + A^n (\underline{Q}^{n+1} - \underline{Q}^n) + O(\Delta t^2) \\ &= \underline{E}^n + A^n \Delta \underline{Q}^n + O(\Delta t^2) \end{aligned} \quad (5.4.7a)$$

or

$$\Delta \underline{E}^n = A^n \Delta \underline{Q}^n + O(\Delta t^2) \quad (5.4.7b)$$

where

$$A^n = \left(\frac{\partial \underline{E}}{\partial \underline{Q}} \right)^n \quad (5.4.8)$$

Substituting Eq. (5.4.7b) into the vector equation, Eq. (5.4.4), yields

$$\left[I + \frac{\Delta t}{(1+\xi)} \theta \frac{\partial}{\partial x} A^n \right] \Delta Q^n = \frac{\xi}{(1+\xi)} \Delta Q^{n-1} - \frac{\Delta t}{(1+\xi)} \frac{\partial \tilde{E}^n}{\partial x} \quad (5.4.9)$$

where I is the identity matrix,

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.4.10)$$

This three-level Beam and Warming scheme, often called the Δ -form (delta-form), contains Q^{n-1} , Q^n and Q^{n+1} . For $\xi = 0$ and $\theta = 1/2$, it reduces to the one-step trapezoidal (Crank–Nicolson) scheme.

$$\left[1 + \frac{\Delta t}{2} \frac{\partial}{\partial x} A^n \right] \Delta Q^n = -\Delta t \frac{\partial \tilde{E}^n}{\partial x} \quad (5.4.11)$$

With the application of central differences to the spatial derivatives ($\partial A^n / \partial x$) and ($\partial \tilde{E}^n / \partial x$) in Eq. (5.4.11), this equation can be written in the tridiagonal form discussed in subsection 4.4.2 for the unsteady heat conduction equation, that is,

$$\Delta Q_i^n + \frac{\Delta t}{\Delta x} \frac{1}{4} (A_{i+1} \Delta Q_{i+1}^n - A_{i-1} \Delta Q_{i-1}^n) = -\frac{1}{2} \frac{\Delta t}{\Delta x} (E_{i+1}^n - E_{i-1}^n) \quad (5.4.12)$$

When Eq. (5.1.4) is linear, that is, $A = c$, then the above tridiagonal system has the form

$$\Delta Q_i^n + \frac{1}{4} \sigma \Delta Q_{i+1}^n - \frac{1}{4} \sigma \Delta Q_{i-1}^n = -\frac{\sigma}{2} (E_{i+1}^n - E_{i-1}^n) \quad (5.4.13)$$

The application of the one-step trapezoidal scheme to the vector equation (5.1.4) allows Eq. (5.4.11) to be generalized to the form

$$\left[I + \frac{\Delta t}{2} \frac{\partial}{\partial x} A^n \right] \Delta Q^n = -\Delta t \frac{\partial \tilde{E}^n}{\partial x} \quad (5.4.14)$$

where A is the matrix given by Eq. (5.1.5c). With the application of central differences to the spatial derivatives, Eq. (5.4.14) can again be written in a tridiagonal form; the elements a_i , b_i and c_i in the resulting equation, however, are no longer scalars but blocks, as discussed in Section 4.5. The solution of the vector equation with the Beam–Warming scheme for two-dimensional flows will be discussed in Section 12.4.

Example 5.4. Solve the inviscid Burger's equation, Eq. (4.2.8), subject to the initial and boundary conditions

$$\begin{aligned} t &= 0, & u(x, 0) &= x & 0 \leq x \leq 1 \\ x &= 0, & u(0, t) &= 0 \end{aligned} \quad (\text{E5.4.1})$$

with the Beam–Warming method (one step trapezoidal scheme). Take $\Delta x = 0.02$ and $\Delta t = 0.01$ and perform the calculations for 200 time steps. Use backward differencing for the numerical boundary condition at $x = 1$.

Solution:

First rewrite Burger's equation (4.2.8) in vector form similar to Eq. (5.1.2),

$$\frac{\partial u}{\partial t} + \frac{\partial(u^2/2)}{\partial x} = 0 \quad (\text{E5.4.2})$$

and then substitute

$$E = u^2/2, \quad A = u \quad (\text{E5.4.3})$$

in Eq. (5.4.8). The finite difference approximation to this equation is given by Eq. (5.4.9) which is applicable for all i except at the outflow boundary, $i = I$, at which we represent $\frac{\partial}{\partial x}$ with backward differencing and write

$$\Delta u_I^n + \frac{\Delta t}{\Delta x} \frac{u_I^n \Delta u_I^n - u_{I-1}^n \Delta u_{I-1}^n}{2} = -\frac{\Delta t}{\Delta x} (E_I^n - E_{I-1}^n) \quad (\text{E5.4.4})$$

Figure E5.1 shows the solutions at $t = 0, 0.5, 1, 1.5, 2$.

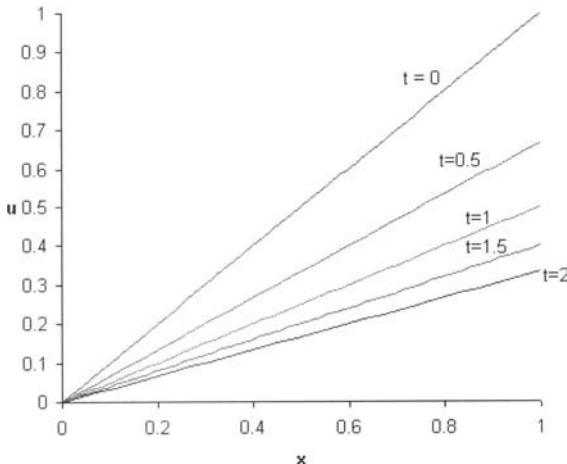


Fig. E5.1. Solution of the inviscid Burger's equation using the Beam–Warming method with initial and boundary conditions.

5.5 Upwind Methods

In the characteristics analysis of the nonlinear Euler equation, Eq. (5.1.2), Section 5.1, we have shown that the eigenvalues of this equation give the shape of the characteristics lines and indicate how information propagates along them. For example, l_1 indicates that information is propagated by a fluid element moving at velocity u ; the eigenvalues l_2 and l_3 indicate that information is propagated to the right and left, respectively, along the x -axis at the local speed of sound relative to the moving fluid element.

As we also discussed in the same section, the characteristic lines play an important role in the development of the numerical methods and, for convenience of explanation, we consider the linear convection equation, Eq. (5.1.1). In this case, information propagates along the characteristics curves specified by $dt/dx = 1/c$ [see Eq. (5.1.9)] from either the right or the left side of the solution point depending on whether $c < 0$ or $c > 0$, respectively. The propagation of information of this type is referred to as *upwind propagation* since the information comes from the direction from which the convection velocity comes, that is, the *upwind direction*. Finite difference methods that account for the upwind influence are called *upwind methods*.

To describe an upwind method for Eq. (5.1.1), we use a *one-sided space differencing* in the characteristic direction, and this results in,

$$\frac{u_i^{n+1} - u_i^n}{t^{n+1} - t^n} = -c^+ \frac{u_i^n - u_{i-1}^n}{x_i - x_{i-1}} - c^- \frac{u_{i+1}^n - u_i^n}{x_{i+1} - x_i} \quad (5.5.1)$$

where

$$c^+ = \max(c, 0) = \frac{c + |c|}{2} \quad \text{and} \quad c^- = \min(c, 0) = \frac{c - |c|}{2} \quad (5.5.2)$$

For a uniform grid in space and time, Eq. (5.5.1) becomes

$$u_i^{n+1} = u_i^n - c^+ \frac{\Delta t}{\Delta x} (u_i^n - u_{i-1}^n) - c^- \frac{\Delta t}{\Delta x} (u_{i+1}^n - u_i^n) \quad (5.5.3)$$

Note that if central differences are used for u_x in Eq. (5.1.1), the preferred paths of information are ignored and the explicit scheme becomes unstable as will be discussed in Section 5.7. However, the explicit upwind method for Eq. (5.1.1) is stable provided

$$0 \leq \left| c \frac{\Delta t}{\Delta x} \right| \leq 1 \quad (5.5.4)$$

Equation (5.5.3) can be written as

$$u_i^{n+1} - u_i^n = -\frac{\tau c}{2} (u_{i+1}^n - u_{i-1}^n) + \psi_i^n \quad (5.5.5)$$

where $\tau = \Delta t / \Delta x$ and ψ_i^n is called numerical dissipation given by

$$\psi_i^n = \frac{\tau}{2} |c| (u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (5.5.6)$$

The extension of the above approach for the linear convection equation to nonlinear partial differential equations PDEs, large systems of PDEs and multi-dimensional problems can become rather complicated. The *flux-vector-splitting* method of Steger and Warming [5] and the *flux-difference-splitting* method discussed by Osher [6] and Roe [7] provide a systematic approach to identify the direction of information propagation of each spatial derivative, and allow the

solution of nonlinear equations and multi-dimensional problems with upwind methods.

In the flux-vector-splitting method, we first determine the eigenvalues and eigenfunctions of the Jacobian matrix A , which, in the case of the nonlinear Euler equation, Eq. (5.1.2), is given by Eq. (5.1.5). This allows the system of equations to be expressed in one-dimensional form given by Eq. (5.1.19) which is similar to the convective equation, (5.1.1).

$$X^{-1}AX = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} = A = \begin{bmatrix} u & & \\ & u+c & \\ & & u-c \end{bmatrix} \quad (5.5.7)$$

where c is given in Section 5.1 and X is the product of M and N defined by

$$M = \begin{bmatrix} 1 & 0 & 0 \\ u & \varrho & 0 \\ \frac{u^2}{2} & \varrho u & \frac{1}{\gamma-1} \end{bmatrix}, \quad N = \begin{bmatrix} 1 & \frac{\varrho}{\sqrt{2}c} & \frac{\varrho}{\sqrt{2}c} \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ 0 & \frac{\varrho c}{\sqrt{2}} & \frac{\varrho c}{\sqrt{2}} \end{bmatrix} \quad (5.5.8)$$

The inverse of these two matrices are

$$M^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{-u}{\varrho} & \frac{1}{\varrho} & 0 \\ (\gamma-1)\frac{u^2}{2} & (1-\gamma)u & \gamma-1 \end{bmatrix} \quad N^{-1} = \begin{bmatrix} 1 & 0 & \frac{-1}{c^2} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}\varrho c} \\ 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}\varrho c} \end{bmatrix} \quad (5.5.9)$$

Similar to Eq. (5.5.2), we next decompose Eq. (5.5.7) and write it as

$$A^\pm = \frac{1}{2} \begin{bmatrix} \lambda_1 \pm |\lambda_1| & & \\ & \lambda_2 \pm |\lambda_2| & \\ & & \lambda_3 \pm |\lambda_3| \end{bmatrix} = \frac{1}{2}(A \pm |A|) \quad (5.5.10)$$

Based on this decomposition, we express matrix A in two parts

$$A = A^+ + A^-, \quad A^+ = X A^+ X^{-1}, \quad \text{and} \quad A^- = X A^- X^{-1} \quad (5.5.11)$$

so that, similar to Eq. (5.5.2), we can identify A^+ and A^- as corresponding to positive and negative characteristic directions.

Assuming that $E(Q)$ is a linear function of Q , Steger and Warming [5] define the flux-vector-splitting by

$$E^+ = A^+ Q \quad \text{and} \quad E^- = A^- Q \quad (5.5.12)$$

which satisfies the relations

$$E(Q) = AQ = E^+ + E^- \quad (5.5.13a)$$

$$|E| = E^+ - E^- = |A|Q \quad (5.5.13b)$$

$$\frac{\partial |E|}{\partial Q} = |A| \quad (5.5.13c)$$

In terms of the flux-vector-splitting, the upwind method for Eq. (5.1.2) can be written in a form similar to Eq. (5.5.1), that is

$$Q_i^{n+1} - Q_i^n = -\tau(E_i^+ - E_{i-1}^+)^n - \tau(E_{i+1}^- - E_i^-)^n \quad (5.5.14)$$

or with the relations given in Eq. (5.5.13),

$$Q_i^{n+1} - Q_i^n = -\frac{\tau}{2}(E_{i+1} - E_{i-1})^n + \psi_i^n \quad (5.5.15)$$

where ψ_i^n is the numerical dissipation given by

$$\psi_i^n = \frac{\tau}{2}(|E|_{i+1} - 2|E|_i + |E|_{i-1})^n \quad (5.5.16)$$

To show that Eq. (5.5.15) is first-order accurate in space and time, we apply Taylor series expansion to Eq. (5.5.13b) and obtain

$$|E|_{i+1} = |E|_i + \frac{\partial |E|_i}{\partial Q} \frac{\partial Q_i}{\partial x} \Delta x_i + O(\Delta x^2)$$

which can also be written as

$$|E|_{i+1} - |E|_i = \frac{\partial |E|_i}{\partial Q} \Delta Q_i + O(\Delta x^2)$$

or

$$|E|_{i+1} - |E|_i = |A|_{i+1/2}(Q_{i+1} - Q_i) + O(\Delta x)(Q_{i+1} - Q_i) \quad (5.5.17)$$

In terms of Eq. (5.5.17), Eq. (5.5.15) can be written as

$$\begin{aligned} Q_i^{n+1} - Q_i^n &= -\frac{\tau}{2}(E_{i+1} - E_{i-1})^n \\ &\quad + \frac{\tau}{2}[|A|_{i+1/2}(Q_{i+1} - Q_i) - |A|_{i-1/2}(Q_i - Q_{i-1})]^n \end{aligned} \quad (5.5.18)$$

In terms of the numerical flux $\tilde{E}_{i+1/2}$ defined by

$$\begin{aligned} \tilde{E}_{i+1/2} &= \frac{1}{2}\{[E(Q_{i+1}) + E(Q_i)] - |A|_{i+1/2}(Q_{i+1} - Q_i)\} \\ &= \frac{1}{2}\{[E(Q_{i+1}) + E(Q_i)] - (A_{i+1/2}^+ - A_{i+1/2}^-)\Delta Q_{i+1/2}\} \\ &= \frac{1}{2}\{[E(Q_{i+1}) + E(Q_i)] - (\Delta E_{i+1/2}^+ - \Delta E_{i+1/2}^-)\} \end{aligned} \quad (5.5.19)$$

Equation (5.5.18) can be written in a compact form,

$$Q_i^{n+1} - Q_i^n = -\frac{\Delta t}{\Delta x}(\tilde{E}_{i+1/2}^n - \tilde{E}_{i-1/2}^n) \quad (5.5.20)$$

where

$$\Delta Q_{i+1/2} = Q_{i+1} - Q_i \quad (5.5.21a)$$

$$\Delta E_{i+1/2}^\pm = A^\pm(\bar{Q}_{i+1/2})\Delta Q_{i+1/2} \quad (5.5.21b)$$

$$\bar{Q}_{i+1/2} = \frac{Q_{i+1} + Q_i}{2} \quad (5.5.21c)$$

Thus, in an upwind method using the flux-vector-splitting method, the derivative of the convective flux in Eq. (5.1.2) is approximated by

$$\frac{\partial E}{\partial x} \approx \frac{\tilde{E}_{i+1/2} - \tilde{E}_{i-1/2}}{x_{i+1/2} - x_{i-1/2}} = \frac{\tilde{E}_{i+1/2} - \tilde{E}_{i-1/2}}{\Delta x} \quad (5.5.22)$$

Equation (5.5.19) can be generalized and expressed in the form

$$\tilde{E}_{i+1/2} = \frac{1}{2}\{[E(Q_{i+1}) + E(Q_i)] - \phi_{i+1/2}\}, \quad (5.5.23)$$

where $\phi_{i+1/2}$ denotes the dissipation term which can be of first-order, second-order or higher-order. A summary of dissipation terms of different orders are given below.

Case 1. If there is no numerical dissipation, that is

$$\phi_{i+1/2} = 0, \quad (5.5.24)$$

Equation (5.5.22) becomes

$$\frac{\partial E}{\partial x} \approx \frac{E_{i+1} - E_{i-1}}{2\Delta x}$$

The corresponding numerical method

$$Q_i^{n+1} - Q_i^n = -\frac{\Delta t}{2\Delta x}(E_{i+1} - E_{i-1})$$

is unstable.

Case 2. For a *first-order* accurate upwind method (described by Eq. (5.5.19)),

$$\phi_{i+1/2} = \Delta E_{i+1/2}^+ - \Delta E_{i+1/2}^- \quad (5.5.25)$$

First-order upwind methods introduce significant amounts of implicit numerical diffusion into the solution. Higher-order upwind methods give more accurate solutions.

Case 3. For the *third-order* accurate upwind method,

$$\phi_{i+1/2} = -\frac{1}{3}(\Delta E_{i-1/2}^+ - \Delta E_{i+1/2}^+ + \Delta E_{i+1/2}^- - \Delta E_{i+3/2}^-) \quad (5.5.26)$$

Case 4. For the *fifth-order* accurate upwind method

$$\begin{aligned} \phi_{i+1/2} = & -\frac{1}{30}(-2\Delta E_{i-3/2}^+ + 11\Delta E_{i-3/2}^+ - 6\Delta E_{i+1/2}^+ - 3\Delta E_{i+3/2}^+ \\ & + 2\Delta E_{i+5/2}^- - 11\Delta E_{i+3/2}^- + 6\Delta E_{i+1/2}^- + 3\Delta E_{i-1/2}^-) \end{aligned} \quad (5.5.27)$$

The flux-difference-splitting method discussed by Osher [6] and Roe [7] is based on Godenov and Riemann solver as discussed in detail in [1]. Unfortunately this method is beyond the scope of this text, and only a very brief description is given below.

If we use the exact formula for $|E|_{i+1} - |E|_i$ in Eq. (5.5.17)

$$|E|_{i+1} - |E|_i = \int_{Q_i}^{Q_{i+1}} \left| \frac{\partial E}{\partial Q} \right| dQ = \int_{Q_i}^{Q_{i+1}} |A| dQ, \quad (5.5.28)$$

Equation (5.5.15) can be written as

$$Q_i^{n+1} - Q_i^n = -\frac{\tau}{2}(E_{i+1} - E_{i-1})^n + \frac{\tau}{2} \left[\int_{Q_i}^{Q_{i+1}} |A| dQ - \int_{Q_{i-1}}^{Q_i} |A| dQ \right]^n$$

Then the numerical flux $\tilde{E}_{i+1/2}$ becomes

$$\tilde{E}_{i+1/2} = \frac{1}{2} \left\{ [E(Q_{i+1}) + E(Q_i)] - \int_{Q_i}^{Q_{i+1}} |A| dQ \right\} \quad (5.5.29)$$

5.6 Finite-Volume Methods

While finite-difference methods are based on a discretization of the differential form of the conservation equations, the finite-volume methods are based on a discretization of the integral forms of the conservation equations. To examine the numerical solution of the conservation equations with this approach, consider the generic form of the conservation integral equation given by Eq. (2.2.24), that is,

$$\frac{\partial}{\partial t} \iiint_{\Omega} U d\Omega + \iint_S \vec{F} \cdot d\vec{S} = \iiint_{\Omega} Q_v d\Omega + \iint_S Q_S dS \quad (2.2.24)$$

where, according to the usual sign convention for control volumes, \vec{S} is perpendicular to the control surface in a direction away from the control volume.

In order to discretize this equation, as with the finite-difference methods, it is necessary to divide the physical space into a discrete network of cells. Two

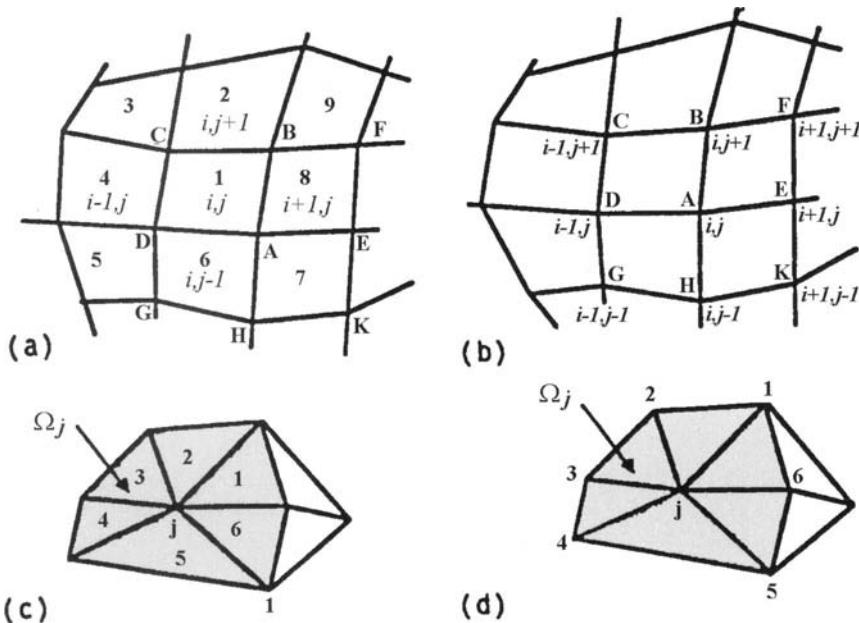


Fig. 5.2. Two-dimensional finite-volume mesh systems. (a) Cell centered structured finite-volume mesh; (b) cell vertex structured finite-volume mesh; (c) cell centered unstructured finite-volume mesh; (d) cell vertex unstructured finite-volume mesh.

kinds of meshes corresponding to structured and unstructured meshes, shown in Fig. 5.2, can be used for this purpose. The structured meshes refer to the type of meshes (Fig. 5.2a,b) where all mesh points lie on the intersection of two (or three) families of lines. The unstructured meshes are those formed by combinations of triangular (Fig. 5.2c,d) and quadrilateral cells (or tetrahedra and pyramids in three dimensions) and the mesh points cannot be identified with coordinate lines. Therefore, they cannot be represented by a set of integers such as i, j but must be numbered individually in a certain order. For this reason, the use of unstructured meshes require more computer memory and computer time. While they are more suitable for complex geometries, the structured meshes are efficient to use for simpler geometries.

The generation of structured meshes or grids will be considered in Chapter 9. This section considers the structured grid shown in Fig. 5.2a and describes the numerical solution of Eq. (2.2.24) with the finite-volume approach. The purpose of the indices i, j here is different from the purpose of indices used in the finite-difference methods. There they refer to a series of discrete grid points, while indices in Fig. 5.2a serve to identify specific cells and do not coincide with any fixed points in space. The points A, B, C and D, on the other hand, represent fixed points in space, specifying the location of the vertices of the cell denoted by (i, j) .

The cells shown in a two-dimensional x, y -plane in Fig. 5.2a are planar quadrilaterals and should be visualized such that each cell has a unit depth perpendicular to the page; hence each cell has a finite volume equal to the area of the cell times the unit perpendicular distance. Thus for the cell identified by (i, j) , the volume denoted by Ω_{ij} actually refers to the area of cell (i, j) .

To elaborate on the discretization of the integral conservation equations in the form given by Eq. (2.2.24), consider the volume Ω_{ij} as the control volume and evaluate Eq. (2.2.24) over this control volume. The first term of Eq. (2.2.24) becomes

$$\frac{\partial}{\partial t} \iiint_{\Omega} U d\Omega = \frac{\partial}{\partial t} (U_{ij} \Omega_{ij}) \quad (5.6.1a)$$

The remaining terms of Eq. (2.1.24) can be written as

$$\iint_S \vec{F} \cdot d\vec{S} = \sum_{\text{sides}} (\vec{F} \cdot \vec{S}) \quad (5.6.1b)$$

$$\iiint_{\Omega} Q_v d\Omega = (Q_v)_{ij} \Omega_{ij} \quad (5.6.1c)$$

$$\iint_S Q_s dS = \sum_{\text{sides}} (Q_s)_{ij} S \quad (5.6.1d)$$

where the sum of the flux terms refers to all the external sides of the control volume Ω_{ij} . Inserting Eqs. (5.6.1) into Eq. (2.2.24), we obtain the discretized form of the conservation equation given by Eq. (2.2.24), that is,

$$\frac{\partial}{\partial t} (U_{ij} \Omega_{ij}) + \sum_{\text{sides}} (\vec{F} \cdot \vec{S}) = (Q_v)_{ij} \Omega_{ij} + \sum_{\text{sides}} (Q_s)_{ij} S \quad (5.6.2)$$

for each cell in the physical space.

Consider the application of this equation in the absence of the source term Q_v to the Euler equations expressed in a Cartesian coordinate system in the form given by the reduced form of Eq. (2.2.30),

$$\frac{\partial \underline{Q}}{\partial t} + \frac{\partial \underline{E}}{\partial x} + \frac{\partial \underline{F}}{\partial y} = 0 \quad (5.6.3)$$

Here \underline{E} and \underline{F} represent the flux terms defined in Eq. (2.2.32a). To comply with the notation in Eq. (2.2.24) or (5.6.2), rewrite Eq. (5.6.3) as

$$\frac{\partial \underline{Q}}{\partial t} + \frac{\partial \underline{E}_1}{\partial x} + \frac{\partial \underline{F}_1}{\partial y} = - \left(\frac{\partial \underline{E}_2}{\partial x} + \frac{\partial \underline{F}_2}{\partial y} \right) \quad (5.6.4)$$

by defining \underline{E}_1 , \underline{E}_2 , \underline{F}_1 and \underline{F}_2 by

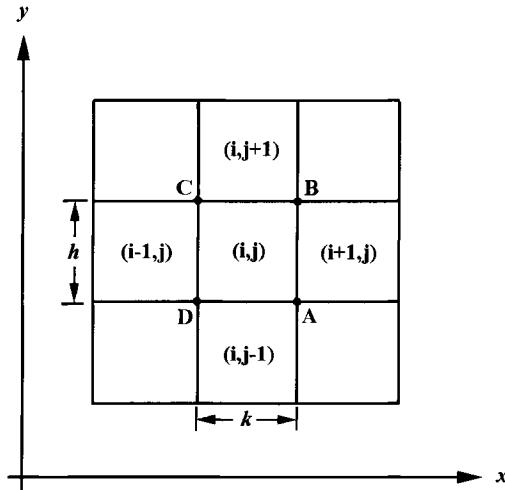


Fig. 5.3. Grid for the rectangular cell-centered geometry.

$$\underline{E}_1 = \begin{vmatrix} \varrho u \\ \varrho u^2 \\ \varrho uv \\ E_t u \end{vmatrix}, \quad \underline{F}_1 = \begin{vmatrix} \varrho v \\ \varrho uv \\ \varrho v^2 \\ E_t v \end{vmatrix} \quad (5.6.5a)$$

and

$$\underline{E}_2 = \begin{vmatrix} 0 \\ p \\ 0 \\ pu \end{vmatrix}, \quad \underline{F}_2 = \begin{vmatrix} 0 \\ 0 \\ p \\ pv \end{vmatrix} \quad (5.6.5b)$$

Represent the components of the flux vectors \underline{E}_1 and \underline{F}_1 for one equation with e and f so that the flux vector \vec{F} in Eq. (5.6.2) can be written as

$$\vec{F} = e\vec{i} + f\vec{j} \quad (5.6.6)$$

For simplicity, consider a rectangular cell-centered geometry shown in Fig. 5.3, where k and h are the same for all cells but may not be equal to each other. A similar geometry can also be considered for cell-vertex mesh. Recalling that elemental surface area $d\vec{S}$ is normal to the surface and (by convention) is positive away from the surface and the unit vectors \vec{i} and \vec{j} are in the positive directions of x and y , write Eq. (5.6.2) for the control cell ABCD in Fig. 5.3 as

$$\sum_{\text{sides}} \vec{F} \cdot \vec{S} = e_{AB}h + f_{BC}k - e_{CD}k - f_{DA}k \\ = (f_{BC} - f_{DA})k + (e_{AB} - e_{CD})h \quad (5.6.7)$$

The evaluation of flux components along the sides f_{BC} , e_{AB} , depends on the location of the flow variables with respect to the mesh and on the selected

scheme, central scheme, or the upwind scheme discussed in Section 5.5. For a central cell-centered finite-volume method there are several choices for evaluating the flux terms as discussed by Hirsch [1]. Consider here the choice of averaging the fluxes; for example, write e_{AB} as

$$e_{AB} = \frac{1}{2}(e_{i+1,j} + e_{i,j}) = e_{i+1/2,j} \quad (5.6.8a)$$

and f_{BC} as

$$f_{BC} = \frac{1}{2}(f_{i,j+1} + f_{i,j}) = f_{i,j+1/2} \quad (5.6.8b)$$

Eq. (5.6.7) can then be written as

$$\sum_{\text{sides}} \vec{F} \cdot \vec{S} = (f_{i,j+1/2} - f_{i,j-1/2})k + (e_{i+1/2,j} - e_{i-1/2,j})h \quad (5.6.9)$$

Apply Eq. (5.6.1d) to the x - and y -components of the momentum equations since, for the continuity equation, $Q_s = 0$. For the x -component of the momentum equation, write Eq. (5.6.1d) for the control volume cell ABCD as

$$\sum_{\text{sides}} Q_s S = -(p_{AB} - p_{CD})h = \frac{1}{2}(p_{i-1,j} - p_{i+1,j})h \quad (5.6.10a)$$

and for the y -component

$$\sum_{\text{sides}} Q_s S = -(p_{BC} - p_{DA})h = \frac{1}{2}(p_{i,j-1} - p_{i,j+1})k \quad (5.6.10b)$$

Similarly for the energy equation, write

$$\begin{aligned} \sum_{\text{sides}} Q_s S &= -[(pu)_{AB} - (pu)_{CD}]h - [(pv)_{BC} - (pv)_{DA}]k \\ &= \frac{1}{2}[(pu)_{i-1,j} - (pu)_{i+1,j}]h + \frac{1}{2}[(pv)_{i,j-1} - (pv)_{i,j+1}]k \end{aligned} \quad (5.6.11)$$

Inserting the expression given by Eq. (5.6.9) into Eq. (5.6.2) and noting that $\Omega_{ij} = kh$, write the left-hand side of the generic form of the conservation integral equation (2.2.24) as

$$\frac{\partial U_{ij}}{\partial t} + \frac{e_{i+1/2,j} - e_{i-1/2,j}}{k} + \frac{f_{i,j+1/2} - f_{i,j-1/2}}{h}$$

or

$$\frac{\partial U_{ij}}{\partial t} + \frac{e_{i+1,j} - e_{i-1,j}}{2k} + \frac{f_{i,j+1} - f_{i,j-1}}{2h} \quad (5.6.12)$$

Equation (5.6.2) with its right-hand side given by the expressions in Eq. (5.6.10) leads to second-order accurate space discretizations on Cartesian meshes. It should be noted that U_{ij} does not represent the quantity at a specific grid point; it represents an average value of the quantity for cell (i, j) . Similarly,

$e_{i+1,j}$ denotes an average value of e for cell $i + 1, j$, etc. Also Eq. (5.6.2) is treated as an equation for solving U_{ij} as the unknown.

Equation (5.6.2) can be applied to the continuity, momentum and energy equations. Since for the continuity equation, $Q_s = 0$, Eq. (5.6.2) becomes

$$\frac{\partial \varrho_{ij}}{\partial t} + \frac{(\varrho u)_{i+1,j} - (\varrho u)_{i-1,j}}{2k} + \frac{(\varrho v)_{i,j+1} - (\varrho v)_{i,j-1}}{2h} = 0 \quad (5.6.13)$$

since $U = \varrho$, $e = \varrho u$ and $f = \varrho v$. Similarly the x -component of the momentum equation can be written as

$$\frac{\partial}{\partial t}(\varrho u)_{ij} + \frac{(\varrho u^2)_{i+1,j} - (\varrho u^2)_{i-1,j}}{2k} + \frac{(\varrho uv)_{i,j+1} - (\varrho uv)_{i,j-1}}{2h} = \frac{p_{i-1,j} - p_{i+1,j}}{2k} \quad (5.6.14)$$

and the energy equation as

$$\begin{aligned} \frac{\partial}{\partial t}(E_t)_{ij} &+ \frac{(E_t u)_{i+1,j} - (E_t u)_{i-1,j}}{2k} + \frac{(E_t v)_{i,j+1} - (E_t v)_{i,j-1}}{2h} \\ &= \frac{(pu)_{i-1,j} - (pu)_{i+1,j}}{2k} + \frac{(pv)_{i,j-1} - (pv)_{i,j+1}}{2h} \end{aligned} \quad (5.6.15)$$

The solution of Eqs. (5.6.12) to (5.6.14) together with the y -component equation can be obtained conveniently with an explicit time-marching solution based, for example, on a standard Runge–Kutta numerical integration with respect to time. As in time-marching solution, the flowfield is assumed to be at time t^n . In Eqs. (5.6.12) to (5.6.14), the right-hand sides are evaluated at t^n , thus they are known. These equations are of the form

$$\left(\frac{dU_{ij}}{dt} \right)^n = C \quad (5.6.16)$$

with C denoting a constant which is different for each conservation equation. This equation represents a system of first-order ordinary differential equations in t that can be integrated conveniently with a fourth-order Runge–Kutta method described in Chapter 12 to calculate the values of U_{ij} at the next time step, t^{n+1} as shall be discussed in some detail in Chapter 12.

Example 5.5. Use the finite volume method with (a) central and (b) upwind differencing to solve the one-dimensional steady convection and diffusion equation

$$\begin{aligned} \frac{d}{dx}(u\phi) &= \frac{d}{dx}\left(\nu \frac{d\phi}{dx}\right) \quad 0 < x < L \\ \phi(0) &= 1 \quad \phi(L) = 0 \end{aligned} \quad (E5.5.1)$$

where $u = 2.5$ and $\nu = 0.1$. Compare your results with the exact solution $\phi(x)$

$$\frac{\phi(x) - \phi(0)}{\phi(L) - \phi(0)} = \frac{\exp(ux/\nu) - 1}{\exp(uL/\nu) - 1} \quad (E5.5.2)$$

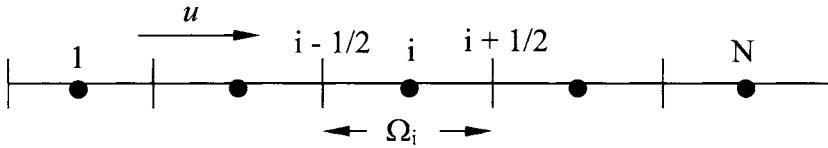


Fig. E5.2. Grid for finite volume.

Solution:

(a) Central differences

First integrate Eq. (E5.5.1) [see Eq. (2.1.24)] in the interval Ω_i , $x_{i-1/2} \leq x \leq x_{i+1/2}$, for the grid shown in Fig. E5.2.

$$\int_{\Omega_i} \frac{d}{dx}(u\phi)dx = \int_{\Omega_i} \frac{d}{dx} \left(\nu \frac{d\phi}{dx} \right) dx \quad (\text{E5.5.3})$$

Applying central differences to the above equation yields,

$$(u\phi)_{i+1/2} - (u\phi)_{i-1/2} = \left(\nu \frac{d\phi}{dx} \right)_{i+1/2} - \left(\nu \frac{d\phi}{dx} \right)_{i-1/2} \quad (\text{E5.5.4})$$

and

$$u_{i+1/2} \frac{\phi_{i+1} + \phi_i}{2} - u_{i-1/2} \frac{\phi_i + \phi_{i-1}}{2} = \nu_{i+1/2} \frac{\phi_{i+1} - \phi_i}{\Delta x} - \nu_{i-1/2} \frac{\phi_i - \phi_{i-1}}{\Delta x} \quad (\text{E5.5.5})$$

for $1 < i < N$. For $i = 1$ and $i = N$, (E5.5.4) becomes

$$u_{1+1/2} \frac{\phi_2 + \phi_1}{2} - u_{1-1/2} \phi(0) = \nu_{1+1/2} \frac{\phi_2 - \phi_1}{\Delta x} - \nu_{1-1/2} \frac{\phi_1 - \phi(0)}{\Delta x/2} \quad (\text{E5.5.6})$$

and

$$u_{N+1/2} \phi(L) - u_{N-1/2} \frac{\phi_N + \phi_{N-1}}{2} = \nu_{N+1/2} \frac{\phi(L) - \phi_N}{\Delta x/2} - \nu_{N-1/2} \frac{\phi_N - \phi_{N-1}}{\Delta x} \quad (\text{E5.5.7})$$

respectively.

Rearrange Eqs. (E5.5.6), (E5.5.5) and (E5.5.7) in the form

$$\left(\frac{u_{1+1/2}}{2} + \frac{\nu_{1+1/2}}{\Delta x} + \frac{2\nu_{1-1/2}}{\Delta x} \right) \phi_1 + \left(\frac{u_{1+1/2}}{2} - \frac{\nu_{1+1/2}}{\Delta x} \right) \phi_2 = \left(u_{1-1/2} + \frac{2\nu_{1-1/2}}{\Delta x} \right) \phi(0) \quad (\text{E5.5.8a})$$

$$\left(-\frac{u_{i-1/2}}{2} - \frac{\nu_{i-1/2}}{\Delta x} \right) \phi_{i-1} + \left(\frac{u_{i+1/2}}{2} - \frac{u_{i-1/2}}{2} + \frac{\nu_{i+1/2}}{\Delta x} + \frac{\nu_{i-1/2}}{\Delta x} \right) \phi_i + \left(\frac{u_{i+1/2}}{2} - \frac{\nu_{i+1/2}}{\Delta x} \right) \phi_{i+1} = 0 \quad (\text{E5.5.8b})$$

$$\begin{aligned} \left(-\frac{u_{N-1/2}}{2} - \frac{\nu_{N-1/2}}{\Delta x} \right) \phi_{N-1} + \left(-\frac{u_{N-1/2}}{2} + \frac{2\nu_{N+1/2}}{\Delta x} + \frac{\nu_{N-1/2}}{\Delta x} \right) \phi_N \\ = \left(-u_{N+1/2} + \frac{2\nu_{N+1/2}}{\Delta x} \right) \phi(L) \end{aligned} \quad (\text{E5.5.8c})$$

The above system has a tridiagonal form and is solved with the Thomas algorithm given in Table 4.1.

Figure E5.3 compares the numerical and exact solutions for $N = 10$ and shows that the numerical solutions oscillate in the region $0.7 \leq x \leq 1$.

(b) Upwind method

Unlike central differences, the upwind method represents the convective term with the values from upstream. Since u is positive, we can write

$$(u\phi)_{i+1/2} = u_{i+1/2}\phi_i, \quad (u\phi)_{i-1/2} = u_{i-1/2}\phi_{i-1} \quad (\text{E5.5.9})$$

and Eq. (E5.5.4) is represented by

$$u_{i+1/2}\phi_i - u_{i-1/2}\phi_{i-1} = \nu_{i+1/2}\frac{\phi_{i+1} - \phi_i}{\Delta x} - \nu_{i-1/2}\frac{\phi_i - \phi_{i-1}}{\Delta x} \quad (\text{E5.5.10})$$

At nodes 1 and N , Eq. (E5.5.4) becomes

$$u_{1+1/2}\phi_1 - u_{1-1/2}\phi(0) = \nu_{1+1/2}\frac{\phi_2 - \phi_1}{\Delta x} - \nu_{1-1/2}\frac{\phi_1 - \phi(0)}{\Delta x/2} \quad (\text{E5.5.11})$$

and

$$u_{N+1/2}\phi_N - u_{N-1/2}\phi_{N-1} = \nu_{N+1/2}\frac{\phi(L) - \phi_N}{\Delta x/2} - \nu_{N-1/2}\frac{\phi_N - \phi_{N-1}}{\Delta x} \quad (\text{E5.5.12})$$

Express Eqs. (E5.5.10–12) in a tridiagonal form and solve

$$\left(u_{1+1/2} + \frac{\nu_{1+1/2}}{\Delta x} + \frac{\nu_{1-1/2}}{\Delta x/2}\right)\phi_1 - \frac{\nu_{1+1/2}}{\Delta x}\phi_2 = \left(u_{1-1/2} + \frac{\nu_{1-1/2}}{\Delta x/2}\right)\phi(0) \quad (\text{E5.5.13a})$$

$$\left(-u_{i-1/2} - \frac{\nu_{i-1/2}}{\Delta x}\right)\phi_{i-1} + \left(u_{i+1/2} + \frac{\nu_{i+1/2}}{\Delta x} + \frac{\nu_{i-1/2}}{\Delta x}\right)\phi_i - \frac{\nu_{i+1/2}}{\Delta x}\phi_{i+1} = 0 \quad (\text{E5.5.13b})$$

$$\left(-u_{N-1/2} - \frac{\nu_{N-1/2}}{\Delta x}\right)\phi_{N-1} + \left(u_{N-1/2} + \frac{2\nu_{N+1/2}}{\Delta x} + \frac{\nu_{N-1/2}}{\Delta x}\right)\phi_N = \frac{2\nu_{N+1/2}}{\Delta x}\phi(L) \quad (\text{E5.5.13c})$$

The calculated results obtained with the Thomas algorithm are given below. Note that while the upwind method produces smooth solutions in the region $0.7 \leq x \leq 1$, the solutions do not agree well with the exact solutions in that region. This is because the upwind scheme is first-order accurate.

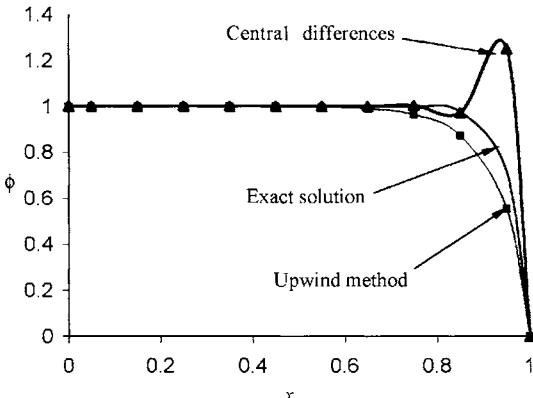


Fig. E5.3. Comparison of numerical and exact solutions.

5.7 Convergence and Stability

If U denotes the exact solution of a partial-differential equation with independent variables x and t , and the exact solution of the difference equations is u , then the difference $(U - u)$ is called the discretization error. Its magnitude at any grid point depends on the mesh widths and on the number of finite-differences in the truncated series used to approximate the derivatives. The finite-difference approximations are said to be converged when the discretization error approaches to zero as mesh widths Δt and Δx both approach to zero.

The round-off error refers to the difference between the exact solution of the difference equations, u and its approximating difference equation, N . It will be zero if in the solution of the finite-difference equations the calculations can be performed for an infinite number of decimal places. In practice this is not possible and the calculations are performed for a finite-number of decimal places, leading to round-off errors. In general, the solution of the finite-difference equations is stable when the cumulative effect at all the rounding errors is negligible.

More specifically, if errors ε_i^n are introduced at the grid points (i, n) , and each value of $|\varepsilon_i^n|$ is less than δ , then the difference equations are stable when the maximum value of the round-off error $(u - N)$ approaches zero as δ approaches zero and does not increase exponentially with the number of columns or rows in the calculation. i.e., with 1 or n . The latter condition is necessary because in certain cases the errors may not decrease exponentially with i or n but persist as linear combinations of the initial errors. In many cases they are numerically tolerable provided their sum remains much smaller than u .

The growth of errors in the computations can be examined by expressing the equations in matrix form and analyzing the eigenvalues of an associated matrix or by expressing them in a finite Fourier series. Here, due to its simplicity, the Fourier series method, which is also known as the von Neumann analysis, is discussed. For simplicity, the discussion is restricted to a single equation. The stability analysis for systems of equations can be found in several references, see for example, Hirsch [1].

To describe the stability analysis of a numerical scheme for a linear equation with the von Neumann analysis, consider the explicit approximation to the unsteady heat conduction equation, Eq. (4.4.3a). If \bar{T}_i^n is the exact solution of the difference equation and T_i^n the actual computed solution, then ε_i^n in

$$T_i^n = \bar{T}_i^n + \varepsilon_i^n \quad (5.7.1)$$

represents the error at time step n at grid point i . Substituting Eq. (5.7.1) into the difference equation, Eq. (4.4.3a),

$$\begin{aligned}\frac{1}{\Delta t} (\bar{T}_i^{n+1} - \bar{T}_i^n) + \frac{1}{\Delta t} (\varepsilon_i^{n+1} - \varepsilon_i^n) &= \frac{\alpha}{(\Delta x)^2} (\bar{T}_{i+1}^n - 2\bar{T}_i^n + \bar{T}_{i-1}^n) \\ &\quad + \frac{\alpha}{\Delta x^2} (\varepsilon_{i+1}^n - 2\varepsilon_i^n + \varepsilon_{i-1}^n)\end{aligned}\quad (5.7.2)$$

Since the exact solution \bar{T}_i^n satisfies Eq. (4.4.3a), then the equation

$$\frac{1}{\Delta t} (\varepsilon_i^{n+1} - \varepsilon_i^n) = \frac{\alpha}{(\Delta x)^2} (\varepsilon_{i+1}^n - 2\varepsilon_i^n + \varepsilon_{i-1}^n) \quad (5.7.3)$$

represents the errors ε_i^n and is identical to the basic scheme. Thus, the errors ε_i^n evolve over time in the same manner as the numerical solution T_i^n . If the errors ε_i decrease or stay the same, at some stage of the solution, the solution will be stable as the solution progresses from time step n to $n+1$; on the other hand, if ε_i 's become larger (amplify) during the progression from n to $n+1$, then the solution is unstable. Thus, for a stable solution,

$$\left| \frac{\varepsilon_i^{n+1}}{\varepsilon_i^n} \right| \leq 1 \quad (5.7.4)$$

In the stability analysis of a linear equation by the Fourier series method, the error can be written in the form

$$\varepsilon(x, t) = \sum_{m=1} e^{at} e^{ik_m x} \quad (5.7.5)$$

where a is a constant and k_m is the wave number given by

$$k_m = \frac{m\pi}{L} \quad m = 1, 2, 3, \dots \quad (5.7.6)$$

with L corresponding to the length of the domain on which the equation is being solved.

Since the difference equation under consideration is linear, and separate solutions are additive (method of superposition), it is necessary to consider only the propagation of the error due to one term of the series,

$$\varepsilon_m(x, t) = e^{at} e^{ik_m x} \quad (5.7.7)$$

Substituting Eq. (5.7.7) into Eq. (5.7.3), we obtain

$$e^{a(t+\Delta t)} e^{ik_m x} - e^{at} e^{ik_m x} = \frac{\alpha \Delta t}{(\Delta x)^2} [e^{at} e^{ik_m (x+\Delta x)} - 2e^{at} e^{ik_m x} + e^{at} e^{ik_m (x-\Delta x)}] \quad (5.7.8)$$

If we divide by $e^{at} e^{ik_m x}$ and use the relation

$$\cos(k_m \Delta x) = \frac{1}{2} (e^{ik_m \Delta x} + e^{-ik_m \Delta x})$$

then, Eq. (5.7.8) becomes

$$e^{a\Delta t} = 1 + \frac{2\alpha\Delta t}{(\Delta x)^2} [\cos(k_m \Delta x) - 1] \quad (5.7.9)$$

Recalling that

$$\sin^2\left(\frac{k_m \Delta x}{2}\right) = \frac{1 - \cos(k_m \Delta x)}{2}.$$

Eq. (5.7.9) becomes

$$e^{a\Delta t} = 1 - \frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) \quad (5.7.10)$$

From Eq. (5.7.4)

$$\frac{\varepsilon_i^{n+1}}{\varepsilon_i^n} = \frac{e^{a(t+\Delta t)} e^{ik_m x}}{e^{at} e^{ik_m x}} = e^{a\Delta t} \quad (5.7.11)$$

Combining Eqs. (5.7.4), (5.7.10) and (5.7.11), we obtain

$$\left| \frac{\varepsilon_1^{n+1}}{\varepsilon_i^n} \right| = |e^{a\Delta t}| = \left| 1 - \frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) \right| \leq 1 \quad (5.7.12)$$

the requirement to have a stable solution. The factor

$$\left| 1 - \frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) \right| \quad (5.7.13)$$

is called the *amplification factor* and will be denoted by G . In evaluating the inequality in Eq. (5.7.12), two possible cases must be considered.

(1) If

$$\left[1 - \frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) \right] \leq 1$$

then

$$\frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) \geq 0 \quad (5.7.14)$$

Since $4\alpha\Delta t/(\Delta x)^2$ is always positive, the condition expressed by Eq. (5.7.14) is always satisfied.

(2) If

$$\left[1 - \frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) \right] \geq -1$$

then

$$\frac{4\alpha\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k_m \Delta x}{2}\right) - 1 \leq 1 \quad (5.7.15)$$

For the above condition to hold,

$$\frac{\alpha\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad (5.7.16)$$

This equation gives the stability requirement for the solution of the difference equation, Eq. (4.4.3a), by placing a constraint on the size of the time step relative to the size of the mesh spacing.

The von Neumann or Fourier stability method can also be applied to examine the stability characteristics of hyperbolic equations. To illustrate this, consider the linear convection equation, (5.1.1),

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad c > 0 \quad (5.1.1)$$

and replace the time derivative with a forward difference, Eq. (4.3.8), and the spatial derivative with a central difference, Eq. (4.3.7): this gives the simple Euler explicit scheme

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0 \quad (5.7.17)$$

which can be shown to be unstable by applying the von Neumann stability analysis (see Problem 5.7). If in the time derivative, the term u_i^n in Eq. (4.7.17) is replaced by an average value between grid points $(i+1)$ and $(i-1)$, so that Eq. (5.7.17) becomes

$$\frac{u_i^{n+1} - \frac{1}{2}(u_{i+1}^n + u_{i-1}^n)}{\Delta t} + c \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0 \quad (5.7.18)$$

the scheme becomes stable by $|c\Delta t/\Delta x| \leq 1$. This differencing in the time derivative is called the *Lax Method*.

To apply the von Neumann stability analysis to Eq. (5.7.18), again assume an error in the form given by Eq. (5.7.7) and substitute this form into the difference equation, (5.7.18); the amplification factor becomes

$$e^{a\Delta t} = \cos(k_m \Delta x) - i\sigma \sin(k_m \Delta x) \quad (5.7.19a)$$

and in terms of phase angle ϕ , it can be written as

$$G = |G| e^{i\phi} = \sqrt{\cos^2(k_m \Delta x) + \sigma^2 \sin^2(k_m \Delta x)} e^{i\phi} \quad (5.7.19b)$$

where

$$\phi = \tan^{-1}[-\sigma \tan(k_m \Delta x)]$$

The stability requirement $|e^{a\Delta t}| \leq 1$, when applied to Eq. (5.7.19) yields

$$\sigma = c \frac{\Delta t}{\Delta x} \leq 1 \quad (5.7.20)$$

where $\sigma = c\Delta t/\Delta x$ is called the Courant number. Eq. (5.7.20) is known as the *Courant-Friedrichs-Lowy* condition, generally referred to as the CFL condition.

Figure 5.4 shows a polar plot of Eq. (5.7.19b) for several Courant numbers. For $\sigma = 1$ all frequency components are propagated without attenuation in the

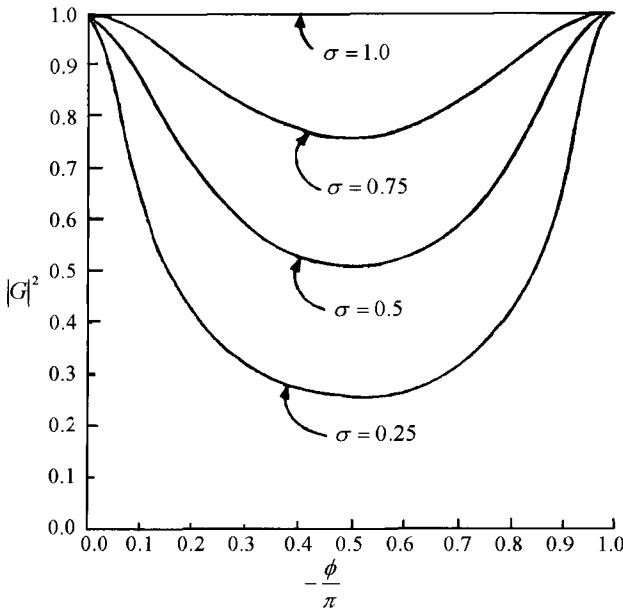


Fig. 5.4. Amplitude-phase plot for the amplification factor of the Lax scheme.

grid. For $\sigma < 1$ the low and high frequency components are slightly affected, while the mid-range frequency signal content is severely attenuated.

A physical interpretation of the relation given by Eq. (5.7.20) for hyperbolic equations is important. To show this significance, consider the second-order wave equation

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad (5.7.21)$$

and its characteristics

$$x + ct = \text{constant} = c_1 \quad (5.7.22a)$$

$$x - ct = \text{constant} = c_2 \quad (5.7.22b)$$

A direct approximation of Eq. (5.7.21) can be obtained by using centered-difference quotients, as in Section 4.4. to replace derivatives. At point (x_i, t_j) , Eq. (5.7.21) can be written as

$$u_{i,j+1} = 2 \left[1 - \left(c \frac{\Delta t}{\Delta x} \right)^2 \right] u_{i,j} + \left(c \frac{\Delta t}{\Delta x} \right)^2 (u_{i+1,j} + u_{i-1,j}) - u_{i,j-1} \quad (5.7.23)$$

A glance at Eq. (5.7.23) indicates that the solution at any fixed net point, (x^*, t^*) depends only on the values of u at the net points in the triangle formed by the initial line and the two lines with slopes $\pm \Delta t / \Delta x$, such as $x \pm (\Delta x / \Delta t)t = \text{constant}$, which pass through (x^*, t^*) . This region, shown in Fig. 5.5, may be

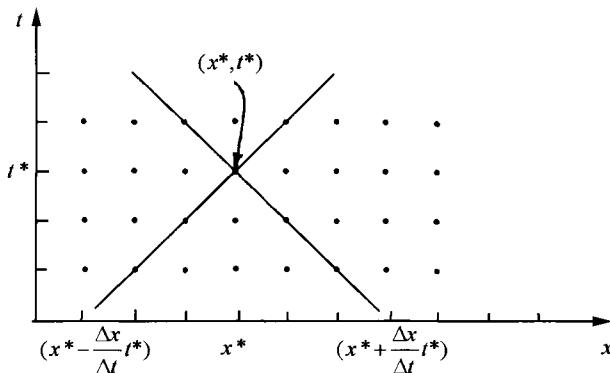


Fig. 5.5. Net points and numerical domain of dependence for difference scheme, Eq. (5.7.23).

called the *numerical domain of dependence* for the difference equations in Eq. (5.7.23).

Clearly, the numerical domain of dependence will be greater than or equal to the domain of dependence of the wave equation for the same point (x^*, t^*) if

$$\frac{\Delta t}{\Delta x} \leq \frac{1}{c}$$

The term $1/c$ is referred to as the *characteristic slope* and $\Delta t/\Delta x$ as the *net slope*. Therefore, if the characteristic slope is greater than or equal to the net slope, then the numerical domain of dependence includes the domain of dependence of the wave equation. The ratio of these slopes is

$$\lambda \equiv \frac{\text{net slope}}{\text{characteristic slope}} \equiv \frac{c\Delta t}{\Delta x} \quad (5.7.24)$$

and thus the above condition becomes $\lambda \leq 1$. Since c is the speed of propagation of a signal or wave for the wave equation, λ is the ratio of the distance such that a signal travels in one time step to the length of a special step of the net. Thus, if such signals cannot move more than the distance Δx in the time Δt , then the numerical domain contains the analytical domain of dependence.

5.8 Numerical Dissipation and Dispersion: Artificial Viscosity

In Chapters 11 and 12 the numerical methods for solving the Euler and Navier–Stokes equations will be discussed. The basis for this discussion will be the numerical methods described in Sections 5.1, 5.2, 5.3, 5.4 and 5.6 for model equations. Since the solutions of these equations are being obtained within an

accuracy determined by the truncation and round-off errors, it is useful to discuss the numerical dissipation and dispersion errors these equations have. The numerical dissipation arises as a result of the even derivative term that appears in the truncation error and the numerical dispersion arises as a result of the odd derivative term that appears in the truncation error. The numerical solutions can be distorted by either (or both) of these errors. With the concept of artificial viscosity, however, the stability of the numerical solutions can be improved, as will be discussed in Chapters 10 and 12.

Consider the one-dimensional wave equation given by Eq. (5.1.1), that is

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad c > 0 \quad (5.1.1)$$

Using a first-order forward difference in time and a first-order backward difference in space, this equation can be written as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \quad (5.8.1)$$

This is a first-order accurate method with truncation error of $O(\Delta t, \Delta x)$.

We use Taylor's series expansion, see Eq. (4.3.1), and write

$$u_i^{n+1} = u_i^n + \left(\frac{\partial u}{\partial t} \right)_i^n \Delta t + \left(\frac{\partial^2 u}{\partial t^2} \right)_i^n \frac{(\Delta t)^2}{2} + \left(\frac{\partial^3 u}{\partial t^3} \right)_i^n \frac{(\Delta t)^3}{6} + \dots \quad (5.8.2a)$$

$$u_{i-1}^n = u_i^n - \left(\frac{\partial u}{\partial t} \right)_i^n \Delta x + \left(\frac{\partial^2 u}{\partial x^2} \right)_i^n \frac{(\Delta x)^2}{2} - \left(\frac{\partial^3 u}{\partial x^3} \right)_i^n \frac{(\Delta x)^3}{6} + \dots \quad (5.8.2b)$$

Substituting the above relations into Eq. (5.8.1) and rearranging, we get

$$\begin{aligned} \left(\frac{\partial u}{\partial t} \right)_i^n + c \left(\frac{\partial u}{\partial x} \right)_i^n &= - \left(\frac{\partial^2 u}{\partial t^2} \right)_i^n \frac{\Delta t}{2} - \left(\frac{\partial^3 u}{\partial t^3} \right)_i^n \frac{(\Delta t)^2}{6} + \left(\frac{\partial^2 u}{\partial x^2} \right)_i^n c \frac{(\Delta x)}{2} \\ &\quad - \left(\frac{\partial^3 u}{\partial x^3} \right)_i^n c \frac{(\Delta x)^2}{6} \end{aligned} \quad (5.8.3)$$

The left-hand side of this equation corresponds to Eq. (5.1.1) and the right-hand side is the truncation error associated with Eq. (5.8.1). The importance of terms in the truncation error can be better appreciated if we first differentiate Eq. (5.8.3) with respect to t ; multiply Eq. (5.8.3) with c after differentiating with respect to x and subtracting the resulting equation from the equation differentiated with respect to t , we obtain

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} &= c^2 \frac{\partial^2 u}{\partial x^2} + \frac{\Delta t}{2} \left[- \frac{\partial^3 u}{\partial t^3} + c \frac{\partial^3 u}{\partial t^2 \partial x} + O(\Delta t) \right] \\ &\quad + \frac{\Delta x}{2} \left[c \frac{\partial^3 u}{\partial x^2 \partial t} - c^2 \frac{\partial^3 u}{\partial x^3} + O(\Delta x) \right] \end{aligned} \quad (5.8.4)$$

Here, for simplicity, we have dropped the subscript i and superscript n , with the understanding that all derivatives are being evaluated at point i and at time n .

Using a similar procedure, the following expressions can be obtained,

$$\frac{\partial^3 u}{\partial t^3} = -c^3 \frac{\partial^3 u}{\partial x^3} + O(\Delta t, \Delta x) \quad (5.8.5a)$$

$$\frac{\partial^3 u}{\partial t^2 \partial x} = c^2 \frac{\partial^3 u}{\partial x^3} + O(\Delta t, \Delta x) \quad (5.8.5b)$$

$$\frac{\partial^3 u}{\partial x^2 \partial t} = -c \frac{\partial^3 u}{\partial x^3} + O(\Delta t, \Delta x) \quad (5.8.5c)$$

Substituting Eqs. (5.8.5) into Eq. (5.8.4), we get

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} &= c^2 \frac{\partial^2 u}{\partial x^2} + \frac{\Delta t}{2} \left[c^3 \frac{\partial^3 u}{\partial x^3} + c^3 \frac{\partial^3 u}{\partial x^3} + O(\Delta t, \Delta x) \right] \\ &\quad + \frac{\Delta x}{2} \left[-c^2 \frac{\partial^3 u}{\partial x^3} - c^2 \frac{\partial^3 u}{\partial x^3} + O(\Delta t, \Delta x) \right] \end{aligned} \quad (5.8.6)$$

Substituting Eqs. (5.8.6) and (5.8.5a) into Eq. (5.8.3), and after rearranging, we obtain

$$\begin{aligned} \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} &= \frac{c \Delta x}{2} (1 - \sigma) \frac{\partial^2 u}{\partial x^2} + \frac{c (\Delta x)^2}{6} (3\sigma - 2\sigma^2 - 1) \frac{\partial^3 u}{\partial x^3} \\ &\quad + O[(\Delta t)^3, (\Delta t)^2 (\Delta x), (\Delta t) (\Delta x)^2, (\Delta x)^3] \end{aligned} \quad (5.8.7)$$

where σ is the Courant number defined by Eq. (5.2.2),

$$\sigma = c \frac{\Delta t}{\Delta x} \quad (5.8.8)$$

The above equation known as the *modified equation*, is the partial-differential equation that is actually solved when a finite-difference method is applied to Eq. (5.1.1). It is important to note that when Eq. (5.8.1) is being used to obtain a numerical solution of Eq. (5.1.1), actually the difference equation is solving Eq. (5.8.7) rather than Eq. (5.1.1).

The modified equation (5.8.7) can provide useful information on the behavior of the numerical solution of the difference equation. To discuss this point further, consider the viscous term in the one-dimensional incompressible Navier–Stokes equation.

$$\frac{\partial}{\partial x} (\sigma_{xx}) = \frac{4}{3} \mu \frac{\partial^2 u}{\partial x^2} \quad (5.8.9)$$

Here the coefficient of $\partial^2 u / \partial x^2$ represents the dissipative aspect of the physical viscosity μ on the flow. The term $\partial^2 u / \partial x^2$ in Eq. (5.8.7) also acts as a dissipative term, much like the viscous terms in the Navier–Stokes equations. Unlike the viscous terms in the Navier–Stokes equations, however, this term is

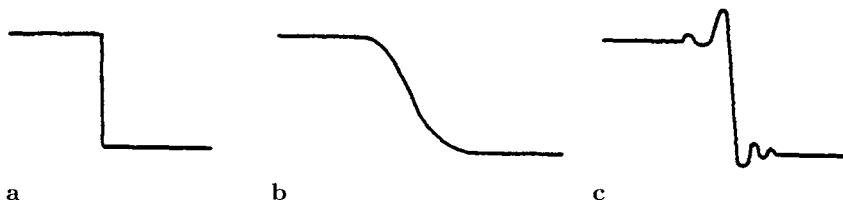


Fig. 5.6. Effects of dissipation and dispersion. (a) Exact solution (b) numerical solution distorted primarily by dissipation errors (typical of first-order methods). (c) Numerical solution distorted primarily by dispersion errors (typical of second-order methods) [1].

of purely *numerical origin* and has no physical significance. The appearance of this term in the numerical solution is called *numerical dissipation*. The coefficient of $\partial^2 u / \partial x^2$, namely, $c\Delta x / 2(1 - \sigma)$ and those like it, which act like the physical viscosity, are called the *artificial viscosity*.

Numerical dispersion is another effect also observed in numerical schemes. It arises as a result of the odd derivative terms that appear in the truncation error. It creates a numerical behavior different from that of numerical dissipation. Dispersion results in a distortion of the different phases of a wave which shows up as “wiggles” in front of and behind the wave. The combined effect of dissipation and dispersion is sometimes referred to as *diffusion*. Diffusion tends to spread-out sharp dividing lines which may appear in the computational domain. Figure 5.6 illustrates the effects of dissipation and dispersion on the computation of a discontinuity taken from [1]. In general, if the lowest-order term in the truncation error contains an even derivative, the resulting solution will predominately exhibit dissipative errors. On the other hand, if the leading term is an odd derivative, the resulting solution will predominately exhibit dispersive errors.

In Chapters 10 and 12 we will discuss the incorporation of additional terms to the Euler and Navier–Stokes equations in order to reduce or eliminate the dissipation and dispersion errors and have stable numerical solutions.

References

- [1] Hirsch, C.: *Numerical Computation of Internal and External Flows*, Vol. 1, John Wiley and Sons, N.Y., 1988.
- [2] Kress, H. O.: On difference approximations of the dissipative type for hyperbolic differential equations, *Comm. Pure and Applied Mathematics* **17**, 335–353, 1964.
- [3] Briley, W. R., McDonald, H.: Solution of the three-dimensional compressible Navier–Stokes equations by an implicit technique, Proc. Fourth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 35, Springer-Verlag, Berlin Heidelberg, 1975.
- [4] Beam, R. M. and Warming R. F.: Alternating direction implicit methods for parabolic equations with a mixed derivative, *SIAM J. of Sci. Stat. Comp.* **1**, 131–159, 1980.

- [5] Jameson, A., Schmidt, W. and Turkel, E.: Numerical simulation of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes, AIAA Paper 81-1259, 1981.
- [6] Steger, J. L. and Warming, R. F.: Flux vector splitting of the inviscid gas dynamic equations with applications to finite-difference methods. *J. Comput. Phys.* **40**, 263–293, 1981.
- [7] Osher, S. and Solomon, F.: Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comput.* **38**(158), 339–374, 1982.
- [8] Roe, P. L.: Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**, 357–372, 1981.

Problems

5-1. Solve Eq. (5.1.1) subject to the following boundary and initial conditions

$$\begin{aligned} x = 0, \quad u &= 0 \\ t = 0, \quad u &= \begin{cases} \sin 2\pi x & 0 \leq x \leq 1 \\ 0 & 1 < x \leq 5 \end{cases} \end{aligned}$$

by the two-step Lax-Wendroff method for $\Delta x = 0.1$ at $t = 4$ for $\Delta t = 0.01, 0.1$ and 0.2 . Take $c = 1$. Compare your solutions with the analytical solution

$$u = \begin{cases} \sin 2\pi(x - t) & t \leq x \leq t + 1 \\ 0 & \text{otherwise} \end{cases}$$

and determine the percentage error. To implement the numerical boundary condition, use the first-order backward difference formula (4.3.5), $u_0 - u_1 = 0$.

5-2. Repeat Problem 5.1 using the Crank-Nicolson method.

5-3. Solve the inviscid Burger's equation, Eq. (4.2.8), subject to the boundary and initial conditions

$$u(0, t) = 0, \quad t \geq 0$$

$$u(x, 0) = x, \quad 0 \leq x \leq 1$$

by the McCormack method. Take $\Delta x = 0.02$ and $\Delta t = 0.01$ and perform the calculations for 200 time steps.

5-4. Repeat Problem 5.3 using the Beam-Warming method (trapezoidal scheme $\theta = \frac{1}{2}$, $\xi = 0$) and the Euler implicit scheme. Experiment with different CFL numbers.

5-5. Show that for one-dimensional flow, the Jacobian matrix A of the flux vector \underline{E} can be written in the form given by Eq. (5.1.5c).

Hint: Eliminate p in Eq. (5.1.3b) with the expression given by Eq. (P2.16.1) for one-dimensional flow.

5-6. For two-dimensional inviscid flows, Eq. (2.2.30) can be written as

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial x} + \frac{\partial \tilde{F}}{\partial y} = 0 \quad (\text{P5.6.1})$$

where \tilde{Q} , \tilde{E} and \tilde{F} are given by Eq. (2.2.32a). It can also be written in the form

$$\frac{\partial \tilde{Q}}{\partial \tilde{t}} + A \frac{\partial \tilde{Q}}{\partial \tilde{x}} + B \frac{\partial \tilde{Q}}{\partial \tilde{y}} = 0 \quad (\text{P5.6.2})$$

with the Jacobian matrices A and B of the flux vectors \tilde{E} and \tilde{F} defined by

$$A = \frac{\partial \tilde{E}}{\partial \tilde{Q}}, \quad B = \frac{\partial \tilde{F}}{\partial \tilde{Q}} \quad (\text{P5.6.3})$$

Show that for a perfect gas, A and B can be written as

$$A = \begin{vmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}u^2 + \frac{\gamma-1}{2}v^2 & (3-\gamma)u & (1-\gamma)v & \gamma-1 \\ -uv & v & u & 0 \\ -\frac{\gamma e u}{\varrho} + (\gamma-1)u(u^2+v^2) & \frac{\gamma e}{\varrho} + \frac{1-\gamma}{2}(3u^2+v^2) & (1-\gamma)uv & \gamma u \end{vmatrix} \quad (\text{P5.6.4})$$

$$B = \begin{vmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{\gamma-3}{2}v^2 + \frac{\gamma-1}{2}u^2 & (1-\gamma)u & (3-\gamma)v & \gamma-1 \\ -\frac{\gamma e v}{\varrho} + (\gamma-1)v(u^2+v^2) & (1-\gamma)uv & \frac{\gamma e}{\varrho} + \frac{1-\gamma}{2}(3v^2+u^2) & \gamma v \end{vmatrix} \quad (\text{P5.6.5})$$

5-7. Apply the von Neumann stability analysis to show that the Euler explicit method, Eq. (5.7.17), is unstable.

5-8. Show that the implicit method

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{\sigma}{2}(u_{i+1}^{n+1} - u_{i-1}^{n+1}) = 0$$

for Eq. (5.1.1) is unconditionally stable.

5-9. Show that the explicit upwind method, Eq. (5.5.1) is stable $0 < \sigma \leq 1$.

5-10. Show that

$$u(x, t) = f(x - ct) \quad (\text{P5.10.1})$$

satisfies the linear convection Eq. (5.1.1) and the initial condition

$$u(x, 0) = f(x) \quad (\text{P5.10.2})$$

5-11. Solve Eq. (5.1.1) subject to the initial and boundary conditions

$$u(x, 0) = \begin{cases} 1 & x \leq 10 \\ 0 & x > 10 \end{cases} \quad (\text{P5.11.1})$$

and

$$u(0, t) = 1 \quad (\text{P5.11.2})$$

by using the Lax–Wendroff method for $\Delta t/\Delta x = 1.0, 0.6$ and 0.3 . Take $c = 1$, choose a 41 grid point mesh with $\Delta x = 1$ and compute up to $t = 18$. Compare your solutions with the exact solution (P5.10.1) graphically.

5-12. Repeat Problem 5.11 with the MacCormack method.

5-13. Repeat Problem 5.11 with the Beam–Warming method.

5-14. Repeat Problem 5.11 with the upwind method Eq. (5.5.3).

5-15. Repeat Problem 5.11 with the upwind method Eq. (5.5.26).

5-16. Show that

$$u(x, t) = \exp(-k^2 \nu t) \sin[k(x - ct)] \quad (\text{P5.16.1})$$

is a solution of the linearized Burger's equation.

$$u_t + cu_x = \nu u_{xx} \quad (\text{P5.16.2})$$

for the initial condition

$$u(x, 0) = \sin(kx) \quad 0 \leq x \leq 2\pi \quad (\text{P5.16.3})$$

and the periodic boundary condition.

$$u(0, t) = u(2\pi, t) \quad (\text{P5.16.4})$$

5-17. Use the Lax–Wendroff method to solve Eq. (P5.16.2–4) with $k = 2$, $c = \frac{1}{2}$ for values of $\nu = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4} . Take $\Delta t/\Delta x = 1.0$ and $\Delta t = 0.02$. Compare your solution graphically with the exact solution.

5-18. Repeat Problem 5.17 with the MacCormack method.

5-19. Repeat Problem 5.17 with the Beam–Warming method.

5-20. Repeat Problem 5.17 with the upwind method Eq. (5.5.3).

5-21. Repeat Problem 5.17 with the upwind method Eq. (5.5.26).

5-22. Find the accuracy of the following approximations to u_x at i ,

(a)

$$(u_x)_{i-1} = \frac{-3u_{i-1} + 4u_i - u_{i+1}}{3\Delta x_i - \Delta x_{i+1}} \quad (\text{P5.22.1})$$

(b)

$$(u_x)_i = \frac{u_{i+1} - u_{i-1}}{\Delta x_{i+1} + \Delta x_i} \quad (\text{P5.22.2})$$

(c)

$$(u_x)_{i+1} = \frac{u_{i-1} - 4u_i + 3u_{i+1}}{3\Delta x_{i+1} - \Delta x_i} \quad (\text{P5.22.3})$$

6

Inviscid Flow Equations for Incompressible Flows

6.1 Introduction

In this chapter we address the solution of the inviscid flow equations for incompressible flows and postpone the discussion on compressible flows to Chapter 10. For incompressible irrotational flows the Euler equations of subsection 2.4.1 simplify further and, in terms of either velocity potential, ϕ , or stream function, ψ , they reduce to the Laplace equation in ϕ or ψ , as discussed in Section 6.2. This equation can be solved by finite-difference methods discussed in Section 6.3. It can also be solved by superposition of flows, called panel methods, rather than finite-difference methods. In Section 6.4, a popular and useful panel method, developed by Hess and Smith, is described for two-dimensional flows. Section 6.5 presents and describes a computer program based on this method. Applications of the panel method to several problems are discussed in Section 6.6.

6.2 Laplace Equation and Its Fundamental Solutions

For an incompressible irrotational flow, the inviscid flow equations of subsection 2.4.1 can be expressed in forms usually more convenient for mathematical treatment. For example, in terms of the velocity potential ϕ , which for a two-dimensional flow is,

$$u = \frac{\partial \phi}{\partial x}, \quad v = \frac{\partial \phi}{\partial y} \quad (6.2.1)$$

the continuity equation becomes Laplace's equation in ϕ ,

$$\nabla^2 \phi = 0 \quad (6.2.2)$$

where the operator ∇^2 is defined by

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (6.2.3)$$

The use of the stream function ψ ,

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \quad (6.2.4)$$

allows the equation resulting from the irrotationality condition,

$$\frac{\partial u}{\partial y} = \frac{\partial v}{\partial x} \quad (6.2.5)$$

to be written as Laplace's equation in ψ ,

$$\nabla^2 \psi = 0 \quad (6.2.6)$$

The assumption of an irrotational flow is a useful one in that it removes the nonlinearity in the momentum equations and allows them to be replaced by the Bernoulli equation (2.4.16), which provides an algebraic relation between velocity and pressure. For a two-dimensional flow, it is given by

$$p + \frac{1}{2} \varrho(u^2 + v^2) = \text{const.} \quad (6.2.7)$$

Equations (6.2.2) and (6.2.6) apply to any incompressible irrotational flow and can be used to compute the *velocity field* about a given body. What distinguishes one flow from another are the *boundary conditions*. For example, to predict the flowfield about a body at rest in an onset flow, V_∞ , moving in the increasing x -direction (an onset flow is the flow that would exist if the body is not present), it is necessary to impose the condition that the surface of the body is a streamline of the flow, that is,

$$\psi = \text{constant} \quad \text{or} \quad \frac{\partial \phi}{\partial n} = 0 \quad (6.2.8)$$

at the surface on which n is the direction of the normal, and that far away from the body. The velocity components are

$$u = \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} = V_\infty \quad (6.2.9a)$$

$$v = \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} = 0 \quad (6.2.9b)$$

since the onset flow is irrotational.

The requirement of an irrotational flow and the conditions imposed by Eq. (6.2.9) are independent of the body shape. The relation given by Eq. (6.2.8), which does not allow any flow through the body surface, brings in the body shape about which a given circulation exists; in other words, the uniqueness of

the irrotational flow about a given body is established if the boundary conditions on the body and at infinity are satisfied and, in addition, if the magnitude of the circulation around the body is specified. Thus, the problem of determining the flowfield of an incompressible inviscid flow about a body is reduced to a purely mathematical one of finding a suitable solution of Laplace's equation in either ϕ or ψ .

The solution of the Laplace equation for the potential function, Eq. (6.2.2), or for the stream function, Eq. (6.2.6), subject to the boundary conditions that the resultant velocity is equal to the freestream value at points far from the surface and that the component of the velocity normal to the surface is zero, can be obtained by the finite-difference methods discussed in Section 4.5. In this case, for a given transformation, the Laplace equation in the physical plane can be written in the form shown in Problem 2.18. With the metrics determined from the transformation, the transformed Laplace equation can then be solved in the computational plane subject to its boundary conditions. In the following section we shall describe this choice to compute the external flow over a circular cylinder.

A more efficient choice for external flows, however, is to use a panel method which takes account of the linearity of the Laplace equation and avoids the need to generate the grid in both physical and computational planes. The panel method is based on the superposition of flows: functions that individually satisfy the Laplace equation may be added together to describe the desired flowfield. A popular approach is to express the flowfield in terms of the velocity potential based on two or more elemental flows in the presence of an onset flow. The two elemental flows (sometimes called singularities) often used in this approach, referred to as the surface-singularity method, include a source, sink and a vortex. The first is the radially symmetric solution of the Laplace equation, and the second is its complement, the solution independent of radius. A source flows radially outward (see Fig. 6.1) such that the continuity equation is satisfied everywhere but at the singularity that exists at the source's center. The potential function for a two-dimensional source centered at the origin is

$$\phi_s = \frac{q}{2\pi} \ln r \quad (6.2.10)$$

where r is the radial coordinate from the center of the source and q is the source strength.

A sink is a negative source, that is, fluid flows *into* a sink along radial streamlines. The potential function for a sink centered at the origin is the same as Eq. (6.2.10) except for the minus sign ($-$) on the right-hand side.

A potential vortex is defined as a singularity about which fluid flows along concentric streamlines (see Fig. 6.2). The potential function for a vortex centered at the origin is

$$\phi_v = -\frac{\Gamma}{2\pi}\theta \quad (6.2.11)$$

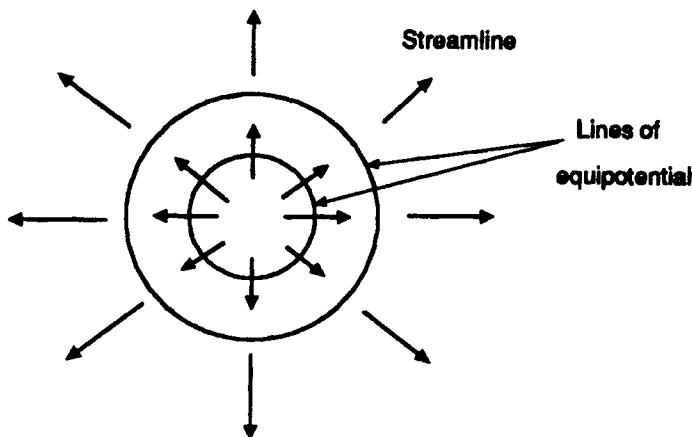


Fig. 6.1. Equipotential lines and streamlines for a source flow.

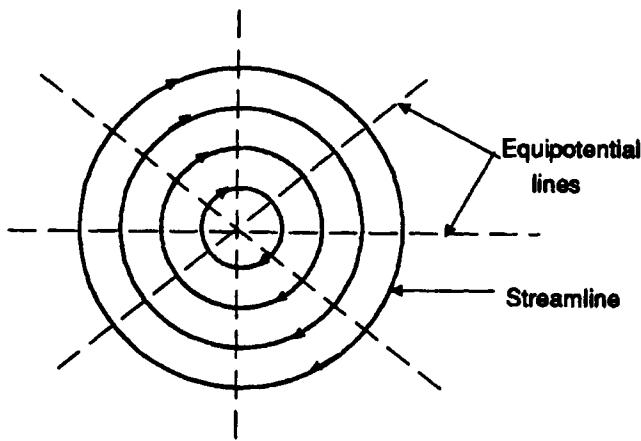


Fig. 6.2. Equipotential and streamlines for a potential vortex flow.

where Γ is the strength of the vortex. The minus sign on the right-hand side of Eq. (6.2.11) represents a vortex with clockwise circulation.

In Section 6.4 we shall use one form of this approach (due to Hess and Smith [1]), which is called a panel method, to solve the Laplace equation to describe the flowfield around an airfoil at an angle of attack α .

6.3 Finite-Difference Method

To illustrate the finite-difference solution of the Laplace equation for an external flow, consider the flow over a circular cylinder and, for convenience, use a polar coordinate system (r, θ) rather than a Cartesian coordinate system (x, y) . In

this case, the solution of the Laplace equation expressed in polar coordinates, Eq. (P2.19.2), which can be written as

$$\frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r} \frac{\partial \phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \theta^2} = 0 \quad (6.3.1)$$

is well known, see for example, Anderson [2]. For a cylinder radius of r_0 and freestream velocity of V_∞ (Fig. 6.3) it is given by

$$\phi = V_\infty r \cos \theta + V_\infty r_0^2 \frac{\cos \theta}{r} \quad (6.3.2)$$

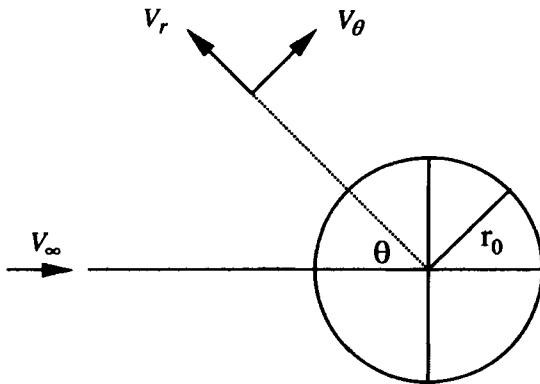


Fig. 6.3. Flow over a circular cylinder.

The total velocity V is composed of radial V_r and circumferential V_θ velocity components related to the velocity potential ϕ by

$$V_r = \frac{\partial \phi}{\partial r}, \quad V_\theta = \frac{1}{r} \frac{\partial \phi}{\partial \theta} \quad (6.3.3)$$

The solution in Eq. (6.3.2) obeys the boundary conditions at the body surface and at infinity, see Eqs. (6.2.8) and (6.2.9), which in our case can be written as

$$r = r_0, \quad \frac{\partial \phi}{\partial r} = 0 \quad (6.3.4a)$$

$$r \rightarrow \infty, \quad \phi \rightarrow V_\infty r \cos \theta \quad (6.3.4b)$$

Before we discuss the numerical solution of the Laplace equation expressed in polar coordinates, Eq. (6.3.1), it is useful to express this equation and its boundary conditions in dimensionless forms. For this purpose, define

$$\tilde{r} = \frac{r}{r_0}, \quad \tilde{\phi} = \frac{\phi}{r_0 V_\infty} \quad (6.3.5)$$

and write Eqs. (6.3.1) and (6.3.4) as

$$\frac{\partial^2 \tilde{\phi}}{\partial \tilde{r}^2} + \frac{1}{\tilde{r}} \frac{\partial \tilde{\phi}}{\partial \tilde{r}} + \frac{1}{\tilde{r}^2} \frac{\partial \tilde{\phi}}{\partial \theta^2} = 0 \quad (6.3.6)$$

$$\tilde{r} = 1, \quad \frac{\partial \tilde{\phi}}{\partial \tilde{r}} = 0 \quad (6.3.7a)$$

$$\tilde{r} \rightarrow \infty, \quad \tilde{\phi} \rightarrow \tilde{r} \cos \theta \quad (6.3.7b)$$

Since the solution of an elliptic equation requires the specification of boundary conditions along the entire *boundary* of the domain, additional boundary conditions are needed other than those given by Eq. (6.3.7). Since the flow is symmetric about a horizontal line through the center of the circle, it is sufficient to compute the flow only in the upper half of the physical plane and set

$$\frac{\partial \phi}{\partial \theta} = 0$$

or

$$\frac{\partial \tilde{\phi}}{\partial \theta} = 0 \quad (6.3.8)$$

along the symmetry line at $\theta = 0$ and $\theta = \pi$ (Fig. 6.4).

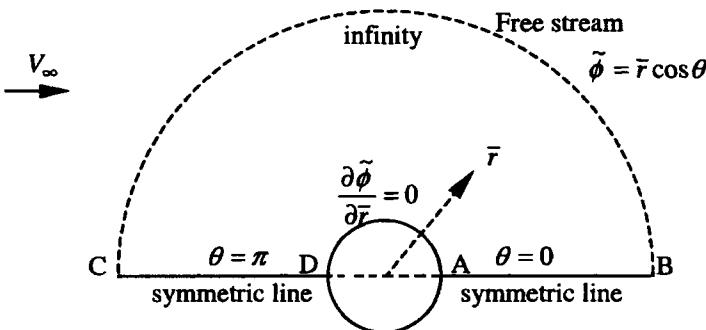


Fig. 6.4. Physical plane for flow over a circular cylinder.

The boundary condition in Eq. (6.3.7b) requires that the velocity be uniform at infinity. While this requirement presents no difficulties for the analytical solution in Eq. (6.3.2), it can cause problems for the numerical solutions since the numerical solution must be carried out in a finite computational plane. To partially resolve this dilemma, we introduce a transformation which transforms infinity into zero; we define a new independent variable η by

$$\eta = \frac{1}{\tilde{r}} \quad (6.3.9a)$$

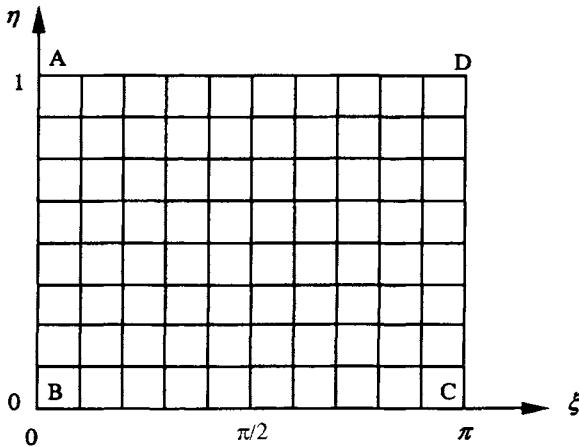


Fig. 6.5. Computational plane for flow over a circular cylinder.

and with

$$\xi = \theta \quad (6.3.9b)$$

perform numerical calculations in the computational plane (ξ, η) (Fig. 6.5). Here the upper boundary line, $\eta = 1$, represents the body surface and the lower boundary line, $\eta = 0$, represents the freestream at infinity. The left and right symmetry lines at $\xi(\equiv \theta) = 0$ and $\xi = \pi$ become boundary conditions represented by AB and DC, respectively.

To express Eq. (6.3.6) and its boundary conditions given by Eqs. (6.3.7) and (6.3.8) in the computational plane, we use the transformation in Eq. (6.3.9) and write the Laplace equation as (see Problem 2.20)

$$\eta^2 \frac{\partial^2 \tilde{\phi}}{\partial \eta^2} + \eta \frac{\partial \tilde{\phi}}{\partial \eta} + \frac{\partial^2 \tilde{\phi}}{\partial \xi^2} = 0 \quad (6.3.10)$$

and its boundary conditions as

$$\frac{\partial \tilde{\phi}}{\partial \eta} = 0 \quad \text{along AD} \quad (6.3.11a)$$

$$\tilde{\phi} = \frac{\cos \xi}{\eta} \quad \text{along BC} \quad (6.3.11b)$$

$$\frac{\partial \tilde{\phi}}{\partial \xi} = 0 \quad \text{along AB and DC} \quad (6.3.11c)$$

We now use the procedure in Section 4.5 to write the finite-difference approximations to Eqs. (6.3.10) and (6.3.11). For convenience, we drop the tilde on ϕ . Replacing each second derivative in Eq. (6.3.10) by Eq. (4.3.10) and the first derivative by Eq. (4.3.7), all at (i, j) , we obtain

$$\eta_i^2 \left[\frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta\eta)^2} \right] + \eta_i \left[\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta\eta} \right] + \left[\frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{(\Delta\xi)^2} \right] = 0 \quad (6.3.12a)$$

or with $\beta = \Delta\xi/\Delta\eta$

$$\alpha_1\phi_{i,j} + \alpha_2\phi_{i+1,j} + \alpha_3\phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} = 0 \quad (6.3.12b)$$

where

$$\alpha_1 = -2(\eta_i^2\beta^2 + 1), \quad \alpha_2 = \eta_i^2\beta^2 + \frac{1}{2}\eta_i(\Delta\xi)\beta, \quad \alpha_3 = \eta_i^2\beta^2 - \frac{1}{2}\eta_i(\Delta\xi)\beta \quad (6.3.13)$$

The boundary conditions are

$$i = 0, \quad \phi_{0,j} = \cos \xi_j / \eta_0 \quad (6.3.14a)$$

$$i = I + 1, \quad \frac{\partial \phi}{\partial \eta} = 0 \quad (6.3.14b)$$

$$j = 0, \quad \frac{\partial \phi}{\partial \xi} = 0 \quad (6.3.14c)$$

$$j = J + 1, \quad \frac{\partial \phi}{\partial \xi} = 0 \quad (6.3.14d)$$

Using Eq. (6.3.14a), Eq. (6.3.12b) at $i = 1$ can be written as

$$\alpha_1\phi_{1,j} + \alpha_2\phi_{2,j} + \phi_{1,j+1} + \phi_{1,j-1} = -\alpha_3\phi_{0,j} = -\alpha_3 \cos \xi_j / \eta_0 \quad (6.3.15a)$$

Using Eq. (4.5.14) in Eq. (6.3.14b), Eq. (6.3.12b) at $i = I$ can be written as

$$\left(\alpha_1 + \frac{4}{3}\alpha_2 \right) \phi_{I,j} + \left(\alpha_3 - \frac{1}{3}\alpha_2 \right) \phi_{I-1,j} + \phi_{I,j+1} + \phi_{I,j-1} = 0 \quad (6.3.15b)$$

Using Eq. (4.5.14) in Eq. (6.3.14c), Eq. (6.3.12b) at $j = 1$ can be written as

$$\left(\alpha_1 + \frac{4}{3} \right) \phi_{i,1} + \alpha_2\phi_{i+1,1} + \alpha_3\phi_{i-1,1} + \frac{2}{3}\phi_{i,2} = 0 \quad (6.3.15c)$$

Using Eq. (4.5.14) in Eq. (6.3.14d), Eq. (6.3.12b) at $j = J$ can be written as

$$\left(\alpha_1 + \frac{4}{3} \right) \phi_{i,J} + \alpha_2\phi_{i+1,J} + \alpha_3\phi_{i-1,J} + \frac{2}{3}\phi_{i,J-1} = 0 \quad (6.3.15d)$$

Using Eq. (4.5.6), with \mathbf{U} now equal to ϕ , $U_j = \underline{\phi}_j$

$$\boldsymbol{\phi} = \begin{vmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_j \\ \vdots \\ \phi_J \end{vmatrix}, \quad \underline{\phi}_j = \begin{vmatrix} \phi_{1,j} \\ \phi_{2,j} \\ \vdots \\ \phi_{i,j} \\ \vdots \\ \phi_{I,j} \end{vmatrix} \quad (6.3.16a)$$

and with $f_{i,j} = 0$

$$\mathbf{F} = \begin{vmatrix} \underline{F}_1 \\ \underline{F}_2 \\ \vdots \\ \underline{F}_j \\ \vdots \\ \underline{F}_J \end{vmatrix}, \quad \underline{F}_j = \begin{vmatrix} -\alpha_3 \phi_{0,j} \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{vmatrix} \quad (6.3.16b)$$

the A_j , B_j and C_j matrices of the coefficient matrix A in Eq. (4.5.7) are given by

$$A_1 = \begin{vmatrix} a_1^* & \alpha_2 \\ \alpha_3 & a_1^* & \alpha_2 \\ \alpha_3 & a_1^* & \alpha_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha_3 & a_1^* & \alpha_2 \\ a_2^* & a_3^* \end{vmatrix} \quad (6.3.17a)$$

$$A_j = \begin{vmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_1 & \alpha_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha_3 & \alpha_1 & \alpha_2 \\ a_2^* & a_4^* \end{vmatrix} \quad (6.3.17b)$$

$$A_J = \begin{vmatrix} a_1^* & \alpha_2 \\ \alpha_3 & a_1^* & \alpha_2 \\ \alpha_3 & a_1^* & \alpha_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha_3 & a_1^* & \alpha_2 \\ a_2^* & a_3^* \end{vmatrix} \quad (6.3.17c)$$

$$B_j = C_j = I_I \quad 2 \leq j \leq J-1 \quad (6.3.17d)$$

$$C_1 = B_J = \frac{2}{3}I_I \quad (6.3.17e)$$

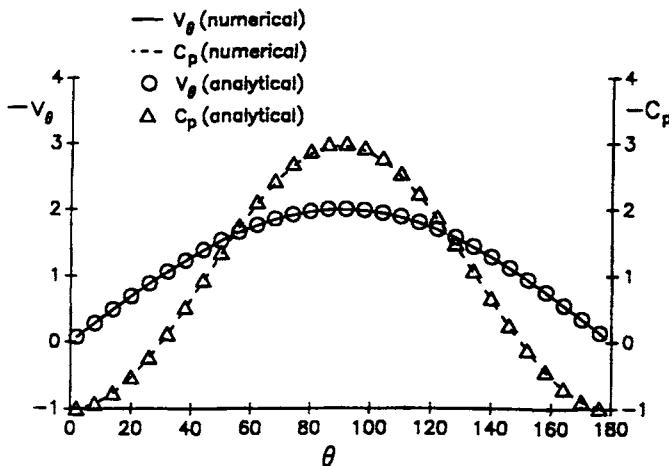
where

$$a_1^* = \alpha_1 + \frac{4}{3}, \quad a_2^* = \alpha_3 - \frac{1}{3}\alpha_2, \quad a_3^* = a_1^* + \frac{4}{3}\alpha_2, \quad a_4^* = \alpha_1 + \frac{4}{3}\alpha_2 \quad (6.3.18)$$

Table 6.1 and Fig. 6.6 show the results obtained for a grid composed of 91 points in the ξ -direction and 46 points in the η -direction with $\eta_0 = 0.1$. Several

Table 6.1. Numerical and analytical dimensionless velocity potential results

η	$\xi = 2^\circ$		$\xi = 90^\circ$		$\xi = 178^\circ$	
	$\tilde{\phi}_{\text{num}}$	$\tilde{\phi}_{\text{an}}$	$\tilde{\phi}_{\text{num}}$	$\tilde{\phi}_{\text{an}}$	$\tilde{\phi}_{\text{num}}$	$\tilde{\phi}_{\text{an}}$
0.12	8.38	8.45	0.00	0.00	-8.38	-8.45
0.2	5.16	5.20	0.00	0.00	-5.16	-5.20
0.3	3.61	3.63	0.00	0.00	-3.61	-3.63
0.4	2.88	2.90	0.00	0.00	-2.88	-2.90
0.5	2.48	2.50	0.00	0.00	-2.48	-2.50
0.6	2.25	2.26	0.00	0.00	-2.25	-2.26
0.7	2.12	2.13	0.00	0.00	-2.12	-2.13
0.8	2.04	2.05	0.00	0.00	-2.04	-2.05
0.9	2.00	2.01	0.00	0.00	-2.00	-2.01
0.98	1.99	2.00	0.00	0.00	-1.99	-2.00

**Fig. 6.6.** Comparison of numerical and analytical results for the circular cylinder.

calculations for different values of η_0 (i.e., 0.01, 0.05, 0.1) showed that in comparison with the analytical results given by Eq. (6.3.2), best results are obtained for $\eta_0 = 0.1$, which clearly showed the importance of η_0 in the calculations.

Table 6.1 shows a comparison between the computed and analytical values of the dimensionless velocity potential $\tilde{\phi}$ as a function of η at $\xi (\equiv \theta)$ locations corresponding to 2° , 90° and 178° . The analytical values of $\tilde{\phi}$ follow from Eq. (6.3.2), which, in terms of dimensionless variables, can be written as

$$\tilde{\phi} = \left(\eta + \frac{1}{\eta} \right) \cos \xi \quad (6.3.19)$$

Figure 6.6 shows the variation of the dimensionless circumferential velocity \bar{V}_θ ($\equiv V_\theta/V_\infty$) and pressure coefficient C_p along the surface of the circular

cylinder. The solid and dashed lines correspond to the numerical results and the symbols denote the analytical results. For example, from the definition of V_θ in Eq. (6.3.3) and from the exact solution given by Eq. (6.3.2), it follows that

$$\bar{V}_\theta = -2 \sin \theta \quad (6.3.20a)$$

and the pressure coefficient C_p is

$$C_p = 1 - \bar{V}_\theta^2 = 1 - 4 \sin^2 \theta \quad (6.3.20b)$$

Since the numerical solutions are obtained in terms of the dimensionless velocity potential, the circumferential velocity component \bar{V}_θ is computed by a second-order central difference,

$$(\bar{V}_\theta)_{I+1,j} = \frac{\tilde{\phi}_{I+1,j+1} - \tilde{\phi}_{I+1,j-1}}{2(\Delta\xi)} \quad (6.3.21)$$

Overall, the numerical results are in good agreement with the analytical results. An error of less than one percent is largely due to the accuracy of the numerical scheme.

6.4 Hess-Smith Panel Method

We consider an airfoil at rest in an onset flow of velocity V_∞ . We assume that the airfoil is at an angle of attack, α (the angle between its chord line and the onset velocity), and that the upper and lower surfaces are given by functions $Y_u(x)$ and $Y_l(x)$, respectively. These functions can be defined analytically, or (as often is the case) by a set of (x, y) values of the airfoil coordinates. We denote the distance of any field point (x, y) measured from an arbitrary point, b , on the airfoil surface by r , as shown in Fig. 6.7. Let \vec{n} also denote the unit vector normal to the airfoil surface and directed from the body into the fluid and \vec{t} , a unit vector tangential to the surface, and assume that its inclination to the x -axis is given by θ . It follows from Fig. 6.7 that with \vec{i} and \vec{j} denoting unit vectors in the x - and y -directions, respectively,

$$\begin{aligned} \vec{n} &= -\sin \theta \vec{i} + \cos \theta \vec{j} \\ \vec{t} &= \cos \theta \vec{i} + \sin \theta \vec{j} \end{aligned} \quad (6.4.1)$$

If the airfoil contour is divided into a large number of small segments, ds , then we can write

$$\begin{aligned} dx &= \cos \theta ds \\ dy &= \sin \theta ds \end{aligned} \quad (6.4.2)$$

We next assume that the airfoil geometry is represented by a finite number (N) of short straight-line elements called panels, defined by $(N + 1)(x_j, y_j)$ pairs

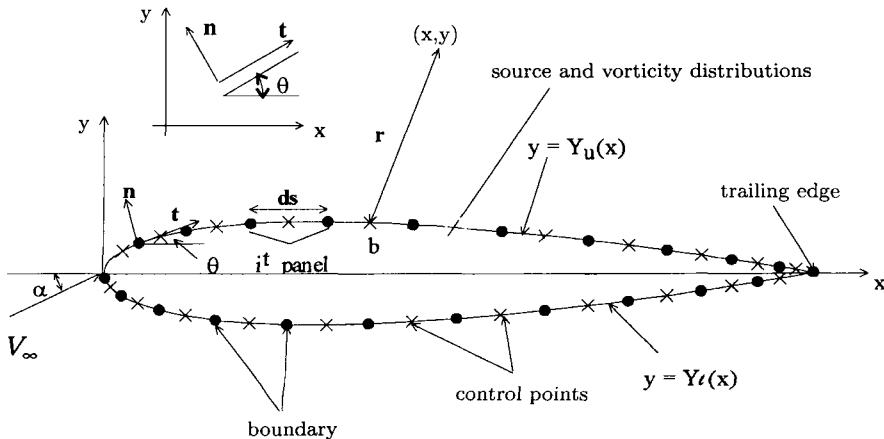


Fig. 6.7. Panel representation of airfoil surface and notation for an airfoil at incidence α .

called boundary points. It is customary to input the (x, y) coordinates starting at the lower surface trailing edge, proceeding clockwise around the airfoil, and ending back at the upper surface trailing edge. If we denote the boundary points by

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N), (x_{N+1}, y_{N+1}) \quad (6.4.3)$$

then the pairs (x_1, y_1) and (x_{N+1}, y_{N+1}) are identical and represent the trailing edge. It is customary to refer to the element between (x_j, y_j) and (x_{j+1}, y_{j+1}) as the j -th panel, and to the midpoints of the panels as the *control points*. Note from Fig. 6.7 that as one traverses from the i -th boundary point to the $(i + 1)$ -th boundary point, the airfoil body is on the right-hand side. This numbering sequence is consistent with the common definition of the unit normal vector \vec{n}_i and unit tangential vector \vec{t}_i for all panel surfaces, i.e., \vec{n}_i is directed from the body into the fluid and \vec{t}_i from the i -th boundary point to the $(i + 1)$ -th boundary point with its inclination to the x -axis given by θ_i .

In the Hess-Smith panel method, the velocity \vec{V} at any point (x, y) is represented by

$$\vec{V} = \vec{U} + \vec{v} \quad (6.4.4)$$

where \vec{U} is the velocity of the uniform flow at infinity

$$\vec{U} = V_\infty (\cos \alpha \vec{i} + \sin \alpha \vec{j}) \quad (6.4.5)$$

and \vec{v} is the disturbance field due to the body which is represented by two elementary flows corresponding to source and vortex flows. A source or vortex on the j -th panel causes an induced source velocity \vec{v}_s at (x, y) or an induced vortex velocity \vec{v}_v at (x, y) , respectively, which are obtained by taking gradients of corresponding potentials given by Eq. (6.2.10) and (6.2.11) so that, with integrals applied to the airfoil surface,

$$\vec{v}(x, y) = \int \vec{v}_s q_j(s) ds_j + \int \vec{v}_v \tau_j(s) ds_j \quad (6.4.6)$$

Here $q_j ds_j$ is the source strength for the element ds_j on the j -th panel. Similarly, $\tau_j ds_j$ is the vorticity strength for the element ds_j on the same panel.

Each of the N panels are represented by similar sources and vortices distributed on the airfoil surface. The induced velocities in Eq. (6.4.6) satisfy the irrotationality condition and the boundary condition at infinity, Eq. (6.2.9). For uniqueness of the solutions, it is also necessary to specify the magnitude of the circulation around the body. To satisfy the boundary conditions on the body, which correspond to the requirement that the surface of the body is a streamline of the flow given by Eq. (6.2.8). It is customary to choose the *control points* to numerically satisfy the requirement that the resultant flow is tangent to the surface. If the tangential and normal components of the total velocity at the control point of the i -th panel are denoted by $(V^t)_i$ and $(V^n)_i$, respectively, the flow tangency conditions are then satisfied at panel control points by requiring that the resultant velocity at each control point has only $(V^t)_i$, and

$$(V^n)_i = 0 \quad i = 1, 2, \dots, N \quad (6.4.7)$$

Thus, to solve the Laplace equation with this approach, at the i -th panel control point we compute the normal $(V^n)_i$ and tangential $(V^t)_i$, ($i = 1, 2, \dots, N$) velocity components induced by the source and vorticity distributions on all panels, j ($j = 1, 2, \dots, N$), including the i -th panel itself, and separately sum all the induced velocities for the normal and tangential components together with the freestream velocity components. The resulting expressions, which satisfy the irrotationality condition, must also satisfy the boundary conditions discussed above. Before discussing this aspect of the problem, it is convenient to write Eq. (6.4.6) expressed in terms of its velocity components $(V^n)_i$ and $(V^t)_i$ by

$$(V^n)_i = \sum_{j=1}^N A_{ij}^n q_j + \sum_{j=1}^N B_{ij}^n \tau_j + V_\infty \sin(\alpha - \theta_i) \quad (6.4.8a)$$

$$(V^t)_i = \sum_{j=1}^N A_{ij}^t q_j + \sum_{j=1}^N B_{ij}^t \tau_j + V_\infty \cos(\alpha - \theta_i) \quad (6.4.8b)$$

where A_{ij}^n , B_{ij}^n , A_{ij}^t , B_{ij}^t are known as influence coefficients defined as the velocities induced at a control point (x_{m_i}, y_{m_i}) ; more specifically, A_{ij}^n and A_{ij}^t denote the normal and tangential velocity components, respectively, induced at the i -th panel control point by a unit strength source distribution on the j -th panel, and B_{ij}^n and B_{ij}^t are those induced by unit strength vorticity distribution on the j -th panel. The influence coefficients are related to the airfoil geometry and the panel arrangement; they are given by the following expressions:

$$A_{ij}^n = \begin{cases} \frac{1}{2\pi} \left[\sin(\theta_i - \theta_j) \ln \frac{r_{i,j+1}}{r_{i,j}} + \cos(\theta_i - \theta_j) \beta_{ij} \right] & i \neq j \\ \frac{1}{2} & i = j \end{cases} \quad (6.4.9)$$

$$A_{ij}^t = \begin{cases} \frac{1}{2\pi} \left[\sin(\theta_i - \theta_j) \beta_{ij} - \cos(\theta_i - \theta_j) \ln \frac{r_{i,j+1}}{r_{i,j}} \right] & i \neq j \\ 0 & i = j \end{cases} \quad (6.4.10)$$

$$B_{ij}^n = -A_{ij}^t \quad B_{ij}^t = A_{ij}^n \quad (6.4.11)$$

Here

$$\begin{aligned} r_{i,j+1} &= \left[(x_{m_i} - x_{j+1})^2 + (y_{m_i} - y_{j+1})^2 \right]^{1/2} \\ r_{i,j} &= \left[(x_{m_i} - x_j)^2 + (y_{m_i} - y_j)^2 \right]^{1/2} \\ x_{m_i} &= \frac{1}{2}(x_i + x_{i+1}), \quad y_{m_i} = \frac{1}{2}(y_i + y_{i+1}) \\ \theta_i &= \tan^{-1} \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right), \quad \theta_j = \tan^{-1} \left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j} \right) \\ \beta_{ij} &= \tan^{-1} \left(\frac{y_{m_i} - y_{j+1}}{x_{m_i} - x_{j+1}} \right) - \tan^{-1} \left(\frac{y_{m_i} - y_j}{x_{m_i} - x_j} \right) \end{aligned} \quad (6.4.12)$$

Regardless of the nature of $q_j(s)$ and $\tau_j(s)$, Eq. (6.4.8) satisfies the irrotationality condition and the infinity boundary condition, Eq. (6.2.9). To satisfy the requirements given by Eq. (6.4.7) and the condition related to the circulation, it is necessary to adjust these functions. In the approach adopted by Hess and Smith [1], the source strength $q_j(s)$ is assumed to be constant over the j -th panel and is adjusted to give zero normal velocity over the airfoil, and the vorticity strength τ_j is taken to be constant on all panels ($\tau_j = \tau$) and its single value is adjusted to satisfy the condition associated with the specification of circulation. Since the specification of the circulation renders the solution to be unique, a rational way to determine the solution is required.

The best approach is to adjust the circulation to give the correct force on the body as determined by experiment. However, this requires advance knowledge of that force, and one of the principal aims of a flow calculation method is to calculate the force and not to take it as given. Thus, another criterion for determining circulation is needed.

For smooth bodies such as ellipses, the problem of rationally determining the circulation has yet to be solved. Such bodies have circulation associated with them and resulting lift forces, but there is no rule for calculating these forces. If, on the other hand, we deal with an airfoil having a sharp trailing edge, we can apply the Kutta condition [2, 4]. It turns out that for every value of circulation except one, the inviscid velocity is infinite at the trailing edge. The Kutta condition states that the particular value of circulation that gives a finite

velocity at the trailing edge is the proper one to choose. This condition does not include bodies with nonsharp trailing edges and bodies on which the viscous effects have been simulated by, for example, surface blowing, as discussed in detail in [4]. Thus, the classical Kutta condition is of strictly limited validity. It is customary to apply a “Kutta condition” to bodies outside its narrow definition, but this is an approximation; nevertheless the calculations are often in close accord with experiment.

In the panel method, the Kutta condition is indirectly applied by deducing another property of the flow at the trailing edge that is a direct consequence of the finiteness of velocity; this property is used as “the Kutta condition.” Properties that have been used in lieu of “the Kutta condition” in panel methods include the following:

- (a) A streamline of the flow leaves the trailing edge along the bisector of the trailing-edge angle.
- (b) Upper and lower surface total velocities approach a common limit at the trailing edge. The limiting value is zero if the trailing-edge angle is nonzero.
- (c) Source and/or vorticity strengths at the trailing edge must satisfy conditions to allow finite velocity.

Of the above, property (b) is more widely used. At first it may be thought that this property requires setting both the upper and lower surface velocities equal to zero. This gives two conditions, which cannot be satisfied by adjusting a single parameter. The most reasonable choice is to make these two total velocities in the downstream direction at the 1st and N -th panel control points equal so that the flow leaves the trailing edge smoothly. Since the normal velocity on the surface is zero according to Eq. (6.4.7), the magnitude of the two tangential velocities at the trailing edge must be equal to each other, that is,

$$(V^t)_N = -(V^t)_1 \quad (6.4.13)$$

Introducing the flow tangency condition, Eq. (6.4.7), into Eq. (6.4.8a) and noting that $\tau_j = \tau$, we get

$$\sum_{j=1}^N A_{ij}^n q_j + \tau \sum_{j=1}^N B_{ij}^n + V_\infty \sin(\alpha - \theta_i) = 0, \quad i = 1, 2, \dots, N \quad (6.4.14)$$

In terms of the unknowns, q_j ($j = 1, 2, \dots, N$) and τ , the Kutta condition of Eq. (6.4.13) and Eq. (6.4.14) form a system of algebraic equations whose solution can be obtained by the Gaussian elimination method discussed in Section 4.5. The details of the solution procedure and the computer program are given in the following section.

6.5 A Panel Program for Airfoils

In this section a computer program is described for calculating the inviscid flow-field over an airfoil with the Hess-Smith panel method that was discussed in the previous section (see also Appendix B). Before reviewing the four subroutines and MAIN of this program, it is useful to examine the solution of Eqs. (6.4.13) and (6.4.14), which can be written in the form given by Eq. (4.5.23), that is,

$$A\vec{x} = \underline{b} \quad (4.5.23)$$

Here A is a square matrix of order $(N + 1)$

$$A \equiv \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1N} & a_{1,N+1} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2N} & a_{2,N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{iN} & a_{i,N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{Nj} & \dots & a_{NN} & a_{N,N+1} \\ a_{N+1,1} & a_{N+1,2} & \dots & a_{N+1,j} & \dots & a_{N+1,N} & a_{N+1,N+1} \end{vmatrix} \quad (6.5.1)$$

and $\vec{x} = (q_1, \dots, q_i, \dots, q_N, \tau)^T$ and $\underline{b} = (b_1, \dots, b_i, \dots, b_N, b_{N+1})^T$ with T denoting the transpose. The elements of the coefficient matrix A follow from Eq. (6.4.14)

$$a_{ij} = A_{ij}^n, \quad \begin{matrix} i = 1, 2, \dots, N \\ j = 1, 2, \dots, N \end{matrix} \quad (6.5.2a)$$

$$a_{i,N+1} = \sum_{j=1}^N B_{ij}^n, \quad i = 1, 2, \dots, N \quad (6.5.2b)$$

A_{ij}^n are given by Eq. (6.4.9) and B_{ij}^n by Eq. (6.4.11). The relation in Eq. (6.5.1) follows from the definition of \vec{x} where τ is essentially x_{N+1} .

To find $a_{N+1,j}$ ($j = 1, \dots, N$) and $a_{N+1,N+1}$ in the coefficient matrix A , we use the Kutta condition and apply Eq. (6.4.13) to Eq. (6.4.8b) and, with τ as a constant, we write the resulting expression as

$$\begin{aligned} & \sum_{j=1}^N A_{1j}^t q_j + \tau \sum_{j=1}^N B_{1j}^t + V_\infty \cos(\alpha - \theta_1) \\ &= - \left[\sum_{j=1}^N A_{Nj}^t q_j + \tau \sum_{j=1}^N B_{Nj}^t + V_\infty \cos(\alpha - \theta_N) \right] \end{aligned}$$

or as

$$\begin{aligned} & \sum_{j=1}^N (A_{1j}^t + A_{Nj}^t) q_j + \tau \sum_{j=1}^N (B_{1j}^t + B_{Nj}^t) \\ &= -V_\infty \cos(\alpha - \theta_1) - V_\infty \cos(\alpha - \theta_N) \end{aligned} \quad (6.5.3)$$

so that,

$$a_{N+1,j} = A_{1j}^t + A_{Nj}^t, \quad j = 1, 2, \dots, N \quad (6.5.4a)$$

$$a_{N+1,N+1} = \sum_{j=1}^N (B_{1j}^t + B_{Nj}^t) \quad (6.5.4b)$$

where now A_{1j}^t and A_{Nj}^t are computed from Eq. (6.4.10) and B_{1j}^t and B_{Nj}^t from Eq. (6.4.11).

The components of \vec{b} again follow from Eqs. (6.4.14) and (6.5.2). From Eq. (6.4.14),

$$b_i = -V_\infty \sin(\alpha - \theta_i), \quad i = 1, \dots, N \quad (6.5.5a)$$

and from Eq. (6.5.3),

$$b_{N+1} = -V_\infty \cos(\alpha - \theta_1) - V_\infty \cos(\alpha - \theta_N) \quad (6.5.5b)$$

With all the elements of a_{ij} determined from Eqs. (6.5.2) and (6.5.4) and the elements of \vec{b} from Eq. (6.5.5), the solution of Eq. (4.5.23) can be obtained with subroutine GAUSS in Table 4.2. The elements of \vec{x} are given by

$$x_i = \frac{1}{a_{ii}^{(i-1)}} - \left[b_i^{(i-1)} - \sum_{j=i+1}^{N+1} a_{ij}^{(i-1)} x_j \right] \quad i = N+1, \dots, 1 \quad (6.5.6)$$

where

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)}, \quad \begin{array}{l} k = 1, \dots, N \\ j = k+1, \dots, N+1 \\ i = k+1, \dots, N+1 \end{array} \quad (6.5.7a)$$

$$a_{ij}^{(0)} = a_{ij}$$

$$b_i^{(k)} = b_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)}, \quad \begin{array}{l} k = 1, \dots, N \\ i = k+1, \dots, N+1 \end{array} \quad (6.5.7b)$$

$$b_i^{(0)} = b_i$$

6.5.1 MAIN Program

MAIN contains the input information which comprises (1) the number of panels, N , along the surface of the airfoil, NODTOT, (2) airfoil coordinates normalized with respect to its chord c , x/c , y/c , [X(I) and Y(I)], and (3) angle of attack α (ALPHA) in degrees. The panel slopes are calculated from Eq. (6.4.2). The subroutine COEF is called to compute A and b in Eq. (4.5.23), subroutine GAUSS to compute \vec{x} , subroutine VPDIS to compute the velocity and pressure distributions, and subroutine CLCM to compute the airfoil characteristics corresponding to lift (CL) and pitching moment (CM) coefficients. (The drag coefficient CD for irrotational, incompressible flows is always zero, as stated by D'Alembert Paradox).

6.5.2 Subroutine COEF

This subroutine calculates the elements a_{ij} of the coefficient matrix A from Eqs. (6.5.1) and (6.5.4) and the elements of \underline{b} from Eq. (6.5.5). We note that $N + 1$ corresponds to KUTTA, N to NODTOT and

$$\begin{aligned}\cos(\theta_i - \theta_j) &= \cos \theta_i \cos \theta_j + \sin \theta_i \sin \theta_j = \text{CTIMTJ} \\ \sin(\theta_i - \theta_j) &= \sin \theta_i \cos \theta_j - \cos \theta_i \sin \theta_j = \text{STIMTJ}\end{aligned}$$

6.5.3 Subroutine GAUSS

The solution of Eq. (4.5.23) is obtained with the Gauss elimination method described in Section 4.5.

6.5.4 Subroutine VPDIS

Once \underline{x} is determined by subroutine GAUSS so that source strengths q_i ($i = 1, 2, \dots, N$) and vorticity τ on the airfoil surface are known, the tangential velocity component (V^t) at each control point can be calculated. Denoting q_i with Q(I) and τ with GAMMA, the tangential velocities $(V^t)_i$ are obtained with the help of Eq. (6.4.8b). This subroutine also determines the distributions of the dimensionless pressure coefficient C_p ($\equiv \text{CP}$) defined by

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \rho V_\infty^2} \quad (6.5.8a)$$

which in terms of velocities can be written as

$$C_p = 1 - \left(\frac{V^t}{V_\infty} \right)^2 \quad (6.5.8b)$$

The output of this subroutine includes the distribution of source strength q_i ($\equiv \text{Q}$), C_p ($\equiv \text{CP}$), V^t/V_∞ ($\equiv \text{UE}$) and vorticity strength τ (GAMMA).

6.5.5 Subroutine CLCM

The dimensionless pressure in the appropriate directions is integrated to compute the aerodynamic force and the coefficients for lift (CL) and pitching moment (CM) about the leading edge of the airfoil.

6.6 Applications of the Panel Method

To demonstrate the use of the computer program of Section 6.5, subsection 6.6.1 presents the calculation of the lift and pitching moment coefficients of an airfoil for a range of angles of attack. To demonstrate the application of the panel method to geometries other than an airfoil, in subsection 6.6.2, the calculation of the surface pressure and external velocity distributions on a circular cylinder discussed in Section 6.2 with a finite-difference method is presented and the modifications required to the panel program are described.

The extension of the panel program to multielement airfoils is described in subsection 6.6.3. The modifications for this case are discussed in detail and the resulting program is applied to two-element and three-element airfoils to compute their pressure distributions and lift and pitching moment coefficients.

6.6.1 Flowfield and Section Characteristics of a NACA 0012 Airfoil

Consider a NACA 0012 airfoil that is symmetrical with a maximum thickness of $0.12c$ [3]; the pressure distribution and external velocity distribution on its upper and lower surface is computed and its section characteristics determined with the panel program. Sometimes the airfoil coordinates are given analytically but often in tabular form similar to those given in Appendix B, Chapter 6.

The calculations for this airfoil are performed for angles of attack from 0° to 20° with $\delta\alpha$ increments of 4° . The x/c and y/c values are read in starting on the lower surface trailing edge (TE), traversing clockwise around the nose to the upper surface TE.

In identifying the upper and lower surfaces of the airfoil, it is necessary to determine the x/c -location where $V^t/V_\infty = 0$. This location, called the stagnation point, is easy to determine since the V^t/V_∞ values are positive for the upper surface and negative for the lower surface. In general it is sufficient to take the stagnation point to be the x/c -location where the change of sign in V^t/V_∞ occurs. For higher accuracy, if desired, the stagnation point can be determined by interpolation between the negative and positive values of V^t/V_∞ as a function of the surface distance along the airfoil.

Figures 6.8 and 6.9 show the computed pressure coefficients, C_p , on the lower and upper surfaces and dimensionless external velocity distributions, V/V_∞ , on the upper surface of the airfoil at three angles of attack starting from 0° . As expected, the results show that the pressure and external velocity distributions on both surfaces are identical to each other at $\alpha = 0^\circ$. With increasing incidence angle, the pressure peak moves upstream on the upper surface and downstream on the lower surface. In the former case, with the pressure peak increasing in magnitude with α , the extent of the flow deceleration increases on the upper surface and, as shall be shown later in Chapter 7, it increases the regions of flow separation on the airfoil and in the wake. On the lower surface, on the other

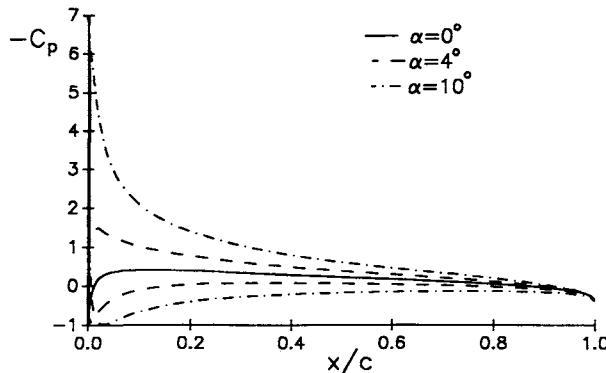


Fig. 6.8. Distribution of pressure coefficients on the NACA 0012 airfoil at three angles of attack.

hand, the region of accelerated flow increases with incidence angle which leads to regions of more laminar flow than turbulent flow.

These results indicate that the use of inviscid flow theory becomes increasingly less accurate at higher angles of attack since, due to flow separation, the viscous effects neglected in the panel method become increasingly more important. This is indicated in Fig. 6.10, which shows the calculated inviscid lift coefficients for this airfoil together with the experimental data reported in [3] for chord Reynolds numbers, R_c ($\equiv V_\infty c/\nu$), of 3×10^6 and 9×10^6 . As can be seen, the calculated results agree reasonably well with the measured values at low and modest angles of attack. With increasing angle of attack, the lift coefficient reaches a maximum, called the maximum lift coefficient, $(c_l)_{\max}$, at an angle of attack, α , called the stall angle. After this angle of attack, while the experimental lift coefficients begin to decrease with increasing angle of attack, the calculated lift coefficient, independent of Reynolds number, continuously increases with increasing α . The inviscid lift slope is not influenced by R_c , but $c_{l\max}$ is dependent upon R_c .

Figure 6.11 shows the moment coefficient c_m about the aerodynamic center. In general, moments on an airfoil are a function of angle of attack. However, there is one point on the airfoil about which the moment is independent of α ; this point is referred to as the aerodynamic center. As illustrated by Fig. 6.11, the moment coefficient is insensitive to R_c except at higher angles of attack.

6.6.2 Flow Over a Circular Cylinder

The computer program of Section 6.5 can also be used for two-dimensional geometries other than airfoils. To demonstrate this, consider flow over a circular cylinder of radius r_0 and compute its external velocity distribution with the panel program. The Kutta–Joukowski theorem states that the lift per unit

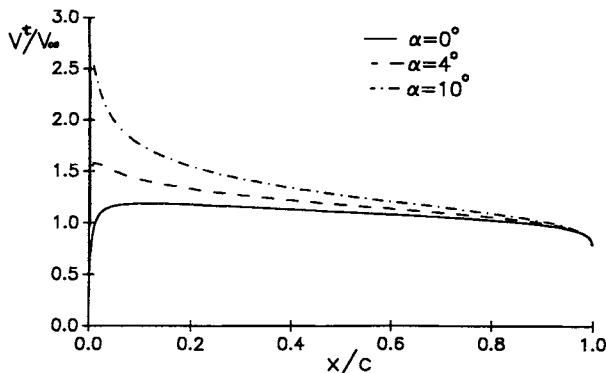


Fig. 6.9. Distribution of dimensionless external velocity distribution V^t/V_∞ on the upper surface of a NACA 0012 airfoil at three angles of attack.

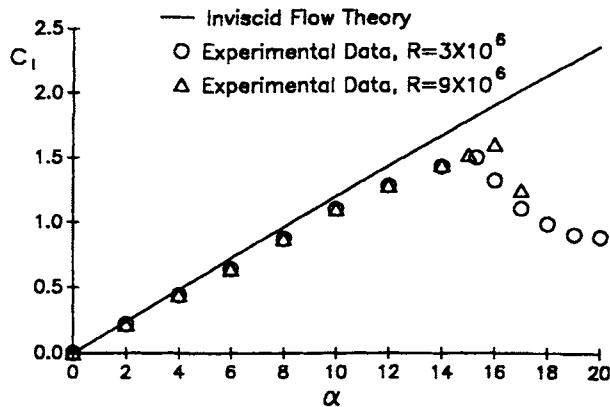


Fig. 6.10. Comparison of calculated and experimental lift coefficients for the NACA 0012 airfoil.

span of an isolated two-dimensional body in an incompressible inviscid flow is proportional to the net circulation Γ around the body, that is

$$L = \rho V_\infty \Gamma \quad (6.6.1)$$

Since for smooth bodies such as circular cylinders, the Kutta condition does not apply as discussed in Section 6.4, one must specify the circulation instead of determining it by the Kutta condition. For this reason, values of net circulation Γ (equal to $2\pi r_0 \tau$ in the panel method) are specified and Eq (6.4.14) is written as

$$\sum_{j=1}^N A_{ij}^n q_j = -\tau \sum_{j=1}^N B_{ij}^n - V_\infty \sin(\alpha - \theta_i) \quad i = 1, 2, \dots, N \quad (6.6.2)$$

N algebraic equations containing N unknowns, q_j ($j = 1, 2, \dots, N$), must be solved, again with the Gauss elimination method. Appendix B presents the

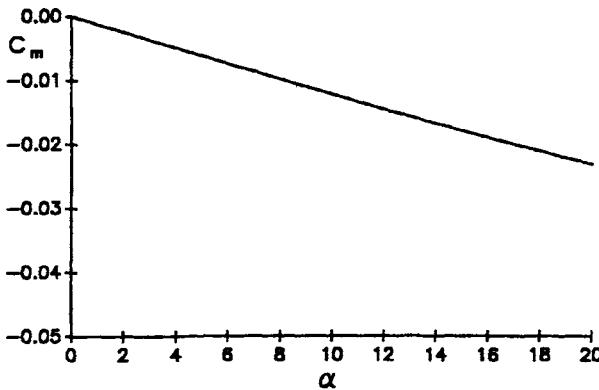


Fig. 6.11. Variation of the moment coefficient c_m with angle of attack, α for the NACA 0012 airfoil.

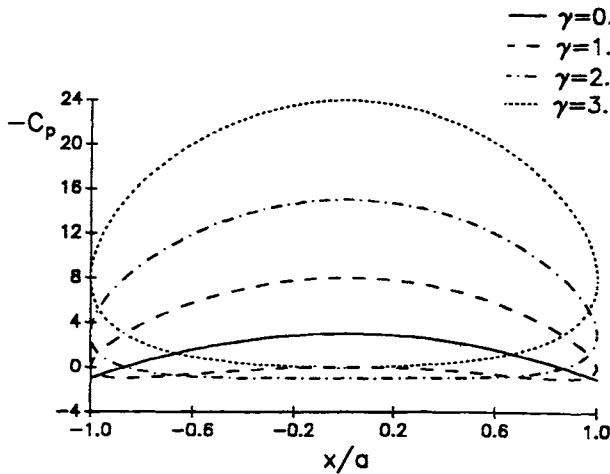


Fig. 6.12. Distribution of the pressure coefficients around a circular cylinder for four values of γ .

revised computer program and contains the modifications required to each subroutine (except GAUSS and CLCM) of the panel program and to its MAIN program.

Figure 6.12 shows the computed pressure coefficients around the circular cylinder for values of τ (GAMMA) equal to 0, 1, 2 and 3. These results are in excellent agreement with the analytical solution with

$$\frac{V}{V_\infty} = 2 \sin \theta + \frac{\Gamma}{2\pi V_\infty r_0} \quad (6.6.3)$$

obtained from inviscid flow theory as described, for example, in Anderson [2].

6.6.3 Multielement Airfoils

As was discussed in Section 1.3, the maximum lift coefficient, $(c_l)_{\max}$, has an important impact on the performance of an airplane. For a complete flight vehicle, the maximum lift coefficient determines the stalling speed of the aircraft. To show this, consider a steady equilibrium flight where the lift L equals the aircraft weight, w . Thus,

$$L = w = \frac{1}{2} \rho V_\infty^2 S c_l \quad (6.6.4)$$

This means that the lowest possible speed of the aircraft, the stalling speed, V_{stall} , occurs when the lift coefficient is maximum, that is,

$$V_{\text{stall}} = \sqrt{\frac{2w}{\rho S (c_l)_{\max}}} \quad (6.6.5)$$

Thus, it is very important to increase the maximum lift coefficient of an airfoil in order to obtain either a lower stalling speed or a higher payload weight at the same speed.

Since for an airfoil $(c_l)_{\max}$ is a function of its shape at a specified Reynolds number, it is necessary to resort to some special measures in order to increase its $(c_l)_{\max}$ beyond its value fixed by the airfoil shape and Reynolds number. Such special measures include the use of flaps and/or leading-edge slats; these are referred to as high-lift devices or multielement airfoils. Their flowfields and section lift and moment coefficients can be calculated with modifications to the panel program, as discussed in some detail below.

For a multielement airfoil configuration at an angle of attack α , each element is represented by a finite number of panels separately. Assuming that the total number of elements is N , and the number of panels for the l -th element is n_l , the source strength q_{jl} is taken to be constant over the j -th panel of the l -th element and the vorticity strength τ_l to be constant on all panels of the l -th element. The flow tangency condition, similar to Eq. (6.4.14), can be written for $i = 1, 2, \dots, n_k$, $k = 1, 2, \dots, N$ as

$$\sum_{l=1}^N \sum_{j=1}^{n_l} A_{ik,jl}^n q_{jl} + \sum_{l=1}^N \tau_l \sum_{j=1}^{n_l} B_{ik,jl}^n + V_\infty \sin(\alpha - \theta_{ik}) = 0 \quad (6.6.6)$$

The Kutta conditions for each element are similar to that given by Eq. (6.5.3) and can be written for $k = 1, 2, \dots, N$ with $m = n_k$ as

$$\begin{aligned} & \sum_{l=1}^N \sum_{j=1}^{n_l} (A_{1k,jl}^t + A_{mk,jl}^t) q_{jl} + \sum_{l=1}^N \tau_l \sum_{j=1}^{n_l} (B_{1k,jl}^t + B_{mk,jl}^t) \\ & = -V_\infty \cos(\alpha - \theta_{1k}) - V_\infty \cos(\alpha - \theta_{mk}) \end{aligned} \quad (6.6.7)$$

Again, the Gaussian elimination method is used to solve for the M ($\equiv \sum_{l=1}^N n_l + N$) unknowns q_{jl} , τ_l from the M equations given above.

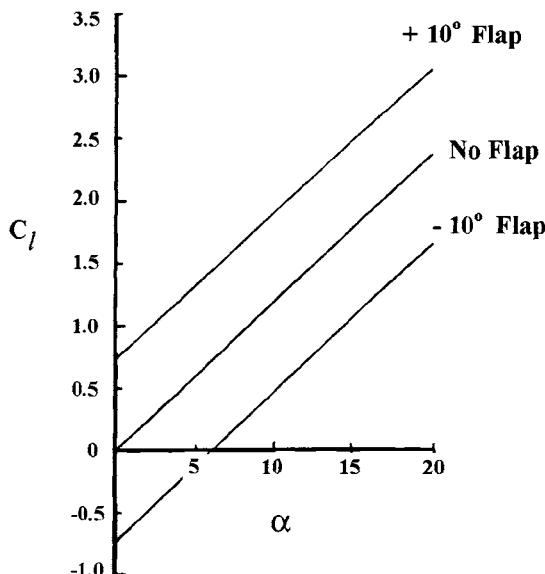


Fig. 6.13. Effect of flap deflection on the lift coefficient of a NACA 0012 airfoil.

The modifications to the computer program of Section 6.5 are relatively minor as indicated in the revised program given in Appendix B, where subroutine GAUSS remains unchanged.

To demonstrate the application of the multielement airfoil program, we consider a simple trailing-edge flap which is a portion of the trailing-edge section of an airfoil that is hinged and which can be deflected upward and downward. We assume that the airfoil section is again the NACA 0012 airfoil and compute its lift coefficients for flap deflection angles of $\pm 10^\circ$.

Figure 6.13 shows the calculated lift curves as a function of angle of attack for different flap deflections. Results show that flap deflection has no effect on the calculated lift curve slope and the curves are simply shifted up or down depending on the direction of the flap deflection. Results also show that the zero lift angle is shifted away from $\alpha = 0$ with flap deflection. Thus, the airfoil experiences a lifting force although the nominal angle of attack remains at zero. Flap deflection can be viewed as effectively changing the mean line, or the camber of the airfoil.

Appendix 6A Finite Difference Program for a Circular Cylinder

The finite difference program for the circular cylinder (Appendix B) consists of MAIN and two subroutines GAUSS and POISSON discussed in subsection 4.5.1 and given by Tables 4.2 and 4.3, respectively. While subroutine GAUSS

remains unchanged, subroutine POISSON (modified subroutine POISSON) has some minor changes as discussed below.

The statements in the beginning of this subroutine starting from

ELEMENTS OF DIAGONAL VECTORS IN THE BLOCK MATRICES A, B AND C require changes, as indicated below, up to the statement

ELEMENTS OF W-VECTOR, SEE EQ. (4.5.2)

The remaining statements stay unchanged. The listing is given in Appendix B, Chapter 6.

Appendix 6B Panel Program for an Airfoil

6B.1 MAIN Program

In the revised MAIN program given below, we read the value of τ ($\equiv \text{GAMMA}$), the total number of grid points (NODTOT) in the interval $0 \leq \theta \leq 2\pi$. The listing is given in Appendix B, Chapter 6.

6B.2 Subroutine COEF

Since the Kutta condition is not used, the statements regarding this condition are removed as shown in Appendix B, Chapter 6.

The index KUTTA, which represents the total number of equations, is replaced by NEQNS = NODTOT, and Eq. (6.5.2) is incorporated in the statement under “ELEMENTS OF VECTOR B for I = 1 to N.”

6B.3 Subroutine VPDIS

With values of GAMMA given as the input and passed from the MAIN program, the revised subroutine is given in Appendix B, Chapter 6.

Appendix 6C Panel Program for Multielement Airfoils

6C.1 MAIN Program

The number of elements (NBODY) the number of panels for the K -th element (NODT(K)), airfoil coordinates ($X(I,K)$, $Y(I,K)$) for the K -th element, the angle of attack (ALPHA) and the reference length (CHORD) are input to the revised MAIN Program. The listing of MAIN and other subroutines are given in Appendix B, Chapter 6.

6C.2 Subroutine COEF

This subroutine is modified according to Eqs. (6.5.2) and (6.5.4).

6C.3 Subroutine VPDIS

This subroutine is also modified to take into account the distribution of all the sources and vorticities from each element.

6C.4 Subroutine CLCM

In this subroutine, the lift CL and pitching moment CM coefficients of each element are first determined and then the total lift CLT and total pitching moment CMT coefficients are calculated by adding together the coefficients of each element.

References

- [1] Hess, J. L. and Smith, A. M. O.: "Calculation of Potential Flow About Arbitrary Bodies," *Progress in Aerospace Sciences*, Vol. 5, Pergamon Press, N.Y., 1966.
- [2] Anderson, J. D.: *Fundamentals of Aerodynamics*, McGraw Hill, Inc., N.Y., 1991.
- [3] Abbott, J. H. and von Doenhoff, A. E.: *Theory of Wing Sections*, Dover, 1959.
- [4] Cebeci, T., *An Engineering Approach to the Calculation of Aerodynamic Flows*, Horizons Pub., Long Beach, Calif., and Springer, Heidelberg, 1999.

Problems

6-1. Consider a two-dimensional, steady, fully-developed laminar flow in a channel of rectangular cross-section with a width of $2L$ in the z -direction and the distance between upper and lower surfaces zl . Determine the velocity distribution by the direct method discussed in subsection 4.5.1 by solving

$$0 = -\frac{1}{\mu} \frac{dp}{dx} + \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (\text{P6.1.1})$$

subject to the boundary conditions in one symmetric quadrant of the flow

$$\frac{\partial u(0, z)}{\partial y} = 0, \quad u(l, z) = 0 \quad (\text{P6.1.2a})$$

$$\frac{\partial u(y, 0)}{\partial z} = 0, \quad u(y, L) = 0 \quad (\text{P6.1.2b})$$

Hint: First express the above equations in dimensionless form by defining

$$u^* = \frac{u}{-\frac{dp}{dx} \frac{l^2}{\mu}}, \quad m = \frac{l}{L}, \quad \eta = \frac{y}{L}, \quad \zeta = \frac{z}{L}$$

so that Eqs. (P6.1.1) and (P6.1.2) can be written as

$$\frac{\partial^2 u^*}{\partial \eta^2} + m^2 \frac{\partial^2 u^*}{\partial \zeta^2} = -1 \quad (\text{P6.1.3})$$

$$\frac{\partial u^*(0, \zeta)}{\partial \eta} = 0, \quad u^*(1, \zeta) = 0 \quad (\text{P6.1.4a})$$

$$\frac{\partial u^*(\eta, 0)}{\partial \zeta} = 0, \quad u^*(\eta, 1) = 0 \quad (\text{P6.1.4b})$$

Compare your results with the analytical solution given by

$$u^* = \frac{1}{2}[1 - \eta^2] - 1 \sum_{n=0}^{\infty} \frac{(-1)^n}{(\lambda_n \ell)^3} \left(\frac{\cosh \lambda_n \ell \zeta / m}{\cosh \lambda_n \ell / m} \right) \cosh \lambda_n \ell \eta \quad (\text{P6.1.5})$$

where

$$\lambda_n \ell = (2n + 1) \frac{\pi}{2}, \quad n = 0, 1, 2, \dots$$

Take $m = 2$, $\Delta\eta = \Delta\zeta = 0.002$ and a convergence criterion of

$$\max_{i,j} |u_{i,j}^{(\nu+1)} - u_{i,j}^{(\nu)}| \leq 10^{-7} \quad (\text{P6.1.6})$$

Note that Subroutine GAUSS (Table 4.2) remains unchanged. Note also the subroutine POISSON requires changes given below,

```

SUBROUTINE POISSON(II,JJ,TX,TY,F,U)
DIMENSION A(100,100),B(100,100),C(100,100),D(100,100),E(100,100)
DIMENSION DELTA(100,100,100), DELTT(100,100), BM(100,100)
DIMENSION DM(100,100),F(100,100),W(100,100),UT(100),U(100,100)
C ELEMENTS OF DIAGONAL VECTORS IN THE BLOCK MATRICES A,B AND C
DO 5 I=1,II
DO 5 J=1,JJ
  A(I,J) = -TX
  B(I,J) = 1.
  C(I,J) = -TX
  D(I,J) = -TY
  E(I,J) = -TY
5 CONTINUE
DO 6 I=1,II
  B(I,1) = 1.-4./3.*TY
  E(I,1) = -2./3.*TY
6 CONTINUE
DO 7 J=1,JJ
  B(1,J) = B(1,J)-4./3.*TX
  C(1,J) = -2./3.*TX
7 CONTINUE

```

The remaining statements in this subroutine are the same as those in Table 4.3 starting with

C ELEMENTS OF W-VECTOR, SEE EQ. (4.5.21a)

6-2. Repeat Problem 6.1 by using the accelerated Gauss-Seidel method discussed in subsection 4.5.2. Assume that the initial solution is given by

$$u^* = \frac{1}{2}(1 - \eta^2) \left[1 - \frac{\cosh(\sqrt{3}/l)z}{\cosh(\sqrt{3}/l)L} \right]$$

6-3. Repeat Problem 6.1 by using the ADI method of subsection 4.5.2. Take $\omega = 0.05$.

6-4. The NACA 0012 airfoil is a conventional airfoil which has a favorable pressure distribution on the upper surface up to about a quarter chord point at $\alpha = 0^\circ$; with increasing incidence angle, say $\alpha = 8^\circ$, the gradient becomes unfavorable over practically the entire surface.

(a) Compute the pressure distribution on this airfoil with the panel program of Section 6.5 for angles of attack of $\alpha = 0^\circ, 4^\circ, 10^\circ$ and plot C_p vs x/c and V/V_∞ vs x/c for each α .

(b) Compute the lift coefficients for angles of attack of α from 0° to 20° at 4° degree intervals and compare them with the experimental data in Fig. P6.1 obtained for a chord Reynolds number R_c ($\equiv V_\infty c/\nu$) of 6×10^6 . Discuss the numerical results with experimental data.

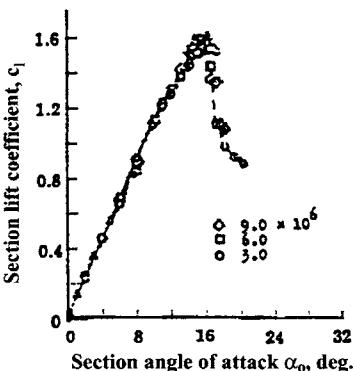
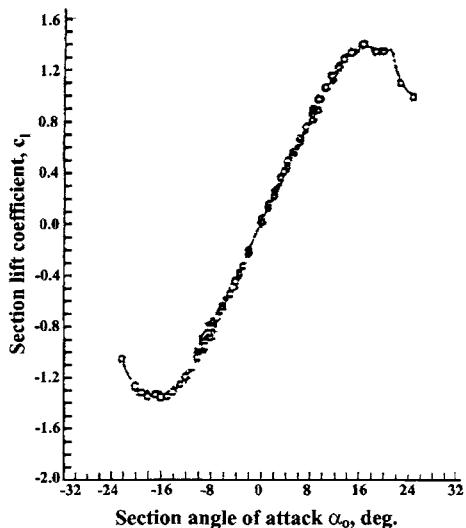


Fig. P6.1. Experimental data for the NACA 0012 airfoil.

6-5. The NACA 653-018 airfoil (Table P6.1) is an example of a low drag or laminar flow airfoil. The subscript 3 indicates that the drag coefficient is a minimum over a range of lift coefficients of 0.3 on either side of the design lift coefficient, which for this airfoil section is zero. The performance characteristics of this airfoil differ from a conventional airfoil, like the NACA 0012 airfoil, in that near the design lift coefficient, the low-drag airfoil has a laminar boundary

Table P6.1. Coordinates of the NACA 65₃-018 airfoil.

x/c	y/c	x/c	y/c
0.005	0.01337	0.45	0.08901
0.0075	0.01608	0.50	0.08568
0.0125	0.02014	0.55	0.08008
0.0250	0.02751	0.60	0.07267
0.050	0.03866	0.65	0.06395
0.075	0.04733	0.70	0.05426
0.10	0.05457	0.75	0.04396
0.15	0.06606	0.80	0.03338
0.20	0.07476	0.85	0.02295
0.25	0.08129	0.90	0.01319
0.30	0.08595	0.95	0.00490
0.35	0.08886	1.0	0
0.40	0.08999		

**Fig. P6.2.** Experimental data for the NACA 65₃-018 airfoil.

layer over a considerable portion of its area. The late transition is accomplished by designing the airfoil so that the pressure gradient is favorable over a larger area. A favorable pressure gradient tends to delay transition. The main geometrical difference that accounts for the change in the pressure distribution is a movement rearward of the point of maximum thickness.

(a) Repeat Problem 6.4 for this airfoil and compare your calculated results with the experimental data (Fig. P6.2) obtained for a chord Reynolds number of 6×10^6 .

(b) Discuss the difference in results for both airfoils.

6-6. The NASA GA(W)-2 airfoil (see Table P6.2), which is a general aviation airfoil designed by Whitcomb of NASA Langley Research Center, is an example of a supercritical airfoil. Supercritical airfoils were originally developed to delay the drag rise Mach number in the transonic flow regime. It was later found that airfoils of this type were also useful in the lower speed range of the general aviation aircraft, and exhibited good high-lift capabilities. Their only drawback is their relatively high pitching moment coefficient caused by the maximum camber being located far aft along the chord line. With proper design, however, this drawback can be avoided.

Table P6.2 presents the airfoil coordinates and Fig. P6.3 presents experimental data for a range of angles of attack at a chord Reynolds number of 4.1×10^6 .

Table P6.2. Airfoil coordinates for the NASA GA(W)-2 airfoil, $c = 61.0\text{ cm}$.

x/c	$(y/c)_{\text{upper}}$	$(y/c)_{\text{lower}}$	x/c	$(y/c)_{\text{upper}}$	$(y/c)_{\text{lower}}$
0.0	0.0	0.0	0.54954	0.08025	-0.03803
0.00199	0.00922	-0.00486	0.59950	0.07609	-0.03326
0.01246	0.02365	-0.01385	0.64946	0.01035	-0.02745
0.03747	0.03957	-0.02196	0.69942	0.06305	-0.02107
0.07494	0.05230	-0.02904	0.74938	0.05446	-0.01460
0.12490	0.06323	-0.03528	0.79933	0.4476	-0.00851
0.17485	0.07080	-0.03769	0.84929	0.3417	-0.00357
0.24980	0.07857	-0.04353	0.89925	0.02296	-0.00086
0.34971	0.08357	-0.04508	0.94921	0.01112	-0.00143
0.39967	0.08441	-0.04475	0.97419	0.00497	-0.00377
0.49958	0.08294	-0.04149	0.99917	-0.00143	-0.00720
			1.0	-0.00164	-0.00732

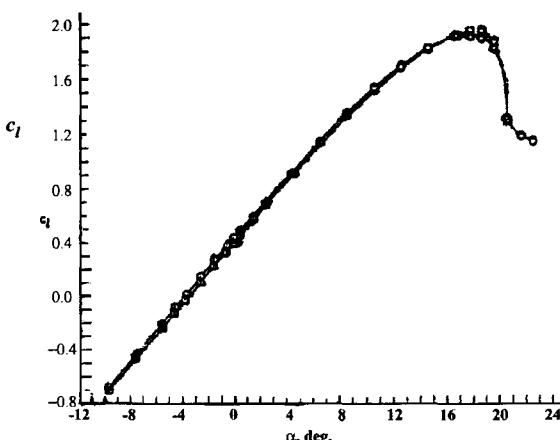


Fig. P6.3. Experimental data for the NACA GA(W)-2 airfoil, $R_c = 4.1 \times 10^6$.

The measurements were made without and with a roughness strip located near the leading edge.

- (a) With the panel program of Section 6.5, compute the pressure distribution on this airfoil for $\alpha = 0^\circ, 4^\circ$ and 8° , and plot the results C_p vs x/c and V/V_∞ vs x/c for each angle of attack.
- (b) Compute the lift and moment coefficients for $\alpha = -8^\circ, -4^\circ, 0^\circ, 4^\circ, 8^\circ, 12^\circ, 16^\circ$ and 20° , and compare them with the experimental data in Fig. P6.3. Discuss the calculated results with measurements as well as with those obtained for the NACA 0012 airfoil in Problem 6.4.

7

Boundary-Layer Equations

7.1 Introduction

The solution of the boundary-layer equations of subsection 2.4.3 can be obtained for boundary conditions that include a priori specification of the external pressure or velocity distributions either from experimental data or from inviscid flow theory (called the standard problem). The solution of the boundary-layer equations can also be obtained for boundary conditions that include a priori specification of an alternative boundary condition which may be the longitudinal variation of the cross-sectional area of a duct or of a displacement thickness (called the inverse problem) or the determination of the freestream boundary condition by iteration between solutions of inviscid and boundary-layer equations (called the interaction problem).

Section 7.2 describes the standard and inverse problems for two-dimensional laminar and turbulent flows. It also discusses a brief description of the interaction problem. This section is followed by a description of the numerical procedures used to solve the boundary-layer equations in standard mode. Section 7.4 presents and describes a computer program for two-dimensional incompressible external flows. Applications of this program for boundary conditions of relevance to engineering for a sample of flows are discussed in Section 7.5.

In the solution of the boundary-layer equations for turbulent flows, we make use of the eddy-viscosity concept discussed in Section 3.1 so that the momentum equation given by Eq. (2.4.34) can be written as

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = - \frac{1}{\varrho} \frac{dp}{dx} + \nu \frac{\partial}{\partial y} \left(b \frac{\partial u}{\partial y} \right) \quad (7.1.1)$$

where

$$b = 1 + \varepsilon_m^+, \quad \varepsilon_m^+ = \frac{\varepsilon_m}{\nu} \quad (7.1.2)$$

In this way the solution procedure for the momentum equation and continuity equation, Eq. (2.4.33), which is renumbered for convenience

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (7.1.3)$$

is the same for laminar and turbulent flows with an algebraic eddy-viscosity formulation. The solution procedure is general and can be used to solve the turbulent boundary-layer equations with turbulence models other than those based on an algebraic eddy viscosity formulation as discussed in [1, 2].

7.2 Standard, Inverse and Interaction Problems

Equations (7.1.1) and (7.1.3) apply to internal and external flows. In the latter case they are often solved for prescribed pressure $p(x)$ or external velocity $u_e(x)$ (see Section 2.7) that are related to each other by Bernoulli's equation,

$$-\frac{1}{\rho} \frac{dp}{dx} = u_e \frac{du_e}{dx} \quad (7.2.1)$$

so that a specification of one variable implies the other. This procedure, sometimes referred to as the *standard problem*, can be used to solve Eqs. (7.1.1) and (7.1.3) subject to the boundary conditions discussed in Section 2.7 provided that boundary-layer separation, which corresponds to vanishing wall shear, does not occur. If the wall shear vanishes at some x -location, the solutions break down and convergence cannot be obtained. This is referred to as the singular behavior of the boundary-layer equations at separation. For laminar flows, the behavior of the wall shear τ_w close to the separation point x_s , has been shown to be of the form

$$\left(\frac{\partial u}{\partial y} \right) \sim (x_s - x)^{1/2} \quad (7.2.2)$$

by Goldstein [3], who considered a uniformly retarded flow past a semi-infinite plate and showed that, with the relation given above, there is no real solution downstream of separation; the normal velocity component v becomes infinite at x_s . Goldstein also pointed out that the pressure distribution around the separation point cannot be taken arbitrarily and must satisfy conditions associated with the existence of reverse flow downstream of separation. This is consistent with the fact that the standard boundary-layer equations are parabolic in space, whereas flow separation introduces the elliptic nature of the flowfield.

The inviscid-pressure distribution for flow past a circular cylinder leads to a velocity maximum at 90° and separation at about 105° from the front stagnation point, whereas viscous flow in the subcritical range of Reynolds number leads to a velocity maximum at 70° and separation at 83° . As can be seen from Fig. 7.1, viscosity modifies the inviscid flow, which cannot serve even as a first

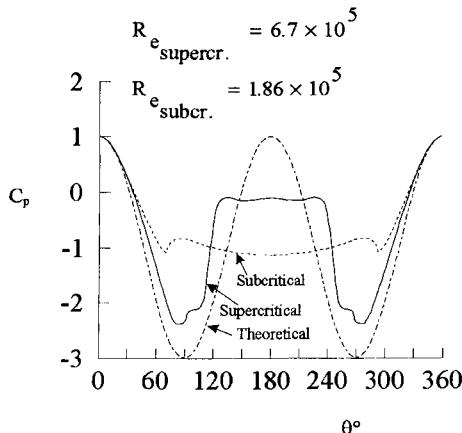


Fig. 7.1. Pressure distribution around a circular cylinder in the subcritical and supercritical range of Reynolds numbers $q_\infty = 1/2\varrho V_\infty^2$.

approximation to the actual flow. In the case of streamlined bodies, where separation takes place close to the rear stagnation point, the solution of the inviscid flow equations can serve as a good approximation to real flow as discussed in detail in [4]. For completeness, it is briefly discussed below.

The boundary-layer equations are not singular at separation when the external velocity or pressure is computed as part of the solution. Catherall and Mangler [5] were the first to show that in two-dimensional steady laminar flows, the modification of the external velocity distribution near the region of flow separation leads to solutions free of numerical difficulties. Prescribing the displacement thickness as a boundary condition at the boundary-layer edge, that is,

$$y = \delta, \quad \delta^*(x) = \text{given} \quad (7.2.3)$$

in addition to those given by Eq. (2.7.3) with no mass transfer, they were able to integrate the boundary-layer equations through the separation location and into a region of reverse flow without any evidence of singularity at the separation point. This procedure for solving the boundary-layer equations for a prescribed displacement thickness distribution, with the external velocity or pressure computed as part of the solution is known as the *inverse problem*. This observation of Catherall and Mangler has led to other studies by various investigators of inverse solutions of the boundary-layer equations; these are obtained by prescribing distributions of displacement thickness or wall shear. Furthermore, it has been demonstrated in [6] that for flows with separation bubbles, these solutions are in good agreement with the solutions of the Navier-Stokes equations.

A problem associated with the use of these inverse techniques for external flows is the lack of a priori knowledge of the required displacement thickness or wall shear. The appropriate value must be obtained as part of the overall problem from interaction between the boundary layer and the inviscid flow. In

the case of internal flows, the problem is somewhat easier because as discussed in [4], the conservation of mass in integral form can be used to relate pressure $p(x)$ to velocity $u(x, y)$ in terms of mass balance in the duct.

For two-dimensional external flows, two procedures have been developed to couple the solutions of the inviscid and viscous equations for airfoil flows. In the first procedure, developed by Le Balleur [7] and Carter and Wornom [8], the solution of the boundary-layer equations is obtained by the standard method, and a displacement-thickness, $\delta^{*0}(x)$, distribution is determined. If this initial calculation encounters separation, $\delta^{*0}(x)$ is extrapolated to the trailing edge of the airfoil. For the given $\delta^{*0}(x)$ distribution, the boundary-layer equations are then solved in the inverse mode to obtain an external velocity $u_{ev}(x)$. An updated inviscid velocity distribution, $u_{ei}(x)$, is then obtained from the inviscid flow method with the added displacement thickness. A relaxation formula is introduced to define an updated displacement-thickness distribution,

$$\delta^*(x) = \delta^{*0}(x) \left\{ 1 + \omega \left[\frac{u_{ev}(x)}{u_{ei}(x)} - 1 \right] \right\} \quad (7.2.4)$$

where ω is a relaxation parameter, and the procedure is repeated with this updated mass flux.

In the second approach, developed by Veldman [9], the external velocity $u_e(x)$ and the displacement thickness $\delta^*(x)$ are treated as unknown quantities, and the equations are solved in the inverse mode simultaneously in successive sweeps over the airfoil surface. For each sweep, the external boundary condition for the boundary-layer equation dimensionless form, with $u_e(x)$ normalized with u_∞ , is written as

$$u_e(x) = u_e^0(x) + \delta u_e(x) \quad (7.2.5a)$$

Here $u_e^0(x)$ denotes the inviscid velocity and δu_e the perturbation due to the displacement thickness, which is calculated from the Hilbert integral

$$\delta u_e = \frac{1}{\pi} \int_{x_a}^{x_b} \frac{d}{d\sigma} (u_e \delta_*) \frac{d\sigma}{x - \sigma} \quad (7.2.5b)$$

The term $\frac{d}{d\sigma} (u_e \delta_*)$ in the above equation denotes the blowing velocity used to simulate the boundary-layer in the region (x_a, x_b) . This approach is more general and can be used for two- and three-dimensional flows as discussed in [4].

Predicting the flowfield by solutions based on inviscid-flow theory is usually adequate as long as the viscous effects are negligible. A boundary layer that forms on the surface causes the irrotational flow outside it to be on a surface displaced into the fluid by a distance equal to the displacement thickness δ^* , which represents the deficiency of mass within the boundary layer. Thus, a new boundary for the inviscid flow, taking the boundary-layer effects into consideration, can be formed by adding δ^* to the body surface. The new surface is called the displacement surface and, if its deviation from the original surface is not

negligible, the inviscid flow solutions can be improved by incorporating viscous effects into the inviscid flow equations [4].

A convenient and popular approach is based on the concept that the displacement surface can also be formed by distributing a blowing or suction velocity on the body surface. The strength of the blowing or suction velocity v_b is determined from the boundary-layer solutions according to

$$v_b = \frac{d}{dx}(u_e \delta^*) \quad (7.2.6)$$

where x is the surface distance of the body, and the variation of v_b on the body surface simulates the viscous effects in the potential flow solution. This approach can be used for both incompressible and compressible flows as discussed in [4].

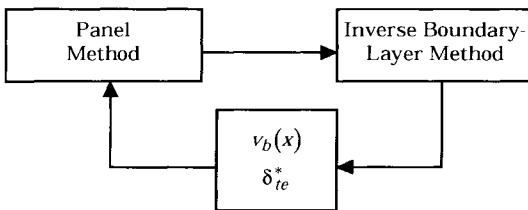


Fig. 7.2. Interactive boundary-layer scheme.

In the application of this interaction problem for an airfoil in subsonic flow for a given airfoil geometry and freestream flow conditions, we first obtain the inviscid velocity distribution with a panel method such as the one described in Chapter 6, we then solve the boundary-layer equations in the inverse mode so that the blowing velocity distribution, $v_b(x)$, is computed from Eq. (7.2.6) and the displacement thickness distribution $\delta^*(x)$ on the airfoil and in the wake are then used in the panel method to obtain an improved inviscid velocity distribution with viscous effects as described in detail in [2]. The δ_{te}^* is used to satisfy the Kutta condition in the panel method at a distance equal to δ_{te}^* ; this is known as the off-body Kutta condition (Fig. 7.2). In the first iteration between the inviscid and the inverse boundary-layer methods, $v_b(x)$ is used to replace the zero blowing velocity at the surface. At the next and following iterations, a new value of $v_b(x)$ in each iteration is used as a boundary condition in the panel method. This procedure is repeated for several cycles until convergence is obtained, which is usually based on the lift and total drag coefficients of the airfoil. Studies discussed in [4] show that with three boundary-layer sweeps for one cycle, convergence is obtained in less than 10 cycles.

7.3 Numerical Method for the Standard Problem

In this section the momentum and continuity equations are considered and their solutions for the standard problem corresponding to an external flow in which the boundary conditions are given by Eq. (2.7.3) are discussed; the boundary conditions are renumbered for convenience

$$y = 0, \quad u = 0, \quad v = v_w(x) \quad (7.3.1a)$$

$$y = \delta, \quad u = u_e(x) \quad (7.3.1b)$$

Since the boundary-layer equations are parabolic, their solution procedure employs a marching scheme similar to the unsteady heat conduction equation discussed in Section 4.4. It begins with initial conditions, say at $x = x_0$ and proceeds in the positive x -direction with dependent variables determined in sequence normal to the flow at each x -station subject to the boundary conditions.

The continuity and momentum equations given by Eqs. (7.1.3) and (7.1.1) can be solved in the form they are expressed. They can also be solved after they are expressed as a third order equation by using the definition of stream function $\psi(x, y)$. Noting that

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \quad (7.3.2)$$

Eqs. (7.1.3) and (7.1.1), with a prime denoting differentiation with respect to y can be written as

$$\nu[b\psi'']' + u_e \frac{du_e}{dx} = \psi' \frac{\partial \psi'}{\partial x} - \psi'' \frac{\partial \psi}{\partial x} \quad (7.3.3)$$

In either form, for given initial conditions, say at $x = x_0$ and eddy viscosity distribution, these equations are solved subject to their boundary conditions in the interval 0 to δ at each specified x -location greater than x_0 . The boundary-layer thickness $\delta(x)$, however, increases with increasing downstream distance x for both laminar and turbulent flows; to maintain computational accuracy, it is necessary to take small steps in the streamwise direction to maintain computational accuracy.

Transformed coordinates employing similarity variables are advantageous in the solution of boundary-layer equations since they can reduce the growth of $\delta(x)$ and thus allow larger steps to be taken in the streamwise direction. Furthermore, and as shown below, for laminar flows they can also be used to generate initial conditions.

There are several transformations that can be used for this purpose, and even though most of them have been developed for laminar flows, they can also be used for turbulent flows. The Falkner–Skan transformation discussed in [4] is a convenient choice. In this transformation, a dimensionless similarity variable η and a dimensionless stream function $f(x, \eta)$ are defined by

$$\eta = \sqrt{u_e/\nu x} y, \quad \psi(x, y) = \sqrt{u_e \nu x} f(x, \eta) \quad (7.3.4)$$

Noting the chain rule relations (subsection 2.2.4), it can be shown that, with a prime denoting differentiation with respect to η , and with

$$\xi = \frac{x}{L}, \quad \bar{u}_e = \frac{u_e}{u_\infty} \quad (7.3.5)$$

Eq. (7.3.3) and its boundary conditions, Eq. (7.3.1), can be written as

$$(bf'')' + \frac{m+1}{2} ff'' + m[1 - (f')^2] = \xi \left(f' \frac{\partial f'}{\partial \xi} - f'' \frac{\partial f}{\partial \xi} \right) \quad (7.3.6)$$

$$\eta = 0, \quad f' = 0, \quad f(\xi, 0) \equiv f_w(\xi) = -\frac{\sqrt{R_L}}{\sqrt{\bar{u}_e \xi}} \int_0^\xi \frac{v_w}{u_\infty} d\xi \quad (7.3.7a)$$

$$\eta = \eta_e, \quad f' = 1 \quad (7.3.7b)$$

Here η_e corresponds to a transformed boundary-layer thickness, $\sqrt{u_e/\nu x} \delta$, or $\sqrt{\bar{u}_e R_L / \xi} \delta / L$; R_L is a Reynolds number based on reference velocity u_∞ and length L ; m is a dimensionless pressure-gradient parameter defined by

$$m = \frac{\xi}{\bar{u}_0} \frac{d\bar{u}_e}{d\xi} \quad (7.3.8)$$

The velocity components u and v are related to the dimensionless stream function $f(\xi, \eta)$ by

$$u = u_e f' \quad (7.3.9a)$$

$$v = -\sqrt{u_e \nu x} \left[\frac{f}{\sqrt{u_e x}} \frac{d}{dx} \sqrt{u_e x} + \frac{\partial f}{\partial x} + f' \frac{\partial \eta}{\partial x} \right] \quad (7.3.9b)$$

For laminar flows with the external velocity of the form

$$u_e = Cx^m \quad (7.3.10)$$

with C and m constants, and with boundary conditions on f and f' independent of x , the left-hand side of Eq. (7.3.6) reduces to

$$f''' + \frac{m+1}{2} ff'' + m[1 - (f')^2] = 0 \quad (7.3.11)$$

which is called the Falkner–Skan equation. Its solutions, which are limited to $-0.0904 \leq m \leq \infty$, are independent of ξ and can be used to generate the initial conditions needed for the solution of the boundary-layer equations, as shall be shown later.

There are several numerical methods that can be used to solve the boundary-layer equations. Finite-difference methods offer the greatest flexibility, and those of Crank–Nicolson and Keller, discussed in Section 4.4, have been widely used. The latter method provides significant advantages over the former method and will be used to solve the boundary-layer equations in this chapter, as well as the stability equation to be discussed in Chapter 8.

To solve Eqs. (7.3.6) and (7.3.7) with Keller's box method, we follow the four steps discussed in subsection 4.4.3.

7.3.1 Numerical Formulation

In order to express Eqs. (7.3.6) and (7.3.7) as a system of first-order equations, we define new variables $u(\xi, \eta)$ and $v(\xi, \eta)$ by

$$f' = u \quad (7.3.12a)$$

$$u' = v \quad (7.3.12b)$$

and write Eqs. (7.3.6) and (7.3.7) as

$$(bv)' + \frac{m+1}{2}fv + m(1-u^2) = \xi \left(u \frac{\partial u}{\partial \xi} - v \frac{\partial f}{\partial \xi} \right) \quad (7.3.12c)$$

$$\eta = 0, \quad u = 0, \quad f = f_w(x); \quad \eta = \eta_e, \quad u = 1 \quad (7.3.13)$$

We denote the net points of the net rectangle shown in Fig. 4.6, modified below due to a slight change in notation, by

$$\begin{aligned} \xi_0 &= 0, & \xi_n &= \xi_{n-1} + k_n, & n &= 1, 2, \dots, N \\ \eta_0 &= 0, & \eta_j &= \eta_{j-1} + h_j, & j &= 1, 2, \dots, J \end{aligned} \quad (7.3.14)$$

and write the difference equations that are to approximate Eqs. (7.3.12) by considering one mesh rectangle as in Fig. 7.3. We again start by writing the finite-difference approximations of the ordinary differential equations (7.3.12a,b) for the midpoint $(\xi^n, \eta_{j-1/2})$ of the segment P_1P_2 , using centered-difference derivatives (see subsection 4.4.3),

$$\frac{f_j^n - f_{j-1}^n}{h_j} = \frac{u_j^n + u_{j-1}^n}{2} \equiv u_{j-1/2}^n \quad (7.3.15a)$$

$$\frac{u_j^n - u_{j-1}^n}{h_j} = \frac{v_j^n + v_{j-1}^n}{2} \equiv v_{j-1/2}^n \quad (7.3.15b)$$

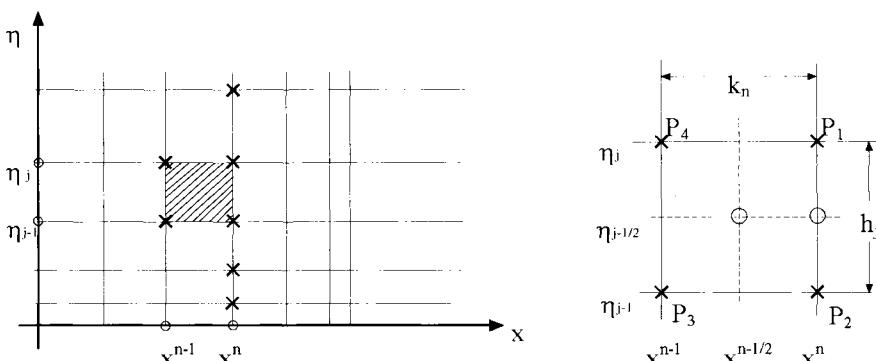


Fig. 7.3. Net rectangle for difference approximations.

Similarly, the partial differential equation (7.3.12c) is approximated by centering about the midpoint $(\xi^{n-1/2}, \eta_{j-1/2})$ of the rectangle $P_1P_2P_3P_4$. This can be done in two steps. In the first step we center it about $(\xi^{n-1/2}, \eta)$ without specifying η . If we denote its left-hand side by L , then the finite-difference approximation to Eq. (7.3.12c) is

$$\frac{1}{2}(L^n + L^{n-1}) = \xi^{n-1/2} \left[u^{n-1/2} \left(\frac{u^n - u^{n-1}}{k_n} \right) - v^{n-1/2} \left(\frac{f^n - f^{n-1}}{k_n} \right) \right] \quad (7.3.16)$$

$$\alpha^n = \frac{\xi^{n-1/2}}{k_n}, \quad \alpha_1 = \frac{m^n + 1}{2} + \alpha^n, \quad \alpha_2 = m^n + \alpha^n \quad (7.3.17a)$$

$$R^{n-1} = -L^{n-1} + \alpha^n [(fv)^{n-1} - (u^2)^{n-1}] - m^n \quad (7.3.17b)$$

$$L^{n-1} \equiv \left[(bv)' + \frac{m+1}{2} fv + m(1-u^2) \right]^{n-1} \quad (7.3.17c)$$

Eq. (7.3.16) can be written as

$$[(bv)']^n + \alpha_1(fv)^n - \alpha_2(u^2)^n + \alpha^n(v^{n-1}f^n - f^{n-1}v^n) = R^{n-1} \quad (7.3.18)$$

The identity sign introduces a useful shorthand: $[]^{n-1}$ means that the quantity in square brackets is evaluated at $\xi = \xi^{n-1}$.

We next center Eq. (7.3.18) about the point $(\xi^{n-1/2}, \eta_{j-1/2})$, that is, we choose $\eta = \eta_{j-1/2}$ and obtain

$$h_j^{-1}(b_j^n v_j^n - b_{j-1}^n v_{j-1}^n) + \alpha_1(fv)_{j-1/2}^n - \alpha_2(u^2)_{j-1/2}^n + \alpha^n(v_{j-1/2}^{n-1}f_{j-1/2}^n - f_{j-1/2}^{n-1}v_{j-1/2}^n) = R_{j-1/2}^{n-1} \quad (7.3.19)$$

where

$$R_{j-1/2}^{n-1} = -L_{j-1/2}^{n-1} + \alpha^n [(fv)_{j-1/2}^{n-1} - (u^2)_{j-1/2}^{n-1}] - m^n \quad (7.3.20a)$$

$$L_{j-1/2}^{n-1} = \left\{ h_j^{-1}(b_j v_j - b_{j-1} v_{j-1}) + \frac{m+1}{2} (fv)_{j-1/2} + m[1 - (u^2)_{j-1/2}] \right\}^{n-1} \quad (7.3.20b)$$

Eqs. (7.3.15) and (7.3.19) are imposed for $j = 1, 2, \dots, J-1$ at given η and the transformed boundary-layer thickness, η_e , is to be sufficiently large so that $u \rightarrow 1$ asymptotically. The latter is usually satisfied when $v(\eta_e)$ is less than approximately 10^{-3} .

The boundary conditions [Eq. (7.3.13)] yield, at $\xi = \xi^n$,

$$f_0^n = f_w, \quad u_0^n = 0, \quad u_J^n = 1 \quad (7.3.21)$$

7.3.2 Newton's Method

Unlike the unsteady heat conduction equation, Eq. (4.2.4), the boundary-layer equations are nonlinear. So after we write the finite-difference approximations, we follow the third step (see subsection 4.4.3) and linearize Eqs. (7.3.15), (7.3.19) and (7.3.21).

If we assume f_j^{n-1} , u_j^{n-1} , and v_j^{n-1} to be known for $0 \leq j \leq J$, then Eqs. (7.3.15), (7.3.19) and (7.3.21) form a system of $3J + 3$ equations for the solution of $3J + 3$ unknowns (f_j^n, u_j^n, v_j^n) , $j = 0, 1, \dots, J$. To solve this nonlinear system, we use Newton's method; we introduce the iterates $[f_j^{(\nu)}, u_j^{(\nu)}, v_j^{(\nu)}]$, $\nu = 0, 1, 2, \dots$, with initial value ($\nu = 0$) equal to those at the previous x -station x^{n-1} (which is usually the best initial guess available). For the higher iterates we set

$$f_j^{(\nu+1)} = f_j^{(\nu)} + \delta f_j^{(\nu)}, \quad u_j^{(\nu+1)} = u_j^{(\nu)} + \delta u_j^{(\nu)}, \quad v_j^{(\nu+1)} = v_j^{(\nu)} + \delta v_j^{(\nu)} \quad (7.3.22)$$

We then insert the right-hand sides of these expressions in place of f_j^n , u_j^n and v_j^n in Eqs. (7.3.15) and (7.3.19) and drop the terms that are quadratic in $\delta f_j^{(\nu)}$, $\delta u_j^{(\nu)}$ and $\delta v_j^{(\nu)}$. This procedure yields the following *linear* system (the superscript n is dropped from f_j , u_j , v_j and v from δ quantities for simplicity).

$$\delta f_j - \delta f_{j-1} - \frac{h_j}{2}(\delta u_j + \delta u_{j-1}) = (r_1)_j \quad (7.3.23a)$$

$$\delta u_j - \delta u_{j-1} - \frac{h_j}{2}(\delta v_j + \delta v_{j-1}) = (r_3)_{j-1} \quad (7.3.23b)$$

$$(s_1)_j \delta v_j + (s_2)_j \delta v_{j-1} + (s_3)_j \delta f_j + (s_4)_j \delta f_{j-1} + (s_5)_j \delta u_j + (s_6)_j \delta u_{j-1} = (r_2)_j \quad (7.3.23c)$$

where

$$(r_1)_j = f_{j-1}^{(\nu)} - f_j^{(\nu)} + h_j u_{j-1/2}^{(\nu)} \quad (7.3.24a)$$

$$(r_3)_{j-1} = u_{j-1}^{(\nu)} - u_j^{(\nu)} + h_j v_{j-1/2}^{(\nu)} \quad (7.3.24b)$$

$$(r_2)_j = R_{j-1/2}^{n-1} - \left[\begin{array}{l} h_j^{-1}(b_j^{(\nu)} v_j^{(\nu)} - b_{j-1}^{(\nu)} v_{j-1}^{(\nu)}) + \alpha_1(fv)_{j-1/2}^{(\nu)} \\ -\alpha_2(u^2)_{j-1/2}^{(\nu)} + \alpha^n(v_{j-1/2}^n f_{j-1/2}^{(\nu)} - f_{j-1/2}^n v_{j-1/2}^{(\nu)}) \end{array} \right] \quad (7.3.24c)$$

In writing the system given by Eqs. (7.3.23) we have used a certain order for them. The reason for this choice, as we shall see later, is to ensure that the A_0 matrix in Eq. (7.3.28a) is not singular.

The coefficients of the linearized momentum equation are

$$(s_1)_j = h_j^{-1} b_j^{(\nu)} + \frac{\alpha_1}{2} f_j^{(\nu)} - \frac{\alpha^n}{2} f_{j-1/2}^{n-1} \quad (7.3.25a)$$

$$(s_2)_j = -h_j^{-1} b_{j-1}^{(\nu)} + \frac{\alpha_1}{2} f_{j-1}^{(\nu)} - \frac{\alpha^n}{2} f_{j-1/2}^{n-1} \quad (7.3.25b)$$

$$(s_3)_j = \frac{\alpha_1}{2} v_j^{(\nu)} + \frac{\alpha^n}{2} v_{j-1/2}^{n-1} \quad (7.3.25c)$$

$$(s_4)_j = \frac{\alpha_1}{2} v_{j-1}^{(\nu)} + \frac{\alpha^n}{2} v_{j-1/2}^{n-1} \quad (7.3.25d)$$

$$(s_5)_j = -\alpha_2 u_j^{(\nu)} \quad (7.3.25e)$$

$$(s_6)_j = -\alpha_2 u_{j-1}^{(\nu)} \quad (7.3.25f)$$

The boundary conditions, Eq. (7.3.21) become

$$\delta f_0 = 0, \quad \delta u_0 = 0, \quad \delta u_J = 0 \quad (7.3.26)$$

As discussed in subsection 4.4.3, the linear system given by Eqs. (7.3.23) and (7.3.26) again has a block tridiagonal structure and can be written in matrix-vector form as given by Eq. (4.4.29) where now

$$\vec{\delta}_j = \begin{vmatrix} \delta f_j \\ \delta u_j \\ \delta v_j \end{vmatrix} \quad \vec{r}_j = \begin{vmatrix} (r_1)_j \\ (r_2)_j \\ (r_3)_j \end{vmatrix} \quad 0 \leq j \leq J \quad (7.3.27)$$

and A_j , B_j , C_j are 3×3 matrices defined as

$$A_0 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & -h_1/2 \end{vmatrix} \quad A_j \equiv \begin{vmatrix} 1 & -h_j/2 & 0 \\ (s_3)_j & (s_5)_j & (s_1)_j \\ 0 & -1 & -h_{j+1}/2 \end{vmatrix} \quad 1 \leq j \leq J-1 \quad (7.3.28a)$$

$$A_J \equiv \begin{vmatrix} 1 & -h_J/2 & 0 \\ (s_3)_J & (s_5)_J & (s_1)_J \\ 0 & 1 & 0 \end{vmatrix} \quad B_j \equiv \begin{vmatrix} -1 & -h_j/2 & 0 \\ (s_4)_j & (s_6)_j & (s_2)_j \\ 0 & 0 & 0 \end{vmatrix} \quad 1 \leq j \leq J \quad (7.3.28b)$$

$$C_j \equiv \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & -h_{j+1}/2 \end{vmatrix} \quad 0 \leq j \leq J-1 \quad (7.3.28c)$$

Note that the first two rows of A_0 and C_0 and the last row of A_J and B_J correspond to the boundary conditions [Eq. (7.3.26)]. To solve the continuity and momentum equations for different boundary conditions, only the matrix rows mentioned above need altering.

7.4 Computer Program BLP

This section describes a boundary layer program (BLP) for two-dimensional external flows in which the solutions of the continuity and momentum equations are obtained in terms of Falkner–Skan variables. The program, given in Appendix B, is appropriate to two-dimensional flows without separation and mass transfer but, with minor modifications, can be used for flows with mass transfer, axisymmetric flows, free shear flows, and flows with heat transfer as discussed in [2].

In BLP, the calculations start at the leading edge, $\xi = 0$, where the flow is laminar and becomes turbulent at any ξ -location by specifying the transition location. The solution procedure requires the specification of the dimensionless pressure gradient $m(\xi)$ which can be obtained from the specified dimensionless external velocity distribution $u_e(\xi)$.

BLP consists of a MAIN routine, which contains the logic of the computations, and subroutines: INPUT, IVPL, GROWTH, COEF3, SOLV3, OUTPUT and EDDY. The following subsections describe the function of each subroutine. A listing for each routine is given in Appendix B.

7.4.1 MAIN

BLP solves the linearized form of the equations. Thus an iteration procedure in which the solution of Eqs. (7.3.19) and (7.3.21) is obtained for successive estimates of the velocity profiles is needed with a subsequent need to check the convergence of the solutions. A convergence criterion based on v_0 which corresponds to f_w'' is usually used and the iterations, which are generally quadratic for laminar flows, are stopped when

$$|\delta v_0 (= \text{DELV}(1))| < \varepsilon_1 \quad (7.4.1)$$

with ε_1 taken as 10^{-5} . For turbulent flows, due to the approximate linearization procedure used for the turbulent diffusion term, the rate of convergence is not quadratic and solutions are usually acceptable when the ratio of $|\delta v_0/v_0|$ is less than 0.02. With proper linearization, quadratic convergence of the solutions can be obtained as described in [2].

After the convergence of the solutions, the OUTPUT subroutine is called and the profiles F, U, V and B, which represent the variables f_j , u_j , v_j and b_j are shifted.

7.4.2 Subroutine INPUT

The solution procedure requires the generation of a grid normal to the surface, η -grid, and along the surface, ξ -grid. The latter requirement is usually satisfied

by specifying locations with intervals which can be uniform or nonuniform. Its distribution depends on the variation of u_e with ξ so that the pressure gradient parameter $m(\xi)$ can be calculated accurately. To ensure this requirement, it is necessary to take small $\Delta\xi$ -steps (k_n) where there are rapid variations in $u_e(\xi)$ and where flow approaches separation.

For laminar flows, it is often sufficient to use a uniform grid in the η -direction. A choice of transformed boundary-layer thickness η_e equal to 8 often ensures that the dimensionless slope of the velocity profile at the edge, $f''(\eta_e)$, is sufficiently small ($< 10^{-3}$) and that approximately 41 j -points satisfies numerical accuracy requirements. For turbulent flows, however, a uniform grid is not satisfactory because the boundary-layer thickness η_e and dimensionless wall shear parameter f_w'' are much larger in turbulent flows than laminar flows. Since short steps in η must be taken to maintain computational accuracy when f_w'' is large, the steps near the wall in a turbulent boundary-layer must be shorter than the corresponding steps in a laminar boundary-layer under similar conditions.

A convenient and useful η -grid is a geometric progression having the property that the ratios of lengths of any two adjacent intervals is a constant; that is, $h_j = Kh_{j-1}$ [2]. The distance to the j -th line is given by the formula;

$$\eta_j = h_1 \frac{K^j - 1}{K - 1}, \quad j = 1, 2, \dots, J \quad K > 1 \quad (7.4.2)$$

There are two parameters: h_1 , the length of the first $\Delta\eta$ -step, and K , the ratio of two successive steps. The total number of points, J , can be calculated by the formula:

$$J = \frac{\ln[1 + (K - 1)(\eta_e/h_1)]}{\ln K} + 1 \quad (7.4.3)$$

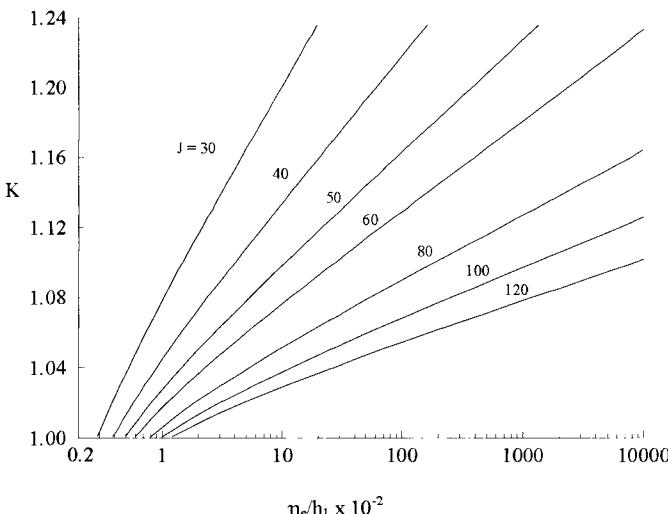


Fig. 7.4. Variation of K with h_1 for different η_e -values.

In most problems, calculations are performed by selecting h_1 and K and calculating the transformed boundary-layer thickness η_e . An idea about the number of points taken across the boundary-layer with the variable η -grid that uses those parameters for different η_e -values can be obtained from Fig. 7.4. For example, for $h_1 = 0.01$, $K = 1.10$ and $\eta_e = 100$, the ratio of η_e/h_1 is 10^4 , and the number of points across the boundary-layer is approximately 70. For a uniform spacing ($K = 1$) with $h_1 = 0.01$ and $\eta_e = 100$, there would be 10,000 points!

The calculation of m is achieved from the given external velocity distribution $u_e(\xi)$ and from the definition of m ($\equiv P2$) except for the first NX-station where P2(1) is read in. The derivative of $du_e/d\xi$ (DUDS) is obtained by using three-point Lagrange interpolation formulas given by ($n < N$):

$$\left(\frac{du_e}{d\xi} \right)_n = -\frac{u_e^{n-1}}{A_1}(\xi_{n+1} - \xi_n) + \frac{u_e^n}{A_2}(\xi_{n+1} - 2\xi_n + \xi_{n-1}) + \frac{u_e^{n+1}}{A_3}(\xi_n - \xi_{n-1}) \quad (7.4.4)$$

Here N refers to the last ξ^n station and

$$\begin{aligned} A_1 &= (\xi_n - \xi_{n-1})(\xi_{n+1} - \xi_{n-1}) \\ A_2 &= (\xi_n - \xi_{n-1})(\xi_{n+1} - \xi_n) \\ A_3 &= (\xi_{n+1} - \xi_n)(\xi_{n+1} - \xi_{n-1}) \end{aligned} \quad (7.4.5)$$

The derivative of $du_e/d\xi$ at the end point $n = N$ is given by

$$\left(\frac{du_e}{d\xi} \right)_N = -\frac{u_e^{N-2}}{A_1}(\xi_N - \xi_{N-1}) + \frac{u_e^{N-1}}{A_2}(\xi_N - \xi_{N-2}) + \frac{u_e^N}{A_3}(2\xi_N - \xi_{N-2} - \xi_{N-1}) \quad (7.4.6)$$

where now

$$\begin{aligned} A_1 &= (\xi_{N-1} - \xi_{N-2})(\xi_N - \xi_{N-2}) \\ A_2 &= (\xi_{N-1} - \xi_{N-2})(\xi_N - \xi_{N-1}) \\ A_3 &= (\xi_N - \xi_{N-1})(\xi_N - \xi_{N-2}) \end{aligned} \quad (7.4.7)$$

In this subroutine we generate the η -grid, calculate the pressure gradient parameters m and m_1 ($\equiv P1$) and specify η_e at $\xi = 0$ and the reference Reynolds number R_L (RL). In addition, the following data are read in and the total number of j -points J(NP) is computed from Eq. (7.4.3).

- NXT Total number of ξ -stations, not to exceed 60
- NTR NX-station for transition location ξ_{tr}
- NPT Total number of η -grid points.
- DETA(1) Δn -initial step size of the variable grid system. Use $\Delta n = 0.01$ for turbulent flows. If desired, it may be changed.
- ETAE Transformed boundary-layer thickness, η_e

VGP	K is the variable-grid parameter. Use $K = 1.0$ for laminar flow and $K = 1.14$ for turbulent flow. For a flow consisting of both laminar and turbulent flows, use $K = 1.14$.
RL	Reynolds number, $\frac{u_\infty L}{\nu}$
ξ	Surface distance, feet or meters, or dimensionless.
u_e	Velocity, feet per second or meter per second, or dimensionless.

7.4.3 Subroutine IVPL

At $\xi = 0$, Eq. (7.3.6) reduces to the Falkner-Skan equation, Eq. (7.3.11), which can be solved subject to the boundary conditions of Eq. (7.3.7). Since the equations are solved in linearized form, initial estimates of f_j , u_j and v_j are needed in order to obtain the solutions of the nonlinear Falkner-Skan equation. Various expressions can be used for this purpose. Since Newton's method is used, however, it is useful to provide as good an estimate as is possible and an expression of the form.

$$u_j = \frac{3}{2} \frac{\eta_j}{\eta_e} - \frac{1}{2} \left(\frac{\eta_j}{\eta_e} \right)^3 \quad (7.4.8)$$

usually satisfies this requirement. The above equation is obtained by assuming a third-order polynomial of the form

$$f' = a + b\eta + c\eta^3$$

and by determining constants a , b , c from the boundary conditions given by Eq. (7.3.7) for the zero-mass transfer case and from one of the properties of momentum equation which requires that $f'' = 0$ at $\eta = \eta_e$.

The other profiles f_j , v_j follow from Eq. (7.4.8) and can be written as

$$f_j = \frac{\eta_e}{4} \left(\frac{\eta_j}{\eta_e} \right)^2 \left[3 - \frac{1}{2} \left(\frac{\eta_j}{\eta_e} \right)^2 \right] \quad (7.4.9)$$

$$v_j = \frac{3}{2} \frac{1}{\eta_e} \left[1 - \left(\frac{\eta_j}{\eta_e} \right)^2 \right] \quad (7.4.10)$$

7.4.4 Subroutine GROWTH

For most laminar-boundary-layer flows the transformed boundary-layer thickness $\eta_e(x)$ is almost constant. A value of $\eta_e = 8$ is sufficient. However, for turbulent boundary-layers, $\eta_e(x)$ generally increases with increasing x . An estimate of $\eta_e(x)$ is determined by the following procedure.

We always require that $\eta_e(x^n) \geq \eta_e(x^{n-1})$, and in fact the calculations start with $\eta_e(0) = \eta_e(x_1)$. When the computations on $x = x^n$ (for any $n \geq 1$) have been completed, we test to see if $|v_j^n| \leq \varepsilon_v$ at $\eta_e(x^n)$ where, say $\varepsilon_v = 5 \times 10^{-4}$.

This test is done in MAIN. If this test is satisfied, we set $\eta_e(x^{n+1}) = \eta_e(x^n)$. Otherwise, we call GROWTH and set $J_{\text{new}} = J_{\text{old}} + t$, where t is a number of points, say $t = 1$. In this case we also specify values of $(f_j^n, u_j^n, v_j^n, b_j^n)$ for the new n_j points. We take the values of $u_j = 1$, $v_j^n = 0$, $f_j^n = (\eta_j - \eta_e)u_j^n + f_j^n$, and $b_j^n = b_j^n$. This is also done for the values of f_j^{n-1} , v_j^{n-1} , and b_j^{n-1} .

7.4.5 Subroutine COEF3

This is one of the most important subroutines of BLP. It defines the coefficients of the linearized momentum equation given by Eqs. (7.3.23) and (7.3.26).

7.4.6 Subroutine SOLV3

The solution of Eq. (4.4.29) by the block-elimination method discussed in subsection 4.4.3 can be obtained by using the recursion formulas given by Eqs. (4.4.32) and (4.4.34), and determining the expressions such as Δ_j , Γ_j , \vec{w}_j and $\vec{\delta}_j$. To describe the procedure let us first consider Eq. (4.4.32). Noting that the Γ_j matrix has the same structure as B_j and denoting the elements of γ_j , by γ_{ik} ($i, k = 1, 2, 3$), we can write Γ_j as

$$\Gamma_j \equiv \begin{vmatrix} (\gamma_{11})_j & (\gamma_{12})_j & (\gamma_{13})_j \\ (\gamma_{21})_j & (\gamma_{22})_j & (\gamma_{23})_j \\ 0 & 0 & 0 \end{vmatrix} \quad (7.4.11a)$$

Similarly, if the elements of Δ_j are denoted by α_{ik} we can write Δ_j as [note that the third row of Δ_j follows from the third row of A_j according to Eq. (4.4.32c)]

$$\Delta_j \equiv \begin{vmatrix} (\alpha_{11})_j & (\alpha_{12})_j & (\alpha_{13})_j \\ (\alpha_{21})_j & (\alpha_{22})_j & (\alpha_{23})_j \\ 0 & -1 & -h_{j+1}/2 \end{vmatrix} \quad 0 \leq j \leq J-1 \quad (7.4.11b)$$

and for $j = J$, the first two rows are the same as the first two rows in Eq. (4.4.32b), but the elements of the third row, which correspond to the boundary conditions at $j = J$, are $(0, 1, 0)$.

For $j = 0$, $\Delta_0 = A_0$; therefore the values of $(\alpha_{ik})_0$ are

$$\begin{aligned} (\alpha_{11})_0 &= 1 & (\alpha_{12})_0 &= 0 & (\alpha_{13})_0 &= 0 \\ (\alpha_{21})_0 &= 0 & (\alpha_{22})_0 &= 1 & (\alpha_{23})_0 &= 0 \end{aligned} \quad (7.4.12a)$$

and the values of $(\gamma_{ik})_1$ are

$$\begin{aligned} (\gamma_{11})_1 &= -1 & (\gamma_{12})_1 &= -\frac{1}{2}h_1 & (\gamma_{13})_1 &= 0 \\ (\gamma_{21})_1 &= (s_4)_1 & (\gamma_{23})_1 &= -2 \left[\frac{(s_2)_1}{h_1} \right] & (\gamma_{22})_1 &= (s_6)_1 + (\gamma_{23})_1 \end{aligned} \quad (7.4.12b)$$

The elements of the Δ_j matrices are calculated from Eq. (4.4.32a). Using the definitions of A_j , Γ_j and C_{j-1} , we find from Eq. (4.4.32c) that for $j = 1, 2, \dots, J$,

$$\begin{aligned} (\alpha_{11})_j &= 1 & (\alpha_{12})_j &= -\frac{h_j}{2} - (\gamma_{13})_j & (\alpha_{13})_j &= \frac{h_j}{2}(\gamma_{13})_j \\ (\alpha_{21})_j &= (s_3)_j & (\alpha_{22})_j &= (s_5)_j - (\gamma_{23})_j & (\alpha_{23})_j &= (s_1)_j + \frac{h_j}{2}(\gamma_{23})_j \end{aligned} \quad (7.4.13a)$$

To find the elements of the Γ_j matrices, we use Eq. (4.4.32b). With Δ_j defined by Eq. (4.4.32c) and B_j by Eq. (4.4.31b), it follows that for $1 \leq j \leq J$,

$$(\gamma_{11})_j = \left\{ (\alpha_{23})_{j-1} + \frac{h_j}{2} \left[\left(\frac{h_j}{2} \right) (\alpha_{21})_{j-1} - (\alpha_{22})_{j-1} \right] \right\} / \Delta_0$$

$$(\gamma_{12})_j = - \left\{ \frac{h_j}{2} \frac{h_j}{2} + (\gamma_{11})_j \left[(\alpha_{12})_{j-1} \frac{h_j}{2} - (\alpha_{13})_{j-1} \right] \right\} / \Delta_0$$

$$(\gamma_{13})_j = [(\gamma_{11})_j(\alpha_{13})_{j-1} + (\gamma_{12})(\alpha_{23})_{j-1}] / \frac{h_j}{2}$$

$$\begin{aligned} (\gamma_{21})_j &= \left\{ (s_2)_j(\alpha_{21})_{j-1} - (s_4)_j(\alpha_{23})_{j-1} \right. \\ &\quad \left. + \frac{h_j}{2}[(s_4)_j(a_{22})_{j-1} - (s_6)_j(\alpha_{21})_{j-1}] \right\} / \Delta_0 \end{aligned}$$

$$(\gamma_{22})_j = \left\{ (s_6)_j \frac{h_j}{2} - (s_2)_j + (\gamma_{21})_j \left[(\alpha_{13})_{j-1} - (\alpha_{12})_{j-1} \frac{h_j}{2} \right] \right\} / \Delta_1 \quad (7.4.13b)$$

$$(\gamma_{23})_j = (\gamma_{21})_j(\alpha_{12})_{j-1} + (\gamma_{22})_j(\alpha_{22})_{j-1} - (s_6)_j$$

$$\begin{aligned} \Delta_0 &= (\alpha_{13})_{j-1}(\alpha_{21})_{j-1} - (\alpha_{23})_{j-1}(\alpha_{11})_{j-1} \\ &\quad - \frac{h_j}{2}[(\alpha_{12})_{j-1}(\alpha_{21})_{j-1} - (\alpha_{22})_{j-1}(\alpha_{11})_{j-1}] \end{aligned}$$

$$\Delta_1 = (\alpha_{22})_{j-1} \frac{h_j}{2} - (\alpha_{23})_{j-1}$$

To summarize the calculation of Γ_j and Δ_j matrices, we first calculate α_{ik} from Eq. (7.4.11b) for $j = 0$, γ_{ik} from Eq. (7.4.12b) for $j = 1$, α_{ik} from Eq. (7.4.13a) for $j = 1$, then γ_{ik} from Eq. (7.4.13b) for $j = 2$, α_{ik} from Eq. (7.4.13a) for $j = 2$, then γ_{ik} from Eq. (7.4.13b) for $j = 3$, etc.

In the second part of the forward sweep we compute \vec{w}_j from the relations given by Eq. (4.4.33). If we denote the components of the vector \vec{w}_j by

$$\vec{w}_j \equiv \begin{vmatrix} (w_1)_j \\ (w_2)_j \\ (w_3)_j \end{vmatrix} \quad 0 \leq j \leq J \quad (7.4.14)$$

Then it follows from Eq. (4.4.33a) that for $j = 0$,

$$(w_1)_0 = (r_1)_0 \quad (w_2)_0 = (r_2)_0 \quad (w_3)_0 = (r_3)_0 \quad (7.4.15a)$$

and from Eq. (4.4.33b) for $1 \leq j \leq J$,

$$\begin{aligned}(w_1)_j &= (r_1)_j - (\gamma_{11})_j(w_1)_{j-1} - (\gamma_{12})_j(w_2)_{j-1} - (\gamma_{13})_j(w_3)_{j-1} \\(w_2)_j &= (r_2)_j - (\gamma_{21})_j(w_1)_{j-1} - (\gamma_{22})_j(w_2)_{j-1} - (\gamma_{23})_j(w_3)_{j-1} \\(w_3)_j &= (r_3)_j\end{aligned}\quad (7.4.15b)$$

In the backward sweep, $\vec{\delta}_j$ is computed from the formulas given by Eq. (4.4.34). With the definitions of $\vec{\delta}_j$, Δ_j and \vec{w}_j , it follows from Eq. (4.4.33a) that

$$\delta u_J = (w_3)_J \quad (7.4.16a)$$

$$\delta v_J = \frac{e_2(\alpha_{11})_J - e_1(\alpha_{21})_J}{(\alpha_{23})_J(\alpha_{11})_J - (\alpha_{13})_J(\alpha_{21})_J} \quad (7.4.16b)$$

$$\delta f_J = \frac{e_1 - (\alpha_{13})_J \delta v_J}{(\alpha_{11})_J} \quad (7.4.16c)$$

where

$$e_1 = (w_1)_J - (\alpha_{12})_J \delta u_J$$

$$e_2 = (w_2)_J - (\alpha_{22})_J \delta u_J$$

The components of $\vec{\delta}$, for $j = J-1, J-2, \dots, 0$, follow from Eq. (4.4.33b)

$$\delta v_j = \frac{(\alpha_{11})_j[(w_2)_j + e_3(\alpha_{22})_j] - (\alpha_{21})_j(w_1)_j - e_3(\alpha_{21})_j(\alpha_{12})_j}{\Delta_2} \quad (7.4.17a)$$

$$\delta u_j = -\frac{h_{j+1}}{2} \delta v_j - e_3 \quad (7.4.17b)$$

$$\delta f_j = \frac{(w_1)_j - (\alpha_{12})_j \delta u_j - (\alpha_{13})_j \delta v_j}{(\alpha_{11})_j} \quad (7.4.17c)$$

where

$$\begin{aligned}e_3 &= (w_3)_j - \delta u_{j+1} + \frac{h_{j+1}}{2} \delta v_{j+1} \\ \Delta_2 &= (\alpha_{21})_j(\alpha_{12})_j \frac{h_{j+1}}{2} - (\alpha_{21})_j(\alpha_{13})_j \\ &\quad - \frac{h_{j+1}}{2} (\alpha_{22})_j(\alpha_{11})_j + (\alpha_{23})_j(\alpha_{11})_j\end{aligned}\quad (7.4.17d)$$

To summarize, one iteration of Newton's method is carried out as follows. The vectors \vec{r}_j defined in Eq. (4.4.30) are computed from Eq. (7.3.24) by using the latest iterate. The matrix elements of A_j , B_j and C_j defined in Eq. (7.3.28) are next determined by Eq. (7.3.25a) to (7.3.25f). Using the relations in Eqs. (4.4.32) and (4.4.33), the matrices Γ_j and Δ_j and vectors \vec{w}_j are calculated. The matrix elements for Γ_j defined in Eq. (4.4.32a) are determined from Eq. (7.4.12). The components of the vector \vec{w}_j defined in Eq. (7.4.14) are determined from Eq. (7.4.15). In the backward sweep, the components of $\vec{\delta}_j$ are computed from Eqs. (7.4.16) and (7.4.17). A subroutine which makes use of these formulas and called SOLV3 is given in Appendix B.

7.4.7 Subroutine OUTPUT

This subroutine prints out the desired profiles such as f_j , u_j , v_j , and b_j as functions of η_j . It also computes the boundary-layer parameters, c_f , δ^* , θ and R_x defined by

$$c_f = \frac{2\tau_w}{\varrho u_e^2} = \frac{2f''_w}{\sqrt{R_x}} \quad (7.4.18a)$$

$$\delta^* = \int_0^\delta \left(1 - \frac{u}{u_e}\right) dy = \frac{x\delta_1^*}{\sqrt{R_x}} \quad (7.4.18b)$$

$$\theta = \int_0^\delta \frac{u}{u_e} \left(1 - \frac{u}{u_e}\right) dy = \frac{x\theta_1}{\sqrt{R_x}} \quad (7.4.18c)$$

$$R_x = \frac{u_e x}{\nu} \quad (7.4.18d)$$

where

$$\delta_1^* = \int_0^{\eta_e} (1 - f') d\eta = \eta_e + f(x, \eta_e) \quad (7.4.19a)$$

$$\theta_1 = \int_0^{\eta_e} f'(1 - f') d\eta \quad (7.4.19b)$$

7.4.8 Subroutine EDDY

For simplicity we use the eddy-viscosity formulation of Cebeci and Smith described in subsection 3.2.1 without the strong pressure gradient effect, that is, $\alpha = 0.0168$ and mass transfer effect. These capabilities, if desired, can easily be incorporated into the formulas defined in the subroutine. In terms of transformed variables defined by Eqs. (7.3.4) and (7.3.5). Eqs. (3.2.1) and (3.2.2) are written as

$$\varepsilon_m^+ = \frac{\varepsilon_m}{\nu} = 0.16 R_x^{1/2} \left[1 - \exp\left(-\frac{y}{A}\right)\right]^2 \eta^2 v \gamma_{\text{tr}} \quad (7.4.20)$$

$$\frac{y}{A} = \frac{N}{26} R_x^{1/4} v_w^{1/2} \eta, \quad p^+ = m R_x^{1/4} (v_w)^{-3/2}, \quad N = (1 - 11.8 p^+)^{1/2}, \quad R_x = \frac{u_e x}{\nu} \quad (7.4.21)$$

Similarly Eq. (3.2.3), with α taken as constant ($\equiv 0.0168$), is written as

$$\varepsilon_m^+ = 0.0168 R_x^{1/2} (\eta_e - f_e) \gamma_{\text{tr}} \gamma \quad (7.4.22)$$

7.5 Applications of BLP

In this section we discuss the applications of the computer program (BLP) of the previous section to a sample of laminar and turbulent flows without separation. These flows deal with momentum transfer in external flows.

7.5.1 Similar Laminar Flows

A certain class of external laminar flows admit similarity solutions, as discussed in Section 7.3, for external velocity, u_e , of the form given by Eq. (7.3.10) and with boundary conditions in transformed variables independent of the downstream distance x . The solutions of the continuity and momentum equations can be obtained by solving the Falkner–Skan equation (7.3.11) subject to the boundary conditions given by Eq. (7.3.7).

To demonstrate the application of BLP, we first consider the case of flow with no mass transfer so that $f_w = 0$, and solve Eq. (7.3.11) for values of m of 1, 1/3, 0, -0.075 and -0.0904. The calculations can be carried out at one ξ -station ($NX = 1$) for each value of m or performed for all the values of m . In the former case, it is necessary to set the value of m at $\xi = 0$, that is, P2(1) in subroutine INPUT. In the latter, P2(NX) is set to require m values at five ξ -stations, including P2(1), rather than calculating them from the input values of ξ and u_e . With NXT = 5, X(I), UE(I) and RL can be assigned any value since the Falkner–Skan equation is independent of these parameters. The parameter α_n (\equiv CEL) [subroutine COEF] is set to zero at all ξ -stations.

The second choice is preferable since the results are obtained from a multi-step calculation, and better convergence is achieved since initial estimates are provided by the results from the previous ξ -station.

The calculated velocity profiles f' ($\equiv u/u_e$) as a function of the similarity variable η [$\equiv (u_e/\nu x)^{1/2}y$] are shown in Fig. 7.5. Of particular note are the m -values of unity and zero corresponding to stagnation and flat-plate flows, $m = 1/3$ corresponding to the stagnation flow on an axisymmetric body, and $m = -0.0904$ corresponding to the strongest adverse pressure gradient for which the flow remains attached with $f''(0)$ equal to zero. These calculations were performed with a uniform grid ($K = 1$) and $h_1 = 0.10$.

To demonstrate the application of BLP to a similar laminar flow with mass transfer, we consider a flat-plate flow with specified values of f_w . It is seen from Eq. (7.3.7a) that for similarity, the mass transfer velocity v_w must vary as $1/\sqrt{x}$. To account for the nonzero value of f_w , one must make changes in the IVPL subroutine since Eqs. (7.4.4) are for zero mass transfer. By rewriting Eq. (7.4.4b) as

$$f_j = f_w + \frac{\eta_e}{4} \left(\frac{\eta_j}{\eta_e} \right)^2 \left[3 - \frac{1}{2} \left(\frac{\eta_j}{\eta_e} \right)^2 \right] \quad (7.5.1)$$

the mass transfer effects can be included in the solution of the Falkner–Skan equation. As in the case of calculations for different values of m , the calculations can also be performed with the two choices described above for flows with $m = 0$ for all ξ -stations without mass transfer and, in this case, it is more convenient to perform them with the first choice.

Figure 7.6 shows the calculated velocity profiles as a function of η for several values of mass transfer parameter f_w with positive values corresponding to

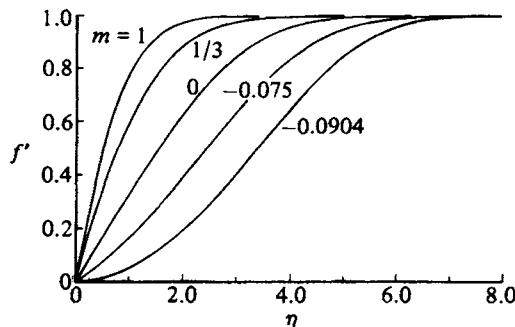


Fig. 7.5. The dimensionless velocity f' as a function of η for various m .

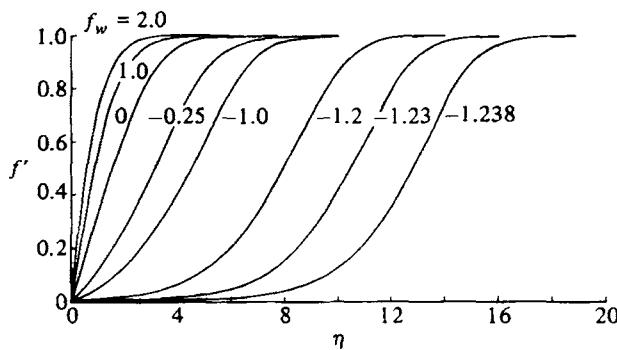


Fig. 7.6. The effect of mass transfer parameter f_w on velocity profiles for $m = 0$.

suction and negative to blowing for zero pressure gradient. The effect of f_w is similar to that of the pressure-gradient parameter so that an increase in blowing rate, like an increase in a positive (“adverse”) pressure gradient, decreases the slope of the velocity profile near the wall until it becomes zero at $f_w = -1.238$. With an increase in suction, as with an increase in a “favorable” (negative) pressure gradient, the slope of the velocity profile increases. It is also seen from the calculations that with an increase in suction, the effective transformed boundary layer thickness becomes smaller, and with an increase in blowing, it becomes bigger.

7.5.2 Nonsimilar Flows

To illustrate the calculation of nonsimilar flows with BLP, we consider a laminar and turbulent flow over an ellipse and a NACA 0012 airfoil.

Calculations for an Ellipse

Consider an ellipse at zero angle of attack with its center at $(a, 0)$, that is,

$$\frac{(x-a)^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (7.5.2a)$$

Inviscid-flow theory provides the external velocity distribution in the form

$$u_e(s) = u_\infty(1+t) \cos \beta \quad (7.5.2b)$$

where s is the surface distance, t the thickness ratio of the ellipse ($\equiv b/a$) and β the angle between the line tangent to the body and the positive x -axis, that is,

$$\beta = \tan^{-1} \frac{dy}{dx} \quad (7.5.3)$$

Since the computer program requires the specification of the dimensionless external velocity distribution as a function of dimensionless surface distance s/L , the following expression is used to compute the dimensionless distance of a two-dimensional body in which the dimensionless coordinates (x/L) and (y/L) are given in tabular form

$$\left(\frac{s}{L}\right)_i = \left(\frac{s}{L}\right)_{i-1} + \sqrt{\left[\left(\frac{x}{L}\right)_i - \left(\frac{x}{L}\right)_{i-1}\right]^2 + \left[\left(\frac{y}{L}\right)_i - \left(\frac{y}{L}\right)_{i-1}\right]^2} \quad (7.5.4)$$

Here $(s/L)_i$ denotes the distance at a given surface location, $(x/L)_i$, $(y/L)_i$ ($i = 1, 2, \dots, I$) with the surface distance calculations starting from the stagnation point.

If we shift the center of the ellipse to $x/2a = 0.5$ so that the reference length $L = 2a$, then in an interval $0 \leq x/2a \leq 1$, we can specify the total number of ξ -stations to be, for example, 81 corresponding to a uniform $\Delta\xi$ -step length of 0.0125. Next $y/2a$ is computed from Eq. (7.5.2a) for each value of $x/2a$ specified at 1/8 intervals, and the distance is calculated from Eq. (7.5.4) and the dimensionless external velocity u_e/u_∞ obtained from Eq. (7.5.2b) for a specified value of thickness ratio t .

The computer program also requires the specification of the onset of transition, which can be either input or computed. For this flow, it is specified at different $x/2a$ locations depending on the Reynolds number, which must also be specified in the calculations.

Figure 7.7 shows the external velocity distribution over the ellipse for a thickness ratio 1 to 4, and Fig. 7.8 shows the variation of the wall shear parameter f_w'' ($\equiv c_f/2\sqrt{R_x}$) with $x/2a$ for specified transition locations of $(x/2a)_{tr} = 0.658$ and 0.784 at Reynolds numbers R_{2a} ($\equiv u_\infty 2a/\nu$) of 10^7 and 10^6 , respectively. It is seen from Fig. 7.7 that the flow starts as a stagnation point flow ($m = 1$) with $f_w'' = 1.23259$ and is relatively constant in the region where the flow accelerates. If the slope of the u_e and s curve in Fig. 7.7 were constant, then the wall shear parameter f_w'' would have been constant and equal to its value at $m = 1$. Thus, as m decreases from its value at the stagnation point, it reflects the decrease in flow acceleration which occurs around $x/2a = 0.20$ (see Fig. 7.7).

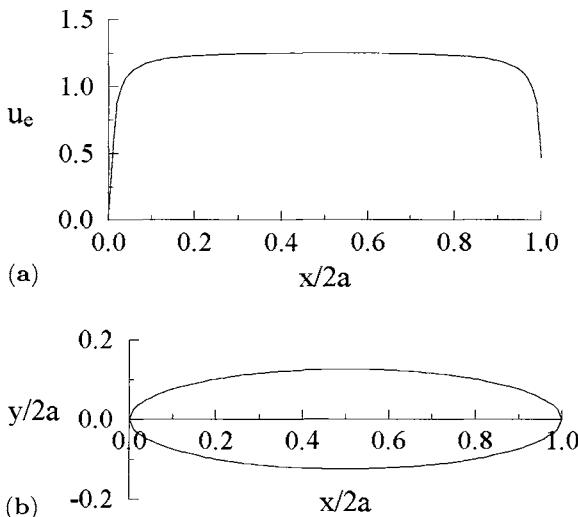


Fig. 7.7. (a) Inviscid velocity distribution about (b) an ellipse of thickness ratio 0.25.

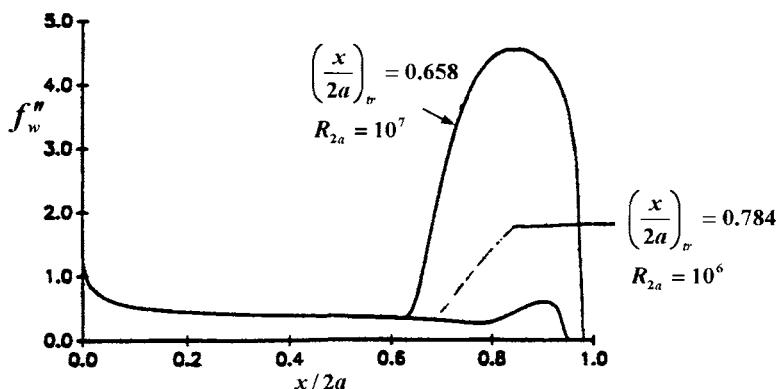


Fig. 7.8. Variation of the wall shear parameter f_w'' over the ellipse at two Reynolds numbers.

With flow deceleration, f_w'' begins to decrease, which can lead to flow separation at some ξ -location if transition does not occur. The calculations in Fig. 7.8 show that for $R_{2a} = 10^6$, laminar separation takes place at $x/2a = 0.784$. If the flow calculations were not performed for turbulent flow at this location, they would have been terminated due to the singular nature of the boundary-layer equations for a specified external velocity distribution. With transition specified at the location of laminar separation, calculations can then be performed with wall shear parameter increasing with increasing $x/2a$, reaching a maximum around $x/2a = 0.90$ and then beginning to decrease quickly with turbulent-flow separation taking place at $x/2a = 0.91$.

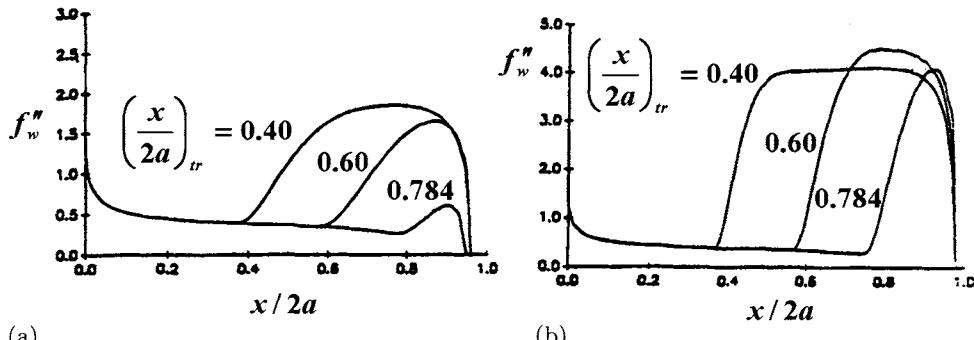


Fig. 7.9. Effect of transition on all shear parameter f_w'' . (a) $R_{2a} = 10^6$ and (b) $R_{2a} = 10^7$.

The flow with $R_{2a} = 10^7$, which would otherwise separate at $x/2a = 0.784$ (because laminar separation location is independent of Reynolds number for a given pressure distribution), becomes turbulent at $x/2a = 0.658$. The wall shear parameter increases sharply with $x/2a$, reaching a maximum at $x/2a = 0.85$, and then begins to decrease, becoming zero and thus indicating turbulent-flow separation at $x/2a = 0.98$. Note that the flow separation at the lower Reynolds number takes place early because the boundary layer is thicker, principally due to the greater growth rate in the laminar region.

From these calculations we see that increasing Reynolds numbers moves the transition location forward and delays the turbulent-flow separation.

Figure 7.9 shows the effect of transition on f_w'' for two Reynolds numbers. We see from Fig. 7.9a that for $R_{2a} = 10^6$, f_w'' increases with decreases in the x -location of transition in the range of $x/2a$ from 0.784 to 0.40. The separation location is, however, nearly unaffected. The results in Fig. 7.9b indicate similar increases in wall shear for $R_{2a} = 10^7$ as the transition location is moved from 0.784 to 0.40. However, this also shows that the flow separation is delayed with the increase in Reynolds number.

Calculations for a NACA 0012 Airfoil

The calculations for a NACA 0012 airfoil are performed at several angles of attack. As in subsection 6.6.1, the inviscid velocity distribution for each α is obtained from the panel program of Section 6.4 with the x/c and y/c coordinates of the airfoil given in Appendix B, Chapter 6.

For this flow, the onset of transition on the airfoil is also calculated. While the e^n -method discussed in Chapter 8 is an accurate method for this, a correlation formula from Michel [4] is used due to its simplicity and ease of use. This formula, which is given by

$$R_{\theta_{tr}} = 1.174 \left(1 + \frac{22.400}{R_{x_{tr}}} \right) R_{x_{tr}}^{0.46} \quad (7.5.5)$$

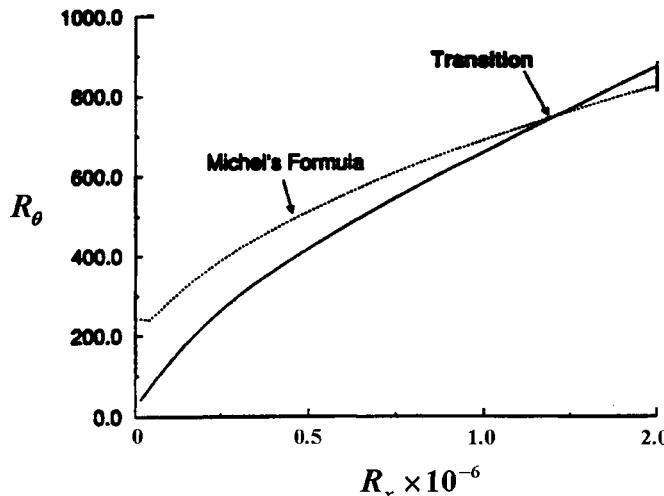


Fig. 7.10. Prediction of transition from Eq. (7.5.5) on the NACA 0012 airfoil; $\alpha = 0^\circ$, $R_c = 3 \times 10^6$. Solid line denotes R_θ obtained from boundary-layer calculations.

is valid for unseparated flows with chord Reynolds numbers, R_c , greater than around 1×10^6 . According to this equation, the development of R_θ ($\equiv u_e \theta / \nu$) is computed as a function of R_x ($\equiv u_e x / \nu$), and transition is determined from the values of R_θ and R_x that satisfy Eq. (7.5.5).

With increasing incidence angle, the location of onset of transition moves upstream on the upper surface of the airfoil and moves downstream on the lower surface. At higher angles of attack, the transition location on the upper surface generally occurs almost at the pressure peak (maximum velocity) and close to the trailing edge on the lower surface. In the former case, it is sufficient to take the pressure peak to be the transition location rather than compute it since the Reynolds number is rather low. In the latter case, the pressure gradient is favorable (accelerating flow) and it is sufficient to take transition to be either at the trailing edge or close to it.

Sometimes, before the onset of transition is computed with Eq. (7.5.5), laminar separation may occur. In that case, it is sufficient to take the laminar separation point to be the transition point, since in high Reynolds number flows transition takes place before laminar separation. This, however, is not the case for low Reynolds number flows when transition can occur after the laminar separation. Its calculation is not appropriate with Eq. (7.5.5), as discussed in detail in [4].

Figures 7.10 and 7.11 show the results from the boundary-layer calculations performed for a total number of approximately 50 ξ -stations on each surface at two Reynolds numbers. At first calculations were carried out for laminar flow in order to determine the onset of transition from Eq. (7.5.5). Figure 7.10 shows the development of R_θ as a function of R_x at $\alpha = 0^\circ$ for a chord Reynolds number

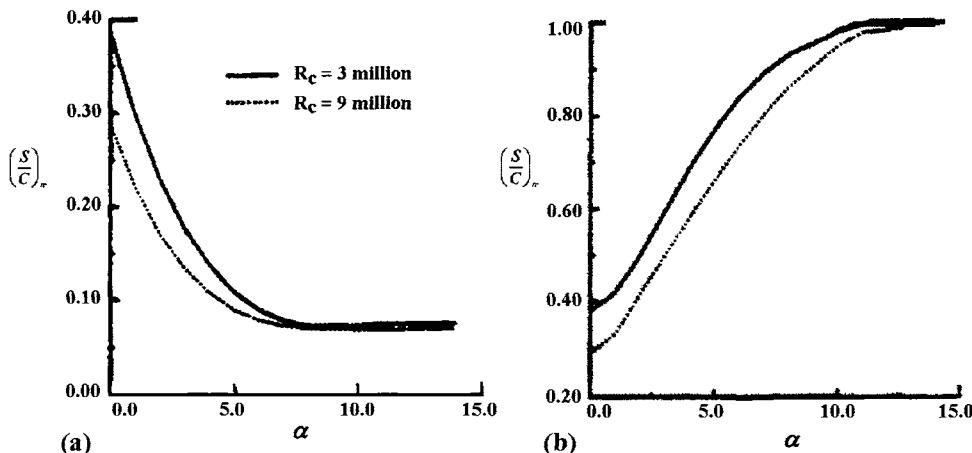


Fig. 7.11. Effect of Reynolds number and angle of attack on the location of transition on the (a) upper, and (b) lower surfaces of the NACA 0012 airfoil.

of 3×10^6 , indicating that the computed values intersect the curve represented by Eq. (7.5.5) at $R_{x_{tr}} = 1.20 \times 10^6$. Note that before transition, the (R_θ, R_x) values are under the curve given by Eq. (7.5.5).

Figure 7.11a shows the variation of the onset of transition, $(s/c)_{tr}$, with incidence angle α on the upper surface for both Reynolds numbers. For values of α up to 6° , the onset of transition was calculated from Eq. (7.5.5); at higher incidence angles, transition location was assumed to be at the pressure peak. On the lower surface, Fig. 7.11b, the onset of transition occurs at $(s/c)_{tr} = 0.28$ at $\alpha = 0^\circ$ for $R_c = 3 \times 10^6$, moves downstream with increasing α and can be calculated with Eq. (7.5.5) for values of α up to 9° . At higher angles, the onset of transition was assumed to be at the trailing edge.

The computed transition locations in Fig. 7.11 confirm the influence of Reynolds number: at the higher Reynolds number, the onset of transition occurs earlier than that computed at the lower Reynolds number.

Figure 7.12 shows the variation of the local skin-friction coefficient, c_f , and displacement thickness, δ^* , for both laminar and turbulent flows as a function of angle of attack on the upper surface of the airfoil for $R_c = 3 \times 10^6$. Whereas there is no flow separation at lower angles of attack, turbulent trailing-edge separation takes place at $\alpha = 6^\circ$ and moves upstream. Since the solution procedure employs the standard method, the calculations are terminated at the location of flow separation.

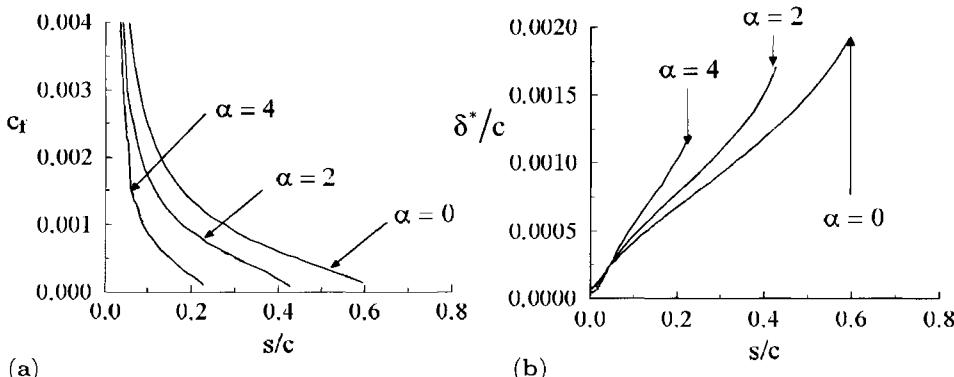


Fig. 7.12. Variation of (a) local skin-friction coefficient c_f and (b) displacement thickness δ^*/c with angle of incidence on the upper surface of the NACA 0012 airfoil for $R_c = 3 \times 10^6$.

References

- [1] Cebeci, T.: *Turbulence Models and Their Application*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 2004.
- [2] Cebeci, T.: *Analysis of Turbulent Flows*, Elsevier, London, 2004.
- [3] Goldstein, S.: "On Laminar Boundary Layer Flow near a Point of Separation," *Quart. J. Mech. & Appl. Math.*, Vol. 1, pp. 43–69, 1948.
- [4] Cebeci, T.: *An Engineering Approach to the Calculation of Aerodynamic Flows*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 1999.
- [5] Catherall, D. and Mangler, K. W.: "The Interpretation of Two-Dimensional Laminar Boundary Layer Equations Past the Point of Vanishing Skin Friction," *J. Fluid Mech.*, Vol. 26, p. 163, 1966.
- [6] Briley, W. R.: "A Numerical Study of Laminar Separation Bubbles Using the Navier-Stokes Equations," *J. Fluid Mech.*, Vol. 47, no. 4, pp. 713–736, 1971.
- [7] LeBalleur, J. C.: "Couplage Visqueux-Non Visqueux: Méthode Numérique et Applications Aux Ecoulements Bidimensionnels Transsoniques et Supersoniques," La Recherche Aérospatiale, March-April 1978.
- [8] Carter, J. and Wornom, S. F.: "Solutions for Incompressible Separated Boundary Layers Including Viscous-Inviscid Interaction," In: *Aerodynamic Analysis Requiring Advanced Computers*, NASA SP-347, p. 125, 1975.
- [9] Veldman, A. E. P.: "A Numerical Method for the Calculation of Laminar Incompressible Boundary Layers with Strong Inviscid Interaction," NLRTR 79023L, 1979.
- [10] Smith, A. M. O.: "Improved Solutions of the Falkner and Skan Boundary-Layer Equations," Sherman M. Fairchild Fund paper FF-10, Inst. Aero. Sci., 1954.
- [11] Iglish, R.: "Exact Calculation of Laminar Boundary Layer in Longitudinal Flow Over a Flat Plate with Homogeneous Suction," NACA Tech. Memo. 1205, 1949.

Problems

- 7-1.** Using the computer program of Section 7.4, study the numerical accuracy of the box method for Falkner-Skan flows $m = 1$ and 0 . Compare the wall

shear stress parameter f_w'' with those given by Smith [10] which are accurate to six decimal places. Use four uniform spacings of ($h = 0.8, 0.4, 0.2, 0.1$) which correspond, respectively, to 11, 21, 41 and 81 points for a transformed boundary-layer thickness of 8.

Wall shear values of Smith [10]

m	f_w''
1	1.232588
0	0.332000

7-2. The accuracy of the Box scheme, which is of second-order, can be improved by using Richardson extrapolation as discussed, for example, in [10]. The purpose of attempting to improve accuracy is to get reasonable answers from a coarse mesh of net points rather than to acquire more significant digits in the solution. With few net point, the iterations converge more rapidly and the computations are generally more efficient.

To describe the application of the Richardson extrapolation, consider the numerical solution of the momentum equation (7.3.6) subject to boundary conditions given by Eqs. (7.3.7) and denote the solution $[f_j^n(r), u_j^n(r), v_j^n(r)]$ on a net of spacings $h^{(r)}$ and $k^{(r)}$. Now let two independent computations be made on net with spacings $(h, k) = [h^{(0)}, k]$ and $(h, k) = [h^{(1)}, k]$. Then we form

$$f_j^n[h^{(0)}, h^{(1)}; k] \equiv \frac{h^{(0)2} f_j^n[h^{(1)}; k] - h^{(1)2} f_j^n[h^{(0)}; k]}{h^{(0)2} - h^{(1)2}} \quad (\text{P7.2.1a})$$

to obtain solutions that are of fourth-order accuracy in h , $O(h^4)$. Similar results also hold for u_j^n, v_j^n . An even more accurate approximation can be obtained, regarding the order of errors in h , if calculations are made by a third admissible net with $(h, k) = [h^{(2)}, k]$. We now also compute $f_j^n[h^{(1)}, h^{(2)}, k]$ in obvious analog with Eq. (P7.2.1a) and form

$$f_j^n[h^{(0)}, h^{(1)}, h^{(2)}; k] \equiv \frac{h^{(0)2} f_j^n[h^{(1)}, h^{(2)}; k] - h^{(2)2} f_j^n[h^{(0)}, h^{(1)}; k]}{h^{(0)2} - h^{(2)2}} \quad (\text{P7.2.1b})$$

to obtain solutions that are of sixth-order accuracy in h , $O(h^6)$.

Clearly, the procedure could be reversed and only the ξ spacing defined in an admissible way. More important, the Richardson extrapolation can be used to get, say, $O(h^6 + k^6)$ accuracy, because then

$$\frac{h^{(1)2}}{h^{(0)2} - h^{(1)2}} = \frac{k^{(1)2}}{k^{(0)2} - k^{(1)2}}, \quad \text{etc.} \quad (\text{P7.2.2})$$

Even higher-order accuracy can be obtained by additional computations and extrapolations.

To demonstrate the power of Richardson extrapolation, consider the solutions of Eq. (7.3.11) and compare the computed f_w'' values with $h = 0.1$ with those obtained by Richardson extrapolation. Use three different uniform spacings, $h^{(0)} = 0.8$, $h^{(1)} = 0.4$, and $h^{(2)} = 0.2$. Perform extrapolation with $f_w''[h^{(0)}]$ and $f_w''[h^{(1)}]$ to get the results for $f_w''[h^{(0)}, h^{(1)}]$; with $f_w''[h^{(1)}]$ and $f_w''[h^{(2)}]$ for $f_w''[h^{(1)}, h^{(2)}]$; and with $f_w''[h^{(0)}, h^{(1)}]$ and $f_w''[h^{(1)}, h^{(2)}]$ for $f_w''[h^{(0)}, h^{(1)}, h^{(2)}]$. Explicitly, in order to determine $f_w''[h^{(0)}, h^{(1)}]$, use Eq. (P7.2.1a) to get

$$f_w''[h^{(0)}, h^{(1)}] = \frac{h^{(0)2} f_w''[h^{(1)}] - h^{(1)2} f_w''[h^{(0)}]}{h^{(0)2} - h^{(1)2}} \quad (\text{P7.2.3a})$$

Since $h^{(1)} = \frac{1}{2}h^{(0)}$ in this case, Eq. (P7.2.3a) can also be written as

$$f_w''[h^{(0)}, h^{(1)}] = \frac{4}{3} f_w''[h^{(1)}] - \frac{1}{3} f_w''[h^{(0)}] \quad (\text{P7.2.3b})$$

7-3. Compute the boundary-layer development of a laminar flow over a circular cylinder of radius r_0 placed normal to the freestream velocity u_∞ . Take the inviscid velocity distribution with $\xi = x/r_0$ as

$$u_e = 2u_\infty \sin \xi$$

with 2, 4, 6, 8 degree intervals in ξ . Plot $\tau_w/\varrho u_\infty^2 (u_\infty r_0/\nu)^{1/2}$ as a function of ξ . Identify the location of flow separation. Note that from the definition of m ,

$$m = \frac{\xi \cos \xi}{\sin \xi}$$

which indicates that the flow starts as a stagnation point flow (why?).

Take $h = 0.2$.

7-4. Apply the Richardson extrapolation to compute f_w'' values obtained in Problem 7.3 for streamwise spacings, $k^{(0)} = 8^\circ$, $k^{(1)} = 6^\circ$ and $k^{(2)} = 4^\circ$ and compare the results with those corresponding to $k_n = 2^\circ$. Take $h = 0.2$ in all cases.

7-5. Consider a laminar flow past a flat plate with uniform suction. The boundary conditions follow from Eq. (7.3.1) and can be written as

$$y = 0, \quad u = 0, \quad v = v_w = \text{const} \quad (\text{P7.5.1a})$$

$$y = \delta, \quad u = u_e = u_\infty \quad (\text{P7.5.1b})$$

At a certain distance from the leading edge, the boundary-layer thickness δ , which in general is a function of x , becomes constant and stays constant with increasing x . As a result the streamwise velocity component u varies only with

y , that is, $\partial u / \partial x = 0$. The momentum equation becomes an ordinary differential equation and can be written as

$$v_w \frac{du}{dy} = \nu \frac{d^2 u}{dy^2} \quad (\text{P7.5.2})$$

since from the continuity equation $v(x, y) = v_w = \text{const.}$

(a) Show that the solution of Eq. (7.5.2) is

$$\frac{u(y)}{u_\infty} = 1 - \exp(v_w y / \nu) \quad (\text{P7.5.3})$$

This velocity profile is known as the asymptotic suction profile.

(b) Using Eq. (7.5.3), show that

$$\delta^* = -\frac{\nu}{v_w}, \quad \theta = -\frac{1}{2} \frac{\nu}{v_w}, \quad \tau(0) = \tau_w = -\varrho v_w u_\infty \quad (\text{P7.5.4})$$

7-6. The idealized flow discussed in P7.5, does not exist close to the leading edge of the plate, even though uniform suction starts at the leading edge. After a certain distance, x downstream, however, the asymptotic suction will materialize.

The distance x from the leading edge necessary to have the asymptotic-suction flow has been determined by Iglisch [11] to be

$$-\frac{v_w}{u_\infty} \left(\frac{u_\infty x}{\nu} \right)^{1/2} = -\frac{v_w}{u_\infty} (R_L \xi)^{1/2} - 2 \quad (\text{P7.6.1})$$

(a) Use the computer program of Section 7.4 and with appropriate changes examine the way in which the boundary-layer solutions approach the asymptotic values of

$$H = 2, \quad \tau_w / \varrho u_\infty^2 = -\frac{v_w}{u_\infty} \quad (\text{P7.6.2})$$

(b) Compare the calculated wall-shear parameter f_w'' with those given by Iglisch for the region before the asymptotic suction profile is reached at the ξ -stations shown in Table P7.1.

Hint: Since the boundary-layer equations are being solved in the transformed plane (ξ, η) , from Eq. (7.3.4)

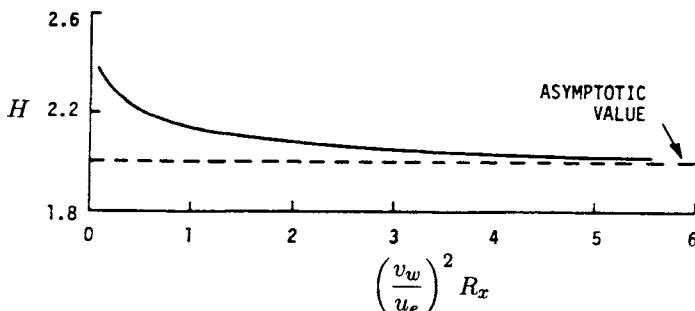
$$f_w(\xi) = -\frac{v_w}{u_\infty} (R_L \xi)^{1/2} \quad (\text{P7.6.3})$$

From Eqs. (P7.6.1) and (P7.6.3), we see that when $f_w = 2$, we have the asymptotic suction profile according to Iglisch's expression (P7.6.1).

Figure P7.1 shows the computed values of H for a flat plate with uniform suction and Table P7.2 shows the computed f_w'' values at the desired ξ -stations.

Table P7.1. Wall-shear parameter f_w'' for a laminar flat-plate with uniform suction.

ξ	f_w'' Iglisch
0	0.332
0.020	0.422
0.080	0.519
0.180	0.622
0.405	0.787
0.720	0.963
1.280	1.208

**Fig. P7.1.** Computed values of H for a flat-plate flow with uniform suction.**Table P7.2.** Computed f_w'' values for a laminar flat plate with uniform suction.

ξ	f_w''
0	0.332
0.020	0.419
0.080	0.523
0.180	0.626
0.405	0.790
0.720	0.963
1.280	1.208

8

Stability and Transition

8.1 Introduction

The current methods for predicting transition from laminar to turbulent flow are based on the solution of the unsteady Navier–Stokes equations (DNS) discussed in [1], on the solutions of the parabolized stability equations (PSE) discussed in [2, 3] and on the solutions of the linear stability equations discussed in Section 2.5. While DNS approach offers exciting possibilities, it is currently limited to some simple flows. Its computer requirements are large for transition calculations on complex bodies. The only engineering calculation method for predicting transition at this time is the e^n -method based on the solutions of the Orr–Sommerfeld (OS) equation given by Eq. (2.5.13)

$$\phi^{iv} - 2\alpha^2\phi'' + \alpha^4\phi = iR(\alpha u - \omega)(\phi'' - \alpha^2\phi) - iR\alpha u''\phi \quad (2.5.13)$$

subject to boundary conditions, which for wall boundary-layer flows are given by Eqs. (P2.10.1) and (P2.10.2), or

$$\bar{y} = 0, \quad \phi = \phi' = 0 \quad (8.1.1a)$$

$$\bar{y} = \delta, \quad (D^2 - \xi_1^2)\phi + (\xi_1 + \xi_2)(D + \xi_1)\phi = 0 \quad (8.1.1b)$$

$$(D + \xi_2)(D^2 - \xi_1^2)\phi = 0$$

Before we discuss the solution procedure it is useful to review the properties of the OS equation by rewriting Eq. (2.5.13) as

$$\phi^{iv} - 2\alpha^2\phi'' + \alpha^4\phi = i\alpha R(u - c)(\phi'' - \alpha^2\phi) - iR\alpha u''\phi \quad (8.1.2)$$

where c is the dimensionless complex propagation velocity of the disturbance related to its dimensional value c^* by

$$c = \frac{\omega}{\alpha} = \left(\frac{\omega^*}{\alpha^*} \right) \frac{1}{u_0} = \frac{c^*}{u_0} \quad (8.1.3)$$

The solution of the Orr–Sommerfeld equation and its boundary conditions may be obtained by temporal or spatial amplification theories. The former takes ω to be complex ($\equiv \omega_r + i\omega_i$) so that the amplitude of the disturbance varies with time as $\exp(\omega_i t)$, in contrast to the spatial amplification theory which takes ω to be real and α to be complex [Eq. (2.5.13)], or ω to be real and c to be complex [Eq. (8.1.2)], so that the amplitude varies with x as $\exp(-\alpha_i x)$. Note that for $\omega_i = 0$ and given α_r , a complex value of c implies values of the real frequency, ω_r , and the spatial amplification rate ($-\alpha_i$). Also, if α and ω are both real, then the disturbance propagates through the parallel mean flow with constant amplitude $|\phi(\bar{y})|$; if α and ω are complex, the disturbance amplitude will vary in both time and space.

While both procedures have advantages, it is more convenient to use the spatial amplification theory since the amplitude change of disturbance with distance can be measured in a steady mean flow. The amplitude at a fixed point is independent of time and spatial theory gives the amplitude change in a more direct manner than does the temporal theory.

The solution of the system given by Eqs. (2.5.13) and (8.1.1), with the real parts of ξ_1, ξ_2 strictly positive, exists only for certain combinations of Reynolds number R and the parameters of the disturbance α and ω since all the boundary conditions are homogeneous. Thus the problem is an *eigenvalue problem* in which values of R , α and ω are the eigenvalues and the corresponding amplitude functions are eigenfunctions. Hence, in general, no nontrivial solution of this system exists. Only if α, ω, R satisfy one or more relations of the form

$$F(\alpha, \omega, R) = 0 \quad (8.1.4)$$

can such a solution be found.

The eigenvalues of the OS equation for the spatial amplification case are often presented in (α, R) and (ω, R) diagrams that describe the three states of a disturbance at a given Reynolds number R as damped, neutral or amplified. For two-dimensional flows the locus $\alpha_i = 0$, or $c_i = 0$, called the *curve of neutral stability*, separates the damped (stable) region from the amplified (unstable) region. The point on this curve at which R has its smallest value is of special interest, because at values of R less than this value, all disturbances are stable. This smallest Reynolds number is known as the critical Reynolds number, R_{cr} . The neutral curve is the same in both temporal and spatial amplification theories. It is defined by a characteristic length L , velocity u_0 and kinematic viscosity ν , that is

$$R_{cr} = \frac{u_0 L}{\nu} \quad (8.1.5)$$

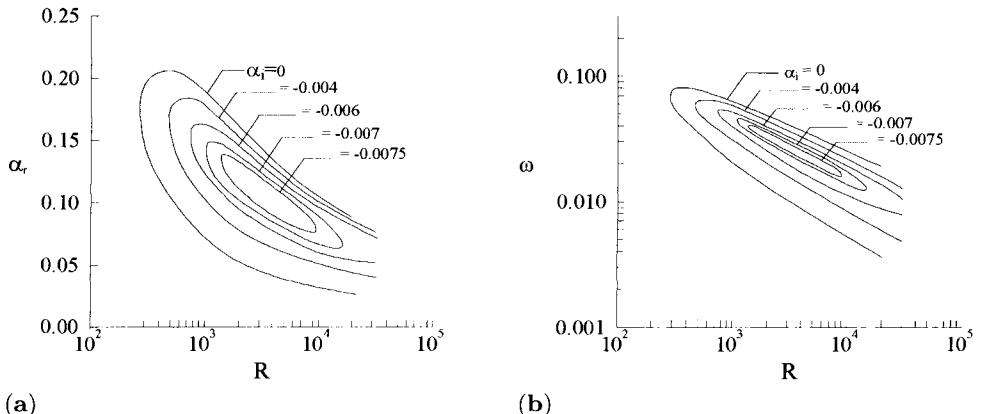


Fig. 8.1. Stability diagrams for Blasius flow. (a) α_r vs R . (b) ω vs R .

Figure 8.1 shows typical (α, R) and (ω, R) stability diagrams for Blasius flow obtained with the computer program of Section 8.4 using the eigenvalue procedure described in subsection 8.2.1. The length scale L used in Eq. (8.1.5) is chosen to correspond to that used in the definition of the similarity parameter in Eq. (7.3.4), that is,

$$L = \sqrt{\frac{\nu x}{u_e}} \quad (8.1.6)$$

and the velocity scale u is taken to correspond to the external velocity u_e .

As a result, the Reynolds number in the Orr–Sommerfeld equation is defined by

$$R = \sqrt{\frac{u_e x}{\nu}} = \sqrt{R_x} \quad (8.1.7)$$

Figure 8.2 and Table 8.1 show the variation of the critical Reynolds number, $R_{\delta_{cr}^*}$, with dimensionless pressure gradient parameter β and shape factor H for

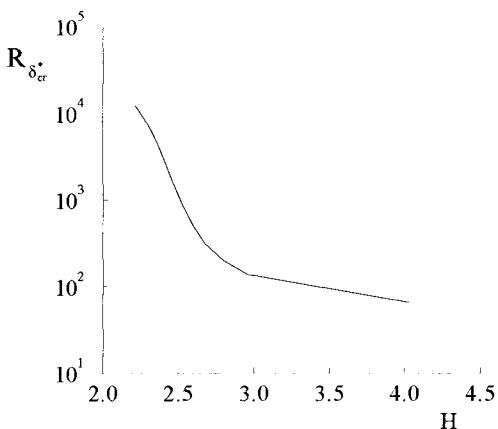


Fig. 8.2. Variation of critical Reynolds number, $R_{\delta_{cr}^*}$ with H .

Table 8.1. Critical Reynolds number for Falkner–Skan flows [3].

$R_{\delta_{cr}^*}$	β	H
12490	1.0	2.216
10920	0.8	2.240
8890	0.6	2.274
7680	0.5	2.297
6230	0.4	2.325
4550	0.3	2.362
2830	0.2	2.411
1380	0.1	2.481
865	0.05	2.529
520	0.0	2.591
318	-0.05	2.676
199	-0.10	2.801
138	-0.14	2.963
67	-0.1988	4.029

Falkner–Skan flows given by Eq. (7.3.11). Note that in this figure the critical Reynolds number is based on displacement thickness δ^* , and local velocity u_e . In terms of Falkner–Skan variables it can be expressed as

$$R_{\delta^*} = \sqrt{R_x} \delta_1^* \quad (8.1.8)$$

with δ_1^* denoting a dimensionless displacement thickness parameter expressed in terms of Falkner–Skan variables by

$$\delta_1^* = \int_0^{\eta_e} (1 - f') d\eta = \eta_e - f_e \quad (8.1.9)$$

From Fig. 8.2 it is seen that accelerating flows can tolerate larger Reynolds numbers than decelerating flows and, as a result, have a higher critical Reynolds number. On the other hand, decelerating flows have less tolerance to higher Reynolds numbers, indicating that $R_{\delta_{cr}^*}$ becomes smaller as H becomes bigger which corresponds to smaller values of β .

8.2 Solution of the Orr–Sommerfeld Equation

There are several numerical methods for obtaining solutions of the stability equations for two- and three-dimensional flows. In general, finite-difference methods are more efficient and flexible than the initial-value schemes which may have problems associated with parasitic error growth as discussed in [3]. An efficient and accurate finite-difference method is the box-scheme discussed in subsection 4.4.3 for the unsteady heat conduction equation and in Chapter 7 for the boundary layer equations. Here we discuss its application to the Orr–Sommerfeld equation.

8.2.1 Numerical Formulation

To formulate the numerical scheme employing the box method, we consider Eq. (2.5.13) and its boundary conditions given by (8.1.1) and reduce them to an equivalent first-order system. We define

$$\phi' = f \quad (8.2.1a)$$

$$f' = s + \xi_1^2 \phi \quad (8.2.1b)$$

$$s' = g \quad (8.2.1c)$$

and write Eq. (2.5.13) as

$$g' = \xi_2^2 s - \xi_3 \phi \quad (8.2.1d)$$

With these variables, the boundary conditions become

$$y = 0, \quad \phi = 0, \quad f = 0 \quad (8.2.2a)$$

$$\begin{aligned} y = \delta; \quad & s + (\xi_1 + \xi_2)f + \xi_1(\xi_1 + \xi_2)\phi = 0; \\ & g + \xi_2 s = 0 \end{aligned} \quad (8.2.2b)$$

where ξ_2 , defined in Eq. (P2.7.6b) attains its value at $y = \delta$.

We now consider a nonuniform mesh with $y = 0$ represented by y_0 and $y = \delta$ by y_J and approximate the quantities (f, s, g, ϕ) at points y_j by (f_j, s_j, g_j, ϕ_j) . As in subsection 4.4.3, we write the finite-difference approximations of Eqs. (8.2.1) for the midpoint $y_{j-\frac{1}{2}}$ using centered-difference derivatives to obtain

$$\phi_j - \phi_{j-1} - c_3(f_j + f_{j-1}) = (r_1)_j = 0 \quad (8.2.3a)$$

$$f_j - f_{j-1} - c_3(s_j + s_{j-1}) - c_1(\phi_j + \phi_{j-1}) = (r_3)_{j-1} = 0 \quad (8.2.3b)$$

$$s_j - s_{j-1} - c_3(g_j + g_{j-1}) = (r_2)_j = 0 \quad (8.2.3c)$$

$$g_j - g_{j-1} - c_4(s_j + s_{j-1}) - c_2(\phi_j + \phi_{j-1}) = (r_4)_{j-1} = 0 \quad (8.2.3d)$$

Here, with h_{j-1} denoting $y_j - y_{j-1}$,

$$c_3 = \frac{h_{j-1}}{2}, \quad c_1 = \xi_1^2 c_3, \quad c_2 = -(\xi_3)_{j-\frac{1}{2}} c_3, \quad c_4 = (\xi_2^2)_{j-\frac{1}{2}} c_3 \quad (8.2.4)$$

As in the procedure for solving the boundary-layer equations, Eqs. (8.2.3) are again written in a sequence which ensures that the A -matrix in Eq. (4.4.30) is not singular.

Equations (8.2.3) are imposed for $j = 1, 2, \dots, J$ and additional conditions at $j = 0$ and $j = J$ are obtained from the appropriate boundary conditions which, for an external flow, follow from Eqs. (8.2.2) and can be written as

$$\phi_0 = (r_1)_0 = 0, \quad f_0 = (r_2)_0 = 0 \quad (8.2.5a)$$

$$f_J + \tilde{c}_3 s_J + \tilde{c}_1 \phi_J = (r_3)_J = 0, \quad g_J + \tilde{c}_4 s_J = (r_4)_J = 0 \quad (8.2.5b)$$

where now

$$\tilde{c}_3 = \frac{1}{\xi_1 + \xi_2}, \quad \tilde{c}_1 = \xi_1, \quad \tilde{c}_4 = \xi_2 \quad (8.2.6)$$

The difference equations (8.2.3) and (8.2.5) have trivial solutions $f_j = s_j = \phi_j = g_j = 0$ for all j and we shall use the iteration procedure described below to find the special parameter values for which nontrivial solutions exist.

Since the Orr-Sommerfeld equation and the boundary conditions are homogenous, the trivial solution $\phi(y) = 0$ is valid for all values of α, β, ω and R . For this reason to compute the eigenvalues and the eigenfunctions we first replace the boundary condition $\phi'(0) = 0$ (that is $f_0 = 0$) of Eq. (8.2.5) by $\phi''(0) = 1$, that is $s_0 = 1$. Now the difference equations have a non-trivial solution since $\phi''(0) \neq 0$ and we seek to adjust or to determine parameter values so that the original boundary condition is satisfied. This is achieved by an iteration scheme based on Newton's method. Specifically, we first write the wall boundary conditions

$$\phi_0 = (r_1)_0 = 0, \quad s_0 = (r_2)_0 = 1 \quad (8.2.7)$$

and the edge boundary conditions in Eq. (8.2.5b) and Eq. (8.2.3) in matrix-vector form as in Eq. (4.4.29), with $\vec{\delta}_j$ and \vec{r}_j defined by

$$\vec{\delta}_j = \begin{vmatrix} \phi_j \\ s_j \\ f_j \\ g_j \end{vmatrix}, \quad \vec{r}_j = \begin{vmatrix} (r_1)_j \\ (r_2)_j \\ (r_3)_j \\ (r_4)_j \end{vmatrix} \quad (8.2.8)$$

and the A_j, B_j, C_j denote 4×4 matrices given by

$$\begin{aligned} A_0 &\equiv \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ (c_1)_1 & (c_3)_1 & 1 & 0 \\ (c_2)_1 & (c_4)_1 & 0 & 1 \end{vmatrix}, \quad A_j \equiv \begin{vmatrix} 1 & 0 & -(c_3)_j & 0 \\ 0 & 1 & 0 & -(c_3)_j \\ (c_1)_{j+1} & (c_3)_{j+1} & 1 & 0 \\ (c_2)_{j+1} & (c_4)_{j+1} & 0 & 1 \end{vmatrix}, \quad 1 \leq j \leq J-1 \\ A_J &= \begin{vmatrix} 1 & 0 & -(c_3)_J & 0 \\ 0 & 1 & 0 & -(c_3)_J \\ (\tilde{c}_1) & (\tilde{c}_3) & 1 & 0 \\ 0 & (\tilde{c}_4) & 0 & 1 \end{vmatrix}, \\ B_j &= \begin{vmatrix} -1 & 0 & -(c_3)_j & 0 \\ 0 & -1 & 0 & -(c_3)_j \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad 1 \leq j \leq J \\ C_j &= \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ (c_1)_{j+1} & (c_3)_{j+1} & -1 & 0 \\ (c_2)_{j+1} & (c_4)_{j+1} & 0 & -1 \end{vmatrix} \quad 0 \leq j \leq J-1 \end{aligned} \quad (8.2.9)$$

We note from the difference equations defined by Eqs. (8.2.3) and (8.2.5) that the solution of Eq. (4.4.29) depends upon the three parameters α , ω and R and we can denote this dependence by writing

$$\vec{\delta} = \vec{\delta}(\alpha, \omega, R) \quad (8.2.10)$$

Recalling that α is complex, and that ω is real in spatial-amplification theory, the above relation implies that the solution of Eq. (4.4.29) depends upon four scalars. With any two of these scalars fixed, the remaining scalars can be computed in such a way that the missing boundary condition $\phi'(0) = 0$ is satisfied. In our finite-difference notation, this corresponds to the condition

$$f_0(\alpha, \omega, R) = 0 \quad (8.2.11)$$

8.2.2 Eigenvalue Procedure

To discuss the solution procedure for Eq. (8.2.11), let us consider the computation of α_r , ω and R for a specified value of α_i . In this case since ω is real, the solution of Eq. (8.2.11) can be obtained by fixing one parameter and solving for the remaining two. The choice of the fixed parameter depends on the slope of $d\alpha_r/dR$ so that, for example, it is more convenient to fix R and solve for α_r and ω away from the critical Reynolds number where $d\alpha_r/dR$ is small. In the region of critical Reynolds number, $d\alpha_r/dR$ is large and increases as $R \rightarrow R_{cr}$ and it is necessary to specify α_r and solve for ω and R . To explain these points further, consider the *eigenvalue problem corresponding to small $d\alpha_r/dR$* . Since f_0 is complex, α_i is given and R is fixed, Eq. (8.2.11) represents two equations with two unknowns (α_r, ω) and the equations can be solved by Newton's method. Specifically, if (α^ν, ω^ν) are the ν -th iterates, the $(\nu + 1)$ -th iterates are determined by using

$$\alpha_r^{\nu+1} = \alpha_r^\nu + \delta\alpha_r^\nu \quad (8.2.12a)$$

$$\omega^{\nu+1} = \omega^\nu + \delta\omega^\nu \quad (8.2.12b)$$

in Eq. (8.2.11), expanding f_0 about α_r^ν and ω^ν and retaining only linear terms in the expansion. This gives the linear system by taking $f_r^{\nu+1} = 0$ and $f_i^{\nu+1} = 0$

$$f_r^\nu + \left(\frac{\partial f_r}{\partial \alpha_r} \right)^\nu \delta\alpha_r^\nu + \left(\frac{\partial f_r}{\partial \omega} \right)^\nu \delta\omega^\nu = 0 \quad (8.2.13a)$$

$$f_i^\nu + \left(\frac{\partial f_i}{\partial \alpha_r} \right)^\nu \delta\alpha_r^\nu + \left(\frac{\partial f_i}{\partial \omega} \right)^\nu \delta\omega^\nu = 0 \quad (8.2.13b)$$

Here, for convenience, we have dropped the subscript 0 on f and used r and i to denote the real and imaginary parts of f at the wall. The solution of Eqs. (8.2.13) is,

$$\delta\alpha_r^\nu = \frac{1}{\Delta_0} \left[f_i^\nu \left(\frac{\partial f_r}{\partial \omega} \right)^\nu - f_r^\nu \left(\frac{\partial f_i}{\partial \omega} \right)^\nu \right] \quad (8.2.14a)$$

$$\delta\omega^\nu = \frac{1}{\Delta_0} \left[f_r^\nu \left(\frac{\partial f_i}{\partial \alpha_r} \right)^\nu - f_i^\nu \left(\frac{\partial f_r}{\partial \alpha_r} \right)^\nu \right] \quad (8.2.14b)$$

where

$$\Delta_0 = \left(\frac{\partial f_r}{\partial \alpha_r} \right)^\nu \left(\frac{\partial f_i}{\partial \omega} \right)^\nu - \left(\frac{\partial f_i}{\partial \alpha_r} \right)^\nu \left(\frac{\partial f_r}{\partial \omega} \right)^\nu \quad (8.2.14c)$$

To evaluate the derivatives of f_r and f_i with respect to α_r and ω , we need only to differentiate Eq. (4.4.29) and, since the vector \vec{r} is independent of α_r and ω , we get:

$$A \left(\frac{\partial \vec{\delta}}{\partial \alpha_r} \right)^\nu = - \left(\frac{\partial A}{\partial \alpha_r} \right)^\nu \vec{\delta}^\nu = \vec{r} \quad (8.2.15a)$$

$$A \left(\frac{\partial \vec{\delta}}{\partial \omega} \right)^\nu = - \left(\frac{\partial A}{\partial \omega} \right)^\nu \vec{\delta}^\nu = \vec{r} \quad (8.2.15b)$$

We shall refer to the above equations as *the variational equations of Eq. (4.4.29) with respect to α_r and ω* , respectively. Thus, to obtain the required derivatives, we need to solve only two linear systems with the same coefficient matrix A already computed and factored for Eq. (4.4.29). The vectors on the right-hand side of Eqs. (8.2.15a) are determined from Eqs. (8.2.3), (8.2.5b) and (8.2.7). For Eq. (8.2.15a) $(r_1)_j = (r_2)_j = 0$ for $0 \leq j \leq J$, but $(r_3)_j$ and $(r_4)_j$ for $1 \leq j \leq J$ are given by

$$(r_3)_{j-1} = 2 \left(\frac{\partial c_1}{\partial \alpha_r} \right) \phi_{j-\frac{1}{2}} \quad (8.2.16a)$$

$$(r_4)_{j-1} = 2 \left(\frac{\partial c_4}{\partial \alpha_r} \right) s_{j-\frac{1}{2}} + 2 \left(\frac{\partial c_2}{\partial \alpha_r} \right) \phi_{j-\frac{1}{2}} \quad (8.2.16b)$$

$$(r_3)_J = - \left(\frac{\partial \tilde{c}_3}{\partial \alpha_r} \right)_J s_J - \left(\frac{\partial \tilde{c}_1}{\partial \alpha_r} \right)_J \phi_J \quad (8.2.16c)$$

$$(r_4)_J = - \left(\frac{\partial \tilde{c}_4}{\partial \alpha_r} \right)_J s_J \quad (8.2.16d)$$

For Eq. (8.2.15b), again $(r_1)_j = (r_2)_j = 0$ for $0 \leq j \leq J$, but with c_1 , c_2 and c_3 being independent of ω for $j \leq J-1$, the coefficients $(r_3)_j$ and $(r_4)_j$ for $1 \leq j \leq J$ are given by

$$(r_3)_{j-1} = 0 \quad (8.2.17a)$$

$$(r_4)_{j-1} = 2 \left(\frac{\partial c_4}{\partial \omega} \right) s_{j-\frac{1}{2}} \quad (8.2.17b)$$

$$(r_3)_J = - \left(\frac{\partial \tilde{c}_3}{\partial \omega} \right)_J s_J \quad (8.2.17c)$$

$$(r_4)_J = - \left(\frac{\partial \tilde{c}_4}{\partial \omega} \right)_J s_J \quad (8.2.17d)$$

To summarize one part of the iteration process for a fixed value of R and for assumed values of α_r and ω , we solve Eq. (4.4.29). If the initial estimates of α_r and ω satisfy Eq. (8.2.11) and that $f_0 = 0$, then of course there is no need to compute new values of α_r and ω . On the other hand, if $f_0 \neq 0$, then new estimates of α_r and ω are obtained by Newton's method. With the right-hand side of Eqs. (8.2.15a,b) given by Eqs. (8.2.16) and (8.2.17), we solve Eqs. (8.2.15a,b) to compute $\partial f_r / \partial \alpha_r$ and $\partial f_r / \partial \omega$. Then we compute $\delta \alpha_r$ and $\delta \omega$ from Eqs. (8.2.14) and insert them in Eqs. (8.2.12) so that we can solve Eq. (4.4.29) with new estimates of α_r and ω and satisfy Eq. (8.2.11). This process is repeated until the increments $|\delta \omega|$ and $|\delta \alpha_r|$ are less than a specified tolerance parameter.

Once a solution to Eq. (8.2.11) is obtained at a specified Reynolds number R_0 , we can determine $(\partial \alpha_r / \partial R)_0$ and $(\partial \omega / \partial R)_0$ at $R = R_0$ and decide whether we are going to solve the eigenvalue problem in which we compute (α_r, ω) for a fixed R , as we did above, or compute (R, ω) for a fixed α_r . For this purpose we take the total derivative of Eq. (8.2.11) with respect to R and after separating the real and imaginary parts of the resulting expression, we get:

$$\left(\frac{\partial f_r}{\partial \alpha_r} \right)_0 \left(\frac{\partial \alpha_r}{\partial R} \right)_0 + \left(\frac{\partial f_r}{\partial \omega} \right)_0 \left(\frac{\partial \omega}{\partial R} \right)_0 = - \left(\frac{\partial f_r}{\partial R} \right)_0 \quad (8.2.18a)$$

$$\left(\frac{\partial f_i}{\partial \alpha_r} \right)_0 \left(\frac{\partial \alpha_r}{\partial R} \right)_0 + \left(\frac{\partial f_i}{\partial \omega} \right)_0 \left(\frac{\partial \omega}{\partial R} \right)_0 = - \left(\frac{\partial f_i}{\partial R} \right)_0 \quad (8.2.18b)$$

It follows from Eqs. (8.2.18) that

$$\left(\frac{\partial \alpha_r}{\partial R} \right)_0 = \frac{1}{\Delta_0} \left[\left(\frac{\partial f_i}{\partial R} \right)_0 \left(\frac{\partial f_r}{\partial \omega} \right)_0 - \left(\frac{\partial f_r}{\partial R} \right)_0 \left(\frac{\partial f_i}{\partial \omega} \right)_0 \right] \quad (8.2.19a)$$

$$\left(\frac{\partial \omega}{\partial R} \right)_0 = \frac{1}{\Delta_0} \left[\left(\frac{\partial f_r}{\partial R} \right)_0 \left(\frac{\partial f_i}{\partial \alpha_r} \right)_0 - \left(\frac{\partial f_i}{\partial R} \right)_0 \left(\frac{\partial f_r}{\partial \alpha_r} \right)_0 \right] \quad (8.2.19b)$$

where

$$\Delta_0 = \left(\frac{\partial f_r}{\partial \alpha_r} \right)_0 \left(\frac{\partial f_i}{\partial \omega} \right)_0 - \left(\frac{\partial f_r}{\partial \omega} \right)_0 \left(\frac{\partial f_i}{\partial \alpha_r} \right)_0 \quad (8.2.19c)$$

To evaluate the derivatives of f_r and f_i with respect to R , this time we differentiate Eq. (4.4.29) to obtain the variational equations with respect to R ,

$$A \left(\frac{\partial \vec{\delta}}{\partial R} \right)^{\nu} = - \left(\frac{\partial A}{\partial R} \right)^{\nu} \vec{\delta}^{\nu} = \vec{r} \quad (8.2.20)$$

The right hand side vectors of the above equation with $(r_1)_j = (r_2)_j = 0$ for $0 \leq j \leq J$, and with $(r_3)_j$ and $(r_4)_j$ for $1 \leq j \leq J$ are:

$$(r_3)_{j-1} = 0 \quad (8.2.21a)$$

$$(r_4)_{j-1} = 2 \left(\frac{\partial c_4}{\partial R} \right) s_{j-\frac{1}{2}} + 2 \left(\frac{\partial c_2}{\partial R} \right) \phi_{j-\frac{1}{2}} \quad (8.2.21b)$$

$$(r_3)_J = - \left(\frac{\partial \tilde{c}_3}{\partial R} \right)_J s_J \quad (8.2.21c)$$

$$(r_4)_J = - \left(\frac{\partial \tilde{c}_4}{\partial R} \right)_J s_J \quad (8.2.21d)$$

As before, with the coefficient matrix A already known, we now compute the relations in Eqs. (8.2.21) so that $\partial f / \partial R$ can be obtained from the solution of Eq. (8.2.20). Since $\partial f / \partial \alpha_r$ and $\partial f / \partial \omega$ are already known from the solutions of Eqs. (8.2.15), $\partial \alpha_r / \partial R$ and $\partial \omega / \partial R$ can be computed from Eqs. (8.2.19) to decide on the choice of the eigenvalue procedure.

It should be mentioned that the values of $\partial \alpha_r / \partial R$ and $\partial \omega / \partial R$ can also be very useful in estimating the initial values of α_r and ω at a different Reynolds number where new sets of α_r and ω are to be obtained. To discuss this point further, we expand α_r and ω by Taylor's series and by retaining only the first term, we write

$$\alpha_r = (\alpha_r)_0 + \left(\frac{\partial \alpha_r}{\partial R} \right)_0 \delta R \quad (8.2.22a)$$

$$\omega = \omega_0 + \left(\frac{\partial \omega}{\partial R} \right)_0 \delta R \quad (8.2.22b)$$

where $\delta R = R - R_0$ and subscript 0 denotes the values of α_r and ω at $R = R_0$.

If the slope $\partial \alpha_r / \partial R$ is greater than a specified quantity, then the strategy of computing the eigenvalues for the specified value of α_i needs to be changed. This can be done by incrementing α_r by small specified values and, for each value, ω and R that satisfies Eq. (8.2.11) are computed. As in the procedure that led to Eqs. (8.2.13), we now expand f in Eq. (8.2.11) about ω^ν and R^ν and retain only linear terms in the expansion,

$$f_r^\nu + \left(\frac{\partial f_r}{\partial \omega} \right)^\nu \delta \omega^\nu + \left(\frac{\partial f_r}{\partial R} \right)^\nu \delta R^\nu = 0 \quad (8.2.23a)$$

$$f_i^\nu + \left(\frac{\partial f_i}{\partial \omega} \right)^\nu \delta \omega^\nu + \left(\frac{\partial f_i}{\partial R} \right)^\nu \delta R^\nu = 0 \quad (8.2.23b)$$

Solving for $\delta \omega^\nu$ and δR^ν , we obtain essentially the same equations as those given by Eqs. (8.2.14) if we replace α_r by R ,

$$\delta R^\nu = \frac{1}{\Delta_0} \left[f_i^\nu \left(\frac{\partial f_r}{\partial \omega} \right)^\nu - f_r^\nu \left(\frac{\partial f_i}{\partial \omega} \right)^\nu \right] \quad (8.2.24a)$$

$$\delta \omega^\nu = \frac{1}{\Delta_0} \left[f_r^\nu \left(\frac{\partial f_i}{\partial R} \right)^\nu - f_i^\nu \left(\frac{\partial f_r}{\partial R} \right)^\nu \right] \quad (8.2.24b)$$

where now

$$\Delta_0 = \left(\frac{\partial f_r}{\partial R} \right)^\nu \left(\frac{\partial f_i}{\partial \omega} \right)^\nu - \left(\frac{\partial f_i}{\partial R} \right)^\nu \left(\frac{\partial f_r}{\partial \omega} \right)^\nu \quad (8.2.24c)$$

To find the eigenvalues at a new chosen value of α_r , we may again obtain better estimates of ω and R by writing

$$R = R_0 + \left(\frac{\partial R}{\partial \alpha_r} \right)_0 \delta \alpha_r \quad (8.2.25a)$$

$$\omega = \omega_0 + \left(\frac{\partial \omega}{\partial \alpha_r} \right)_0 \delta \alpha_r \quad (8.2.25b)$$

However, the additional work required to compute $\partial \omega / \partial \alpha_r$ may not justify this advantage since the calculations are being performed near the critical Reynolds number region for small increments in α_r which make the initial guess and the eigenvalue problem easier.

The calculation of ω and R for a specified value of α_r is very similar to the one in which ω and α_r are computed for a specified R . With initial estimates of ω and R , Eq. (4.4.29) can be solved to see whether Eq. (8.2.11) is satisfied. If not, new estimates are computed by using Newton's method. Equations similar to (8.2.15b) and (8.2.20) are solved to get $\partial f / \partial \omega$ and $\partial f / \partial R$, respectively, and then new increments in R and ω are obtained from the solution of Eqs. (8.2.24). The procedure is repeated until convergence, when, for example, $|\delta \omega|$ and $|\delta R|$ become less than a specified tolerance parameter.

8.3 e^n -Method

The e^n -method, which utilizes linear stability theory, was first used by Smith [4] and van Ingen [5] and is discussed in detail in [3]. The basic assumption is that transition starts when a small disturbance is introduced at a critical Reynolds number and is amplified by a factor of e^n which, for a typical value of n equal to 9, is about 8000. The calculation of transition with this procedure is relatively straight-forward in two-dimensional (and in axisymmetric) flows, requiring the calculation of the amplification rates ($-\alpha_i$) as a function of x (or R) for a range of dimensional frequencies ω^* . The stability calculations are preceded by boundary-layer calculations and, for a given external velocity distribution $u_e(x)$ and freestream Reynolds number, the laminar boundary-layer equations are solved to obtain the streamwise velocity profile u and its second derivative u'' . The stability calculations begin at a Reynolds number, R_{δ^*} , slightly larger than the critical Reynolds number, $R_{\delta_{cr}^*}$, on the lower branch of the neutral stability curve at an x -location, say $x = x_1$ (see Fig. 8.3). At this point, since u , u'' and R are known, α_r and ω can be computed by the procedure of subsection 8.2.2 and the dimensional frequency ω^* determined from Eq. (2.5.10), that is,

$$\omega^* = \omega \frac{u_0}{L} \quad (8.3.1)$$

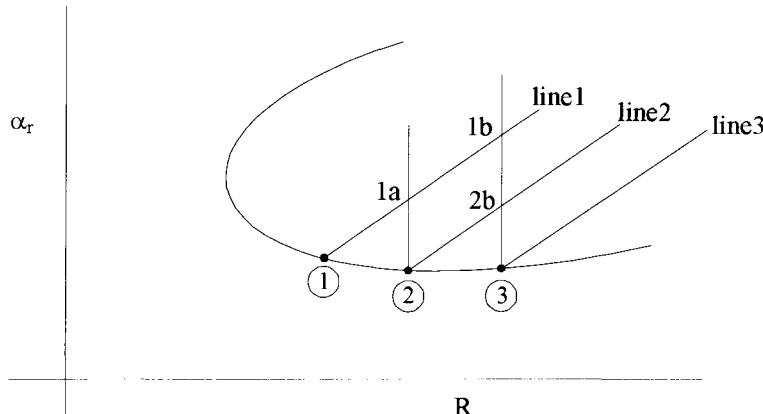


Fig. 8.3. Strategy of calculating transition with the e^n -method.

and kept constant along line 1 defined by this constant dimensional frequency. At the next x -location, x_2 , two separate calculations are performed for the newly computed boundary-layer profiles u and u'' and Reynolds number R . In one set of calculations (point 2 in Fig. 8.3), α_r and ω are computed on the neutral curve with the procedure used to obtain α_r and ω at point 1 so that a new dimensional frequency can be defined on line 2. In the second set of calculations, point 1a, the dimensionless frequency ω is first determined from the dimensional frequency ω^* on line 1, and its characteristic velocity and length scales at point 2, that is,

$$\omega = \omega^* \frac{L}{u_0} \quad (8.3.2)$$

With ω known from Eq. (8.3.2) and R defined at point 2, α can be determined by the eigenvalue procedure for transition described below. The procedure at point 1a is then repeated at points 2b and 1b, and a dimensional frequency is computed for line 3. For example, at point 1b, values of (α_r, α_i) are computed with the known dimensional frequency on line 1 and the specified Reynolds number at point 3; at point 2b, they are computed with the known dimensional frequency on line 2 and the specified Reynolds number at point 3. This procedure is repeated for several lines and the variation of the integrated amplification rate, defined by

$$n = - \int_{x_0}^x \alpha_i dx \quad (8.3.3)$$

with x_0 corresponding to each value of x on the neutral stability curve, is computed for each line. This procedure leads to the curves of constant frequency, Fig. 8.4, and their envelope corresponds to the maximum amplification factors from which transition is computed with a value for n , commonly assumed between 8 and 9.

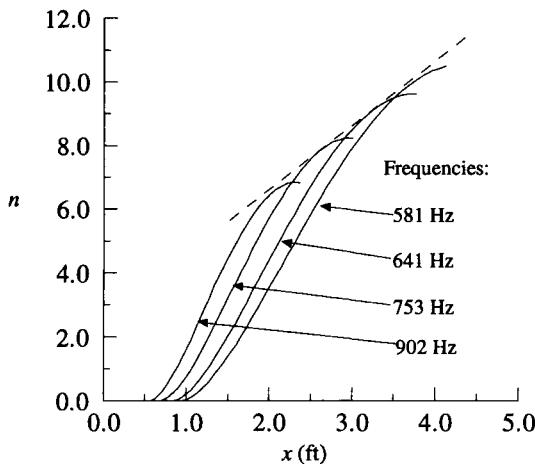


Fig. 8.4. Variation of the integrated amplification factors with distance and frequency for Blasius flow.

The eigenvalue procedure for computing α needed to predict transition with the e^n -method for specified values of ω and R is analogous to that described in the previous subsection. In this case, Eq. (8.2.11) is expanded with the Taylor series rather than Eqs. (8.2.13), and linear terms are retained to give

$$f_r^\nu + \left(\frac{\partial f_r}{\partial \alpha_r} \right)^\nu \delta \alpha_r^\nu + \left(\frac{\partial f_r}{\partial \alpha_i} \right)^\nu \delta \alpha_i^\nu = 0 \quad (8.3.4a)$$

$$f_i^\nu + \left(\frac{\partial f_i}{\partial \alpha_r} \right)^\nu \delta \alpha_r^\nu + \left(\frac{\partial f_i}{\partial \alpha_i} \right)^\nu \delta \alpha_i^\nu = 0 \quad (8.3.4b)$$

The solution of these equations is similar to those given by Eq. (8.2.14),

$$\delta \alpha_r^\nu = \frac{1}{\Delta_0} \left[f_i^\nu \left(\frac{\partial f_r}{\partial \alpha_i} \right)^\nu - f_r^\nu \left(\frac{\partial f_i}{\partial \alpha_i} \right)^\nu \right] \quad (8.3.5a)$$

$$\delta \alpha_i^\nu = \frac{1}{\Delta_0} \left[f_r^\nu \left(\frac{\partial f_i}{\partial \alpha_r} \right)^\nu - f_i^\nu \left(\frac{\partial f_r}{\partial \alpha_r} \right)^\nu \right] \quad (8.3.5b)$$

where

$$\Delta_0 = \left(\frac{\partial f_r}{\partial \alpha_r} \right)^\nu \left(\frac{\partial f_i}{\partial \alpha_i} \right)^\nu - \left(\frac{\partial f_i}{\partial \alpha_r} \right)^\nu \left(\frac{\partial f_r}{\partial \alpha_i} \right)^\nu \quad (8.3.5c)$$

Differentiation of Eq. (4.4.29) with respect to α_r and α_i leads to the derivatives of f_r and f_i with respect to α_r and α_i , and an equation identical to Eq. (8.2.15) with ω replaced by α_i is obtained. Again the vectors on the right-hand side of Eq. (8.2.15) are determined from Eqs. (8.2.3), (8.2.5) and (8.2.7), and $(r_1)_j = (r_2)_j = 0$ for $0 \leq j \leq J$, and $(r_3)_j$ and $(r_4)_j$ for $0 \leq j \leq J$ are given by Eqs. (8.2.16) with c_1 , c_2 , c_3 and c_4 now differentiated with respect to α_r and α_i , respectively.

8.4 Computer Program STP

We now describe a stability-transition program (STP) for calculating the neutral stability curves and transition of two-dimensional flows based on spatial amplification theory and the numerical method discussed in Sections 8.2 and 8.3. The program (Appendix B) requires that the boundary-layer velocity profiles be calculated with the boundary-layer program of Chapter 7 so that u and u'' can be used at each ξ -station as a function of η . For convenience, the velocity u_0 and length L scales in the Orr–Sommerfeld equation are chosen to correspond to

$$u_0 = u_e, \quad L = \sqrt{\frac{\nu x}{u_e}} = \frac{x}{\sqrt{R_x}} \quad (8.4.1)$$

with Reynolds number R now given by

$$R = \frac{u_0 L}{\nu} = \sqrt{R_x} \quad (8.4.2)$$

With this choice, the boundary-layer grid of the velocity profiles can be used in the solution of the Orr–Sommerfeld equation. This means, that u and u'' in the stability-transition program are related to the output of the boundary-layer program by

$$u = f', \quad u'' = f''' \quad (8.4.3)$$

and that the output of the boundary-layer program can be arranged to calculate u'' once the solutions of the boundary-layer equations are known. The parameter f''' can be obtained either by differentiating f'' with respect to η or from the finite-differenced momentum equation, Eq. (7.3.18), which for laminar flow can be written as

$$v'_j \equiv (f'''_j)^n = -\alpha_1(fv)_j^n + \alpha_2(u_j^2)^n - \alpha_n(v_j^{n-1}f_j^n - f_j^{n-1}v_j^n) + R_j^{n-1} \quad (8.4.4)$$

where R_j^{n-1} is given by Eq. (7.3.20a).

With boundary-layer profiles known at each x -station, the stability calculations can be started at any x -station where the critical Reynolds number R_{cr} is less than the local Reynolds number used in the boundary-layer calculations. For external flows, an estimate of $R_{\delta_{cr}^*}$ can be obtained from Fig. 8.2 with R_{δ^*} known from the boundary-layer calculations and included in the output subroutine.

The calculations for transition are first performed for a neutral stability curve at the specified x -location where R ($\equiv \sqrt{R_x}$) is known. The calculation of ω and α requires initial estimates. A convenient procedure to achieve this is the continuation method discussed in [3] but to retain a comparatively simple program, this is not part of the present description. The initial estimates of ω and α for a given R can be obtained for Blasius flow.

To calculate the location of the onset of transition it is usually sufficient to perform neutral stability calculations at four x -locations so that four frequencies can be computed and amplification rates determined on four curves with constant dimensional frequencies. While STP is general and can be used for external flows with suction and injection as well as free shear flows such as jets and wakes, changes are required to accommodate boundary conditions and provide better estimates of initial eigenvalues where required. STP can also be used for flows with heat transfer as discussed in [3].

STP consists of MAIN and four subroutines – VELPRO, CSAVE, NEWTON and NEWTONI – and are described in the following subsections.

8.4.1 MAIN

In MAIN the total number of x -stations (NXT) and the x -station where the stability calculations begin (NXO) are identified together with the requirement of neutral stability (IXT = 0) or transition. In the latter case, it is necessary to specify the number of frequencies, and this is done by setting IXT to 1, 2, or any number other than zero but less than twenty so that, for example, 1 will lead to transition calculations for one frequency and 2 for two frequencies, etc.

To calculate the neutral stability curves, the dimensionless external velocity UE(I) ($\equiv u_e/u_\infty$) is required as a function of surface distance S(I) ($\equiv s/L$), together with Reynolds number R_L ($\equiv u_\infty L/\nu$), a reference length L and velocity UINF ($\equiv u_\infty$). The stability Reynolds number REY(I) ($\equiv R$) is then calculated from Eq. (8.1.7) with R_x given by

$$R_x = \bar{u}_e \left(\frac{s}{L} \right) R_L \quad (8.4.5)$$

The calculation of the eigenvalues α and ω at NX = NXO ($s = s_0$) also requires the specification of their initial estimates at the Reynolds number corresponding to its value at s_0 . For Blasius flow, they can be obtained from the already constructed stability diagrams, such as those given by Fig. 8.1. When other figures are used, care should be taken to ensure that the estimated eigenvalues have the same length and velocity scales as those used in STP.

The velocity profiles u and u'' needed in the stability equation are obtained by calling VELPRO, and the calculations for neutral stability are performed by calling NEWTON and for amplification rates α_i and n -factor by calling NEWTONI.

8.4.2 Subroutine VELPRO

The velocity profiles for STP can be generated by any boundary-layer method. Here we assume that they are provided by BLP of Section 7.4. The read-in values of KX and NP in this subroutine refer to the NX-station and the j -points

(J) where the profiles are computed. With the specification of $U(J)$ ($\equiv f'$) and $V(J)$ ($\equiv f''$) as a function of $\text{ETA}(J)$ ($\equiv n_j$), $\text{UUDP}(J)$ ($\equiv u'_j$) are calculated, by differentiating $V(J)$ with respect to η .

8.4.3 Subroutine CSAVE

This subroutine is used to obtain the solutions of the Orr-Sommerfeld equation for a given set of α and ω when the neutral stability curve is required or α_r and α_i for the determination of the location of the onset of transition. The standard problem refers to the solution of Eq. (4.4.29) subject to the boundary condition that $\phi''(0)$ ($\equiv s_o$) is equal to 1. The definitions of $(c_1)_j$ to $(c_4)_j$ in the Orr-Sommerfeld equation, as well as the edge definitions, are given by Eqs. (8.2.4) and (8.2.6), respectively, and the \vec{r}_j terms by Eqs. (8.2.3) and (8.2.7). All values of \vec{r}_j are zero except for $(r_2)_o$ which is equal to 1 because of the requirement that $s_o \equiv 1$.

This subroutine also contains the coefficients of the variational equations of the standard problem, Eq. (4.4.29), with respect to α , ω and R , together with the right-hand sides of these equations as given by Eq. (8.2.16) for those with respect to α , Eq. (8.2.17) for ω , and Eq. (8.2.21) for R .

The variational equations are written with respect to α , ω and R , but the coefficient matrix A in Eq. (4.4.29) is the same. For this reason, it is only necessary to define those (\vec{r}_j) terms that are not zero.

This subroutine also contains the block-elimination algorithm to solve Eq. (4.4.29) with the procedure described in subsection 4.4.3. Note that AA(1,1,1) and AA(2,2,1) denote $(\alpha_{11})_o$ and $(\alpha_{22})_o$ and correspond to the “wall” boundary conditions.

8.4.4 Subroutine NEWTON

For initial estimates of $\text{ALFA}(\equiv \alpha_r)$ and $\text{OMEGA}(\equiv \omega)$, this subroutine computes the eigenvalues α and ω according to the procedure described in Section 8.2. The perturbation quantities $\text{DALFA}(\equiv \delta\alpha_r)$ and $\text{DOMEGA}(\equiv \delta\omega)$ are computed according to Eq. (8.2.14). Upon convergence of the iterations, the dimensional frequencies $\text{WSO}(\text{IX})(\equiv \omega^*)$ are calculated and printed for that NX-station, and for each corresponding frequency, IX, together with the values of α and ω which serve as initial estimates for the next NX-station (or R), are stored. Also stored are the values of $\text{UM}(\equiv f_o)$, $\text{UMA}(\equiv \partial f_o / \partial \alpha)$, $\text{UMO}(\equiv \partial f_o / \partial \omega)$, $\text{UMR}(\equiv \partial f_o / \partial R)$, UE and REY .

8.4.5 Subroutine NEWTONI

In the calculation of transition, the amplification factor n is computed for a dimensional frequency ω^* according to Eq. (8.3.3) which in terms of the dimensionless quantities used in the solution of the Orr-Sommerfeld equation with

CLOGA corresponding to the exponent n in the e^n -method, can be written as

$$n = - \int_{\xi_o}^{\xi} \left(\frac{\alpha_i R}{\xi} \right) d\xi \quad (8.4.6)$$

8.5 Applications of STP

8.5.1 Stability Diagrams for Blasius Flow

The stability diagrams for Falkner-Skan flows, similar to those given in Fig. 8.1, can be constructed by using BLP and STP. For a given external velocity distribution (constant for Blasius flow), BLP generates the laminar velocity profiles so that $\text{ETA}(J)$, $U(J,2)$ and $V(J,2)$, $J = 1, \dots, \text{NP}$, can be printed from the MAIN program in a format compatible with the READ statement of subroutine VELPRO.

For the neutral stability diagrams, the calculations can be performed at a sufficiently high Reynolds number by specifying initial estimates of α_r and ω . Once a solution is obtained at one Reynolds number, the calculations for other Reynolds numbers can be obtained with the procedure of subsection 8.2.2. The same procedure can also be used for lower Reynolds numbers provided $d\alpha_r/dR$ is small. When this is not the case, α_r is incremented by small specified values and ω and R are computed to satisfy Eq. (8.2.11). The program of Section 8.4 is written for the case of small $d\alpha_r/dR$ and needs to be changed when $d\alpha_r/dR$ becomes large.

To demonstrate the use of STP, in this section we consider the Blasius flow and calculate the eigenvalues α_r and ω at four x -locations. We take $u_e = 160$ ft/sec, $R_L = 5 \times 10^6$, $L = 5$ ft and initiate the calculations at $x_o = 0.54$ ft for $\alpha_i = 0$. This corresponds to a Reynolds number of $R(\equiv \sqrt{R_x}) = 735$ which is sufficiently high for $d\alpha_r/dR$ to be small, see Fig. 8.1a. Initial estimates of α_r and ω follow from Fig. 8.1, and are 0.084 and 0.025 respectively, on the lower branch of the neutral stability curve at $R = 800$.

Figure 8.5 shows the computed eigenvalues α_r and ω along the surface distance x for values of α_i equal to 0 and -0.004 . The calculations for $\alpha_i = -0.004$ are essentially the same as those for the neutral stability curve, $\alpha_i = 0$. The initial estimates of α_r and ω can be obtained from those corresponding to $\alpha_i = 0$ provided that $\Delta\alpha_i$ is small. If this is not the case, then it is best to divide the increment in $\Delta\alpha_i$ to small values and perform the stability calculations for each value of α_i until the desired value of α_i is reached.

8.5.2 Transition Prediction for Flat Plate Flow

To demonstrate the prediction of the onset of transition location with the e^n -method, we first apply STP to a zero-pressure gradient flow which is identical

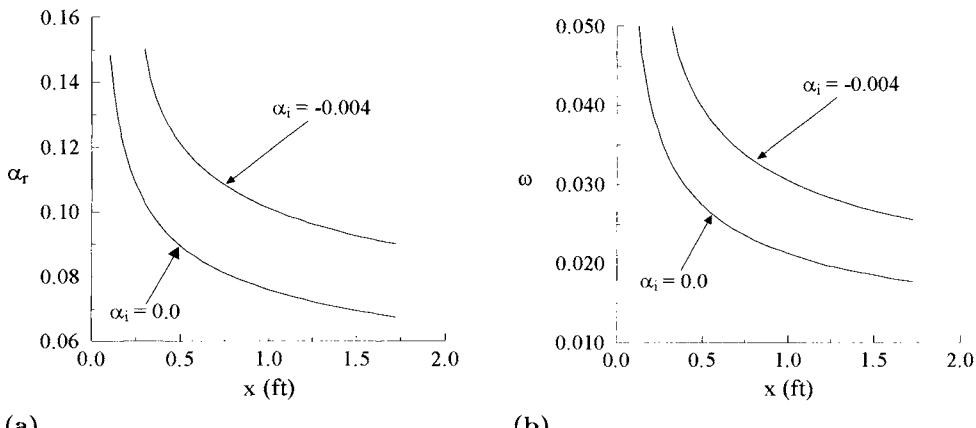


Fig. 8.5. Computed eigenvalues for Blasius flow, (a) α_r and (b) ω_r .

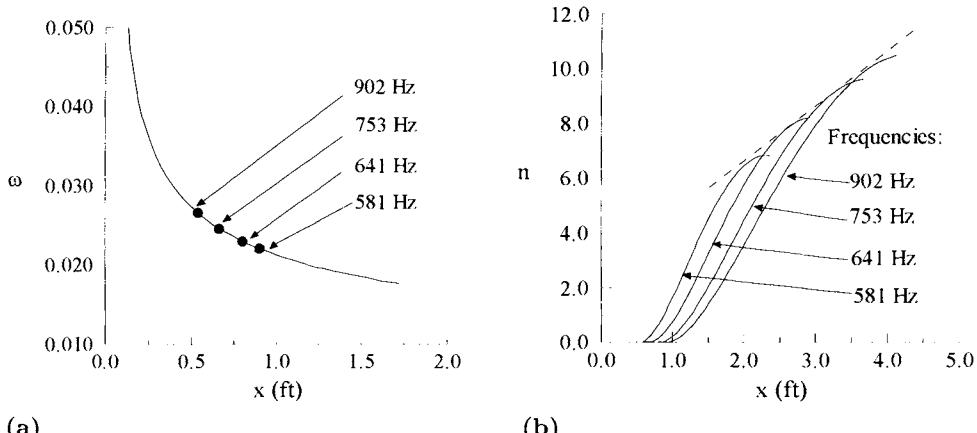


Fig. 8.6. Transition calculation for a flat plate flow, (a) origin of the disturbances on neutral curve and (b) integrated amplification rates.

to the one in the previous subsection except that now the stability calculations are performed to compute α . With dimensional frequencies known at four x -locations, the amplification rate calculations are performed for frequencies of 902, 753, 641 and 581 Hz, Fig. 8.6a. The n -values are obtained from Eq. (8.3.3) with x_o corresponding to each value of x in the neutral stability curve. Fig. 8.6b, which is the same as Fig. 8.4, shows the envelope of the amplification rates (dashed line) and indicates that for a value of $n = 8$, the onset of transition takes place at $x = 2.8$ ft, or at a Reynolds number of $R_x = 2.8 \times 10^6$.

8.5.3 Transition Prediction for Airfoil Flow

The procedure for determining the onset of transition location on an airfoil is similar to the procedure used for the flat plate flow discussed in the previous subsection. To demonstrate this, we consider an NACA 0012 airfoil at two chord Reynolds numbers, $R_c = 10^6$ and 3×10^6 . For the external velocity distribution obtained from the HSPM computer program discussed in Section 6.5, the laminar velocity profiles are computed with BLP for $\alpha = 0^\circ$, 2° and 4° . The neutral stability calculations are then performed for each Reynolds number and angle of attack. These calculations were then followed by amplification rate calculations for each frequency computed on the neutral stability curve. For each calculation, the transition location is determined with respect to the surface distance along the perimeter of the airfoil measured from the stagnation point and the corresponding x/c location is calculated.

Table 8.2 presents a summary of the calculated transition locations at three angles of attack and two chord Reynolds numbers for $n = 8$. The results show that with increasing angle of attack, since the adverse pressure gradient becomes stronger, the transition location moves upstream. The results also show that with increasing Reynolds number, the transition location moves upstream.

Table 8.2. Onset of transition locations on the upper surface of an NACA 0012 airfoil at two Reynolds numbers and three angles of attack.

α	0°	2°	4°
R_c	$(s/c)_{tr} - (x/c)_{tr}$	$(s/c)_{tr} - (x/c)_{tr}$	$(s/c)_{tr} - (x/c)_{tr}$
1×10^6	0.505–0.49	0.33–0.31	0.16–0.13
3×10^6	0.355–0.34	0.21–0.19	0.10–0.075

More applications of STP, including its extension to three-dimensional flows can be found in [3].

References

- [1] Kleiser, L. and Zong, T. A.: "Numerical Simulation of Transition in Wall Bounded Shear Flows," *Annual Review of Fluid Mechanics*, Vol. 23, pp. 495–538, 1991.
- [2] Herbert, T.: "Parabolized Stability Equations," Special Course on Progress in Transition Modeling, AGARD Report 793, April 1994.
- [3] Cebeci, T.: *Stability and Transition: Theory and Application*, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 2004.
- [4] Smith, A. M. O.: "Transition, Pressure Gradient, and Stability Theory," *Proceedings IX International Congress of Applied Mechanics*, Brussels, Vol. 4, pp. 234–24, 1956.
- [5] Van Ingen, J. L.: "A Suggested Semi-empirical Method for the Calculation of the Boundary-Layer Region," Report No. VTH71, VTH74, Delft, Holland, 1956.

Problems

- 8-1.** Compute the eigenvalues α_r and ω on the lower branch of the neutral stability curve for Blasius flow at $700 < R \leq 3000$. Take $u_e = 160 \text{ ft s}^{-1}$, $\nu = 1.6 \times 10^{-4} \text{ ft}^2 \text{ s}^{-1}$. Use BLP to calculate the velocity profiles.
- 8-2.** Repeat Problem 8-1 for $m = -0.01$ and -0.005 .
- 8-3.** Calculate the location of transition for $n = 9$ for the above two problems and discuss the effect of pressure gradient on the results.

9

Grid Generation

9.1 Introduction

The solution of the conservation equations, described in Chapter 2, requires an arrangement of a discrete set of grids or cells in the flow field; their determination for a given body is known as grid generation and is discussed in this chapter.

While the finite-volume methods can be applied to uniform and non-uniform meshes, the finite-difference methods require a uniform rectangular grid. In practically all real-life problems, a uniform rectangular grid is not possible in the physical plane. As a result, it is necessary to use a coordinate transformation and map the irregular region into a regular one in the computational plane. Analytic transformations for this purpose are difficult to construct except for some relatively simple geometries; for most multidimensional cases it is impossible to find such a transformation. Three major classes of techniques corresponding to *algebraic methods*, *differential equation methods* and *conformal mapping methods* can be used to overcome these difficulties. For example, in the numerical grid generation technique advanced by Thompson [1], the coordinate transformation is obtained automatically from the solution of partial-differential equations. In what is referred to as the structured grid approach, a curvilinear mesh is generated over the physical domain such that one member of each family of curvilinear coordinate lines is coincident with the boundary contour of the physical domain. For this reason, the scheme is also called the *boundary-fitted coordinate method*.

While the task of grid generation is an integral part of CFD, it can be separate from the task of numerically calculating the resulting flow field. A numerical method (flow solver) can be developed independently to solve the conservation equations with the grid generated separately. For methods requiring a solution in a computational plane, grid generation amounts to computing the metrics

of the transformation and knowing the one-to-one correspondence of the location of each grid point in both computational and physical planes in order to transform the solution back to the physical plane.

In the brief discussion of grid generation methods presented in this chapter, the basic concepts of the mapping and development of curvilinear coordinates are discussed in Section 9.2. Descriptions of three grid generation techniques, starting with the simplest scheme in one dimension (Section 9.3) are given in Sections 9.4 to 9.6. Section 9.7 will briefly review basic concepts and techniques for generating unstructured grids.

9.2 Basic Concepts in Grid Generation and Mapping

To illustrate the basic concepts of the mapping and development of curvilinear coordinates, otherwise known as the structured grid approach, consider an irregular region ABCDA in the physical plane in the (x, y) Cartesian coordinates (Fig. 9.1a) and determine its mapping into the computational domain in the (ξ, η) Cartesian coordinates such that the mapped region will have a rectangular shape and allow the construction of a square mesh (Fig. 9.1b). In addition, the boundaries of the physical domain must be coincident with the (ξ, η) coordinate lines of the boundaries of the transformed region in the computational domain. One procedure to accomplish this mapping is to set the values of ξ, η along the boundaries of the physical region in the following manner:

- 1) With $\eta = \text{constant}$, vary ξ monotonically along the boundary segments AB and CD of the physical region, and
- 2) With $\xi = \text{constant}$, vary η monotonically along the boundary segments BC and DA of the physical region.

With these requirements on the values of ξ and η along the boundaries of the physical region, the segments AB and CD of the physical region are mapped into the computational domain as horizontal lines, while the segments BC and DA are mapped into the computational domain as vertical lines, as shown in Fig. 9.1b. Note that each boundary segment of the irregular region in the physical domain is mapped into the sides of the rectangular region in the computational domain. Furthermore, the following requirements are placed on the mapping:

1. The mapping must be one to one.
2. Coordinate lines of the same family (i.e., ξ or η) must not cross.
3. The lines of different families must not cross more than once.
4. Grid points should be closely spaced in the physical domain where large numerical errors are expected.

To satisfy these requirements, a proper organization of the grid points along the boundaries of the physical region is needed. That is, if I-grid points are

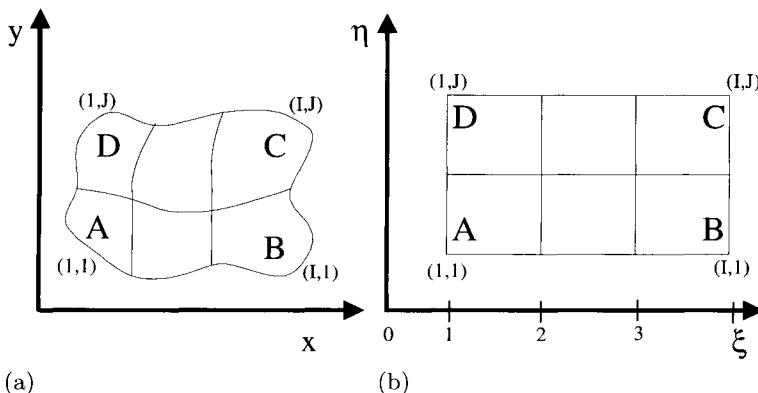


Fig. 9.1. Mapping an irregular simply connected region into the computational domain as a rectangle. (a) Physical plane, (b) computational plane.

placed along the bottom boundary segment AB of the physical domain, I-grid points must be placed along the opposing top boundary segment DC of the physical domain. Similarly, if J-grid points are placed along the right boundary segment BC of the physical domain, J-grid points must also be placed along the left segment AD. The actual values of ξ and η in the computational domain are immaterial because they do not appear in the final expressions. Thus, the coordinates of the grid A in the computational domain can be taken as $\xi = 1$ and $\eta = 1$ and the mesh size as $\Delta\xi = 1$ and $\Delta\eta = 1$. In this way, a square mesh can be constructed over the rectangular transformed region in the computational plane.

In the above example of mapping, an irregular region in the physical plane is mapped as a rectangular region into the computational plane. Depending on the choice of the values of ξ, η along the boundary segments of the physical region, other configurations can be generated in the computational plane. To illustrate this, let us consider an L-shaped irregular region ABCDEFA in the physical plane, as shown in Fig. 9.2a. One possibility is to map the region into an L-shaped regular region, as shown in Fig. 9.2b. This is accomplished by choosing the values of η along various boundary segments in the physical plane as follows:

1. With $\eta = \text{constant}$, vary ξ monotonically along the boundary segments AB, DC and FE of the physical region, and
2. With $\xi = \text{constant}$, vary η monotonically along the boundary segments FA, ED and CB of the physical region.

With these requirements, the L-shaped irregular region can be mapped into the computational plane as an L-shaped regular region. The coordinates of the grid point A in the computational plane can again be chosen as $\xi = \eta = 1$ and the mesh size taken as $\Delta\xi = \Delta\eta = 1$.

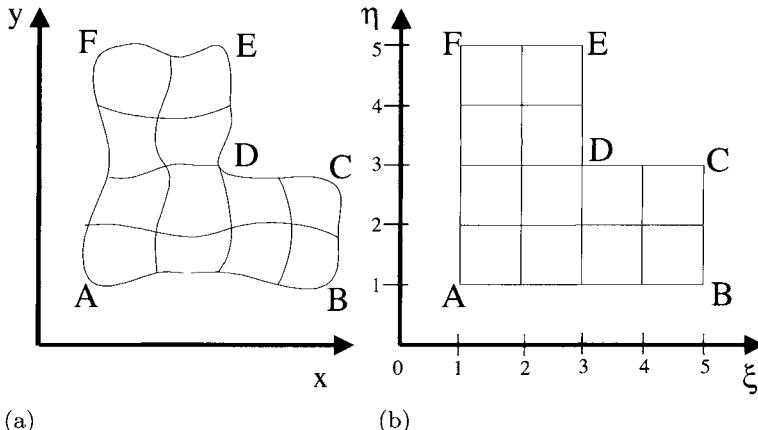


Fig. 9.2. Mapping the L-shaped irregular region into an L-shaped regular region. (a) Physical plane, (b) computational plane.

Another possibility of mapping the L-shaped irregular region in Fig. 9.3a is to map it as a rectangular region, as shown in Fig. 9.3b. This is accomplished by choosing the ξ and η values along various boundary segments of the physical region as follows:

1. With $\eta = \text{constant}$, vary ξ monotonically along the boundary segments FAB and CDE of the physical region, and
2. With $\xi = \text{constant}$, vary η monotonically along the boundary segments BC and EF of the physical region

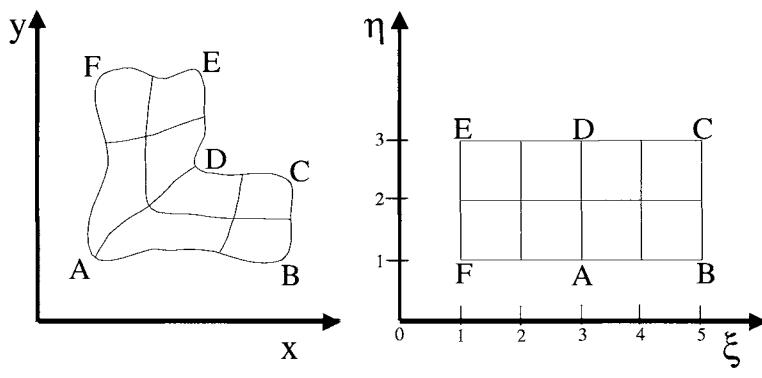


Fig. 9.3. Mapping the L-shaped irregular region into a rectangular region. (a) Physical plane, (b) Computational plane.

9.3 Stretched Grids

Let us assume that we are interested in the viscous flow solution of the conservation equations on a body, where the velocity varies rapidly near the surface of the physical plane. To calculate the details of this flow, a non-uniform grid, which has fine spacing close to the surface and coarse spacing away from the surface, is needed. To overcome the difficulties associated with the use of a non-uniform grid, it is desirable to obtain the flow-field solution in the computational plane where the grid is uniform (Fig. 9.4b) rather than in the physical plane (Fig. 9.4a). We use the following coordinate transformation for this purpose,

$$\xi = x \quad (9.3.1a)$$

$$\eta = 1 - \frac{\ln[A(y)]}{\ln B} \quad (9.3.1b)$$

Where:

$$A(y) = \frac{\beta + (1 - y/h)}{\beta - (1 - y/h)}, \quad B = \frac{\beta + 1}{\beta - 1} \quad (9.3.2)$$

Here β is called the stretching parameter, which assumes values $1 < \beta < \infty$. As β approaches unity, more grids are clustered near the wall in the physical domain. The inverse transformation is:

$$x = \xi \quad (9.3.3a)$$

$$\frac{y}{h} = \frac{(\beta + 1) - (\beta - 1)B^{1-\eta}}{1 + B^{1-\eta}} \quad (9.3.3b)$$

The continuity equation for steady flow in the two-dimensional physical plane, see Eq. (2.2.12b), is

$$\frac{\partial}{\partial x}(\varrho u) + \frac{\partial}{\partial y}(\varrho v) = 0$$

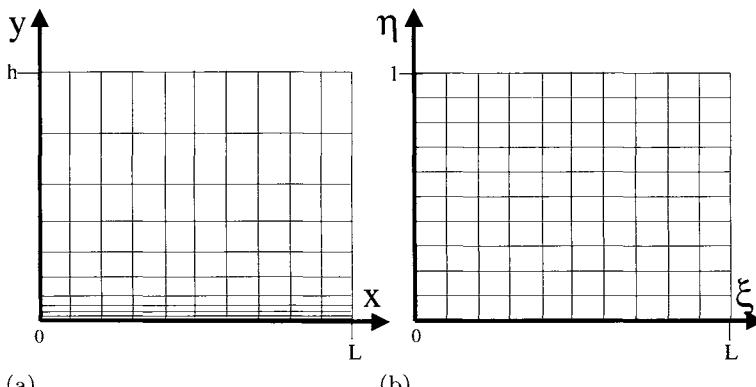


Fig. 9.4. One-dimensional stretching transformation. (a) Physical plane, (b) computational plane.

In the computational plane, this equation can be written as

$$\frac{\partial}{\partial \xi} \frac{\varrho(\xi_x u + \xi_y v)}{J} + \frac{\partial}{\partial \eta} \frac{\varrho(\eta_x u + \eta_y v)}{J} = 0$$

or, noting that the invariants of transformation are equal to zero,

$$\xi_x \frac{\partial}{\partial \xi} (\varrho u) + \eta_x \frac{\partial}{\partial \eta} (\varrho u) + \xi_y \frac{\partial}{\partial \xi} (\varrho v) + \eta_y \frac{\partial}{\partial \eta} (\varrho v) = 0 \quad (9.3.4)$$

The metrics in the above equation can be obtained from the transformation given by Eq. (9.3.1),

$$\xi_x = 1, \quad \xi_y = 0, \quad \eta_x = 0, \quad \eta_y = \frac{2\beta}{h \cdot \ln B} \frac{1}{\beta^2 - (1 - y/h)^2} \quad (9.3.5)$$

Therefore, the form of the continuity equation in the computational plane, with η_y defined in Eq. (9.3.5) and with y in η_y related to η by Eq. (9.3.3b) is:

$$\frac{\partial}{\partial \xi} (\varrho u) + \eta_y \frac{\partial}{\partial \eta} (\varrho v) = 0 \quad (9.3.6)$$

Note that the transformed continuity equation retains its general form, except for the coefficient η_y -term. Therefore, the transformed equation (9.3.6) is slightly more complicated than its original form, but it can now be solved for a uniform grid. Once the solution is obtained in the computational domain, the results can be transformed back to the physical domain with the inverse transformation given by Eq. (9.3.3) for each (ξ, η) location to the corresponding (x, y) location.

9.4 Algebraic Methods

The technique of using algebraic relations to cluster grid points close to the surface can be extended to generate computational grids in two or three dimensions, and to map arbitrary physical regions into a rectangular computational domain.

To illustrate the application of algebraic methods to generate a body-fitted mesh, consider a flow in a diverging nozzle, shown in Fig. 9.5. Let us represent

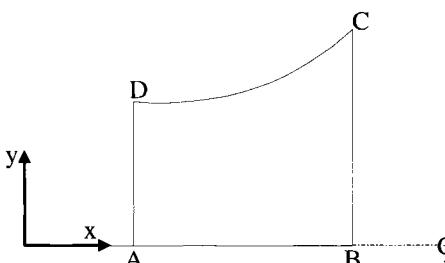


Fig. 9.5. Nozzle geometry.

the centerline by AB and the upper surface by DC, with the ordinate of the nozzle upper surface given by

$$y = x^2 \quad 1 \leq x \leq 2 \quad (9.4.1)$$

It is clear that for this geometry a rectangular grid in the physical plane is not appropriate. A curvilinear coordinate system, shown in Fig. 9.6a, with coordinates lines corresponding to BC and AB, however, nicely fits the boundaries of the nozzle geometry. The transformation of the curvilinear grid in Fig. 9.6a into a rectangular grid in the computational plane can be achieved by the following transformation:

$$\xi = x \quad (9.4.2a)$$

$$\eta = \frac{y}{y_{\max}} \quad (9.4.2b)$$

where y_{\max} denotes the ordinate of the upper surface BC.

Figure 9.6b shows the rectangular grid in the computational plane. Note that all the grid points in the physical plane of the nozzle upper surface, DC, fall along the horizontal line $\eta = 1$ in the computational plane, and those along the centerline AB fall along the horizontal line $\eta = 0$. The metrics of the transformation are easily obtained from Eq. (9.4.2). For the continuity equation given by Eq. (9.3.4),

$$\xi_x = 1 \quad \xi_y = 0 \quad \eta_x = -\frac{2\eta}{\xi} \quad \eta_y = \frac{1}{\xi^2} \quad (9.4.3)$$

Very complex algebraic functions can be used to generate appropriate grid systems. In the method developed by Smith and Weigel [2], the physical coordinates are rectangular and two disconnected boundaries are mapped into the computational plane. Representing these two disconnected boundaries in the physical plane by

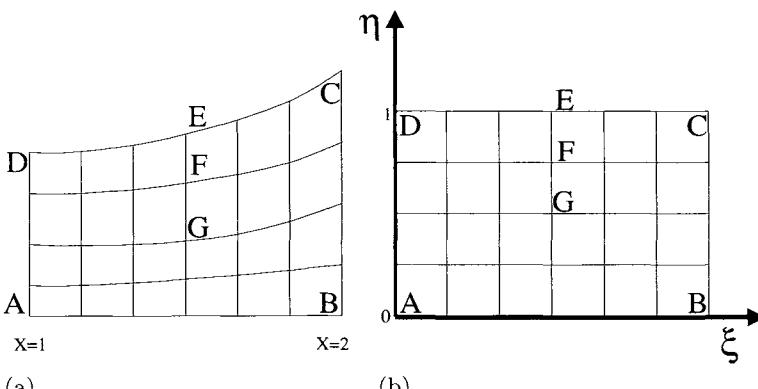


Fig. 9.6. Coordinate system for the nozzle geometry. (a) Physical plane, (b) computational plane.

$$x_{B1} = x_1(\xi), \quad y_{B1} = y_1(\xi) \quad (9.4.4a)$$

and

$$x_{B2} = x_2(\xi), \quad y_{B2} = y_2(\xi) \quad (9.4.4b)$$

The range of ξ in the computational plane is:

$$0 \leq \xi \leq 1$$

and the transformation is defined so that at $\eta = 0$,

$$x_{B1} = x_1(\xi) = x(\xi, 0), \quad y_{B1} = y_1(\xi) = y(\xi, 0) \quad (9.4.5a)$$

and at $\eta = 1$,

$$x_{B2} = x_2(\xi) = x(\xi, 1), \quad y_{B2} = y_2(\xi) = y(\xi, 1) \quad (9.4.5b)$$

A function defined on $0 \leq \eta \leq 1$ with parameters on the two boundaries completes the algebraic relation. This is chosen to be of the form

$$x = x(\xi, \eta) = F\left(x_1, \frac{dx_1}{d\eta}, \dots, x_2, \frac{dx_2}{d\eta}, \dots\right) \quad (9.4.6a)$$

$$y = y(\xi, \eta) = F\left(y_1, \frac{dy_1}{d\eta}, \dots, y_2, \frac{dy_2}{d\eta}, \dots\right) \quad (9.4.6b)$$

Smith and Weigel [2] suggest the use of either linear or cubic polynomials. For a linear function, the relations in Eq. (9.4.6) become

$$x = x_1(\xi)(1 - \eta) + x_2(\xi)\eta \quad (9.4.7a)$$

$$y = y_1(\xi)(1 - \eta) + y_2(\xi)\eta \quad (9.4.7b)$$

To demonstrate this approach, consider the mapping of a trapezoid (Fig. 9.7) into the computational plane centered at the origin. The trapezoid is defined by the equations

$$\begin{aligned} x &= 0, \quad x = 1 \\ y &= 0, \quad y = 1 + x \end{aligned} \quad (9.4.8)$$

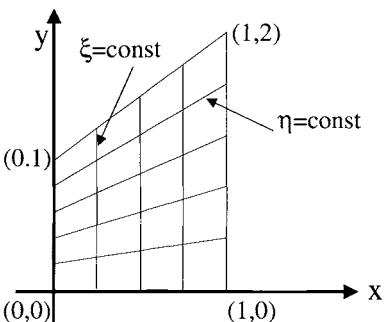


Fig. 9.7. Trapezoid to rectangle mapping.

The upper and lower boundaries may be written as

$$\begin{aligned}x_{B1} &= x_1(\xi) = \xi, \quad y_{B1} = y_1(\xi) = 0 \\x_{B2} &= x_2(\xi) = \xi, \quad y_{B2} = y_2(\xi) = 1 + \xi\end{aligned}\tag{9.4.9}$$

This produces the mapping required in Eq. (9.4.7) and is of the form

$$x = \xi\tag{9.4.10a}$$

$$y = (1 + \xi)\eta\tag{9.4.10b}$$

The metrics of this transformation for the continuity equation (9.3.4) are

$$\xi_x = 1, \quad \xi_y = 0, \quad \eta_x = -\frac{\eta}{1 + \xi}, \quad \eta_y = \frac{1}{1 + \xi}\tag{9.4.11}$$

9.4.1 Algebraic Grid Generation Using Transfinite Interpolation

To generate algebraic grids around more complex configurations, a multi-directional interpolation method called “Transfinite Interpolation” is often used. This method is implemented as a suite of unidirectional interpolations.

Unidirectional Interpolation

In a unidirectional interpolation, the Cartesian coordinate vector $\underline{r}(x, y)$ of each point on a curve is obtained as an interpolation between points that lie on the boundary curves (Fig. 9.8).

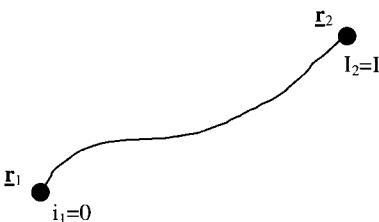


Fig. 9.8. Unidirectional interpolation along a curve with end points specified.

Lagrange Interpolation

The simplest form of unidirectional interpolation is the Lagrange interpolation, which is based on polynomials. Its general form, with $1 < i < I$, can be written:

$$\vec{r}(i) = \sum_{n=1}^N \varphi_n \left(\frac{i}{I} \right) \cdot \vec{r}_n\tag{9.4.12}$$

The Lagrange interpolation polynomials ϕ_n are defined by:

$$\varphi_n\left(\frac{i}{I}\right) = \prod_{k=1}^N \frac{i - i_k}{i_n - i_k} \quad (k \neq n) \quad (9.4.13)$$

Grid points are defined by evaluating the interpolation formula at successive integer values i . Interior points \underline{r}_n for $n = 2, 3, \dots, N - 1$ can be specified to serve as additional parameters to control the distribution. In most cases, however, interpolations are made solely from the boundaries, eliminating the need for additional interior information within the gridded region. The Lagrange interpolation is then reduced to its simplest, linear form:

$$\vec{r}(i) = \left(1 - \frac{i}{I}\right) \vec{r}_1 + \left(\frac{i}{I}\right) \vec{r}_2$$

Therefore $\varphi_1\left(\frac{i}{I}\right) = 1 - \frac{i}{I}$ and $\varphi_2\left(\frac{i}{I}\right) = \frac{i}{I}$

$$\underline{r}_1 = \underline{r} \quad (i = 0) \text{ and } \underline{r}_2 = r \quad (i = I)$$

Hermite Interpolation

Lagrange interpolations match only function values. It is possible to match both the function \underline{r} and the first derivative $\underline{r}' = \underline{r}_i$ by using Hermite interpolation, defined by:

$$\vec{r}(i) = \sum_{n=1}^N \Phi_n\left(\frac{i}{I}\right) \cdot \vec{r}_n + \sum_{n=1}^N \Psi_n\left(\frac{i}{I}\right) \cdot \vec{r}'_n \quad (9.4.14)$$

These polynomials can be obtained from the Lagrange Polynomials by:

$$\Phi_n\left(\frac{i}{I}\right) = \left[1 - 2\varphi'_n\left(\frac{i_n}{I}\right) \cdot \left(\frac{i - i_n}{I}\right)\right] \cdot \varphi_n^2\left(\frac{i}{I}\right) \quad (9.4.15a)$$

$$\Psi_n\left(\frac{i}{I}\right) = \left(\frac{i - i_n}{I}\right) \cdot \varphi_n^2\left(\frac{i}{I}\right) \quad (9.4.15b)$$

In the usual case with $N = 2$, the function matches two boundary values \underline{r}_1 and \underline{r}_2 and the first derivatives \underline{r}'_1 and \underline{r}'_2 at the two boundaries (Fig. 9.9). In this case, we have:

$$\begin{aligned} \Phi_1\left(\frac{i}{I}\right) &= \left(1 + 2\frac{i}{I}\right) \cdot \left(1 - \frac{i}{I}\right)^2 \\ \Phi_2\left(\frac{i}{I}\right) &= \left(3 - 2\frac{i}{I}\right) \cdot \left(\frac{i}{I}\right)^2 \\ \Psi_1\left(\frac{i}{I}\right) &= \left(1 - \frac{i}{I}\right)^2 \cdot \frac{i}{I} \\ \Psi_2\left(\frac{i}{I}\right) &= \left(\frac{i}{I} - 1\right) \cdot \left(\frac{i}{I}\right)^2 \end{aligned}$$

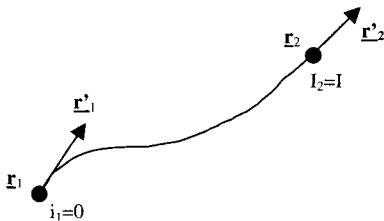


Fig. 9.9. Unidirectional interpolation along a curve with end points and slopes specified.

Interpolation by Splines

Lagrange and Hermite interpolation functions are completely continuous at all points. Both forms fit a single polynomial from one boundary to the other matching the specified values of the coordinates at the boundaries and, for Hermite interpolation, the first derivatives. As more points are included, oscillations may occur. An alternative is to fit a low order polynomial between each of the specified interior points with continuity of as many derivatives as possible. The interpolation function is then a piecewise-continuous polynomial, called a spline. Splines give normally very smooth point distributions. Tension splines can be used to obtain stronger localized curvature around interior points [1]. Another way is the use piecewise continuous functions such as B-Splines [3], which allow the interpolation to be modified locally without affecting the interpolation function outside of a given interval.

Interpolation by Functions Other than Polynomials

Interpolation between two points \underline{r}_1 and \underline{r}_2 can be written in general:

$$\vec{r}(i) = \left[1 - \varphi\left(\frac{i}{I}\right) \right] \vec{r}_1 + \varphi\left(\frac{i}{I}\right) \vec{r}_2 \quad (9.4.16)$$

ϕ can be any function, other than a polynomial, such that $\phi(0) = 0$ and $\phi(1) = 1$. The function ϕ is chosen to match the slope at the boundary or to match interior points and slopes. This interpolation function, used to control the spacing of the grid, is also called a “*stretching function*”. The most used stretching functions are the exponential function, the hyperbolic tangent function and the hyperbolic sine function. The hyperbolic tangent has a good overall distribution. It can be implemented as follows.

Spacing specified at both ends of the curve

Let S be the arc length varying from 0 to 1 as i varies from 0 to I : $S(0) = 0$ and $S(I) = 1$ and let ΔS_1 and ΔS_2 be the spacing specified at both ends $i = 0$ and $i = I$ of the curve.

$$S_i(0) = \Delta S_1 \quad \text{and} \quad S_i(I) = \Delta S_2$$

(the subscript i denotes differentiation with respect to i)

To construct the hyperbolic tangent distribution in this case, first let:

$$A = \frac{\sqrt{\Delta S_2}}{\sqrt{\Delta S_1}} \quad \text{and} \quad B = \frac{1}{I\sqrt{\Delta S_1.\Delta S_2}}$$

Then solve the following non-linear equation for δ :

$$\frac{\sinh \delta}{\delta} = B \quad (9.4.17)$$

The arc length distribution along the curve is then given by:

$$S(i) = \frac{u(i)}{A + (1 - A).u(i)} \quad (9.4.18)$$

With

$$u(i) = \frac{1}{2} \left\{ 1 + \frac{\tanh[\delta(\frac{i}{I} - \frac{1}{2})]}{\tanh(\frac{\delta}{2})} \right\} \quad (9.4.19)$$

Spacing specified at one end of the curve

When the spacing ΔS is specified only at $i = 0$, the distribution is computed as follows. First B is computed from:

$$B = \frac{1}{I\Delta S}$$

Equation (9.4.17) is used to compute δ . The arc length distribution is then given by:

$$s(i) = 1 + \frac{\tanh[\frac{\delta}{2}(\frac{i}{I} - 1)]}{\tanh(\frac{\delta}{2})} \quad (9.4.20)$$

When the spacing ΔS is specified only at $i = I$, the procedure is the same except that equation (9.4.20) is replaced by:

$$s(i) = \frac{\tanh[\frac{\delta}{2}(\frac{i}{I})]}{\tanh(\frac{\delta}{2})} \quad (9.4.21)$$

Spacing specified at an interior point of the curve

If the spacing ΔS is specified only at an interior point $S = p$, B is again calculated as above but δ is determined as the solution of:

$$1 + \left[\frac{1}{B.p.\delta} \right]^2 = \left[\frac{\cosh \delta - 1 + \frac{1}{p}}{\sinh \delta} \right]^2 \quad (9.4.22)$$

The value of i at which $S = p$ is obtained by solving the non-linear equation:

$$\chi = \frac{I}{\delta} \tanh^{-1} \left[\frac{\sinh \delta}{\cosh \delta - 1 + \frac{1}{p}} \right] \quad (9.4.23)$$

The arc length distribution is then given by:

$$s(i) = p \cdot \left\{ 1 + \frac{\sinh[\delta \frac{i-X}{I}]}{\sinh(\delta \frac{X}{I})} \right\} \quad (9.4.24)$$

Multidirectional and Transfinite Interpolation

Let $P_i(\underline{r})$ be a unidirectional interpolation function in the i direction which matches \underline{r} on the N lines $i = i_n$ ($n = 1, 2, \dots, N$). Let $P_j(\underline{r})$ be a unidirectional interpolation function in the j direction which matches \underline{r} on the M lines $j = j_m$ ($m = 1, 2, \dots, M$) (Fig. 9.10).

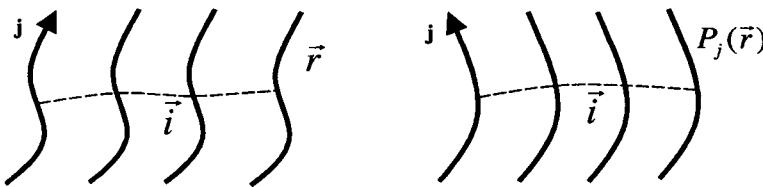


Fig. 9.10. Multidirectional interpolation in i and j directions.

These interpolations are performed by projectors P_i and P_j , which may be simple linear operators. The *product projector* $P_i[P_j(\underline{r})]$ matches the function $P_j(\underline{r})$, instead of \underline{r} , on the N lines $i = i_n$. Since $P_j(\underline{r})$ matches \underline{r} on the M lines $j = j_m$, it follows that the product projector matches \underline{r} at the $N \times M$ points (i_n, j_m) . The same conclusion can be reached with the product projector $P_j[P_i(\underline{r})]$ indicating that the projectors P_i and P_j commute.

The *sum projector* $P_i(\underline{r}) + P_j(\underline{r})$ matches $\underline{r} + P_j(\underline{r})$ on the N lines $i = i_n$ and matches $\underline{r} + P_i(\underline{r})$ on the M lines $j = j_m$. It follows that the projector $P_j(\underline{r}) + P_j(\underline{r}) - P_i[P_j(\underline{r})]$ will match \underline{r} on the N lines $i = i_n$ since $P_i[P_j(\underline{r})]$ matches $P_j(\underline{r})$ on these lines. In the same manner, the projector $P_j(\underline{r}) + P_j(\underline{r}) - P_j[P_i(\underline{r})]$ will match \underline{r} on the M lines $j = j_m$. Therefore, since $P_i P_j = P_j P_i$, the *Boolean sum projector*

$$P_i \oplus P_j = P_i + P_j - P_i P_j \quad (9.4.25)$$

will match r on the $N+M$ lines $I = i_n$ and $j = j_m$ including the entire boundary of the region.

In summary, the individual projectors P_i and P_j interpolate unidirectionally between two opposing boundaries, the product projector $P_i P_j$ interpolates in two directions from the four corners and the Boolean sum projector $P_i \oplus P_j$ interpolates from the entire boundary (Fig. 9.11).

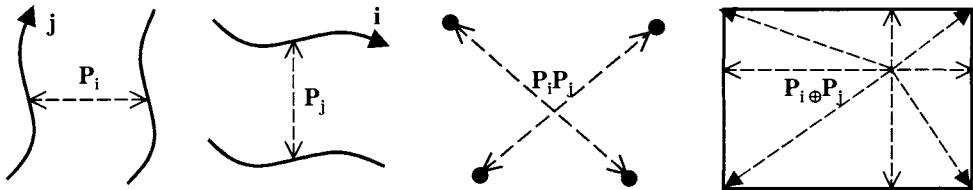


Fig. 9.11. Algebraic interpolation; (a) Individual projector P_i ; (b) Individual projector P_j ; (c) Product projector $P_i P_j$; (d) Boolean projector $P_i \oplus P_j$.

The structure of operators given above allows multidirectional interpolations to be constructed systematically from unidirectional interpolations. This interpolation that matches the function on the entire boundary is called “*Transfinite Interpolation*”.

In two dimensions, the Transfinite Interpolation can be implemented in the following sequence:

1. Interpolation of \underline{r} in the i -direction: $\vec{F}_1 = P_i \vec{r}$
2. Evaluation of the discrepancy between \underline{r} and this result on the j -lines that will be used in the j -interpolation: $(\vec{r} - \vec{F}_1)$
3. Interpolation of the discrepancy in the j -direction: $\vec{F}_2 = P_j (\vec{r} - \vec{F}_1)$
4. Addition of the results of this j -interpolation to the results of the i -interpolation: $\vec{r}(i, j) = \vec{F}_1 + \vec{F}_2$

P_i and P_j can be any one of the unidirectional operators studied earlier, i.e. Lagrange interpolations, Hermite interpolations, splines, or non-polynomial interpolations such as the hyperbolic tangent function.

The methodology described above can be used to generate grids from any four arbitrary bounding curves [4]. The program can handle any grid topology (C, O or H, see Section 9.6) since the boundaries can be any arbitrary curve. Figure 9.12 shows different grids produced around an ellipse using a C-grid topology. Figure 9.12a shows a grid generated using unidirectional interpolation with linear Lagrange polynomials. The input to the program is a sequence of inner, outer, left and right boundaries. The inner ($j = 1$) boundary is the contour of the ellipse plus the branch cut of the C-grid. The outer ($j = \text{JMAX}$) boundary is the far field contour around the grid excluding the right hand vertical boundary. The left ($i = 1$) and right ($i = \text{IMAX}$) boundaries are the lower and upper halves of the downstream vertical boundary. Figure 9.12b shows the grid obtained by unidirectional interpolation using Hermite polynomials. Hermite interpolations allow the slopes of the grid lines at the boundaries to be specified. In this case, they are set to correspond to near-orthogonality at the boundaries. Using Hyperbolic Tangent spacing to concentrate the grid lines near the inner boundary results in grids shown in Fig. 9.12c for Lagrange interpolation and in Fig. 9.12d for Hermite interpolation.

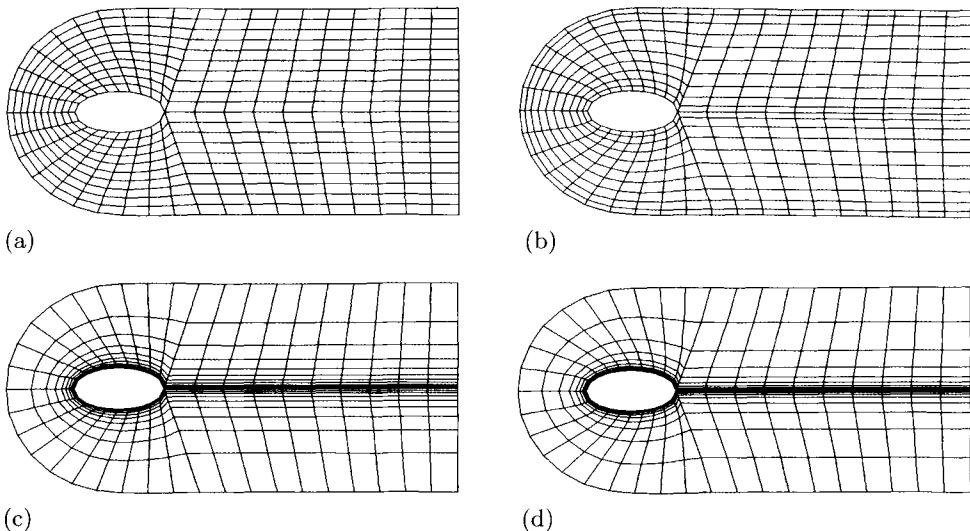


Fig. 9.12. (a) C-grid around ellipse: Unidirectional Lagrange Interpolation, (b) C-grid around ellipse: Unidirectional Hermite Interpolation, (c) C-grid around ellipse: Unidirectional Lagrange Interpolation with Hyperbolic Tangent Spacing, (d) C-grid around ellipse: Unidirectional Hermite Interpolation with Hyperbolic Tangent Spacing.

One problem with algebraic grid generation is that the mapping propagates boundary singularities such as corners into the interior of the domain. Also, if the mapping is not carefully done, situations can occur where grid lines overlap. This can be corrected by refining the parameters defining the boundaries or by adding constraint lines inside the domain. The main advantage of algebraic grid generation is that it is very fast and is the only competitive method in three dimensions. The grids produced by Transfinite Interpolation are usually smoothed using a grid generator solving partial differential equations.

In Appendix B we present a computer program based on algebraic methods for generating grids. The options in this program include transfinite interpolation and Hermite interpolation.

9.5 Differential Equation Methods

Thompson [1] has proposed grid generation and mapping techniques using the solution of differential equations. These methods differ from the algebraic and complex variable methods in that the transformation relations are determined automatically by the finite-difference solution of a set of partial differential equations. For two-dimensional mapping, two elliptic partial-differential equations are solved. The Cartesian coordinate system is usually used with the (x, y) independent variables in the physical plane and the (ξ, η) independent variables in the computational plane. However, the mapping is not limited to Cartesian

coordinates only. For example, mapping can be done from the (r, θ) cylindrical coordinates in the physical plane to the (η, ξ) cylindrical coordinates (i.e., η representing the radial variables and ξ the polar coordinate) in the computational domain.

As discussed in Section 9.2, to map an irregularly shaped region from the physical plane to the computational plane, the values of ξ and η were specified at every boundary segment of the physical region as $\eta = \text{constant}$, varying monotonically and vice versa. As a result, the correspondence between the boundary coordinates of the physical and computational regions is known. Then the problem becomes one of determining the correspondence between the coordinates (x, y) and (ξ, η) at the interior points of the physical and computational regions. The distribution of the points on the interior is determined by solving the following two Poisson equations:

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = P(\xi, \eta) \quad (9.5.1a)$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = Q(\xi, \eta) \quad (9.5.1b)$$

where the non-homogeneous terms $P(\xi, \eta)$ and $Q(\xi, \eta)$ are called the *control functions*. With proper selection of the P and Q terms, the coordinate lines ξ and η can be concentrated towards a specified coordinate line or about a specific grid point. In the absence of these functions, the coordinate lines will tend to be equally spaced in the regions away from the boundaries regardless of the concentration of the grid points along the boundaries. The boundary conditions needed to solve Eqs (9.5.1) are determined from the requirement that the values of ξ and η are specified at every boundary segment of the physical domain.

While Eqs. (9.5.1) describe the basic coordinate transformation between the (x, y) and (ξ, η) coordinate systems, all numerical calculations for the physical problem are performed in the computational plane which has a simple regular geometry. Then, the problem becomes one of seeking the (x, y) values of the physical plane corresponding to the (ξ, η) grid locations in the computational plane. For this reason Eqs. (9.5.1) are transformed to the computational plane and the unknowns (x, y) are determined from the following two elliptic equations:

$$\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2} + J^2 \left(P \frac{\partial x}{\partial \xi} + Q \frac{\partial x}{\partial \eta} \right) = 0 \quad (9.5.2a)$$

$$\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2} + J^2 \left(P \frac{\partial y}{\partial \xi} + Q \frac{\partial y}{\partial \eta} \right) = 0 \quad (9.5.2b)$$

The geometric coefficients α , β and γ and the Jacobian J are given by

$$\alpha = x_\eta^2 + y_\eta^2 \quad (9.5.3a)$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \quad (9.5.3b)$$

$$\gamma = x_\xi^2 + y_\xi^2 \quad (9.5.3c)$$

$$J = x_\xi y_\eta + x_\xi y_\eta \quad (9.5.3d)$$

The boundary conditions for the above equations are the values of the (x, y) positions of the grid points on the boundaries of the physical region in the (x, y) plane when they are transformed to the corresponding locations along the boundaries of the region in the computational plane. Therefore, once (x, y) values are specified along the boundary segments of the region in the computational plane, Eqs. (9.5.2) can be solved by finite-difference methods to determine the values of (x, y) at each grid point at the interior of the region. For further details, the reader is referred to Thompson [1].

The Poisson equations (9.5.2) can be written in generalized form as:

$$g_{22}(\vec{r}_\xi \cdot \vec{r}_\xi + P\vec{r}_\xi) + g_{11}(\vec{r}_\eta \cdot \vec{r}_\eta + Q\vec{r}_\eta) - 2g_{12}\vec{r}_\xi \cdot \vec{r}_\eta = 0 \quad (9.5.4)$$

where, \vec{r} is the coordinate vector in physical space $\vec{r} = x\vec{i} + y\vec{j}$ and the g_{ij} are the covariant metric components:

$$\begin{aligned} g_{11} &= x_\xi^2 + y_\xi^2 = \vec{r}_\xi \cdot \vec{r}_\xi \\ g_{22} &= x_\eta^2 + y_\eta^2 = \vec{r}_\eta \cdot \vec{r}_\eta \\ g_{12} &= x_\xi x_\eta + y_\xi y_\eta = \vec{r}_\xi \cdot \vec{r}_\eta \end{aligned} \quad (9.5.5)$$

Two-Dimensional Grid Generation Using the Poisson Equations

Let us consider the generation of a C-grid around an isolated airfoil. An important feature of grid generation programs is the inclusion of control functions in order to optimize grid stretching and orthogonality. This leads to the solution of the Poisson equations (9.5.2). There are several types of control functions. Some control only the stretching, others only the orthogonality, some others both. Some control functions will cluster grid points or lines in a local area, without altering the overall grid structure. There are different forms of control functions that perform the same task [1]. One example of control functions is the form employed by Sorenson [5]:

$$P(\xi, \eta) = p(\xi) \cdot e^{-a\eta} + r(\xi) \cdot e^{-b(\eta_{max} - \eta)} \quad (9.5.6a)$$

$$Q(\xi, \eta) = q(\xi) \cdot e^{-a\eta} + s(\xi) \cdot e^{-b(\eta_{max} - \eta)} \quad (9.5.6b)$$

where a and b are positive constants. The first terms in the above expressions control the grid characteristics at the inner boundary ($\eta = 0$), and the second terms control the characteristics at the outer boundary ($\eta = \eta_{max}$). The constants a and b , specified by the user, control the degree of propagation of these control functions away from these two boundaries. In all cases, a and b must be

such that the values of the inner boundary control functions become vanishingly small at the outer boundary, and the outer boundary control functions become vanishingly small at the inner boundary.

Thus, we have:

$$\begin{aligned} P(\xi, 0) &= p(\xi) \\ P(\xi, \eta_{\max}) &= r(\xi) \\ Q(\xi, 0) &= q(\xi) \\ Q(\xi, \eta_{\max}) &= s(\xi) \end{aligned} \tag{9.5.7}$$

These components of the control functions are derived as follows. Two conditions are imposed at the boundaries, i.e. orthogonality and specified grid spacing. Since conditions are imposed at the $\eta = 0$ and $\eta = \eta_{\max}$ boundaries, the spacing is along the η coordinate. If S_η is the desired spacing along this coordinate, the condition can be written as:

$$\vec{r}_\eta \cdot \vec{r}_\eta = x_\eta^2 + y_\eta^2 = S_\eta^2 \tag{9.5.8}$$

The orthogonality condition is expressed as follows:

$$\vec{r}_\xi \cdot \vec{r}_\eta = x_\xi x_\eta + y_\xi y_\eta = 0 \tag{9.5.9}$$

On the $\eta = 0$ and $\eta = \eta_{\max}$ boundaries, all the derivatives with respect to ξ are known, since they involve only boundary points. Hence, in the equations (9.5.8) and (9.5.9) above, the x_ξ and y_ξ terms are known, and we thus have two equations in two unknowns, yielding the following expressions for the x_η and y_η terms:

$$x_\eta = -S_\eta y_\xi / (x_\xi^2 + y_\xi^2)^{1/2} \tag{9.5.10a}$$

$$y_\eta = S_\eta x_\xi / (x_\xi^2 + y_\xi^2)^{1/2} \tag{9.5.10b}$$

and

$$\vec{r}_\eta = x_\eta \vec{i} + y_\eta \vec{j} \tag{9.5.10c}$$

Thus, \vec{r}_η is known on the boundary. Furthermore, it can be shown that if equations (9.5.5) and (9.5.9) are combined with equation (9.5.4), the latter reduces to the following on the boundary:

$$(\vec{r}_\eta \cdot \vec{r}_\eta)(\vec{r}_\xi \cdot \vec{r}_\xi) + (\vec{r}_\xi \cdot \vec{r}_\xi)(\vec{r}_\eta \cdot \vec{r}_\eta)|_{\text{boundary}} = 0 \tag{9.5.11}$$

Multiplying the above equation by \vec{r}_ξ or by \vec{r}_η and again using the condition of orthogonality, yields the following two equations for the control functions:

$$P|_{\text{boundary}} = -(\vec{r}_\xi \cdot \vec{r}_\xi \xi) / (\vec{r}_\xi \cdot \vec{r}_\xi) - (\vec{r}_\xi \cdot \vec{r}_\eta \eta) / (\vec{r}_\eta \cdot \vec{r}_\eta)|_{\text{boundary}} \tag{9.5.12a}$$

$$Q|_{\text{boundary}} = -(\vec{r}_\eta \cdot \vec{r}_\eta \eta) / (\vec{r}_\eta \cdot \vec{r}_\eta) - (\vec{r}_\eta \cdot \vec{r}_\xi \xi) / (\vec{r}_\xi \cdot \vec{r}_\xi)|_{\text{boundary}} \tag{9.5.12b}$$

By evaluating the above equations on the boundaries, the p , q , r , and s terms described in equations (5.5.7) can be computed. All quantities in equations

(9.5.12) are known on the boundaries, except $\vec{r}_{\eta\eta}$ (On a $\xi = \text{constant}$ boundary, the same equations for the control functions result, but with $\vec{r}_{\xi\xi}$ as the unknown quantity). The $\vec{r}_{\eta\eta}$ term must be solved for as part of the solution. An iterative procedure is set up in order to evaluate $\vec{r}_{\eta\eta}$ as the solution progresses. This approach, first introduced by Sorenson [5] is now common, with modifications of the basic concept introduced in various codes.

Once the form of the control functions defined, the iterative procedure for the solution of the system (9.5.4) can proceed as follows:

1. A starting grid is generated by unidirectional linear interpolation from the boundary points. However, since the final grid does not, in general, have uniform spacing, the starting grid is stretched using appropriate algebraic formulas according to input values of the cell spacing at the inner and outer boundaries (ΔS_1 and ΔS_2).
2. Once the starting grid is defined, the system (9.5.4) is solved numerically using Successive Line Over-Relaxation. For the first iteration, zero values are assumed for p , q , r , and s . All the fixed derivatives appearing in equations (9.5.12) are computed.
3. Given the initial conditions or the results from the previous iteration, $x_{\eta\eta}$ and $y_{\eta\eta}$, at $j = 1$ and $j = j_{\max}$ are computed using special one-sided difference formulas suggested by Sorenson [5]. The functions p , q , r , and s are then evaluated at the boundaries using equations (9.5.7) and (9.5.12). The control functions P and Q are then evaluated at all grid points using relations (9.5.6). For numerical stability, these control functions are under-relaxed to a degree specified by the user.
4. Another step of Successive Line Over Relaxation solution is then performed on the system of equations (9.5.4).
5. Solutions steps 3 and 4 are repeated until convergence is attained.

A typical C-type algebraic grid is shown in Fig. 9.13a. The outer boundaries for this type of grid consist of a semi-circle and a rectangle. The $\eta = 0$ boundary moves forward from the rear boundary to the trailing edge, clockwise around the airfoil, and then rearward again. The $\eta = \text{constant}$ family of lines form open curves resembling the letter C. The $\xi = \text{constant}$ lines join the inner (airfoil) boundary to the outer boundary. This grid was used as the starting grid for the elliptically generated grid shown in Fig. 9.13b. The grid consists of 70×30 nodes, with a spacing of 0.3 times the airfoil chord c at the outer boundary, and 0.015c at the airfoil boundary. Fifty iterations were performed to obtain this grid, with an over-relaxation factor of 1.3 applied on the main equations, and an under-relaxation factor of 0.05 applied to the control functions. The final grid obtained is relatively smooth and exhibits the proper characteristics. Figure 9.13c provides a closer view of the grid in the vicinity of the airfoil, where the orthogonality of the boundary cells can be observed.

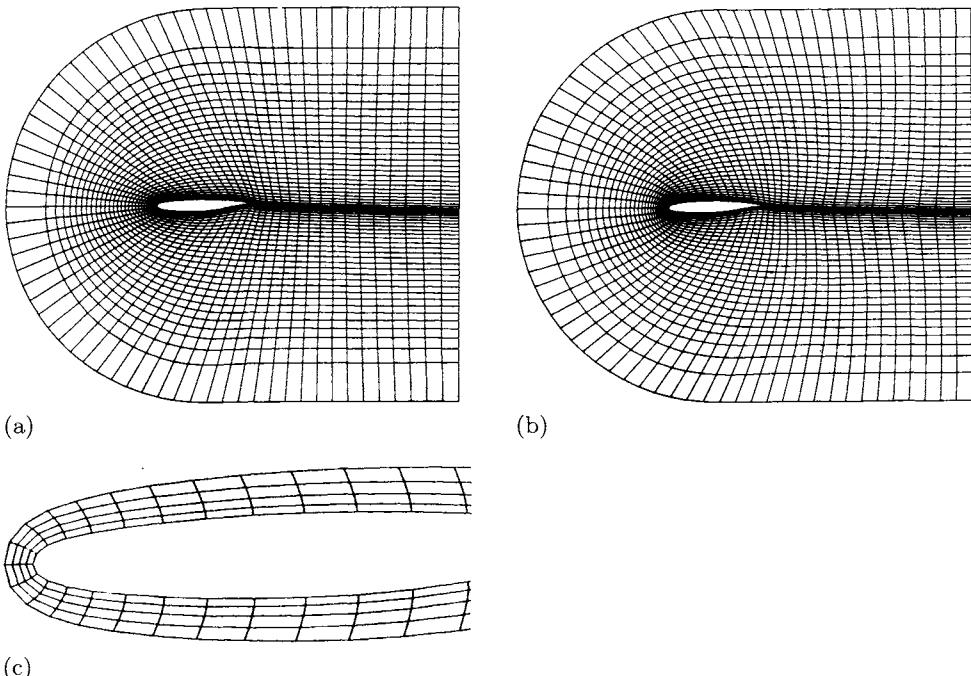


Fig. 9.13. (a) Starting algebraic C-grid around an airfoil section; 70×30 grid points; inner spacing $\Delta S_1 = 0.015c$, outer spacing $\Delta S_2 = 0.3c$, (b) Elliptic C-grid obtained after smoothing the algebraic grid of (a) by the solution of Poisson equations (50 iterations), (c) Close-up of the C-grid showing the application of orthogonality conditions near the leading edge region.

In Appendix B we present a computer program based on the elliptic method for generating grids.

9.6 Conformal Mapping Methods

The methods based on conformal mapping have the advantage that the transformations used are analytical or partially analytical as opposed to the differential equation methods, which are entirely numerical. Their main drawback is their restriction to two-dimensional flows, since they are based on complex variables. Nevertheless, they are very convenient for two-dimensional flows and will now be discussed in some detail.

There are several useful meshes for airfoils, as illustrated in Fig. 9.14. The C-mesh shown in Fig. 9.14a has high density near the leading edge of the airfoil, and good wake resolution. The O-mesh shown in Fig. 9.14b has high density near both the leading edge and trailing edges of the airfoil, and the H-mesh shown in Fig. 9.14c, has two sets of mesh lines similar to the Cartesian mesh and is

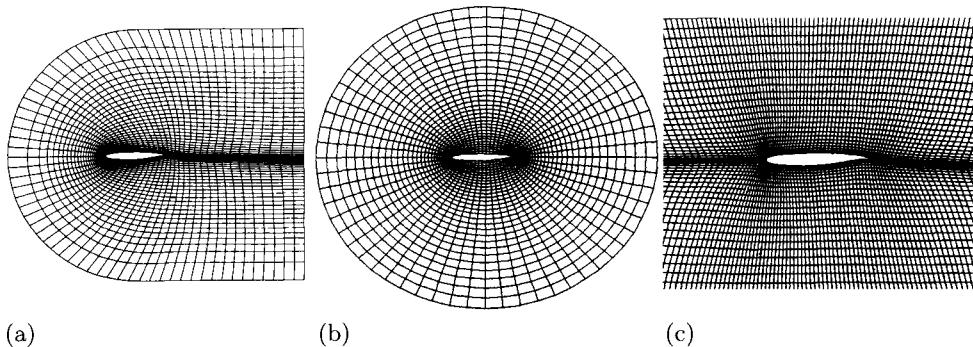


Fig. 9.14. Three common grids for airfoils. (a) C-grid, (b) O-grid, and (c) H-grid.

the easiest to generate. Its mesh lines are aligned well with the approximate streamlines.

9.6.1 Parabolic Mapping Function

The C-mesh can be generated by a number of mapping functions, one being the parabolic mapping function defined by:

$$2(x + iy) = (\xi + i\eta)^2 \quad (9.6.1)$$

or

$$2x = \xi^2 - \eta^2, \quad y = \xi\eta \quad (9.6.2)$$

The inverse transformation can be obtained by solving Eq. (9.6.2) for ξ and η as functions of x and y ,

$$\xi^2 = \sqrt{x^2 + y^2} + x, \quad \eta^2 = \sqrt{x^2 + y^2} - x \quad (9.6.3)$$

The set of $\xi^2 = \text{const}$ and $\eta^2 = \text{const}$ curves form an orthogonal curvilinear net in the x, y coordinate system consisting of two families of intersecting parabolas with common foci at the origin. To facilitate the transformation of coordinate points from the physical plane to the computational plane, it is convenient to use the polar coordinate system in both planes and define

$$\xi = r \cos \theta, \quad \eta = r \sin \theta \quad (9.6.4)$$

where

$$r = \sqrt{\xi^2 + \eta^2}, \quad \theta = \tan^{-1} \frac{\eta}{\xi} \quad (9.6.5)$$

Substitution of Eq. (9.6.4) into Eq. (9.6.1) gives

$$2x = r^2 \cos^2 \theta - r^2 \sin^2 \theta = r^2 \cos 2\theta, \quad y = \frac{1}{2} r^2 \sin 2\theta \quad (9.6.6)$$

in the x, y coordinate system. Noting that,

$$\frac{y}{x} = \tan \phi \quad (9.6.7)$$

the polar angles in both coordinate systems are related by

$$\phi = 2\theta \quad (9.6.8)$$

It follows from Eq. (9.6.3) and from the definition of r in Eq. (9.6.5) that the radius vector in the computational plane is

$$r^2 = 2\sqrt{x^2 + y^2} \quad (9.6.9)$$

Since the radius vector R in the physical plane is

$$R = \sqrt{x^2 + y^2} \quad (9.6.10)$$

the relation between r^2 and R is

$$r^2 = 2R \quad (9.6.11)$$

The (ξ, η) points in the computational plane given by Eq. (9.6.4) can now be written as

$$\xi = \sqrt{2R} \cos \left(\frac{\phi}{2} \right), \quad \eta = \sqrt{2R} \sin \left(\frac{\phi}{2} \right) \quad (9.6.12)$$

In conformal mapping, the singular point is defined to be the point where the mapping function fails. In our case it is at the origin of the coordinate system, $x = 0, y = 0$. It is usually avoided by choosing it to be at a distance half the airfoil nose radius, r_0 , from the leading edge in the physical plane. If the airfoil contour is given analytically the nose radius can be obtained by calculating the curvature at the nose. If only tabular values of airfoil coordinates are available, the nose radius can be computed by fitting a circle to three points nearest to the leading edge. An alternate way is to plot y/\sqrt{x} vs. \sqrt{x} and extrapolate the resulting curve to $\sqrt{x} = 0$, in which case the nose radius r_0 is computed from the extrapolated y/\sqrt{x} value by:

$$r_0 = \frac{1}{2} \left(\frac{y}{\sqrt{x}} \right)_0^2 \quad (9.6.13)$$

Once the origin of the (x, y) coordinate system in the physical plane is fixed, then the (ξ, η) points, corresponding to the airfoil surface coordinates (x_p, y_p) can be computed as follows:

(a) Determine ϕ from Eq. (9.6.7), that is

$$\tan \phi_P = \frac{y_P}{x_P}$$

(b) Compute R from Eq. (9.6.10). that is,

$$R = \sqrt{x_P^2 + y_P^2}$$

(c) Determine ξ and η at p from Eq. (9.6.12), that is,

$$\xi_P = \sqrt{2R} \cos\left(\frac{\phi}{2}\right), \quad \eta_P = \sqrt{2R} \sin\left(\frac{\phi}{2}\right)$$

The calculation of the (ξ, η) net for off-body points requires the determination of $\xi = \text{const}$ and $\eta = \text{const}$ lines. One procedure is to start at $\eta = \eta_P$, where ξ_P is known, and vary η in small increments and compute r and θ from Eq. (9.6.5). This procedure is repeated for other values of ξ_P in uniform increments in η except for the region close to the airfoil surface where the η -increments are determined by the body points. Note that in the region behind the airfoil $y_P = 0$, $\eta_P = 0$ and $\xi_P^2 = 2x_P$.

The ξ_P -constant and η -constant lines determined in this manner produce a parabolic grid in the computational plane. The inverse transformation follows from Eq. (9.6.2); for each value of ξ and η , x and y are determined in the physical plane.

9.6.2 Wind Tunnel Mapping Function

The C-mesh generated by the parabolic mapping function discussed above is essentially a set of confocal, orthogonal parabolas wrapping around the airfoil. With this mapping the outer boundary of the grid given by the last parabola extends far away from the airfoil surface. This feature may be unnecessary and/or undesirable in some practical applications. A different mapping function that addresses this problem is the infinite “wind-tunnel” mapping function defined by

$$Z = \ln(1 - \cosh \zeta) \tag{9.6.14}$$

Where

$$Z = x + iy \tag{9.6.15a}$$

$$\zeta = \xi + i\eta \tag{9.6.15b}$$

and

$$\cosh \zeta = 1 - e^Z \tag{9.6.16}$$

In this transformation, the singular point located at half the leading edge radius is at $(\ln 2, 0)$ and all values of y lie between $\pm\pi$. As shown in Fig. 9.11a, this mapping allows the airfoil and its wake, indicated by the contour ABCDE, to be transformed into a nearly straight $\eta = \pi$ line in the computational plane. The far-field boundaries HI and GF are mapped to the $\eta = 0$ line. Points on the upper surface of the airfoil are mapped to the left of the η -axis and points on the lower surface are mapped to the right of the η -axis.

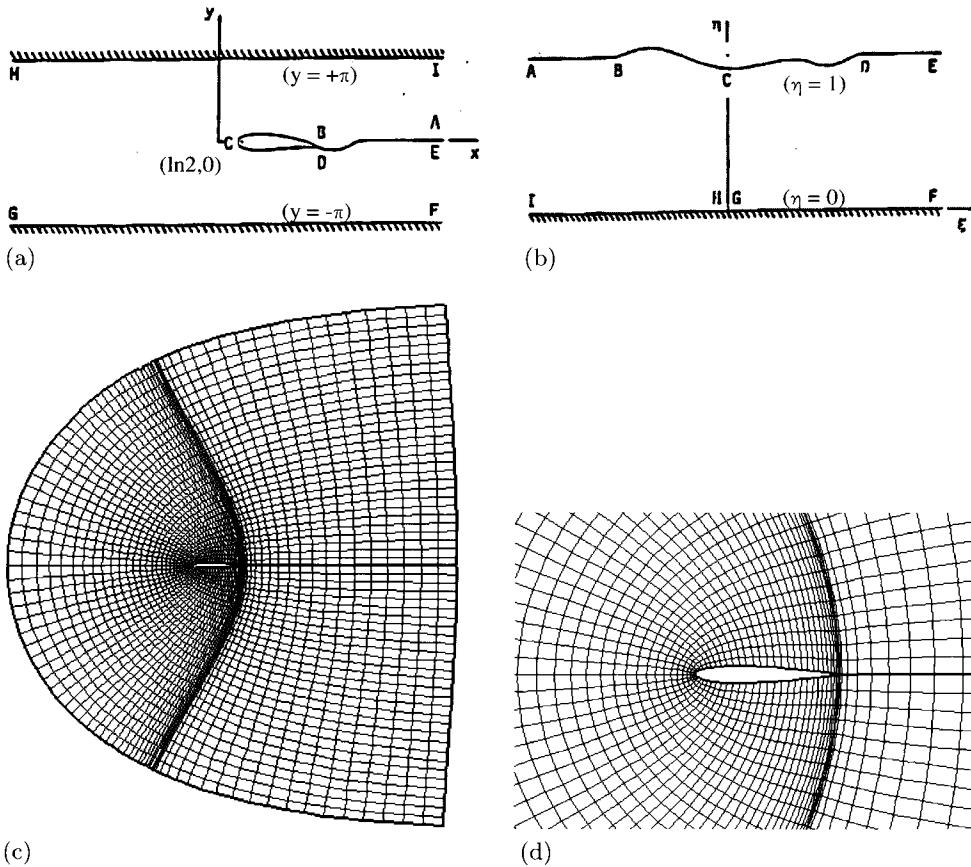


Fig. 9.15. Wind tunnel mapping for an airfoil: (a) physical plane, (b) computational plane. C-grid generated around a NACA airfoil using the wind tunnel mapping for an airfoil (c) physical plane, (d) physical plane, close-up of the grid near the airfoil.

To express the variables x and y as functions of ξ and η , the real and imaginary parts of (9.6.14) are equated and the expressions solved for x and y :

$$x = \frac{1}{2} \ln(K_1^2 + K_2^2) \quad (9.6.17a)$$

$$y = \tan^{-1} \left(\frac{K_2}{K_1} \right) \quad (9.6.17b)$$

Where:

$$K_1 = e^x \cos y = 1 - \cosh \xi \cos \eta \quad (9.6.18a)$$

$$K_2 = e^x \sin y = -\sinh \xi \sin \eta \quad (9.6.18b)$$

The inverse transformation or expressions of ξ and η in terms of x and y can be obtained in the form:

$$\xi = \frac{1}{2} \ln(D_3^2 + D_4^2) \quad (9.6.19a)$$

$$\eta = \tan^{-1} \left(\frac{D_4}{D_3} \right) \quad (9.6.19b)$$

Where:

$$D_3 = R^{1/2} \cos \left(\frac{\theta}{2} \right) - e^x \cos y + 1 \quad (9.6.20a)$$

$$D_4 = R^{1/2} \sin \left(\frac{\theta}{2} \right) - e^x \sin y \quad (9.6.20b)$$

$$R = e^x \left[D_1^2 + D_2^2 \right]^{1/2} \quad (9.6.20c)$$

$$\theta = \tan^{-1} \left(\frac{D_2}{D_1} \right) \quad (9.6.20d)$$

$$D_1 = e^x \cos 2y - 2 \cos y \quad (9.6.20e)$$

$$D_2 = e^x \sin 2y - 2 \sin y \quad (9.6.20f)$$

As a test case, consider the NACA 0012 airfoil, the coordinates of which are given by:

$$\bar{y} = \pm 0.6 \left[0.2969\sqrt{\bar{x}} - 0.126\bar{x} - 0.3516\bar{x}^2 + 0.2843\bar{x}^3 - 0.1015\bar{x}^4 \right] \quad (9.6.21)$$

Here, x is the distance from the leading edge normalized by the chord c and y is normalized thickness. To determine the nose radius, write:

$$\frac{\bar{y}}{\sqrt{\bar{x}}} = 0.6 \left[0.2969 - 0.126\sqrt{\bar{x}} - 0.3516\bar{x}^{3/2} + \dots \right] \quad (9.6.22)$$

In the limit, as $\bar{x} \rightarrow 0$, $\bar{y}/\sqrt{\bar{x}} = 0.6 \times 0.2969 = 0.17814$ and the normalized nose radius is:

$$\bar{r}_0 = \frac{1}{2}(0.6 \times 0.2969)^2 = 0.015867 \quad (9.6.23)$$

The shift in coordinates is obtained by replacing

$$\bar{x} = x + \frac{r_0}{2} \quad (9.6.24)$$

Appendix B gives a listing of a program to generate a C-mesh by the wind tunnel mapping for the NACA0012 airfoil. In this program XW and YW denote ξ and η respectively. NX and NY denote the number of stations in x and y directions, respectively, and the parameter HTC denotes the height-to-chord ratio and is used to control the range of x and ξ . The larger the value of HTC, the smaller the airfoil becomes in the physical domain, as shown in Fig. 9.15a. In general, HTC varies from 15 to 20 for full potential methods and it varies from 50 to 100 for Euler methods. It can also be used to control the grid in the η -direction. As a rule of thumb, the larger the value of HTC, the larger the distance from the wall to the outer boundary. Figure 9.15c shows a sample mesh generated with HTC of 15 and a singular point at the center of the leading edge circle (RLE) of the NACA0012 airfoil. Figure 9.15d shows a close up of the mesh near the airfoil.

9.7 Unstructured Grids

The unstructured grids of the previous section are typically quadrilaterals in two dimensions. The generation of these grids involves complex iterative smoothing techniques that attempt to align elements with the boundaries of physical domains. This task can become very difficult if the geometry becomes very complex. It is often necessary to decompose this complex domain into blocks with simpler topologies and generate structured grids inside each block in turn, a method called the multi-block approach. Unstructured grids can also be used to provide a good representation of complex boundaries. Unstructured grids are usually made of triangles in two dimensions. Because of the immense scope of the field of unstructured mesh generation, this section will simply introduce the more fundamental aspects. Figure 9.16 shows a typical unstructured grid around a two-dimensional multi-element airfoil.

There are several algorithms for generating unstructured grids. The Delaunay triangulation method and the advancing front method are the most popular. The field is in rapid expansion, and there are schools of thought whether the unstructured approach is better or worse than the structured approach to

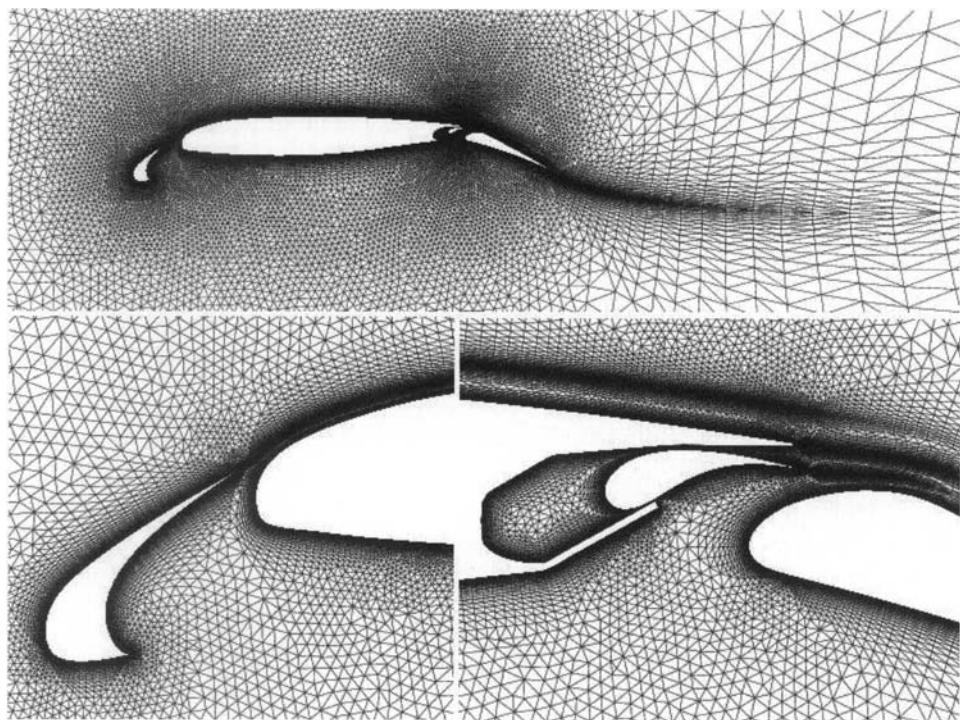


Fig. 9.16. 2D Unstructured grid for Navier–Stokes computations of a multi-element airfoil generated with the hybrid advancing front Delaunay method of Mavriplis [6].

the solution of Partial Differential Equations in fluid dynamics. The advantage of structured grids is the simplicity and the straightforward treatment of the grid in the solution formulation. The disadvantages are the mesh generation constraints for complex configurations. Unstructured grids can be generated faster on most complex domains. Mesh refinement can be done without difficulties, locally and adaptively. Storage of the grid data (it requires information on which node is neighbor to which) takes far more memory than that of a structured grid, and therefore hinders parallelization of computer codes.

9.7.1 Delaunay Triangulation

In structured grids, the connections between points are defined automatically given the (i, j, k) ordering. Such ordering does not exist in unstructured grids. Therefore, connections between points, in addition to the position of points, have to be defined by an unstructured grid method. Delaunay triangulation methods use a particularly simple criterion for connecting points to form conforming, non-intersecting elements. The geometrical construction has been known for many years, but was used only recently for CFD grid generation. The geometrical criterion provides only a mechanism for connecting points. The task of point generation must be considered independently. Grid generation by Delaunay triangulation involves therefore two distinct problems: point connection and point creation.

In 1850, Dirichlet proposed a method for decomposing systematically a given domain, in arbitrary space, into a set of packed convex regions [7]. For a given set of points P , the space is subdivided into regions in such a way that each region is the space closer to a point P than to any other point. This geometrical construction of tiles is known as the *Dirichlet tessellation*. The tessellation of a closed domain results in a set of non-overlapping convex regions called *Voronoi*

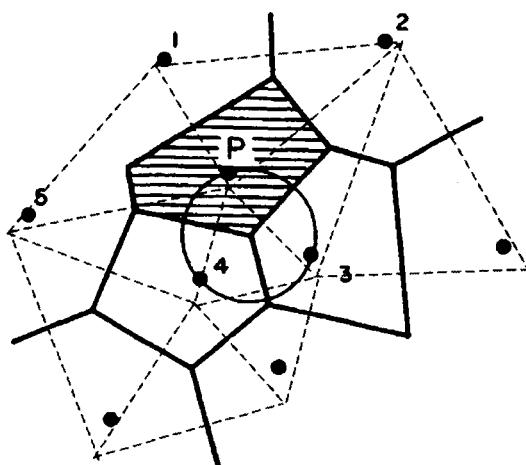


Fig. 9.17. Voronoi diagram and Delaunay triangulation (dashed line triangles) of a set of points.

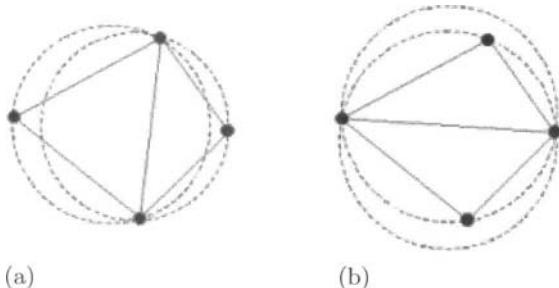


Fig. 9.18. Example of Delaunay criterion; (a) satisfies the criterion while (b) does not.

regions V_i , defined as the set of points that are closer to a point P_i than to any other point [8]. Figure 9.17 illustrates that in two dimensions, the sides of the Voronoi polygon around point P is made of segments of the bisectors of lines joining P to all its neighboring points. If all pairs of points that have a segment of a Voronoi polygon in common are joined by straight lines, the result is a triangulation within the convex hull of the set of points known as the *Delaunay triangulation* $T(P_i)$ [9]. In three dimensions, the territorial boundaries are faces that form Voronoi polyhedra and are equidistant between point pairs. If points with a common face are connected, then a set of tetrahedra is formed that covers the convex hull of points.

One interesting property of the Delaunay triangulation is the in-circle criterion. The circumcircle of a triangle is the circle passing through the three vertices of the triangle. It can be seen that each vertex of a Voronoi diagram is the circumcenter of the triangle formed by three points. *The in-circle criterion states that the circumcircle through each triangle of a Delaunay triangulation contains no points other than its forming points* (Fig. 9.18). This applies to any number of dimensions and is the property used to construct algorithms for the triangulation.

Bowyer-Watson Algorithm

There are several algorithms used to construct the Delaunay triangulation. One approach, which is flexible in that it readily applies to two and three dimensions, is due to Bowyer and Watson and is explained by Baker in Reference [10]. It is an incremental algorithm that exploits the circle criterion of the Delaunay triangulation as follows.

Consider T_n , the Delaunay triangulation of a set of n points, $V_n = \{P_i \mid i = 1 \text{ to } n\}$. For any simplex S belonging to T_n , let R_s be the radius and Q the center of the circumcircle. Now introduce a new point P_{n+1} inside the convex hull of V_n and define $B = \{S \mid S \in T_n, d(P_{n+1}, Q) < R_s\}$ where $d(P, Q)$ is the distance between points P and Q . B is not empty, since P_{n+1} is inside the convex hull of V_n and therefore inside some simplex S' belonging to T_n . It follows that $S' \in B$.

The region C formed when B is removed from T is simply connected, contains P_{n+1} (since P_{n+1} is inside $S' \in B$), and the new point P_{n+1} is visible from all points on the boundary of C . It is therefore possible to generate a triangulation of the set of points $V_{n+1} = V_n \cup \{P_{n+1}\}$ by connecting P_{n+1} to all points on the boundary of C . This triangulation is precisely the Delaunay triangulation T_{n+1} .

The implementation of the Bowyer-Watson algorithm in two dimensions starts with a super-triangle, or super-square partitioned into five triangles, which contains all the other points. The remaining points, which comprise the grid to be triangulated, are introduced one at a time and the Bowyer-Watson algorithm is applied to create the Delaunay triangulation after each point insertion. Two lists are maintained for each triangle in the existing structure: one with the forming points of the triangle, the other with the addresses of the neighboring triangles that have a common edge. The second list, which provides information about the contiguities between triangles, allows all triangles belonging to a cavity to be found by means of a tree search, once one such triangle has been found. Without the contiguity information the algorithm would be extremely inefficient. It is also customary to store the radius and the coordinates of the center of the circumcircle for each triangle. The remaining step in the Bowyer-Watson algorithm is the updating of the data structure. Triangles belonging to the set B are deleted from the lists, and new triangles, obtained by connecting the new point to all edges of the cavity boundary, are added. Finally, it is necessary to determine the contiguities that exist among the new triangles, and also between the new triangle and the old triangles that have edges on the cavity boundary.

Consider the problem of generating a boundary conforming grid within a multiply connected domain defined between two concentric ellipses. The ellipses are defined by a set of discrete points. Following the Bowyer-Watson algorithm, these points can be contained within an appropriate hull and then connected together. The result is shown in Fig. 9.19a. A set of valid triangles has been derived that covers the region of the hull. Two issues can be raised immediately. First, to derive a triangulation in the specified region, triangles outside this region must be deleted. Second, in order for the triangles to provide a boundary conforming grid it is necessary that edges in the Delaunay triangulation form the given geometrical boundaries of the domain. Unfortunately, given a set of points that define pre-specified boundaries there is no guarantee that the resulting Delaunay triangulation will contain the edges defining the domain boundaries. It is necessary, therefore, to check the integrity of boundaries, and if found not to be complete, appropriate steps must be taken.

A combination of *edge swapping* can be used to recover boundary edges in two dimensions. Once the boundary is complete, it is a simple task to delete triangles exterior to the region of interest. Deletion of unwanted triangles in

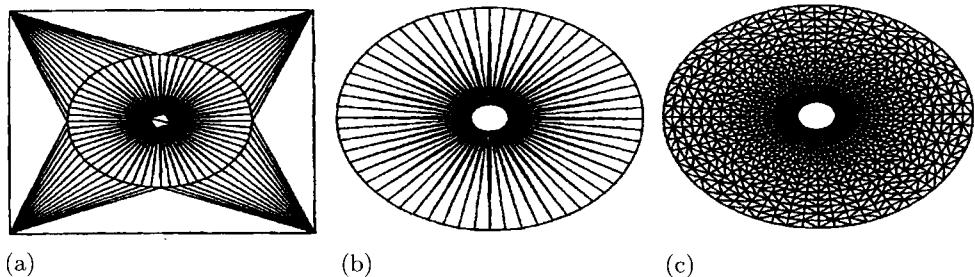


Fig. 9.19. Application of Delaunay triangulation to the generation of unstructured grid in the space between two concentric ellipses; (a) construction of the convex hull and initial Delaunay triangulation; (b) Construction after removal of convex hull points; (c) Final Delaunay triangulation of points from a structured background grid.

Fig. 9.19a yields the triangulation shown in Fig. 9.19b. Although Fig. 9.19b represents a valid triangulation of the points that define the two concentric ellipses, these triangles span the entire region and are inappropriate for any form of analysis. It is necessary therefore necessary to create more points inside the domain.

The additional points for connection by the Delaunay algorithm can be created by a variety of different techniques. For example, in the case of the two ellipses, a *structured background grid* can be generated and the set of points then connected together to form the grid, as shown in Fig. 9.19c. Another method is the creation of points by *grid superposition and successive subdivisions*. The basic idea is to superimpose a regular grid over the domain. The regular grid can be generated using a data structure that allows point density in the regular grid to be consistent with point spacing at the boundary. In general, this approach results in good spatial discretization in the interior of the domain, but in the vicinity of boundaries the grid quality can be poor. Many other techniques for *point insertion* can be found in the extensive literature available on unstructured grids, see [11].

9.7.2 Advancing Front Method

Another very popular family of triangle and tetrahedral grid generation algorithms is the advancing front, or moving front method. One contributor to this method is Rainald Lohner [12]. In this method, the tetrahedra are built progressively inward from the triangulated surface. An active front is maintained where new tetrahedra are formed. Figure 9.20 is a simple two-dimensional example of the advancing front, where triangles have been formed at the boundary. As the algorithm progresses, the front will advance to fill the remainder of the area with triangles. For each triangle edge on the front, an ideal location for a new third node is computed. Also determined are any existing nodes on the front that may form a well-shaped triangle with the edge. The algorithm selects either

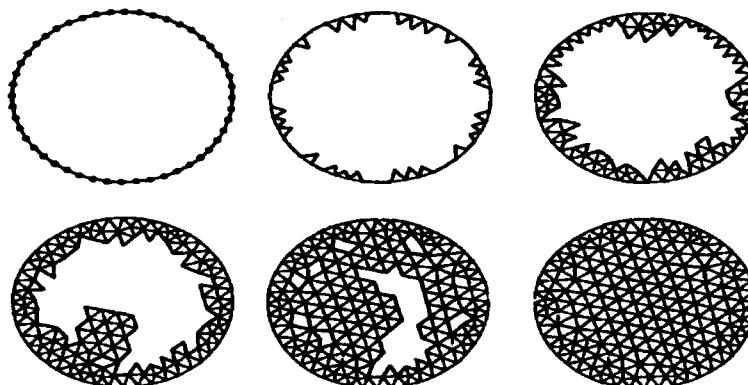


Fig. 9.20. Advancing Front technique for unstructured grid generation.

the new third node or an existing node to form the new triangle based on which will form the best triangle. Also required are intersection checks to ensure that triangles do not overlap as opposing fronts advance towards each other. A sizing function can also be defined in this method to control element sizes. At each step, the front is updated. This front is a dynamic data structure that changes continuously during the generation process. Any straight line segment available to form an element side is termed active. Any segment no longer active is removed from the front. The grid generation is complete when the front is empty. Lohner [12] proposed using a coarse Delaunay mesh of selected boundary nodes over which the sizing function could be quickly interpolated.

In the advancing front method nodes and elements are added simultaneously, with smooth mesh point placement. The method is boundary conforming since the initial front is the boundary. It can fail when there are large variations in grid spacing. The need to search for nearby nodes and edges/faces and the complex intersection checking make the advancing front method slow. In addition, it is better to add points rather than elements because there are, on average, two triangles per point in two dimensions. The Delaunay triangulation method, which adds points, has less searching to perform, and has much simpler geometry checking. It is much faster than the advancing front method and is the preferred method in three-dimensional unstructured grid generation. A more complete overview of the “advancing front” method is given in [13].

References

- [1] Thompson, J. F., Warsi, Z. U. A. and Mastin, C. W., Numerical Grid Generation, Foundations and Applications, North Holland, 1985.
- [2] Smith, R. E. and Weigel, B. L., “Analytic and Approximate Boundary Fitted Coordinate Systems for Fluid Flow Solutions”, AIAA Paper No. 80-0192

- [3] Roberts, A., "Automatic Topology Generation and Generalized B-Spline Mapping", Numerical Grid Generation, Ed. Joe F. Thompson, North-Holland, 465, 1982.
- [4] Boudreau, J., Piperni, P., Mokhtarian, F., Kafyeke, F., "Grid generation and Euler calculations on the CF-18 aircraft", CASI Aerodynamics Symposium, 6th, Toronto, Canada, Apr. 28-30, 1997, Proceedings (A98-38776 10-02), Ottawa, Canadian Aeronautics and Space Institute, pp. 29–34, 1997
- [5] Sorenson, R. L. "A computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equations", NASA TM81198, 1980.
- [6] Mavriplis, D. J., "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness", Journal of Computational Physics, 117, pp. 90–101, 1995.
- [7] Dirichlet, G. L. , "Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen", Z. Reine Angew. Mathematik, Vol. 40, No. 3, pp. 209–227, 1850.
- [8] Voronoi, G., "Nouvelles applications des paramètres continues à la théorie des formes quadratiques, recherche sur les paralléléoedres primitifs," J. Reine Angew. Mathematik, Vol. 134, 1908.
- [9] Delaunay, B. N., "Sur la Sphere Vide." Bulletin of Academic Science URSS, Class. Science Nationale, Vol 7. pp. 793–800, 1934.
- [10] Baker, T. J., "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation", Engineering with Computers, Vol. 5, pp. 161–175, 1989.
- [11] Baker, T. J., "Delaunay-Voronoi Methods", in Handbook of Grid Generation, Ed. Joe F. Thompson, Bharat K. Soni and Nigel P. Weatherill, CRC Press, 1999.
- [12] Lohner, R. "Progress in Grid Generation via the Advancing Front Technique", Engineering with Computers, Vol. 12, pp. 186–210, 1996.
- [13] Peraire, J., Peiro, J. and Morgan, K., "Advancing Front Grid Generation", in Handbook of Grid Generation, Ed. Joe F. Thompson, Bharat K. Soni and Nigel P. Weatherill, CRC Press, 1999.

10 Inviscid Compressible Flow

10.1 Introduction

In this chapter we extend the discussion on inviscid flow equations for incompressible flows to compressible flows. As in incompressible flows, inviscid compressible flow equations and their solutions have played a central role in the development of CFD methods. As discussed in Chapter 6, the incompressible, irrotational inviscid flow equation can be solved using the Laplace equation, which is elliptic in form. Removing the incompressible flow assumption leads to many difficulties. The first is that the panel method of chapter 6 can no longer be used, except for compressibility corrections, which are limited to small Mach numbers. One major advantage of a panel method is that it requires only the generation of a surface mesh. This is no longer the case for compressible flows and mesh generation can become more complex since the solution of the governing PDE will now require the entire field to be discretized. This is discussed in detail in Chapter 9, but some issues pertaining to the discretization of the flow equations are discussed here.

Another difficulty arising from removing the incompressible assumption is that the equations are hyperbolic for supersonic flows. Supersonic flows allow the solution of discontinuous flows, such as shock wave compressions and, while discontinuous expansions are thermodynamically non-physical, the solution of the compressible continuity equation has no means to provide the correct physical behavior. The relation between the upstream and downstream flow properties across a shock wave are derived in Section 10.2 for the Euler, Full-Potential and Transonic Small Disturbance equations. Adequate shock capturing methods ensuring the proper shock jump relations are discussed in Section 10.3.

Adding to the complexity, transonic flows, that is flows that contain both subsonic and supersonic flow regions, are of mixed elliptic-hyperbolic type and require careful computation of the density from the velocity fields so that non-linearity of the flow field is retained [1]. The numerical solutions must therefore reproduce the properties of the local flow equations and this was a considerable

challenge to the CFD community in the 1970's. The Transonic Small Disturbance (TSD) equation was first solved by Murman and Cole in a landmark paper [2], in which they introduce a switch from elliptic to hyperbolic difference operators. The theory is discussed in Section 10.4 and numerical results on a non-lifting airfoil presented in Section 10.5. This type-differencing scheme was extended to solve the full-potential equation by Jameson [3], and Section 10.6 will describe the added complexity of applying such operators when the flow is not aligned with the streamwise coordinates.

All the previously described difficulties are imbedded in the compressible continuity equation, which involves scalar algebra. Solving the equations for conservation of mass and momentum simultaneously involves matrix algebra, as discussed in Chapter 5. Appropriate boundary conditions must then be imposed, especially at the far-field boundaries and a complete theory using the method of Characteristics has been developed. These issues, presented in Section 10.7, were tackled during the 1980's and the methods developed have lead directly to the solution of the incompressible and compressible Navier-Stokes equations in Chapters 11 and 12, respectively.

Stability analyses of the finite-difference operators discussed in Chapter 5 indicated that upwinding must be used to achieve a stable numerical algorithm but upwinding reduces the accuracy of the discretized equations while being unconditionally stable under some circumstances due to the truncation error terms. To retain second-order accuracy, central difference operators must be coupled to added dissipation operators to obtain converged flow solutions. This dissipation can either be added explicitly (artificial dissipation), or implicitly by the truncation error of the finite-difference operators (numerical viscosity), which are discussed in Section 10.8

The explicit MacCormack scheme applied to the one-dimensional compressible Euler equations is presented in Section 10.9. Applications of the scheme to the unsteady 1-D Euler equations and the steady 1D Euler equation with the addition of a source terms on a nozzle flow computation are found in Sections 10.10 and 10.11, respectively.

The implicit method of Beam-Warming with explicit numerical dissipation is presented in Section 10.12, with applications on the same model problems as discussed above presented in Sections 10.13 and 10.14.

10.2 Shock Jump Relations

The compressible flow equations are typically used to solve flow situations that contain discontinuous flow regimes such as a shock wave. Shock waves are produced when the gas undergoes a sudden compression and the second law of thermodynamics models the entropy jump associated with the compression wave. As opposed to elliptic equations, the compressible flow phenomena are non-

reversible, that is expansion waves are unphysical as they would violate the second law of thermodynamics, which states that entropy must increase.

Physical shock waves are continuous across a very narrow viscous region, and the numerical shock wave computed with an inviscid model is represented as a discontinuous jump. Continuous flow solutions are referred as “genuine” solutions, whereas discontinuous flow solutions are referred as “weak” solutions. Inviscid shock jump relations can be derived for each set of compressible governing equations. For the TSD and full-potential equations, they are referred as the *isentropic jump relations* whereas, in the Euler equations, they are called the *Rankine-Hugoniot relations* after their founders. The isentropic relations imply that the entropy rise through the shock wave is assumed negligible. This has direct consequences on the TSD and full-potential solutions, as they allow incorrect expansion shocks as their weak solutions. This is clearly inadmissible and means of correcting for this deficiency have been devised over the years.

The shock jump relations can be derived starting from a generic conservation equation for steady state

$$\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \quad (10.2.1)$$

where E and F are the fluxes. For the TSD equation, they are [Eq. (P2.6.2)]

$$E = (1 - M_\infty^2)\varphi_x - \frac{\gamma + 1}{2} \frac{M_\infty^2}{U_\infty} \varphi_x^2 \quad (10.2.2a)$$

$$F = \varphi_y \quad (10.2.2b)$$

For the full-potential equation, they are [Eqs. (2.1.12b) and (2.3.19)]

$$E = \varrho\varphi_x \quad (10.2.3a)$$

$$F = \varrho\varphi_y \quad (10.2.3b)$$

where in both cases, $\vec{V} = \vec{\nabla}\varphi$

For the Euler equations, the fluxes are [Eq. (2.2.30)]

$$E = \begin{bmatrix} \varrho u \\ \varrho u^2 + p \\ \varrho uv \\ u(E_t + P) \end{bmatrix} \text{ and } F = \begin{bmatrix} \varrho v \\ \varrho uv \\ \varrho v^2 + p \\ v(E_t + P) \end{bmatrix} \quad (10.2.4)$$

The conservation equation (10.2.1) can be written in divergence form

$$\vec{\nabla} \cdot \vec{F} = 0 \quad (10.2.5)$$

Applying the divergence theorem, as in a finite volume representation of the discretized equation (10.2.5) over the discontinuous control volume representing a shock wave (Fig. 10.1), one obtains

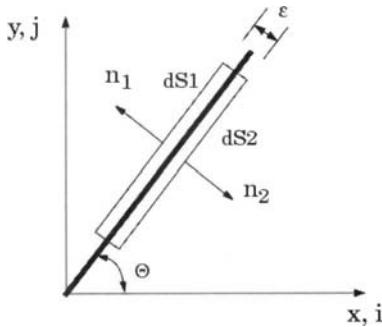


Fig. 10.1. Control volume around a shock wave.

$$\iint (\vec{F} \cdot \vec{n}) dS = 0 = (e_1 \vec{i} + f_1 \vec{j}) \cdot \vec{n}_1 dS_1 + (e_2 \vec{i} + f_2 \vec{j}) \cdot \vec{n}_2 dS_2 \quad (10.2.6)$$

where e and f are the individual fluxes. With the control volume assumed to be infinitely thin, ($\varepsilon \rightarrow 0$)

$$dS_1 = dS_2 = dS \quad (10.2.7)$$

and substituting the geometric relations, i.e. $\vec{i} \cdot \vec{n}_1 = -\sin \theta$ and similar relations for \vec{j} into Eq. (10.2.6), leads to

$$(e_2 - e_1) \tan \theta = f_2 - f_1 \quad (10.2.8)$$

This equation describes the jump conditions of the conservative fluxes across the shock wave. If the fluxes of Eqs. (10.2.2)–(10.2.4) are substituted into (10.2.8) for a one-dimensional flow ($\theta = \pi/2$; $f_2 = f_1 = 0$), the following TSD, Full-Potential and Euler (or Rankine-Hugoniot) normal shock relations are obtained

$$(1 - M_\infty^2)u_1 - \frac{\gamma + 1}{2} \frac{M_\infty^2}{U_\infty} u_1^2 = (1 - M_\infty)u_2 - \frac{\gamma + 1}{2} \frac{M_\infty^2}{U_\infty} u_2^2 \quad (10.2.9)$$

$$\left(1 + \frac{\gamma - 1}{2} M_1^2\right)^{1/(\gamma-1)} u_1 = \left(1 + \frac{\gamma - 1}{2} M_2^2\right)^{1/(\gamma-1)} u_2 \quad (10.2.10)$$

$$(2 + (\gamma - 1)M_1^2)u_1 = (2 + (\gamma - 1)M_2^2)u_2 \quad (10.2.11)$$

In deriving Eq. (10.2.10), the isentropic flow relation

$$\frac{\varrho_0}{\varrho} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{1/(\gamma-1)} \quad (10.2.12)$$

has been applied across the shock, whereas it already was included in the derivation of the TSD equation.

It should be noted that the shock jump is a function of the freestream Mach number for the TSD relation, while it is a function of the local velocity upstream and downstream of the shock for the other two models. All possess the dependence on the ratio of specific heats γ . A graph of the relationship given by Eqs. (10.2.9) to (10.2.11) is shown in Fig. 10.2. As its name implies, the Transonic

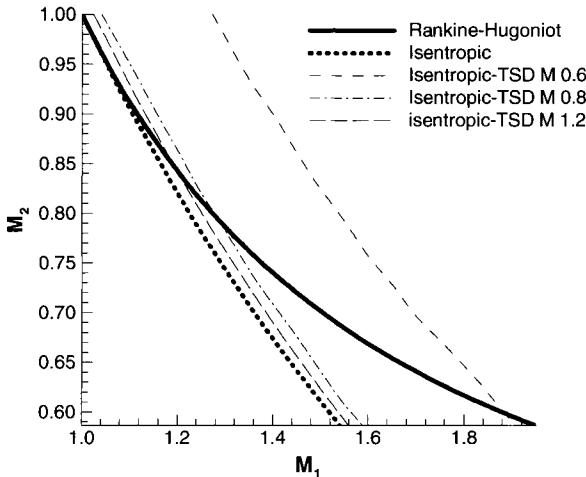


Fig. 10.2. Shock jump relations for the TSD, full-potential and Euler equations.

Small-Disturbance relation is losing accuracy as the freestream Mach number gets away from unity, and for near sonic-flow conditions, all relations are in agreement with the correct Rankine-Hugoniot jump. For Mach numbers above 1.3, the TSD and full-potential jumps are in disagreement with the Euler shock jump. This stems from the use of the isentropic flow relation (10.2.12) across the shock, which results in a momentum imbalance across the shock wave.

10.3 Shock Capturing

A numerical method to compute the correct shock jump as part of the solution to the TSD, Full-Potential or Euler equations is now discussed. Such algorithms are called Shock Capturing schemes. The model equation, Eq. (5.1.1) is considered and was derived from the linear wave equation,

$$\frac{\partial u}{\partial t} + c(u) \frac{\partial u}{\partial x} = 0 \quad (10.3.1)$$

Here u represents a variable to be conserved across the computational domain, including any discontinuities, and c is a function of u . Note that the linear wave equation is recovered if c is a constant. For simplicity, Eq. (10.3.1) is discretized using a forward time, central space procedure:

$$\frac{u_i^{n+1} - u_i^n}{t^{n+1} - t^n} = \frac{-c_i^n}{(x_{i+1} - x_{i-1})} (u_{i+1}^n - u_{i-1}^n) \quad (10.3.2)$$

The application of the discretization of Eq. (10.3.2) to the one-dimensional domain D (Fig. 10.3) leads to

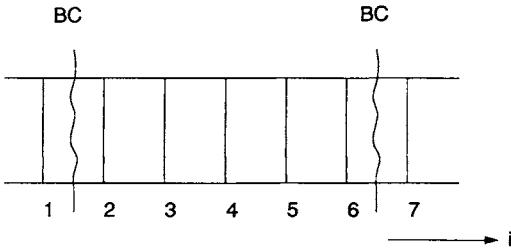


Fig. 10.3. 1-D computational domain for the non-linear wave equation.

$$\sum_i \frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{-1}{2\Delta x} [c_2^n(u_3^n - u_1^n) + c_3^n(u_4^n - u_2^n) + c_4^n(u_5^n - u_3^n) + c_5^n(u_6^n - u_4^n) + c_6^n(u_7^n - u_5^n)] \quad (10.3.3)$$

where $\Delta t = t^{n+1} - t^n$, $\Delta x = x_{i+1} - x_i$ (constant x spacings) and the numerical solution is a function of the interior domain values ($c_2 u_3$, $c_3 u_4$, etc.). For conservation to occur, we must have

$$\text{Flux}_{\text{out}} - \text{Flux}_{\text{in}} = 0 \quad (10.3.4)$$

which expresses that the flux is conserved. Clearly, Eq. (10.3.3) does not satisfy the condition Eq. (10.3.4) since the cross-terms inside the computational domain show up as source terms. This is inadmissible, and these schemes are called non-conservative. It can be shown that a non-conservative equation can be transformed into a conservative equation by the following procedure.

Let us assume that the convective term of Eq. (10.3.1) can be rewritten as

$$c \frac{\partial u}{\partial x} = \frac{\partial wu}{\partial x} \quad (10.3.5)$$

where w is new function of u . The non-linear wave equation becomes

$$\frac{\partial u}{\partial t} + \frac{\partial wu}{\partial x} = 0 \quad (10.3.6)$$

which now can be discretized using the same forward time, central space algorithm used on the non-conservative equation, as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{-1}{2\Delta x} ((wu)_{i+1}^n - (wu)_{i-1}^n) \quad (10.3.7)$$

A flux balance over the Domain D now yields

$$\sum_i \frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{-1}{2\Delta x} [(wu)_3^n - (wu)_1^n + (wu)_4^n - (wu)_2^n + (wu)_5^n - (wu)_3^n + (wu)_6^n - (wu)_4^n + (wu)_7^n - (wu)_5^n] \quad (10.3.8)$$

or

$$\sum_i \frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{-1}{2\Delta x} [((wu)_6^n + (wu)_7^n) - ((wu)_1^n + (wu)_2^n)] \quad (10.3.9)$$

and the solution is not function of the interior discretization, but only a function of the boundary conditions as required by Eq. (10.3.4). Schemes having this property are referred to as conservative and preserve the correct shock jump relations when discontinuities are present.

10.4 The Transonic Small Disturbance (TSD) Equation

The numerical solution of the transonic small disturbance equation has historical significance and Murman and Cole [2] were the first to numerically compute steady transonic flows using this equation. Their breakthrough was to use central difference operators in subsonic flow regions, and upwind difference operators in the supersonic regions. This seemingly simple idea gave rise to extensive research in the field during the 1970's and eventually led to significant advances in CFD. The conservative TSD equation, Eq. (10.2.2), is derived by making an asymptotic expansion of the perturbation velocity potential. Eq. (10.2.2) is rewritten as

$$\left[(1 - M_\infty^2) \varphi_x - \frac{\gamma + 1}{2} M_\infty^2 \varphi_x^2 \right]_x + \varphi_{yy} = 0 \quad (10.4.1)$$

where the perturbation velocity potential has been scaled by U_∞ . The non-conservative form of Eq. (10.4.1) is

$$[1 - M_\infty^2 - (\gamma + 1) M_\infty^2 \varphi_x] \varphi_{xx} + \varphi_{yy} = 0 \quad (10.4.2)$$

The non-conservative form is used in the following description since its type can be readily determined by evaluating the coefficient of the φ_{xx} term. If positive, Eq. (10.4.2) is elliptic whereas the equation is hyperbolic if negative. The limiting case where the coefficient is zero leads to Eq. (10.4.2) being parabolic. Clearly, central difference operators can be used for the elliptic case, while upwind difference operators must be used on the streamwise derivative for the hyperbolic case to retain the proper domain of dependence (see Section 2.6).

Because the body-surface boundary conditions of the TSD equation can be shifted onto a mean plane, with the requirement that the flow direction be tangential to the surface normal, a Cartesian based mesh is enough to discretize the computational domain. One advantage of Cartesian meshes is that they lend themselves to automatic mesh generation, a much more difficult task on body conforming meshes (see Chapter 9). It is also natural to cluster points around the leading and trailing edges of the airfoil, and around the shock region, if any, to increase the flow resolution. If this is the case, appropriate terms need to be added to the discrete form of Eq. (10.4.2) in the computational space.

10.5 Model Problem for the Transonic Small Disturbance Equation: Flow Over a Non-Lifting Airfoil

We consider a non-lifting airfoil placed in a uniform flow (Fig. 10.4).

Since the flow is symmetrical on the upper and lower surfaces of the airfoil, only the upper surface will be considered with a symmetrical boundary condition along the dividing streamline to reproduce the physics of the flow field:

$$\varphi_y = 0 \quad (10.5.1)$$

The surface boundary condition on the perturbation velocity potential is

$$\varphi_y = \frac{\partial y}{\partial x} \quad (10.5.2)$$

where $y = f(x)$ describes the airfoil coordinates. The upstream, downstream and upper boundaries are placed far enough to ensure uniform flow:

$$\varphi = 0 \quad (10.5.3)$$

An alternative boundary condition on the upstream and downstream boundary can be used:

$$\varphi_x = 0 \quad (10.5.4)$$

whereas an alternate boundary condition on the upper boundary can be

$$\varphi_y = 0 \quad (10.5.5)$$

The incoming flow field is assumed to be subsonic, but subsonic or supersonic (transonic) flow conditions can be attained above the airfoil surface. Most of the flowfield will then be subsonic, governed by the theory of elliptic equations, while only a small portion of the computational domain will be governed by hyperbolic equations. Therefore, the block iteration method described in subsection 4.5.2 will be used to solve the discretized flow equations, with the hyperbolic terms properly discretized.

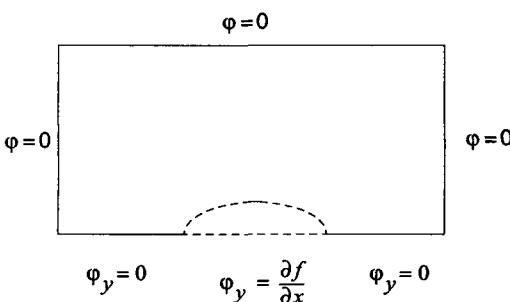


Fig. 10.4. Non-lifting airfoil in a uniform flow.

10.5.1 Discretized Equation

The non-conservative two-dimensional TSD equation [Eq. (10.4.2)] is discretized with the boundary conditions given by Eqs. (10.5.1)–(10.5.5) on a Cartesian mesh with equal mesh spacings ($\Delta x, \Delta y$) in each direction, as shown in Fig. 10.5.

To simplify the notation, Eq. (10.4.2) is written as

$$T\varphi_{xx} + \varphi_{yy} = 0 \quad (10.5.6)$$

with

$$T = 1 - M_\infty^2 - (\gamma + 1)M_\infty^2\varphi_x \quad (10.5.7)$$

All derivatives are discretized with the standard central difference formulas, for example

$$\varphi_x = \frac{(\varphi_{i+1,j} - \varphi_{i-1,j})}{2\Delta x} \quad (10.5.8)$$

and similarly for the terms φ_y while the φ_{yy} terms are written as

$$\varphi_{yy} = \frac{(\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1})}{\Delta y^2} \quad (10.5.9)$$

The φ_{xx} term must also be centrally discretized when $T > 0$, but must be shifted to the following backward scheme when $T < 0$ to retain the physical hyperbolic feature of the flow:

$$\varphi_{xx} = \frac{(\varphi_{i,j} - 2\varphi_{i-1,j} + \varphi_{i-2,j})}{\Delta x^2} \quad (10.5.10)$$

Note that the hyperbolic differencing is first order accurate, while the central difference elliptic operators are second-order accurate.

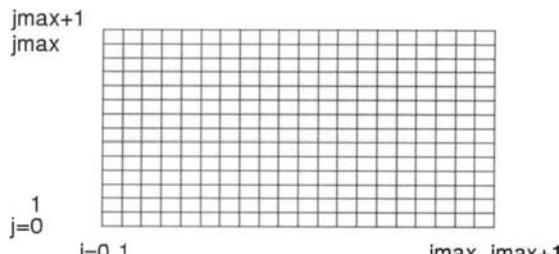


Fig. 10.5. Computational domain (including Halos) for the 2D TSD equation.

10.5.2 Solution Procedure and Sample Calculations

The computer program is given in Appendix B. First, a 2-D grid with equidistant spacing is generated on the computational domain ($i = 1$ to imax , $j = 1$ to jmax). The subroutine also generates the halo cells around the perimeter of the computational grid ($i = 0$, $i = \text{imax} + 1$, $j = 0$ and $j = \text{jmax} + 1$ lines) for reasons which will be discussed below.

The solution is initialized to a zero perturbation state ($\varphi = 0$) everywhere and the boundary conditions are applied immediately, as will be discussed shortly. Applying the ADI method discussed in subsection 4.5.2 in the y -direction and sweeping the domain explicitly in the x -direction, the discretized Eq. (10.5.4) can be written in the form:

$$a\varphi_{i,j-1} + b\varphi_{i,j} + c\varphi_{i,j+1} = \text{RHS} \quad (10.5.11)$$

where the right hand side contains the terms swept previously (station ($i - 1$)) or not yet swept (station ($i + 1$)) (Fig. 10.6). When the hyperbolic operator is activated, the RHS contains terms at station ($i - 2$) (Fig. 10.7)

Equation (10.5.11) is solved using the Thomas algorithm of subsection 4.4.2, and the updated values are under/over relaxed:

$$\varphi^{n+1} = \bar{\varphi} + \omega(\varphi^n - \bar{\varphi}) \quad (10.5.12)$$

where φ^n represents the previous value of the perturbation potential, $\bar{\varphi}$ is the intermediate value obtained from Eq. (10.5.11) and ω is the relaxation factor ($0 < \omega < 2$).

The boundary conditions are applied in terms of first-order backward or forward differences on the appropriate boundary. This would affect the coefficients a , b and c in Eq. (10.5.11) along every boundary. The evaluation of the boundary

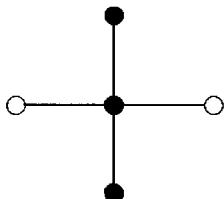


Fig. 10.6. Finite-difference stencil used for SLOR on elliptic TSD equation (● updated, ○ frozen).

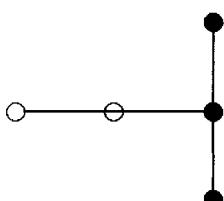


Fig. 10.7. Finite-difference stencil used for SLOR on hyperbolic TSD equation (● updated, ○ frozen).

coefficients introduces “if” statements in the computer logic that slows down the calculations while also making the source code more difficult to read. Another method of applying the boundary conditions consists of using halo cells lying outside the computational domain (hence their name). Figure 10.5 shows the computational space of Fig. 10.4 in the presence of the halo cells. The values of the perturbation potential φ in the halo cells are determined by the boundary conditions, and the algorithm Eq. (10.5.11) remains unaltered throughout the computational domain. As an example, for a computational space extending from $j = 1$ to $j = j_{\max}$, the symmetry and airfoil surface boundary conditions Eqs. (10.5.1) and (10.5.2) are used to update the halo cells located on the $j = 0$ line. Their values are obtained here with a first-order difference formula:

$$\varphi_{i,0} = \varphi_{i,1} \quad (10.5.13)$$

for the symmetry boundary condition and

$$\varphi_{i,0} = \varphi_{i,1} - \frac{\partial y}{\partial x} \Delta y \quad (10.5.14)$$

for the airfoil surface boundary condition.

The overall solution procedure is summarized below:

- (a) generate the grid (subroutine **generate grid**)
- (b) Initialize the perturbation potential φ (i.e. $\varphi = 0$ everywhere, subroutine **initial conditions**)
- (c) apply the boundary conditions (subroutine **boundary conditions**)
- (d) compute the coefficients a , b , c and the RHS in Eq. (10.5.11). In this step, the coefficient of the φ^n term in Eq. (10.5.1) must be computed at each point in order to select appropriate central or upwind differencing. (subroutine **slor**)
- (e) solve Eq. (10.5.11) using the Thomas algorithm (subroutine **tridiagonal**).
- (f) update the values of φ using under/over relaxation [Eq. (10.5.12)]
- (g) repeat steps (c) through (f) until convergence of the iterative process

One of the original test case used by Murman and Cole [2], namely the transonic flow over a non-lifting circular arc airfoil has been programmed. The airfoil coordinates are given by

$$y = \tau(1 - x^2) \text{ for } -1 < x < 1 \quad (10.5.15)$$

where the parameter τ controls the thickness-to-chord ratio. Analysis of the TSD equation shows that a transonic similarity parameter relates τ and the freestream Mach number:

$$K = (1 - M_\infty^2)/\tau^{2/3} \quad (10.5.16)$$

and that the pressure distribution can be scaled as

$$\bar{C}_P = C_P(\tau/M_\infty)^{3/2} \quad (10.5.17)$$

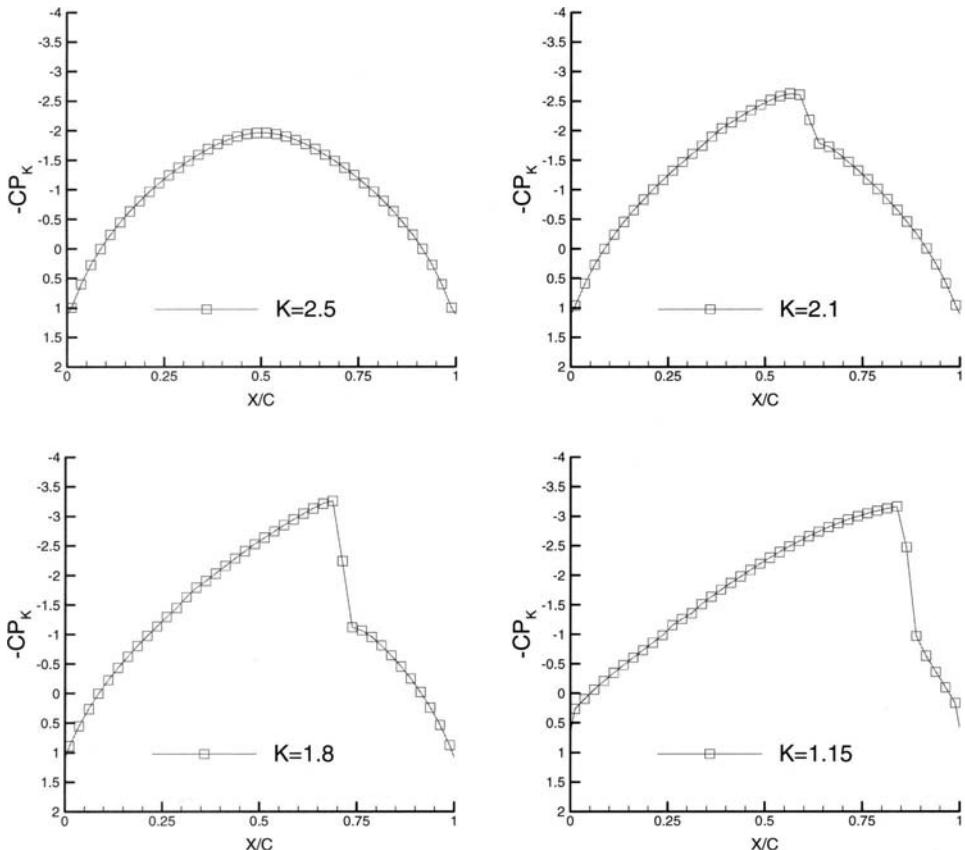


Fig. 10.8. Surface pressure distribution on circular arc airfoil with TSD theory.

which remains identical for airfoil of a given geometry $f(x)$ (see [2]). Note that the pressure is obtained from the isentropic flow relations.

The scaled pressure distribution is shown in Fig. 10.8 for four different values of the transonic similarity parameters ($K = 2.5, 2.1, 1.8$ and 1.45). It can be observed that the subsonic condition ($K = 2.5$) is symmetrical about the airfoil maximum thickness position, which follows the elliptic nature of the TSD equations discretized with a central difference formula. As the sonic zone develops over the airfoil surface with decreasing values of the transonic similarity parameters ($2.1, 1.8$ and 1.45), the shock waves becomes more pronounced and the hyperbolic nature of the TSD equations (with upwind differencing) give rise to the non-symmetric character of the solution. The agreement with the results published by [2] is excellent.

The convergence of the residual change in the values of the perturbation potential for the above test cases are shown in Fig. 10.9 for a fixed relaxation parameters. The convergence rate deteriorates with increasing incoming Mach

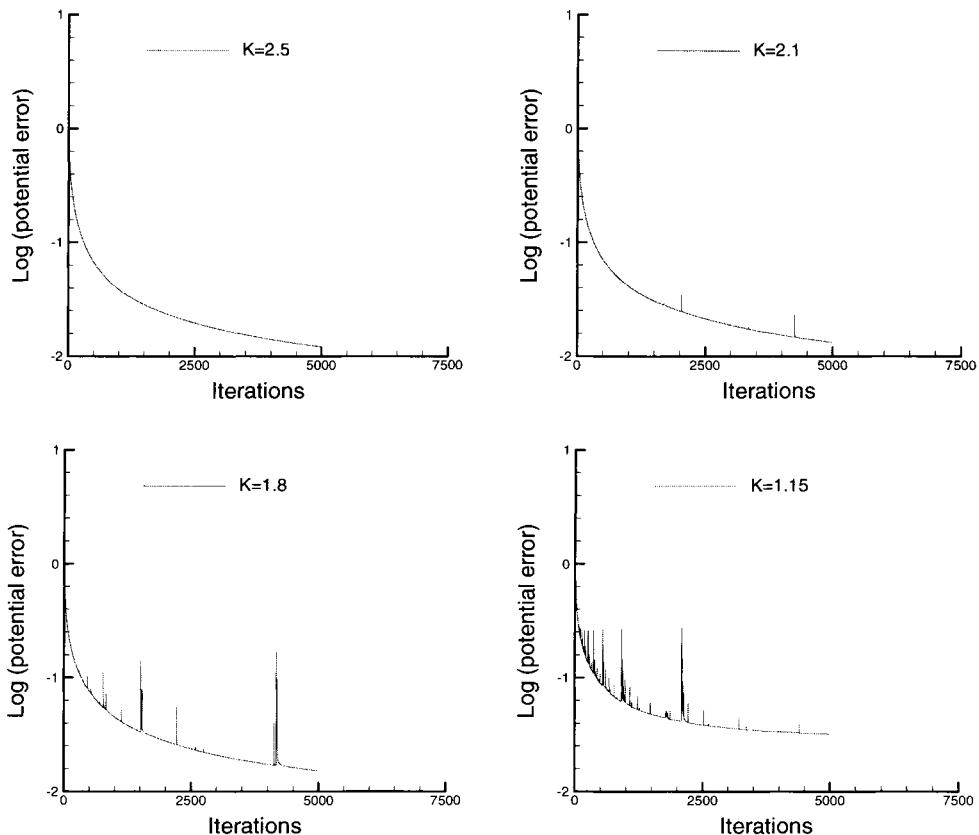


Fig. 10.9. Convergence of the residuals of φ on circular arc airfoil with TSD theory.

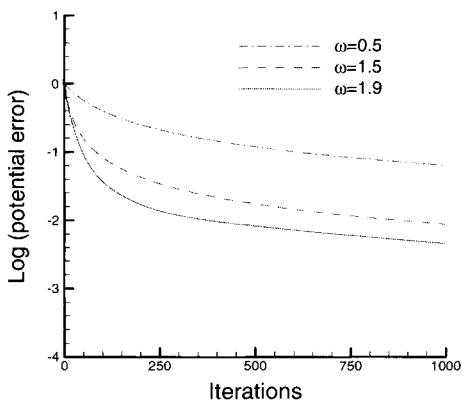


Fig. 10.10. Effect of relaxation factor ω on convergence rate.

numbers (lower K). The presence of spikes on the curve is due to the discontinuous switch from elliptic to hyperbolic difference operators in the supersonic region. The convergence rate variation as a function of the relaxation factor ω

for a fixed value of K is shown in Fig. 10.10. Since the flow is mainly subsonic, high values of the relaxation parameters result in better convergence rates.

10.6 Solution of Full-Potential Equation

The solution of the full-potential equation is the next step in obtaining more accurate flow solutions over an airfoil surface for the following reasons:

- 1) the small perturbation assumption is removed, allowing flow solutions on blunt leading edge airfoils (including the stagnation point).
- 2) the conservative shock relations are no longer a function of the incoming freestream Mach number, which deteriorated the flow solutions for the TSD equations as the Mach number changed from unity.
- 3) It remains valid for irrotational flow only, but the scalar equation is still much simpler than the next available mathematical model, namely the Euler equations.

The non-conservative full-potential equation, derived from the conservative form Eq. (10.2.3) is

$$(a^2 - u^2)\varphi_{xx} - 2uv\varphi_{xy} + (a^2 - v^2)\varphi_{yy} = 0 \quad (10.6.1)$$

with u and v as defined as φ_x and φ_y , respectively and a is the local speed of sound

$$a = \sqrt{a_\infty^2 - \frac{(\gamma - 1)}{2}(u^2 + v^2 - U_\infty^2)} \quad (10.6.2)$$

The presence of large disturbances, especially at the leading edge of an airfoil, presents additional challenges that makes the solution of the full potential equation a much more complex problem than solving the TSD equation. Body fitted grids are the preferred method for discretizing the computational flow domain in order to capture the blunt leading edge of airfoils and this translates into the use of a coordinate transformation to solve the equations in the computational space rather than in the physical space.

Another and more complex problem occurs when we apply upwinding in the hyperbolic flow region. For the TSD case, upwinding is achieved in the x -direction. However, upwinding in the full potential equation could occur in the x and/or y direction due to the fact that the equation type is now determined by the sign of $(u^2 + v^2 - a^2)$. The solution to this problem, provided by Prof. A. Jameson, is called the rotated difference scheme [3], aligning the differences along the streamline by rewriting Eq. (10.6.1) as

$$(a^2 - q^2) \left(\frac{u^2\varphi_{xx} + 2uv\varphi_{xy} + v^2\varphi_{yy}}{q^2} \right) + a^2 \left(\frac{v^2\varphi_{xx} - 2uv\varphi_{xy} + u^2\varphi_{yy}}{q^2} \right) = 0 \quad (10.6.3)$$

where $q = \sqrt{u^2 + v^2}$, which can also be written as

$$(a^2 - q^2)\varphi_{\xi\xi} + a^2\varphi_{\eta\eta} = 0 \quad (10.6.4)$$

with

$$\varphi_{\xi\xi} = \frac{u^2\varphi_{xx} + 2uv\varphi_{xy} + v^2\varphi_{yy}}{q^2} \quad (10.6.5)$$

and

$$\varphi_{\eta\eta} = \frac{v^2\varphi_{xx} - 2uv\varphi_{xy} + u^2\varphi_{yy}}{q^2} \quad (10.6.6)$$

When $a^2 > q^2$ (subsonic flow) is present, central differences are used just as in the elliptic TSD case. However, if $a^2 < q^2$, then upwinding is more complex than in the hyperbolic TSD case. Introducing the forward, backward and central differencing operators Δ , ∇ , δ , respectively, the discretized form of Eq. (10.6.4) can be written as

$$(a^2 - q^2) \left(\frac{u^2 \nabla_x \nabla_x + 2uv \nabla_x \nabla_y + v^2 \nabla_y \nabla_y}{q^2} \right) \varphi_{i,j} + a^2 \left(\frac{v^2 \nabla_x \Delta_x + 2uv \delta_x \delta_y + u^2 \nabla_y \Delta_y}{q^2} \right) \varphi_{i,j} = 0 \quad (10.6.7)$$

where the first term represent upwinding, and the second term central differencing. When $u < 0$, then upwinding is achieved with forward-differencing ($\Delta_x \Delta_x$). The same applies to the y -derivatives if $v < 0$.

The solution of the conservative full potential equation requires the introduction of explicit artificial dissipation, as discussed by Jameson [4]. Alternatively, the lagged density approach developed by Holst [5] provides the necessary upwinding in the supersonic region while simplifying the implementation. The reader is referred to the original papers for details on the subject.

10.7 Boundary Conditions for the Euler Equations

The mathematical formulation of the boundary conditions for the Euler equations is more complex than the scalar wave equation or even the TSD or potential equation, since there are now more than one wave travelling along characteristics. The eigenvalues λ_1 , λ_2 , λ_3 of the one-dimensional Euler equations are respectively u , $u + c$ and $u - c$, as discussed in Section 5.1. For supersonic flows ($u > c$), all waves travel in the same direction ($u > 0$, $u + c > 0$, $u - c > 0$) whereas for subsonic flows, two waves are travelling in the flow direction ($u > 0$, $u + c > 0$) while the third one travels against the flow direction ($u - c < 0$) (Fig. 10.11)

Consistent boundary conditions must be imposed for both types of conditions. If the supersonic waves of Fig. 10.11a travel into the computational

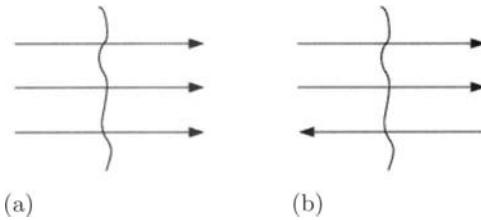


Fig. 10.11. Wave propagation for the 1D Euler equation. (a) supersonic flow $u > c$, (b) subsonic flow $u < c$.

domain, it is necessary to specify *three* boundary conditions, taken from physical data such as experimental values. We refer to these as physical boundary conditions. If the subsonic waves of Fig. 10.11b travel into the computational domain, it is necessary to specify *two* physical boundary conditions. The remaining boundary condition for the subsonic inflow case must be determined numerically by extrapolation of the interior flow domain. We call these numerical boundary conditions. Now, if the supersonic or subsonic waves travel outside the computational domain, it is necessary to specify respectively zero and one physical boundary conditions. The remaining boundary condition for the supersonic and subsonic outflow case must be determined numerically by extrapolation of the interior flow domain. Table 10.1 summarizes the results.

For example, the supersonic flow conditions entering the computational domain might be the freestream Mach number, the total temperature and total pressure. For supersonic outflow conditions, zero-order extrapolation of the primitive variables can be specified. For subsonic inflow and outflow conditions, it is standard procedure to solve the Riemann variables used in the method of characteristics to determine the boundary conditions:

$$\begin{aligned} \text{along } u : \quad s &= \text{constant} \\ \text{along } u + c : \quad u + 2c/(\gamma - 1) &= R_1 \\ \text{along } u - c : \quad u - 2c/(\gamma - 1) &= R_2 \end{aligned} \quad (10.7.1)$$

where s is the entropy, u the flow velocity normal to the surface, c the speed of sound and γ the ratio of specific heat. The incoming and outgoing characteristics are

Table 10.1. Physical and numerical boundary conditions for the 1D Euler equations.

	$M > 1$		$M < 1$	
Physical conditions	Inflow	3	Inflow	2
	Outflow	0	Outflow	1
Numerical conditions	Inflow	0	Inflow	1
	Outflow	3	Outflow	2

$$R_{1\infty} = u_\infty - \frac{2c_\infty}{\gamma - 1} \quad (10.7.2)$$

For a subsonic inflow, we set R_1 and s to their freestream values. R_2 represents the numerical boundary condition extrapolated from the interior of the computational domain. The velocity u is found by matching R_1 and R_2 on the boundary. For a subsonic outflow, R_2 is assigned its freestream value, and s and R_1 are extrapolated from the interior of the computational domain. The velocity is obtained as discussed above. Application of this procedure is given in Section 10.11.

Along a solid wall, the impermeable wall boundary condition is imposed

$$\vec{V} \cdot \vec{n} = 0 \quad (10.7.3)$$

and the density and pressure are extrapolated from the interior. Other choices may be made for the specification of boundary conditions and the reader is referred to [1] for more information.

10.8 Stability Analysis of the Euler Equations

The stability analysis of Section 5.7 indicates that for the scalar wave equation, the maximum time step allowable in the explicit Lax method requires the CFL condition, $\text{CFL} < 1$. The time step restriction disappears in the implicit schemes. In this section, we address the stability of the Euler equations.

Since the Euler equations can be transformed into a system of three scalar equations of the form given by Eq. (5.1.19), the stability bounds for each of the three equations are thus, for the explicit Lax scheme

$$u\Delta t/\Delta x < 1 \text{ for } \lambda = u \quad (10.8.1a)$$

$$(u + c)\Delta t/\Delta x < 1 \text{ for } \lambda = u + c \quad (10.8.1b)$$

$$(u - c)\Delta t/\Delta x < 1 \text{ for } \lambda = u - c \quad (10.8.1c)$$

It is reasonable, although not mathematically strict, to choose the minimum time step of the three equations, that is

$$\text{CFL} = (|u| + c)\Delta t/\Delta x < 1 \quad (10.5.3)$$

Note that this condition must be satisfied at each computational point, and the time step must be taken as the smallest one in the computational domain. Such a system is referred to as being numerically stiff, as there is a wide spread between the smallest and largest eigenvalues as the flow velocity u departs from c .

The analysis is more complicated for implicit methods. It can be shown that implicit schemes have a weak instability at sonic lines, which is damped in the

Lax-Wendroff and MacCormack explicit schemes briefly discussed in Sections 5.2 and 5.3. It then becomes necessary to add explicit artificial dissipation to the implicit methods of Section 5.4. The amount of dissipation must be enough to damp the instabilities and, in addition, to prevent oscillations around shock waves. The model problem of Section 10.13 will show how explicit artificial dissipation is introduced into the formulation of implicit schemes.

10.9 MacCormack Method for Compressible Euler Equations

In Section 5.3 we discussed the explicit MacCormack method for a one-dimensional problem and here we discuss its extension to compressible flows. The Euler equations for one-dimensional flow are considered and from Eq. (2.1.30),

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} = 0 \quad (10.9.1)$$

with Q and E given by Eqs. (2.2.32a) for one-dimensional flow.

The discussion of Section 5.3 is followed, the predictor values at $(x_i, t^{n+1/2})$ is defined by $\bar{Q}_i^{n+1/2}$ ($\equiv \bar{Q}_i$) and the convective flux term E represented with forward differences in the predictor step and backward differences in the corrector step. Thus

predictor:

$$\bar{Q}_i = Q_i^n - \Delta t \left(\frac{E_{i+1}^n - E_i^n}{\Delta x} \right) \quad (10.9.2)$$

corrector:

$$\bar{\bar{Q}}_i = Q_i^n - \Delta t \left(\frac{\bar{E}_i^- - \bar{E}_{i-1}^-}{\Delta x} \right) \quad (10.9.3)$$

updating

$$Q_i^{n+1} = \frac{1}{2} (\bar{Q}_i + \bar{\bar{Q}}_i) \quad (10.9.4)$$

Note that the predictor step could have been obtained with backward differences, while the corrector step with forward differences as in Eqs. (5.3.3), that is

predictor:

$$\bar{Q}_i = Q_i^n - \Delta t \left(\frac{E_i^n - E_{i-1}^n}{\Delta x} \right) \quad (10.9.5)$$

corrector:

$$\bar{\bar{Q}}_i = Q_i^n - \Delta t \left(\frac{\bar{E}_{i+1}^- - \bar{E}_i^-}{\Delta x} \right) \quad (10.9.6)$$

Updating is as described by Eq. (10.9.4). It can be shown that the two procedures are identical for linear equations, while for non-linear equations they are

not. In fact, it will be shown in Problem P10.4 that the first procedure is best suited for discontinuities moving in the $-x$ direction, while the second approach is best suited for discontinuities moving in the $+x$ direction. It can be seen that a numerical boundary scheme is necessary on one side of the predictor step, while a numerical boundary scheme is necessary on the opposite side on the corrector step. This is true for both forward-backward and backward-forward procedures described above.

10.10 Model Problem for the MacCormack Method: Unsteady Shock Tube

To demonstrate the solution of the Euler equation for a one-dimensional compressible flow with the MacCormack method, the model problem of a unsteady shock tube is used as shown in Fig. 10.12.



Fig. 10.12. Shock tube model problem: Initial conditions.

The tube is filled with a gas at different states on the left and right side of a diaphragm. The gas states have different densities and pressures and are at rest. At time $t = 0$, the diaphragm is broken and if it is assumed that viscous effects are negligible and the tube is of infinite length (reflection waves are zero), then the unsteady Euler equations for a one-dimensional flow can be solved analytically with a family of characteristics travelling to the left and right of the diaphragm. If the left side contains the gas at the highest pressure, the right state will expand in the left side region through expansion waves (region 2), whereas a compression wave will travel in the right direction. This will be

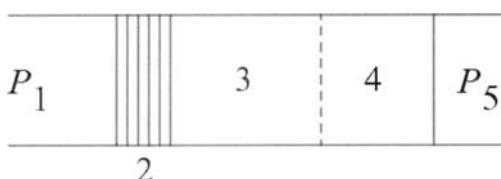


Fig. 10.13. Shock tube model problem: gas states at time $t > 0$.

a shock wave travelling at speed c and leaving behind a contact discontinuity travelling at speed Vc , as shown in Fig. 10.13:

The problem therefore contains three distinct regions (2, 3, 4) in addition to the regions at rest (1 and 5), of which two represents discontinuous regions (region 3-4 and 4-5). The closed form solution of this problem is given in [6].

10.10.1 Initial Conditions

The shock tube is of length 1000, and the diaphragm is located at $x = 500$. The initial conditions at $t = 0$ are:

$$\begin{aligned} u(x, 0) &= 0 && \text{for } 0 < x < 1000 \\ \varrho(x, 0) &= 1, \quad p(x, 0) = 1 && \text{for } x < 500 \\ \varrho(x, 0) &= 4, \quad p(x, 0) = 4 && \text{for } x > 500 \end{aligned}$$

10.10.2 Boundary Conditions

The shock tube is considered of infinite length, which avoids waves being reflected at its end. Therefore, extrapolation from the interior of the domain is used on both boundaries:

$$\begin{aligned} \varrho(i = 0, t) &= \varrho(i = 1, t) & \varrho(i = N, t) &= \varrho(i = N - 1, t) \\ u(i = 0, t) &= u(i = 1, t) & u(i = N, t) &= u(i = N - 1, t) \\ p(i = 0, t) &= p(i = 1, t) & p(i = N, t) &= p(i = N - 1, t) \end{aligned}$$

Note that we make use of halo cells to introduce the boundary conditions, as discussed in subsection 10.5.2.

10.10.3 Solution Procedure and Sample Calculations

The computer program is given in Appendix B. First, a 1-D grid with equidistant spacing is generated (subroutine `generate_grid`). The flux variables are then initialized according to the specified boundary condition of subsection 10.9.2 for a diatomic gas (subroutine `initial_conditions`). Then the algorithm marches in time, with increments satisfying the stability restrictions of Section 10.8 (subroutine `timestep`). The MacCormack predictor-corrector steps of Section 10.9 are computed (subroutine `Maccormack`) to yield the updated flux values. The program ends when the time has reached 250. Note that a subroutine has been devised to compute the Euler fluxes f given the primitive variables (subroutine `flux`). This way, the predictor and corrector steps are simply done by calling the flux subroutine for each of the steps of Eqs. (10.6.5) and (10.6.6). The numerical boundary schemes are used in both steps, but at the opposite end of the tube.

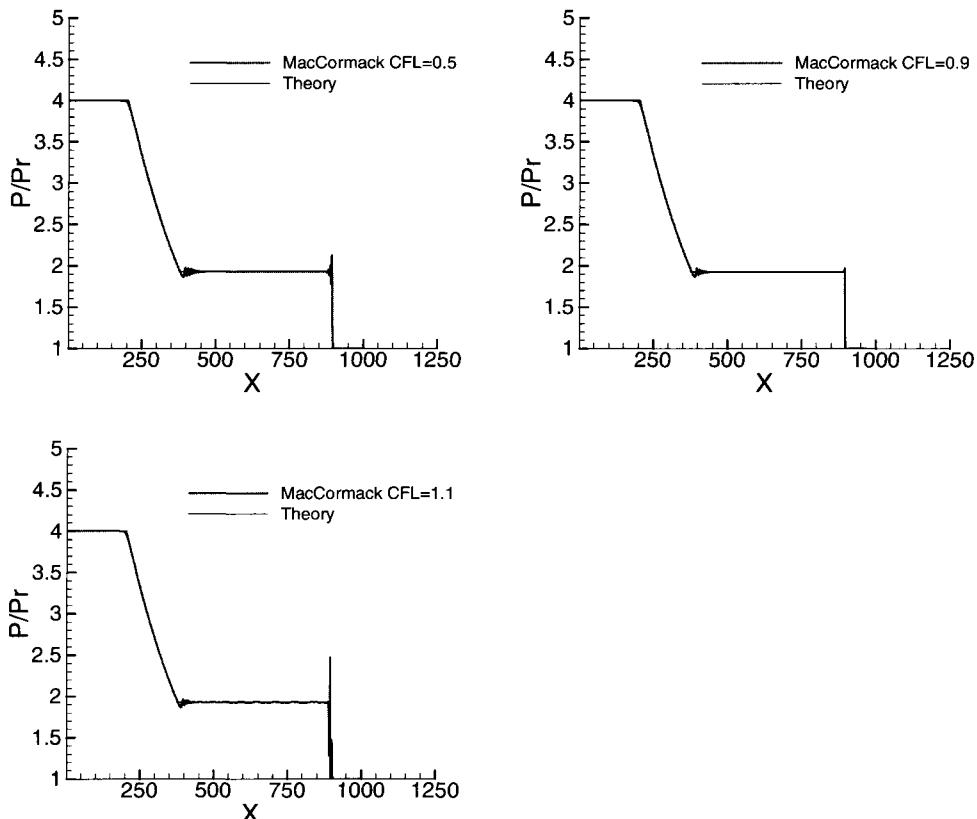


Fig. 10.14. Shock tube model problem: gas states at time $t = 250$.

The numerical results may be compared with the theoretical values for several CFL numbers (0.5, 1.0 and 1.1) in Fig. 10.14, which shows dispersion errors associated with the truncation error acting on third derivatives at every CFL number. The scheme is more accurate at higher CFL number, and the non-linearities allow solutions to be obtained for a CFL number up to 1.1, as opposed to the linear stability analysis, which limits the CFL number to unity. However, dispersion errors are unacceptable at CFL values greater than one and the code diverges as the MacCormack scheme is unable to damp the high-frequency errors at values greater than 1.1.

10.11 Model Problem for the MacCormack Method: Quasi 1-D Nozzle

This model problem allows the computation of steady state solutions for a wide variety of test cases. Since analytical one dimensional nozzle flow solutions can

readily be obtained, and the solutions have many branch points, i.e. fully subsonic, subsonic/supersonic, fully supersonic and lastly subsonic/supersonic and subsonic at the exit (underexpanded jets), it provides a useful way to evaluate the influence of boundary conditions while also yielding information on the convergence speed and accuracy of the selected schemes.

The compressible Euler equations for a quasi 1-D nozzle are

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} = S \quad (10.11.1)$$

where the source term is related to the cross-sectional area of the nozzle. In this equation, the vectors Q and E are very close to their 1D counterpart [Eqs. (2.1.32) and (2.1.33)], but they contain an additional factor, which corresponds to the influence of the cross-sectional nozzle area,

$$Q = \begin{bmatrix} \varrho A \\ \varrho u A \\ eA \end{bmatrix}, \quad E = \begin{bmatrix} \varrho u A \\ (\varrho u^2 + p)A \\ u(e + p)A \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} 0 \\ p \frac{dA}{dx} \\ 0 \end{bmatrix} \quad (10.11.2)$$

The MacCormack scheme is modified to take into account the source term but there are many possibilities, and here the correction will be added at each step:

predictor step:

$$\bar{Q}_i = Q_i^n - \Delta t \left(\frac{E_i^n - E_{i-1}^n}{\Delta x} \right) + \Delta t S_i^n \quad (10.11.3)$$

corrector step:

$$\bar{\bar{Q}}_i = Q_i^n - \Delta t \left(\frac{\bar{E}_{i+1}^- \bar{E}_i}{\Delta x} \right) + \Delta t \bar{S}_i^n \quad (10.11.4)$$

Updating is as described by Eq. (10.6.4). Note that \bar{S} is the source term evaluated with the predicted value from the predictor step. Since there are only slight modifications to the 1D Euler flow equations, the computer program of Section 10.10 is used here, with the proper modifications for the area and source terms made.

10.11.1 Initial Conditions

The nozzle is of length 10, with the incoming flow supersonic ($M = 1.3$) and the outgoing flow subsonic ($M < 1$). The entire flowfield is initialized with the incoming flow values.

10.11.2 Boundary Conditions

The incorporation of the source term poses no difficulty but the non-reflecting boundary conditions must now be changed to subsonic Riemann type boundary conditions when subsonic flow is present and to supersonic boundary conditions when supersonic flow is present. The supersonic boundary conditions are (with right running characteristics)

$$\begin{aligned}\varrho_1 &= \varrho_{\text{inf}} \\ u_1 &= u_{\text{inf}} \\ p_1 &= p_{\text{inf}}\end{aligned}\tag{10.11.5}$$

at the inlet, and

$$\begin{aligned}\varrho_N &= \varrho_{N-1} \\ u_N &= u_{N-1} \\ p_N &= p_{N-1}\end{aligned}\tag{10.11.6}$$

at the outlet. Equations (10.7.1) must be used when subsonic flow is present. Since in our model problem the nozzle entrance will be supersonic and the nozzle exit subsonic, we will develop the appropriate boundary condition for the exit station $i = N$. Referring to Fig. 10.15, the two Riemann invariants meeting at the boundary point J are emanating from the point $i = N - 1$ and from downstream infinity, where the pressure is given.

Along the right running characteristic, we have

$$\begin{aligned}\varrho_R &= \varrho_{N-1} \\ p_R &= p_{N-1}\end{aligned}$$

from which the entropy s_R and the speed of sound c_R are computed. It is known that entropy in an inviscid subsonic flow is constant so that the far-field entropy s_L carried by a left running characteristic is thus equal to the entropy carried by the right wave: $s_L = s_R$. Since the back pressure p_L is given as a boundary condition, the density ϱ_L and the speed of sound c_L are found at once.

The right running Riemann invariant R_1 meets the left running Riemann invariant R_2 at point $i = N$ and, by subtracting the two invariants,

$$R_2 = R_1 - 4c_L/(\gamma - 1)$$

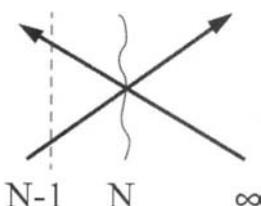


Fig. 10.15. Far-field boundary conditions using Riemann invariants.

and by adding them

$$u_L = (R_1 + R_2)/2$$

from which the energy e_L can be computed. The flux vector Q is then

$$Q = \begin{bmatrix} \varrho_L A_N \\ \varrho_L u_L A_N \\ e_L A_N \end{bmatrix} \quad (10.11.7)$$

10.11.3 Solution Procedure and Sample Calculations

The program is identical to the nozzle flow program, with additional source terms and far-field boundary conditions. The computer program is given separately from the shock tube program in Appendix B. First, a 1-D grid with equidistant spacing is generated (subroutine `generate_grid`). The flux variables are then initialized to their incoming freestream values. Then the algorithm marches in pseudo-time, with increments satisfying the stability restrictions (subroutine `timestep`). The MacCormack predictor-corrector steps are computed (subroutine `Maccormack`) to yield the updated flux values. The program ends when the total number of iterations reached a user specified value. Moreover, a L2-norm of the residual defined as

$$\|u^{n+1} - u^n\|_2 = \sqrt{\sum_j (u_j^{n+1} - u_j^n)^2} \quad (10.11.8)$$

is monitored and written in a separate file (file `residuals`) which allows printing of the convergence curve. The residual at the first iteration is stored, and the residual at every subsequent iteration is then compared to the initial value. The norm is written typically in a logarithmic scale, such that a Newton iteration process on a linear equation would reduce the error by two orders of magnitude in one iteration.

The convergence curve and the Mach number distribution are plotted for several CFL numbers (0.5, 1, and 1.1) in Fig. 10.16, for a nozzle having an area distribution of the form

$$A(x) = 1.398 + 0.347 \tanh(0.8x - 4) \quad (10.11.9)$$

for $0 < x < 10$, a back pressure corresponding to 1.931 times the incoming pressure, and an incoming supersonic flow at Mach 1.3. Some 5500 time steps are needed to achieve convergence to 10^{-10} with a CFL number of 1.0, which is half of those required for the CFL 0.5 case, since the convective waves scale with the time step calculation. As was the case with the shock tube model problem, the dispersion errors are more pronounced as the CFL number is reduced below 1.0. The time marching algorithm diverges for CFL numbers greater than 1.1, which is in close agreement with stability analysis of the linear wave equation (Section 5.7).

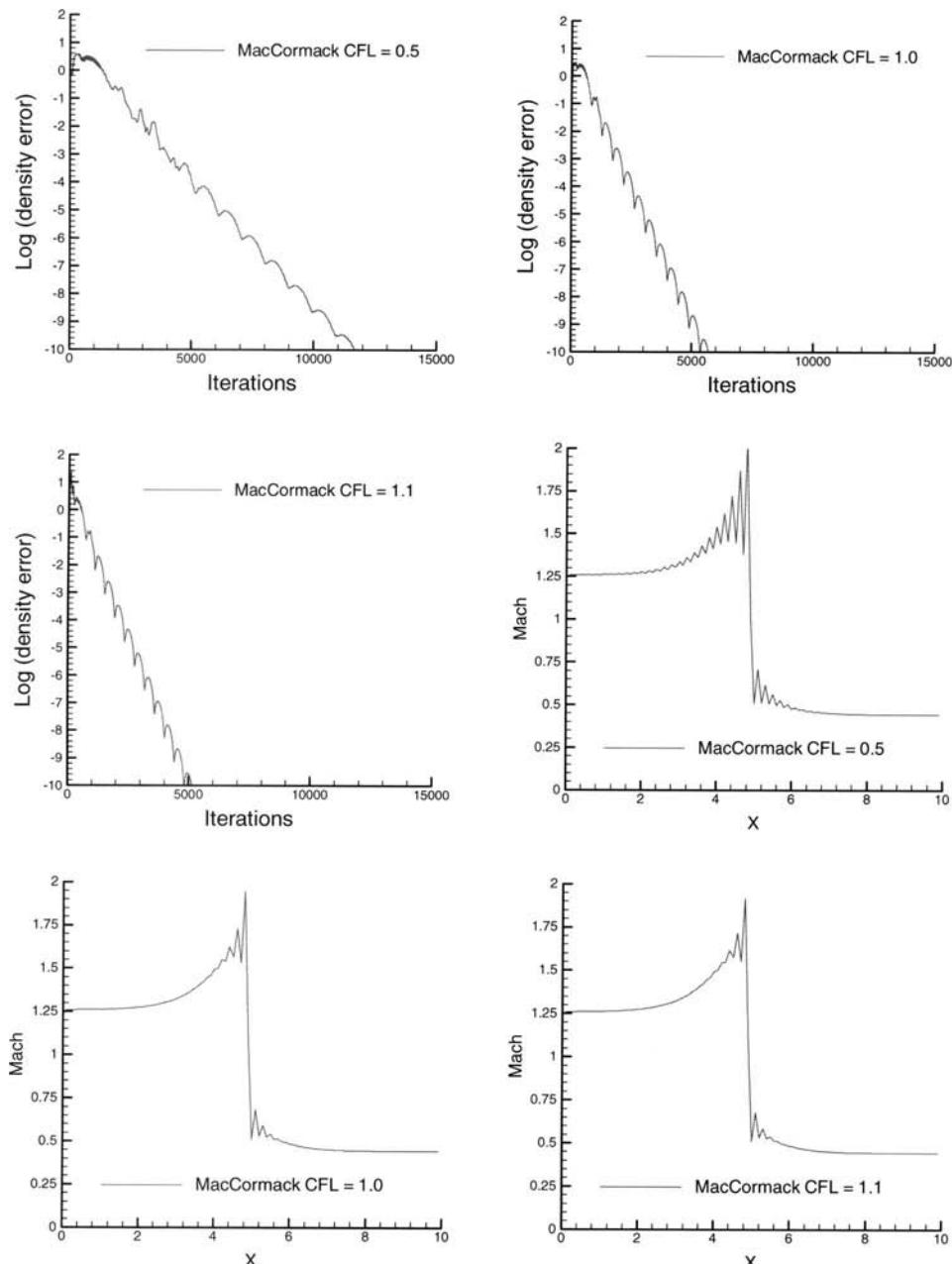


Fig. 10.16. Nozzle model problem: Convergence curve (*top*) and Mach number distributions (*bottom*) as a function of CFL number.

10.12 Beam-Warming Method for Compressible Euler Equations

The 1-D compressible Euler equation

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} = 0 \quad (2.1.30)$$

has been discretized with the implicit procedure in Section 5.4. Here, the case $\varphi = \xi = 0$ is considered, and Eq. (5.4.9) is rewritten as

$$[I + \theta \Delta t \delta_x A^n] \Delta Q^n = -\Delta t \delta_x E \quad (10.12.1)$$

I is the identity matrix of Eq. (5.4.10), A^n is the jacobian matrix of Eq. (5.4.8) and δ_x is the central difference operator.

Explicit artificial dissipation must now be added to stabilize the scheme. Although one would intuitively add a second derivative to the right hand side of the algorithm to reproduce the artificial viscosity terms of the Lax scheme, it has been found that a fourth-derivative term reduces the truncation error while still providing enough stability for the implicit scheme. With a constant coefficient, 4th derivative term added to Eq. (10.12.3), we obtain

$$[I + \theta \Delta t \delta_x A^n] \Delta Q^n = -\Delta t \delta_x E - \varepsilon_e (\nabla \Delta)_x^2 Q^n \quad (10.12.2)$$

where ε_e is the explicit artificial dissipation constant and where Δ and ∇ are the forward and backward difference operators, respectively. Stability analysis of the model scalar equation shows that the bounds on ε_e are

$$\varepsilon_e < 0.125$$

However, in practice this amount of artificial dissipation is too low and a solution is found by adding an equivalent damping term on the implicit operator

$$[I + \theta \Delta t \delta_x A^n - \varepsilon_i (\nabla \Delta)_x] \Delta Q^n = -\Delta t \delta_x E - \varepsilon_e (\nabla \Delta)_x^2 Q^n \quad (10.12.3)$$

where ε_i is the implicit artificial dissipation constant. Unfortunately, this implicit matrix is now a block pentadiagonal matrix, which is expensive to invert. Thus, a second-order operator is preferred and maintains the block tridiagonal matrix of the original implicit system, while considerably increasing the damping effects:

$$[I + \theta \Delta t \delta_x A^n - \varepsilon_i (\nabla \Delta)_x] \Delta Q^n = -\Delta t \delta_x E - \varepsilon_e (\nabla \Delta)_x^2 Q^n \quad (10.12.4)$$

Since the increase in damping is provided on the left hand side of the equation, it does not degrade the accuracy of the solution upon reaching steady state values since the ΔQ term is then close to zero. Linear stability analysis shows that for stability one must have

$$\varepsilon_i > 2\varepsilon_e$$

In addition, the explicit constants must be scaled by Δt to retain dimensional scaling of the equations.

10.13 Model Problem for the Implicit Method: Unsteady Shock Tube

The model problem is the shock tube problem of Section 10.10 and the initial conditions (subsection 10.10.1) and boundary conditions (subsection 10.10.2) are identical.

10.13.1 Solution Procedure and Sample Calculations

The computer program is given in Appendix B and, since the shock tube problem was solved using the explicit algorithm of MacCormack in subsection 10.10.3, the numerical solution of the problem is constructed by borrowing from the subroutines already developed. First, a 1-D grid with equidistant spacing is generated (subroutine `generate grid`). The flux variables are then initialized according to the specified boundary condition of subsection 10.9.1 for a diatomic gas (subroutine `initial conditions`). Then the algorithm marches in time, with increments satisfying the stability restrictions (subroutine `timestep`). The implicit steps are computed (subroutine `implicit`) to yield the updated flux values. The program ends when the time has reached 250. The program uses the same subroutine as in subsection 10.10.3 to compute the Euler fluxes F given the primitive variables (subroutine `flux`).

The Jacobian matrix A is first constructed at all points for the implicit steps and, since it operates on a 1st-derivative central difference operator, it is computed for the lower and upper band of the block tridiagonal matrix. The main band is formed by the identity matrix. The implicit artificial dissipation, which is operated on by a 2nd-derivative central difference operator, is added on the diagonal of each matrix: lower, upper and main diagonal. The explicit artificial dissipation construction is straightforward, and is added to the right hand side vector.

Zero-order extrapolation is applied to the smallest and largest vector element of the right hand side as numerical boundary scheme, which effectively let the waves escape from the computational domain. Once the left side tridiagonal matrix formed, a standard block-tridiagonal subroutine is called to update the primitive quantities.

The numerical results and the theoretical values are presented in Fig. 10.17 for several CFL number (0.5, 0.9, 2.0, 5.0) with the explicit artificial dissipation coefficient held constant at 0.1 and the implicit artificial dissipation coefficient set 2.5 times the explicit one. It can be seen that there are no dispersion errors, but dissipative error remains as expected from the truncation error terms operating on second derivatives. The dissipation increases as the CFL number increased, and the implicit procedures allows CFL number greater than 1, which is consistent with the stability analysis.

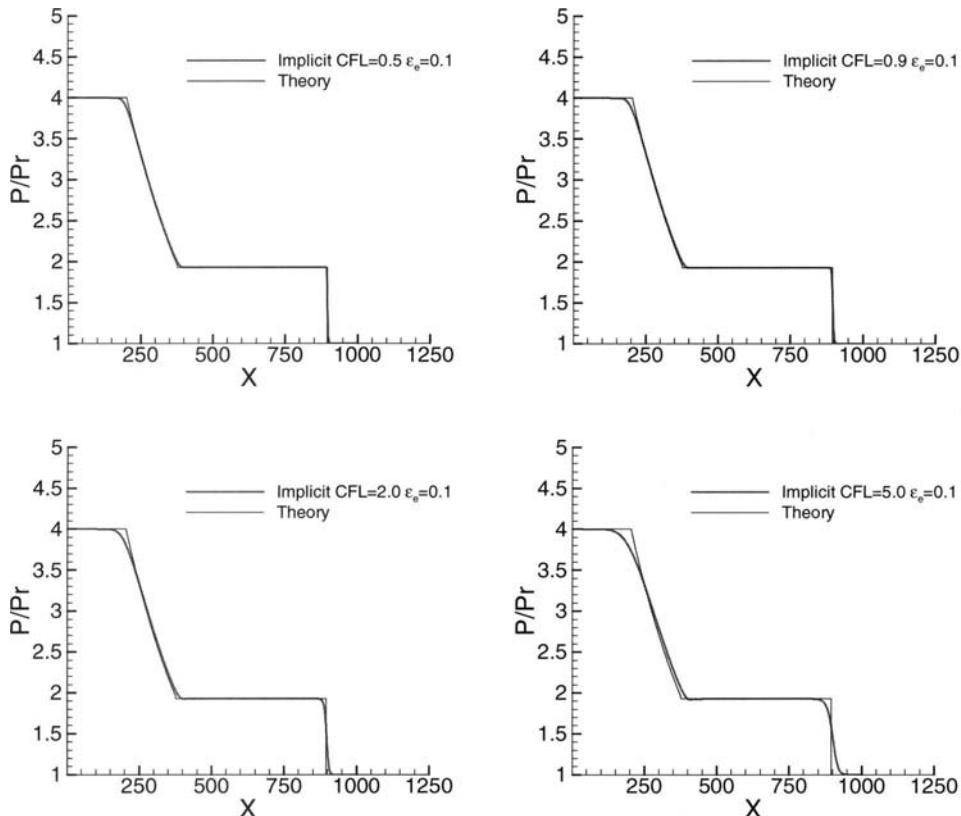


Fig. 10.17. Shock tube model problem: gas states at time $t = 250$; CFL variation.

Another set of calculation was made, but this time the CFL number was kept at 1.0 and the explicit artificial viscosity coefficients were varied from .0001 to 10. The numerical results are be compared with the theoretical values in Fig. 10.18. It can be seen that the dissipation errors increase with the values of ϵ_e , and that the shock wave has almost merged with the expansion wave for the case when $\epsilon_e = 10$.

10.14 Model Problem for the Implicit Method: Quasi-1D Nozzle

The model problem is the quasi-1D nozzle problem of Section 10.11 and the initial conditions (subsection 10.11.1) and boundary conditions (subsection 10.11.2) are identical.

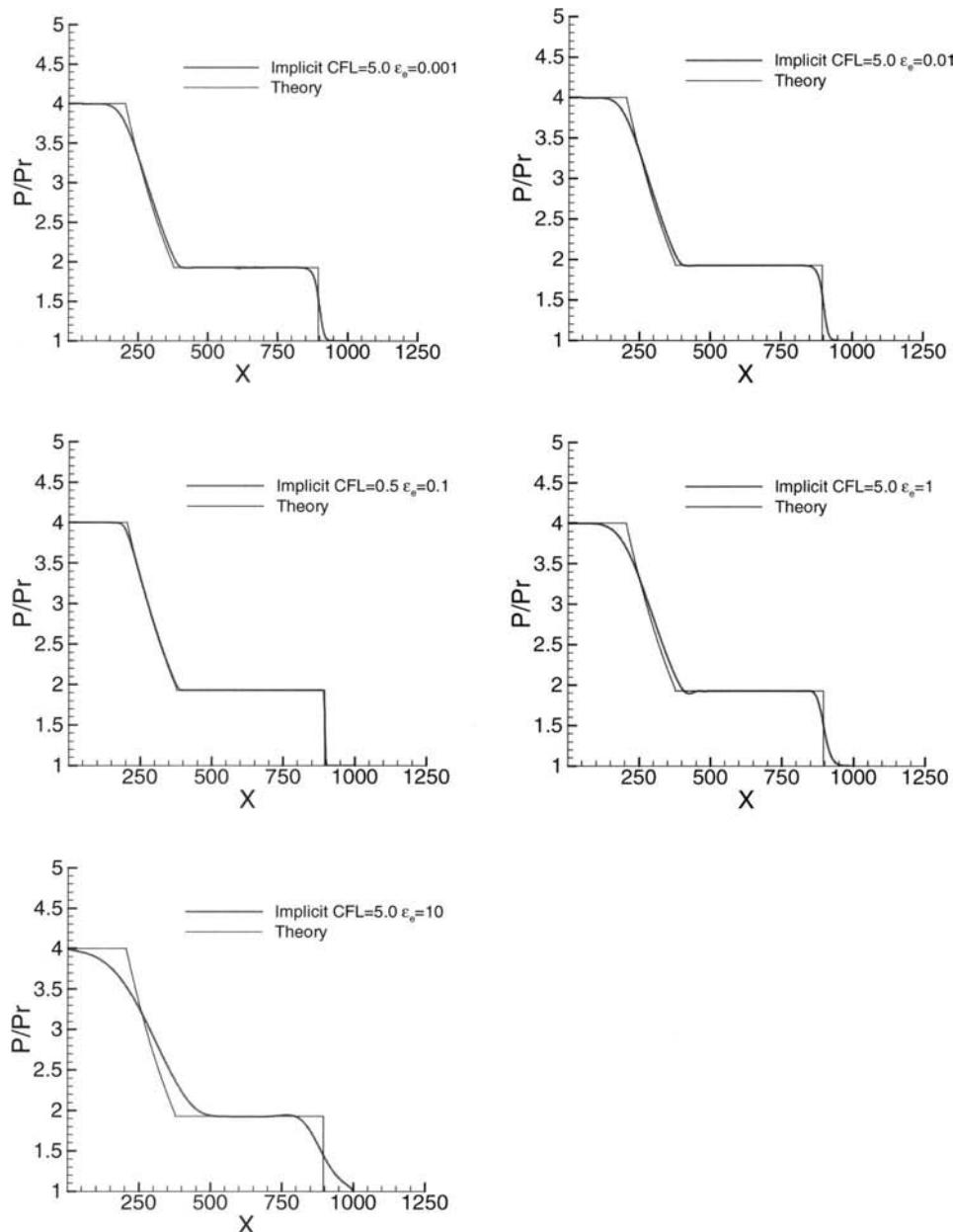


Fig. 10.18. Shock tube model problem: gas states at time $t = 250$; Explicit artificial viscosity variation.

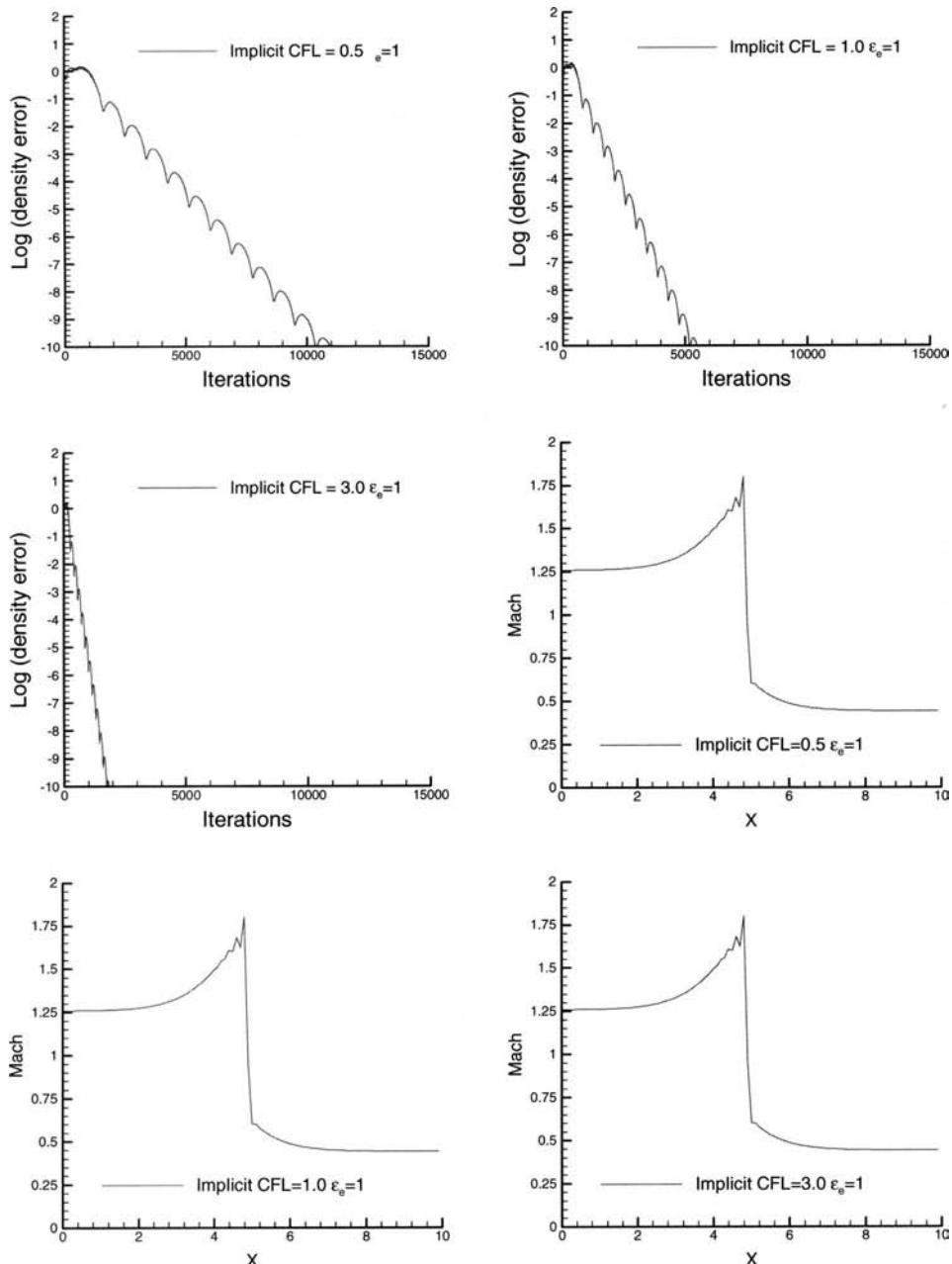


Fig. 10.19. Nozzle model problem: Convergence curve (*top*) and Mach number distributions (*bottom*) as a function of CFL number.

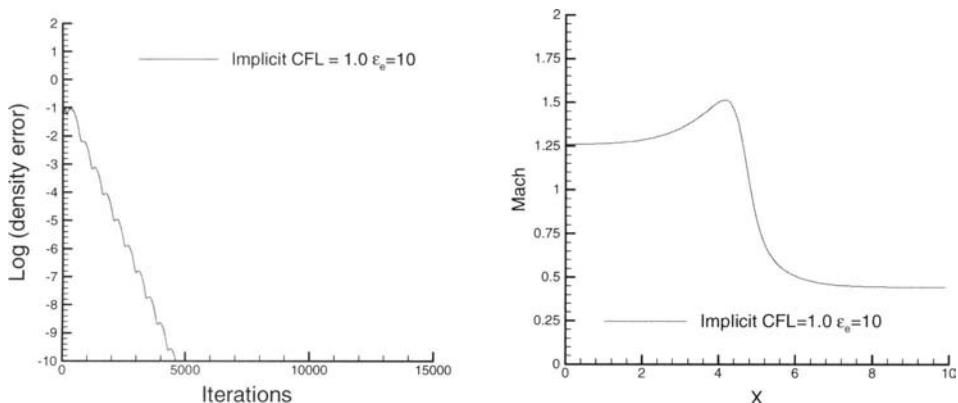


Fig. 10.20. Nozzle model problem: Convergence curve (*top*) and Mach number distributions (*bottom*) for the case $\epsilon_e = 10$.

10.14.1 Solution Procedure and Sample Calculations

The computer program combines the implicit shock tube subroutines with the nozzle flow routines developed for the MacCormack scheme and is given separately in Appendix B. In fact, the artificial dissipation is added to the left-hand side (implicit) and right hand side (explicit) of the equations. Moreover, the dissipation must scale with the ratio of the nozzle area of the corresponding cells, to avoid spurious artificial dissipation terms arising from the non-conservative operators. Also, the Jacobian of the source term must be computed and included in the right hand side evaluation.

The numerical results are plotted for several CFL number (0.5, 1.0, 3.0) in Fig. 10.19 for the same nozzle used in Section 10.11 with the explicit artificial dissipation coefficient held constant at 1.0. The implicit artificial dissipation coefficient was 2.5 times the explicit one. The convergence reached 10 orders magnitude of reduction in approximately 11 000 and 5500 times steps for the cases $CFL = 0.5$ and $CFL = 1.0$, respectively. These numbers are similar to those obtained with the explicit MacCormack scheme in subsection 10.11.3. Since the computing requirements of the implicit scheme are more than those of the explicit scheme, there is no advantage in using the Beam-Warming method with small values of the CFL number. Figure 10.19 shows that with a CFL number of 3.0 the residuals drop to 10^{-10} in 900 iterations, which corresponds to a speedup of 12 compared with the results obtained with the CFL number equal to 0.5.

On another calculation, the explicit artificial parameter was set to 10, and the results are shown in Fig. 10.20 for a CFL number of 1.0. The shock wave is smeared by excessive dissipation present in the implicit algorithm, with no increase in the speedup of the convergence rate compared to the lower explicit artificial dissipation cases.

References

- [1] Hirsch, C.: *Numerical Computation of Internal and External Flows*. John Wiley & Sons, 1990.
- [2] Murman, E. M. and Cole, J. D.: Calculation of Plane Steady Transonic Flows, *AIAA J.* **9**(1), 114–121, 1971.
- [3] Jameson, A.: Iterative Solutions of Transonic Flows over Airfoils and Wings, including flows at Mach 1. *Comm. Pure and Applied Mathematics* **27**, 283–309, 1974.
- [4] Jameson, A.: Transonic Potential Flow Calculations using Conservative Form. Proceedings of the AIAA Second Computational Fluid Dynamics Conference, Hartford, pp. 148–161, 1975.
- [5] Holst, T. L. and Ballhaus, W. F.: Fast Conservative Schemes for the Full Potential Equation Applied to Transonic Flows. *AIAA J.* **17**, 145–152, 1979.
- [6] Liepmann, H. W. and Roshko, A.: *Elements of Gasdynamics*. John Wiley & Sons, 1957.

Problems

10-1. Show that, when central difference operators are applied to the TSD equation, the discretized equations are of similar form whether the conservative Eq. (10.4.1) or non-conservative Eq. (10.4.2) forms are used.

Hint: write the conservative operator $(\frac{1}{2}\varphi_x^2)_x$ using the stencil $\varphi_x = \varphi_{i+1/2} - \varphi_{i-1/2}$ and compare your result with the non-conservative operator $\varphi_x \varphi_{xx}$ written with the standard central difference operator $\delta_x \varphi = \varphi_{i+1} - \varphi_{i-1}$.

10-2. Solve the model problem 10.4 by introducing the boundary conditions through modification of the constants a, b, c in Eq. (10.5.8), instead of using halo cells.

10-3. Solve the model Problem 10.4 using the non-conservative full-potential equation for the case $K = 2.5$, and compare your results with those obtained with the TSD equation.

10-4. Examine the effect of using the reverse predictor-corrector steps (10.6.5)–(10.6.6) on the model Problems 10.11 and 10.14.

10-5. Repeat the model Problems 10.11 and 10.14 for the following inflow/outflow boundary conditions: Supersonic/Supersonic, Subsonic/Subsonic, Subsonic/Supersonic.

11

Incompressible Navier–Stokes Equations

11.1 Introduction

The solution of the incompressible Navier–Stokes equations is discussed in this chapter and that of the compressible form postponed to Chapter 12. It may appear logical to consider the two together and this can be done readily to the boundary layer equations, where the equation representing conservation of energy has to be added to complete the compressible form. The differences between the incompressible and compressible form of the Navier–Stokes equations are greater, the continuity equation for the former is elliptic but is hyperbolic in space and parabolic in time for the latter. These differences are discussed in more detail in Section 11.2, and the influence of the equation type on the boundary conditions are discussed in Section 11.3.

Once the equations and boundary conditions have been formulated, a solution algorithm is required. There are several numerical methods with which to solve the incompressible Navier–Stokes (INS) equations, and they can be divided in two classes. The first class uses the stream function variable to ensure that the conservation of mass is satisfied and this approach is well-established with two-dimensional equations, but extension to three-dimensions is not straightforward since the stream function scalar has to be replaced by a three-dimensional vector potential, which typically is the vorticity vector. Thus, the modified system of equations contains six variables as opposed to the four variables in incompressible Navier–Stokes equations expressed in primitive variables, which corresponds to the second class of solution methods.

The Marker and Cell (MAC) algorithm developed by Harlow and Welch [1] is an early method with which to solve unsteady equations. It uses a staggered grid with the Poisson equation to obtain the pressure at every time step. The method is described in [2] and is not discussed here. Another technique, this time mainly for steady flows, was developed by Patankar and Spalding [3] and

is called SIMPLE (Semi-Implicit Method for Pressure Linked Equations). A discussion of the method and its derivatives (SIMPLER, SIMPLEC and PISO) can be found in [4]. The pseudo or artificial compressibility method developed by Chorin [5] introduces an additional term in the continuity equation which transforms it from elliptic in space only to elliptic in space and hyperbolic in pseudo-time, hence its name. The artificial waves originating from the additional derivative term provide a mechanism for propagating information throughout the domain and drive the divergence of velocity towards zero at steady state. Since the methods of solving a system of travelling waves have been discussed in Chapters 5 and 10 (upwind schemes or central schemes with added artificial dissipation), the artificial compressibility method is discussed in Section 11.4. The discretization procedure of the time derivatives, convective and diffusive fluxes is detailed, and the use of the ADI procedure explained. In Section 11.5, its application to a sudden expansion laminar duct flow is discussed and the convergence of the solutions as a function of time steps and pseudo compressibility parameter β is studied. The application of the method to a laminar and turbulent flow over a flat plate is discussed in Section 11.6 and to multi-element airfoils in Section 11.7.

11.2 Analysis of the Incompressible Navier–Stokes Equations

The incompressible Navier–Stokes equations in tensor notation are given by Eqs. (P2.2.1) and (2.2.2), that is,

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (\text{P2.2.1})$$

$$\frac{\partial u_i}{\partial t} + u_i \frac{\partial u_i}{\partial x_j} = -\frac{1}{\varrho} \frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + f_i \quad (\text{P2.2.2})$$

where the viscous stresses σ_{ij} are given by Eq. (2.2.7). The continuity equation which remains the same for unsteady and steady incompressible flows has an elliptic character. The conservation of mass appears as a kinematic constraint to the solution of the momentum equation. Hence, there is no obvious means to couple the velocity and the pressure, which is done in the case of compressible flows through the density and an equation of state relating the density and the pressure. One way to obtain a relation is to take the divergence of the momentum equation while making use of the continuity constraint; this results in

$$\frac{1}{\varrho} \nabla^2 P = -\vec{\nabla}(\vec{v} \cdot \vec{\nabla}) \vec{v} + \vec{\nabla} \cdot \vec{f} \quad (11.2.1)$$

the Poisson equation for the pressure once the velocity field is known. This equation forms the basis of the pressure correction methods, which first solve the

momentum equations (P2.2.2) to obtain the velocity field for a known pressure, and then solve Eq. (11.2.1) to obtain an updated (or corrected) pressure field. The MAC and SIMPLE algorithms both use the Poisson equation to couple the pressure and velocity fields.

Another way is to modify the continuity equation so that it becomes hyperbolic in time so that the methods developed for compressible flows can be used with only a slight modification accounting for this transformation. The hyperbolic character can be obtained by introducing a time derivative $\partial/\partial t$ to the continuity equation, which would vanish upon reaching steady state. The method can therefore be applicable only to steady flows. The choice of the scalar quantity is obtained by elimination. The density cannot be chosen since it is constant, and the velocity already appears in the time derivative of the momentum equations. The pressure remains the only choice yielding

$$\frac{1}{\beta} \frac{\partial p}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (11.2.2)$$

where β is the artificial compressibility parameter introduced by Chorin [5]. This parameter can be obtained from dimensional analysis and has the dimensions of velocity². Since this is not a non-dimensional constant, it is not universal and its effect on the iterative procedure depends on the problem being considered. Its effect is discussed in subsection 11.5.3.

11.3 Boundary Conditions

The incompressible Navier–Stokes equations require boundary conditions. The system of equations, which contains four variables in three-dimensions (velocity field and temperature) needs the specification of three variables along the inflow and outflow boundaries (Table 11.1). In practice, if the far-field boundaries are placed far enough from the immerse body, then inviscid incompressible flow boundary conditions can be used. Specification of all variables at inflow boundaries is required together with one boundary condition at outflow boundaries. This leads to under specification of the mathematical formulation that could result in non-unique solutions [6]. Note that, in the case of inviscid flow, the

Table 11.1. Physical and numerical boundary conditions for the 3D incompressible flow equations.

	Navier–Stokes		Euler	
Physical conditions	Inflow	3	Inflow	3
	Outflow	3	Outflow	1
Numerical conditions	Inflow	1	Inflow	1
	Outflow	1	Outflow	3

results are consistent with those of Table 10.1 for the subsonic case since the speed of sound is infinite for the incompressible flow equations.

When the artificial compressibility method is used, the hyperbolic system in pseudo-time allows the use of the compressible boundary conditions discussed in Section 10.7 for inviscid and Section 12.2 for viscous flows. The one-dimensional incompressible Navier–Stokes equations in vector-variable form, with the pseudo-time derivative, are obtained from their two-dimensional counterparts (Eqs. (P2.17.2)–(P2.17.4)),

$$\frac{\partial}{\partial \tau} D + \frac{\partial}{\partial x} (E - E_v) = 0 \quad (11.3.1)$$

where

$$D = \begin{bmatrix} p \\ u \end{bmatrix} \quad E = \begin{bmatrix} \beta u \\ u^2 + p \end{bmatrix} \quad E_v = \begin{bmatrix} 0 \\ \tau_{xx} \end{bmatrix} \quad (11.3.2)$$

Rewriting Eqs. (11.3.1) and applying the method of characteristics described in Chapter 5, we obtain

$$\begin{aligned} \frac{\partial}{\partial \tau} D &= -\frac{\partial}{\partial x} (E - E_v) = -\frac{\partial E}{\partial D} \frac{\partial D}{\partial x} + \frac{\partial E_v}{\partial x} \\ &= -A \frac{\partial D}{\partial x} + \frac{\partial E_v}{\partial x} = -X_1 A_1 X_1^{-1} \frac{\partial D}{\partial x} + \frac{\partial E_v}{\partial x} \end{aligned} \quad (11.3.3)$$

If the left-hand side is multiplied by X_1^{-1} and the matrix moved inside the spatial and pseudo-time derivative,

$$\frac{\partial X_1^{-1} D}{\partial \tau} = -A_1 \frac{\partial X_1^{-1} D}{\partial x} + \frac{\partial X_1^{-1} E_v}{\partial x} \quad (11.3.4)$$

a wave equation with an added viscous term is obtained. The sign of the eigenvalues in A_1 determines the direction of the wave and it can be shown (Problem 11.1) that the eigenvalues are

$$\Lambda_{1\pm} = u \pm \sqrt{u^2 + \beta} \quad (11.3.5)$$

These are real and have opposite signs so that the flow remains subsonic with respect to the pseudo-sonic speed which depends on the local flow velocity and the parameter β

$$c = \sqrt{u^2 + \beta} \quad (11.3.6)$$

Chang and Kwak [7] have shown that a good choice for the parameter β is

$$\frac{\beta^2}{u_{\text{ref}}^2} \gg \left(1 + \frac{L}{\delta} \frac{4}{\text{Re}}\right)^2 - 1 \quad (11.3.7)$$

where u and δ are the velocity and length scales of the specific problem to be solved.

11.4 Artificial Compressibility Method: INS2D

The artificial compressibility method INS2D*, widely made popular by Roger and Kwak [8–10], is described in this section. For simplicity, the numerical method is discussed with the equations expressed in Cartesian coordinates for steady flows rather than in transformed form (subsection 2.2.4) for steady and unsteady flows. In this case, the governing equations are given in Problem 2.17,

$$\frac{\partial D}{\partial \tau} + \frac{\partial}{\partial x}(E - E_v) + \frac{\partial}{\partial y}(F - F_v) = 0 \quad (\text{P2.17.2})$$

where

$$D = \begin{bmatrix} p \\ u \\ v \end{bmatrix} \quad E = E_1 = \begin{bmatrix} \beta u \\ u^2 + p \\ uv \end{bmatrix} \quad F = E_2 = \begin{bmatrix} \beta v \\ uv \\ v^2 + p \end{bmatrix} \quad (\text{P2.17.2})$$

$$E_v = \begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \end{bmatrix} \quad F_v = \begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \end{bmatrix} \quad (\text{P2.17.3})$$

The viscous normal and shear stresses are given by Eq. (2.2.8)

$$\sigma_{xx} = 2\mu \frac{\partial u}{\partial x} \quad \sigma_{xy} = \sigma_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad \sigma_{yy} = 2\mu \frac{\partial v}{\partial y} \quad (2.2.8)$$

11.4.1 Discretization of the Artificial Time Derivatives

INS2D uses an implicit scheme to solve Eq. (P2.17.2) subject to the boundary conditions to be described in subsection 11.5.1, and the equation is written as

$$\frac{\partial D}{\partial \tau} + R = 0 \quad (11.4.1)$$

where

$$R = \frac{\partial}{\partial x}(E - E_v) + \frac{\partial}{\partial y}(F - F_v) \quad (11.4.2)$$

Application of a first order backward Euler formulas to Eq. (11.4.1) in pseudo-time τ yields

$$\frac{D^{n+1} - D^n}{\tau^{n+1} - \tau^n} = -R^{n+1} \quad (11.4.3)$$

where the superscript n denotes the pseudo-time iteration count. Application of Taylor series expansion, and noting that R is a function of D , leads to

$$R^{n+1} = R^n + \left(\frac{\partial R}{\partial D} \right)^n (D^{n+1} - D^n) \quad (11.4.4)$$

* INS2D and related software can be obtained from NASA Ames by signing a non-disclosure agreement. For details, see <http://people.nasa.gov/~rogers/home.html>

Substituting Eq. (11.4.4) into Eq. (11.4.3) and noting that the D 's are vectors, leads to the following delta form

$$\left[\frac{I}{\Delta\tau} + \left(\frac{\partial R}{\partial D} \right)^n \right] \Delta D^n = -R^n \quad (11.4.5)$$

where $\Delta D^n = D^{n+1} - D^n$ and $\Delta\tau = \tau^{n+1} - \tau^n$ and I is the 3×3 identity matrix, or in compact form

$$\mathbf{B} \Delta D^n = -R^n \quad (11.4.6)$$

where

$$\mathbf{B} = \left[\frac{I}{\Delta\tau} + \left(\frac{\partial R}{\partial D} \right)^n \right] \quad (11.4.7)$$

11.4.2 Discretization of the Convective Fluxes

The analysis in Section 11.2 showed that the system is hyperbolic, and the upwind method is applied as discussed in Section 5.5. The Jacobian matrices of the convective flux E and F are

$$A = \frac{\partial E}{\partial D} = \begin{bmatrix} 0 & \beta & 0 \\ 1 & 2u & 0 \\ 0 & v & u \end{bmatrix} \quad B = \frac{\partial F}{\partial D} = \begin{bmatrix} 0 & 0 & \beta \\ 0 & v & u \\ 1 & 0 & 2v \end{bmatrix} \quad (11.4.8)$$

and the eigenvalues Λ and eigenvectors X of A and B can be written as

$$\begin{aligned} \Lambda_1 &= X_1^{-1} A X_1 = \begin{bmatrix} \lambda_1^{(1)} & & \\ & \lambda_2^{(1)} & \\ & & \lambda_3^{(1)} \end{bmatrix} = \begin{bmatrix} u & 0 & 0 \\ 0 & u + c_1 & 0 \\ 0 & 0 & u - c_1 \end{bmatrix} \\ \Lambda_2 &= X_2^{-1} B X_2 = \begin{bmatrix} \lambda_1^{(2)} & & \\ & \lambda_2^{(2)} & \\ & & \lambda_3^{(2)} \end{bmatrix} = \begin{bmatrix} v & 0 & 0 \\ 0 & v + c_2 & 0 \\ 0 & 0 & v - c_2 \end{bmatrix} \\ X_1 &= \frac{1}{2\beta c_1^2} \begin{bmatrix} \lambda_1^{(1)} & & \\ & \lambda_2^{(1)} & \\ & & \lambda_3^{(1)} \end{bmatrix} = \begin{bmatrix} 0 & c_1\beta & -c_1\beta \\ 0 & u(c_1 + u) + \beta & u(c_1 - u) + \beta \\ 2\beta & v(c_1 + u) & v(c_1 - u) \end{bmatrix} \\ X_2 &= \frac{1}{2\beta c_2^2} \begin{bmatrix} \lambda_1^{(2)} & & \\ & \lambda_2^{(2)} & \\ & & \lambda_3^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & c_2\beta & -c_2\beta \\ -2\beta & u(c_2 + v) & u(c_2 - v) \\ 0 & v(c_2 + v) + \beta & v(c_2 - v) + \beta \end{bmatrix} \end{aligned} \quad (11.4.9)$$

where

$$c_1 = \sqrt{u^2 + \beta} \quad (11.4.10a)$$

and

$$c_2 = \sqrt{v^2 + \beta}. \quad (11.4.10b)$$

The inverse matrices X_1^{-1} and X_2^{-1} are given by

$$X_1^{-1} = \begin{bmatrix} -v & -uv & u^2 + \beta \\ c_1 - u & \beta & 0 \\ -c_1 - u & \beta & 0 \end{bmatrix} \quad X_2^{-1} = \begin{bmatrix} -u & -v^2 - \beta & vu \\ c_2 - v & 0 & \beta \\ -c_2 - v & 0 & \beta \end{bmatrix} \quad (11.4.11)$$

As described in Section 5.5, A and B can be decomposed as the sum of two parts, one part corresponding to positive eigenvalues and the other to negative eigenvalues,

$$A = A^+ + A^- \quad \text{and} \quad B = B^+ + B^- \quad (11.4.12)$$

where

$$A^\pm = X_1 \begin{bmatrix} \frac{\lambda_1^{(1)} \pm |\lambda_1^{(1)}|}{2} & & \\ & \frac{\lambda_2^{(1)} \pm |\lambda_2^{(1)}|}{2} & \\ & & \frac{\lambda_3^{(1)} \pm |\lambda_3^{(1)}|}{2} \end{bmatrix} X_1^{-1} \quad (11.4.13)$$

$$B^\pm = X_2 \begin{bmatrix} \frac{\lambda_1^{(2)} \pm |\lambda_1^{(2)}|}{2} & & \\ & \frac{\lambda_2^{(2)} \pm |\lambda_2^{(2)}|}{2} & \\ & & \frac{\lambda_3^{(2)} \pm |\lambda_3^{(2)}|}{2} \end{bmatrix} X_2^{-1}$$

Application of the upwind scheme of Section 5.5 to the derivative of the convective flux in the x - and y -directions yields,

$$\frac{\partial E}{\partial x} \approx \frac{\tilde{E}_{i+1/2,j} - \tilde{E}_{i-1/2,j}}{\Delta x} \quad \frac{\partial F}{\partial y} \approx \frac{\tilde{F}_{i,j+1/2} - \tilde{F}_{i,j-1/2}}{\Delta y} \quad (11.4.14)$$

where, referring to Fig. 11.1, $\tilde{E}_{i+1/2,j}$ is a numerical flux at j and $i + 1/2$ is the discrete spatial index for the x -direction. Similarly $\tilde{F}_{i,j+1/2}$ is a numerical flux at i and $j + 1/2$. They are given by

$$\tilde{E}_{i+1/2,j} = \frac{1}{2}[E(D_{i+1,j}) + E(D_{i,j}) - \phi_{i+1/2,j}] \quad (11.4.15a)$$

$$\tilde{F}_{i,j+1/2} = \frac{1}{2}[F(D_{i,j+1}) + F(D_{i,j}) - \phi_{i,j+1/2}] \quad (11.4.15b)$$

When

$$\phi_{i+1/2,j} = 0 \quad \text{and} \quad \phi_{i,j+1/2} = 0 \quad (11.4.16)$$

the convective flux terms in Eq. (11.4.14) are represented by a second-order accurate central difference scheme. A first-order accurate upwind difference scheme is given by

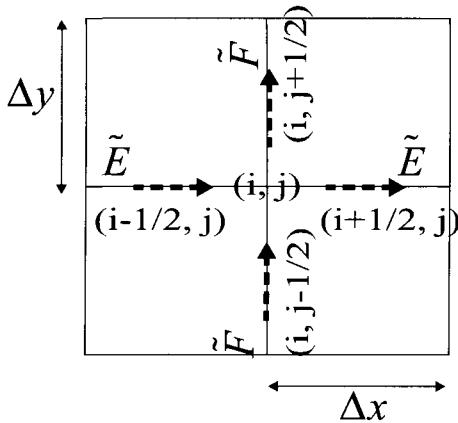


Fig. 11.1. Difference stencil for derivatives in the x -direction.

$$\phi_{i+1/2,j} = \Delta E_{i+1/2,j}^+ - \Delta E_{i+1/2,j}^- \quad (11.4.17a)$$

$$\phi_{i,j+1/2} = \Delta F_{i,j+1/2}^+ - \Delta F_{i,j+1/2}^- \quad (11.4.17b)$$

where ΔE^\pm and ΔF^\pm are the flux differences across the positive or negative traveling waves. As discussed in [9], they are given by

$$\Delta E_{i+1/2,j}^\pm = A^\pm(\bar{D}_{i+1/2,j}) \Delta D_{i+1/2,j} \quad (11.4.18a)$$

$$\Delta F_{i,j+1/2}^\pm = B^\pm(\bar{D}_{i,j+1/2}) \Delta D_{i,j+1/2} \quad (11.4.18b)$$

Here

$$\bar{D}_{i+1/2,j} = \frac{1}{2}(D_{i+1,j} + D_{i,j}) \quad \Delta D_{i+1/2,j} = D_{i+1,j} - D_{i,j} \quad (11.4.19a)$$

$$\bar{D}_{i,j+1/2} = \frac{1}{2}(D_{i,j+1} + D_{i,j}) \quad \Delta D_{i,j+1/2} = D_{i,j+1} - D_{i,j} \quad (11.4.19b)$$

11.4.3 Discretization of the Viscous Fluxes

The discretization of the viscous fluxes is much more simple than the discretization of the convective fluxes. Viscous diffusion occurs in all direction, and the discretization of the viscous terms is always performed with central formulas [11]. Here we approximate the viscous fluxes in Eqs. (P2.17.2) with second-order accuracy on a compact stencil

$$\frac{\partial E_v}{\partial x} = \frac{(E_v)_{i+1/2,j} - (E_v)_{i-1/2,j}}{\Delta x} \quad (11.4.20a)$$

$$\frac{\partial F_v}{\partial y} = \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y} \quad (11.4.20b)$$

11.4.4 System of Discretized Equation

With Eqs. (11.4.14) and (11.4.20), the residual vector R^n -term in Eq. (11.4.5) can be written as

$$\begin{aligned} R_{i,j}^n \approx & \frac{\tilde{E}_{i+1/2,j} - \tilde{E}_{i-1/2,j}}{\Delta x} + \frac{\tilde{F}_{i,j+1/2} - \tilde{F}_{i,j-1/2}}{\Delta y} \\ & - \frac{(E_v)_{i+1/2,j} - (E_v)_{i-1/2,j}}{\Delta x} - \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y} \end{aligned} \quad (11.4.21)$$

The calculation of the exact Jacobian matrix of the residual vector $(\frac{\partial R}{\partial D})^n$ can be very expensive, particularly when higher order upwind methods are used. Therefore, it is more economical to approximate the exact Jacobian from the residual R^n resulting from the first order upwind method. Applying the first order upwind method to the convective terms (Eqs. (11.4.16)–(11.4.19)) and maintaining the central differences for the viscous terms (Eq. 11.4.20) yields the following residual R^n :

$$\begin{aligned} R_{i,j}^n \approx & \frac{E_{i+1,j} - E_{i-1,j}}{2\Delta x} - \frac{\Delta E_{i+1/2,j}^+ - \Delta E_{i+1/2,j}^- - \Delta E_{i-1/2,j}^+ + \Delta E_{i-1/2,j}^-}{2\Delta x} \\ & + \frac{F_{i,j+1} - F_{i,j-1}}{2\Delta y} - \frac{\Delta F_{i,j+1/2}^+ - \Delta F_{i,j+1/2}^- - \Delta F_{i,j-1/2}^+ + \Delta F_{i,j-1/2}^-}{2\Delta y} \\ & - \frac{(E_v)_{i+1/2,j} - (E_v)_{i-1/2,j}}{\Delta x} - \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y} \end{aligned} \quad (11.4.22)$$

The exact Jacobian matrix of the residual given by Eq. (11.4.22) forms a banded matrix \mathbf{B} and can be written in the form

$$\left(\frac{\partial R}{\partial D} \right) = \mathbf{B} \left[\frac{\partial R_{i,j}^n}{\partial D_{i,j-1}}, 0, \dots, 0, \frac{\partial R_{i,j}^n}{\partial D_{i-1,j}}, \frac{\partial R_{i,j}^n}{\partial D_{i,j}}, \frac{\partial R_{i,j}^n}{\partial D_{i+1,j}}, 0, \dots, 0, \frac{\partial R_{i,j}^n}{\partial D_{i,j+1}} \right] \quad (11.4.23)$$

As discussed in [9], the discrete form of Eq. (11.2.5) can be written as

$$\mathbf{B}[V, 0, \dots, 0, X, Y, Z, 0, \dots, 0, W]\Delta D = -R \quad (11.4.24)$$

where V, X, Y, Z and W denote vectors of 3 by 3 blocks which lie on the diagonals of the banded matrix, with the Y vector lying on the main diagonal. These vectors can be approximated as

$$\begin{aligned}
V &\Leftarrow \frac{\partial R_{i,j}^n}{\partial D_{i,j-1}} \approx \frac{1}{2\Delta y}(-B_{i,j-1} - B_{i,j-1/2}^+ + B_{i,j-1/2}^-) + \mu I_m \frac{\partial}{\partial y} \\
X &\Leftarrow \frac{\partial R_{i,j}^n}{\partial D_{i-1,j}} \approx \frac{1}{2\Delta x}(-A_{i-1,j} - A_{i-1/2,j}^+ + A_{i-1/2,j}^-) + \mu I_m \frac{\partial}{\partial x} \\
Y &\Leftarrow \frac{\partial R_{i,j}^n}{\partial D_{i,j}} \approx \frac{1}{2\Delta x}(A_{i+1/2,j}^+ + A_{i-1/2,j}^+ - A_{i+1/2,j}^- - A_{i-1/2,j}^-) \\
&\quad + \frac{1}{2\Delta y}(B_{i,j+1/2}^+ + B_{i,j-1/2}^+ - B_{i,j+1/2}^- - B_{i,j-1/2}^-) \\
Z &\Leftarrow \frac{\partial R_{i,j}^n}{\partial D_{i+1,j}} \approx \frac{1}{2\Delta x}(A_{i+1,j} - A_{i+1/2,j}^+ + A_{i+1/2,j}^-) - \mu I_m \frac{\partial}{\partial x} \\
W &\Leftarrow \frac{\partial R_{i,j}^n}{\partial D_{i,j+1}} \approx \frac{1}{2\Delta y}(-B_{i,j+1} - B_{i,j+1/2}^+ + B_{i,j+1/2}^-) - \mu I_m \frac{\partial}{\partial y}
\end{aligned} \tag{11.4.25}$$

where I_m is a modified identity matrix given by

$$I_m = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11.4.26}$$

The solution of Eq. (11.4.24) is obtained with the ADI method discussed in subsection 4.5.2. Thus, we first write Eq. (11.4.24) for a given time $n+1$. Along the x -direction, this gives

$$\mathbf{B}[X, Y, Z]\Delta D = -R - \Delta D_{j-1}V - \Delta D_{j+1}W \tag{11.4.27}$$

which is solved with the block elimination method discussed in subsection 4.4.3 [see Eqs. (4.4.32) and (4.4.34)]. Similarly, Eq. (4.2.24) is written along the y -direction,

$$\mathbf{B}[V, Y, W]\Delta D = -R - X\Delta D_{i-1} - Z\Delta D_{i+1}, \tag{11.4.28}$$

which is again solved with the block elimination method. Equations (11.4.27) and (11.4.28) are solved iteratively until convergence of the iterative process due to the calculation of the inexact Jacobian. Then the time step is incremented and the above procedure is repeated for the next time step until convergence of the steady-state system (11.4.5).

11.5 Model Problem: Sudden Expansion Laminar Duct Flow

To demonstrate the solution of the incompressible Navier–Stokes equations with the numerical method of Section 11.4, we consider a model problem corresponding to the symmetrical half of a laminar flow downstream of a sudden expansion in a duct of length L and height H as shown in Fig. 11.2.

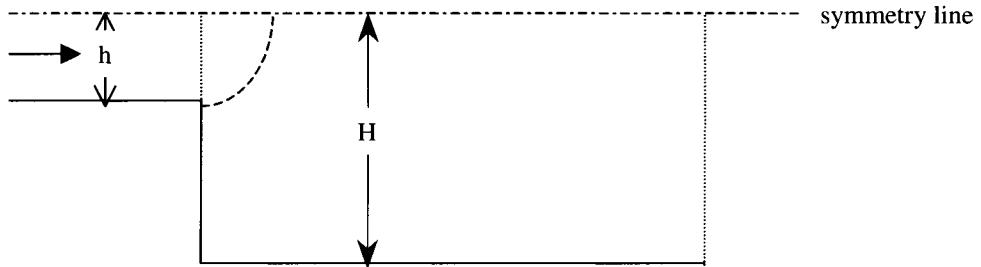


Fig. 11.2. Sudden expansion laminar duct flow.

We assume that the flow is fully developed at $x = 0$ and that the velocity profile u is given by

$$u(0, y) = \begin{cases} 0 & y \leq H - h \\ 1.5 \left[2\frac{y - H + h}{h} - \left(\frac{y - H + h}{h} \right)^2 \right] & H - h \leq y \leq H \end{cases} \quad (11.5.1a)$$

with

$$v = 0, \quad \text{and} \quad \frac{\partial^2 p}{\partial x^2} = 0 \quad (11.5.1b)$$

Along the wall, $y = 0$, it is assumed that

$$u = v = 0, \quad \text{and} \quad \frac{\partial^2 p}{\partial y^2} = 0 \quad (11.5.2)$$

and on the symmetry plane, $y = H$,

$$\frac{\partial u}{\partial y} = v = 0, \quad \text{and} \quad \frac{\partial p}{\partial y} = 0 \quad (11.5.3)$$

As discussed in Section 11.3, it is also necessary to specify three boundary conditions at $x = L$ and since the flow is fully developed, we write

$$\frac{\partial^2 u}{\partial x^2} = v = 0, \quad \frac{\partial^2 p}{\partial x^2} = 0 \quad (11.5.4)$$

11.5.1 Discretization of the Boundary Conditions

In terms of difference approximations, the boundary conditions given by Eqs. (11.5.1)–(11.5.4) can be written as

Inflow:

At $x = 0$ and $0 \leq j \leq J$

$$u_{0,j} = \begin{cases} 0 & y_j \leq H - h \\ 1.5 \left[2\frac{y_j - H + h}{h} - \left(\frac{y_j - H + h}{h} \right)^2 \right] H - h \leq y_j \leq H \end{cases} \quad (11.5.5a)$$

$$v_{0,j} = 0 \quad (11.5.5b)$$

$$p_{0,j} = 2p_{1,j} - p_{2,j} \quad (11.5.5c)$$

No-slip surface:

At $y = 0$ and $0 \leq i \leq I$,

$$u_{i,0} = v_{i,0} = 0 \quad (11.5.6a)$$

and the pressure is approximated with second-order accurate extrapolation

$$p_{i,0} = 2p_{i,1} - p_{i,2} \quad (11.5.6b)$$

Symmetry line:

At the centerline, $j = J$, $0 \leq i \leq I$, $\frac{\partial u}{\partial y}$ and $\frac{\partial p}{\partial y}$ are represented by second order approximations

$$u_{i,J} = \frac{4u_{i,J-1} - u_{i,J-2}}{3} \quad (11.5.7a)$$

$$p_{i,J} = \frac{4p_{i,J-1} - p_{i,J-2}}{3} \quad (11.5.7b)$$

In addition, we set

$$v_{i,J} = 0 \quad (11.5.7c)$$

Outflow:

The boundary conditions at $i = I$, $0 \leq j \leq J$ are written as

$$u_{I,j} = 2u_{I-1,j} - u_{I-2,j} \quad (11.5.8a)$$

$$p_{I,j} = 2p_{I-1,j} - p_{I-2,j} \quad (11.5.8b)$$

$$v_{I,j} = v_{I-1,j} \quad (11.5.8c)$$

11.5.2 Initial Conditions

The initial conditions are:

$$u(x, y, 0) = v(x, y, 0) = p(x, y, 0) = 0 \quad (11.5.9)$$

Here we assume that at the $i = 0$ boundary, $u_{0,j}$ is given by Eq. (11.5.5a) and at $i = I$, $u_{I,j}$ is given by

$$u_{I,j} = 1.5 \frac{h}{H} \left[2\frac{y_j}{H} - \left(\frac{y_j}{H} \right)^2 \right] \quad (11.5.10)$$

The values of $u_{i,j}$ for $0 < i < I$ and $0 < j < J$ are obtained by linear interpolation. From the continuity equation and the definition of $u_{i,j}$, the normal velocity $v_{i,j}$ can be written as

$$v(x, y) = \begin{cases} -\frac{1.5h}{HL} \left(\frac{y^2}{H} - \frac{y^3}{3H^2} \right) & y \leq H-h \\ \frac{1.5}{L} \left[\frac{(y-H+h)^2}{h} - \frac{(y-H+h)^3}{3h^2} - \frac{h}{H} \left(\frac{y^2}{H} - \frac{y^3}{3H^2} \right) \right] & y > H-h \end{cases} \quad (11.5.11)$$

The pressure $p_{i,j}$ is assumed to be zero.

11.5.3 Solution Procedure and Sample Calculations

With the boundary and initial conditions specified, the solution of Eq. (11.4.24) can be obtained by the line iterative (ADI) method discussed in the previous section. The computer program given in Appendix B has nine subroutines and a MAIN program. Here we present sample calculations for this problem and discuss the behavior of the solutions as a function of the pseudo-time step Δt and the parameter β .

The input data, given in MAIN, comprises a uniform grid (I, J) , the geometrical parameters (L, H, h) , the Reynolds number $R_L = \frac{u_0 L}{\nu}$, the parameter β . In addition, the time marching algorithm is provided with an upper limit number of time iterations NLIMIN and a possibility for nonuniform time steps. These at first increase linearly with initial Δt_0 up to a specified Δt_f and remain constant thereafter. The initial conditions (including boundary conditions) are also specified in MAIN. Convergence is based on the solutions of Eq. (11.4.24) obtained by solving Eqs. (11.4.27) and (11.4.28) iteratively. The banded matrix \mathbf{B} in Eq. (11.4.27) is computed in the subroutines FLUX_DX, PHI_IH. Its right hand side is computed in the subroutine RESIDUALX, the banded matrix \mathbf{B} in Eq. (11.4.28) in the subroutines FLUX_DF, PHI_JH, and its right hand side computed in the subroutine RESIDUALY. The unknowns ΔD are obtained by the ADI method in the subroutine SOLVER.

Once a solution to Eq. (11.4.24) is obtained, the boundary conditions are updated in MAIN and, since the boundary conditions at $x = L$ are approximated, it is also necessary to verify the conservation of mass determined at $x = 0$ by integrating Eq. (11.5.1a) across $y = 0$ and $y = H$ at $x = L$.

Figure 11.3 shows the effect of β on the velocity profiles u/u_0 across the duct at several x locations ($x = 0.9, 1.8, 9, 15$). These calculations were made by taking the following parameters: $I = 200$, $J = 100$, $H = 1$, $h = 1/2$, $R_h = 25$, with β varying from 100, 50, 1, (the range as recommended in [8]) $\Delta t_0 = 0.1$, $\Delta t_f = 1$, and for a convergence criteria $|\Delta D| < 10^{-5}$. As can be seen, the velocity profiles are essentially independent of β .

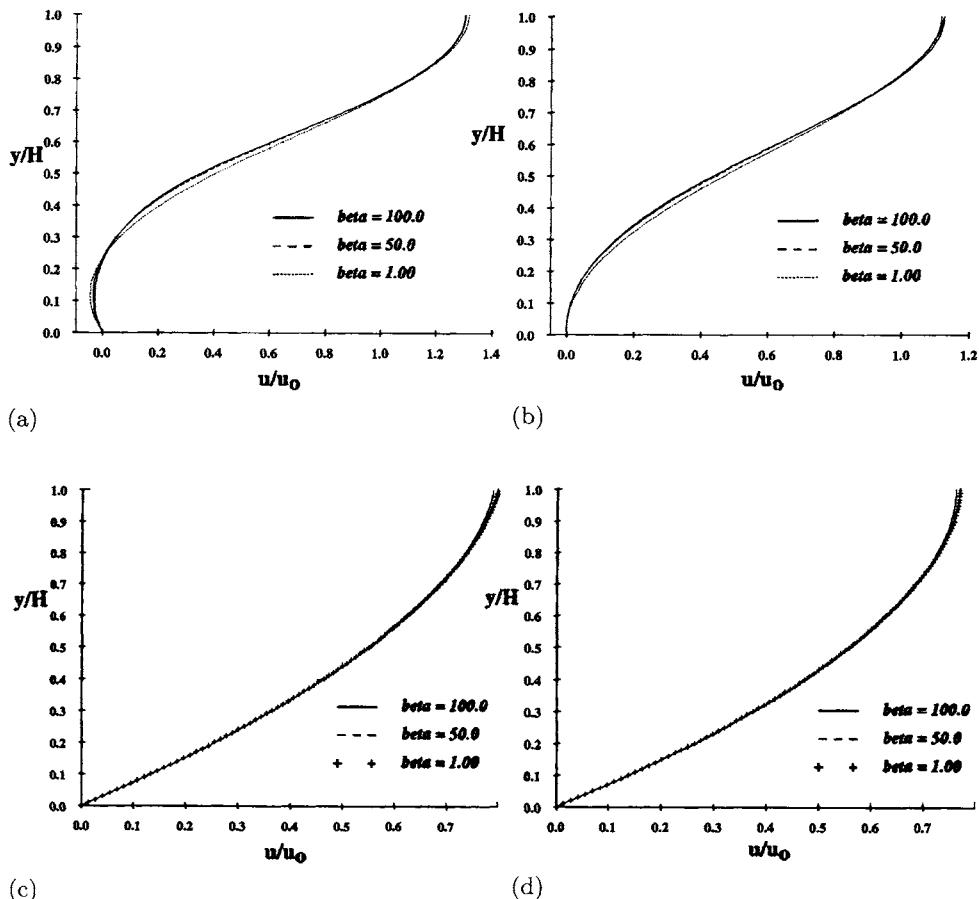


Fig. 11.3. Effect of β on the velocity profiles for $R_h = 25$, $h/H = 0.5$, $L/H = 15$ at x/H (a) 0.9, (b) 1.8, (c) 9 and (d) 15.

Analysis of the flow solutions indicate that steady state convergence within the specified tolerance is reached at about $t = 500$ for $\beta = 100$ and that the reattachment location is essentially independent of β as shown in Table 11.2. However, the error on the conservation of mass defined by

$$\dot{m}_{x=0} - \dot{m}_{x=L}$$

where

$$\dot{m} = \int_0^H u \, dy$$

becomes smaller as β increases. Here, $\dot{m}_{x=0}$ is the mass flow rate at $x = 0$ and $\dot{m}_{x=L}$ at $x = L$.

Table 11.3 shows the effect of the grid size on the reattachment location, x_r/H , and on the error of conservation of mass ($\dot{m}_{x=0} - \dot{m}_{x=L}$) for $R_h = 50$

Table 11.2. Effect of β on attachment location x_r/H for $R_h = 50$.

β	x_r/H	$\dot{m}_{x=0} - \dot{m}_{x=L}$
1	2.9250	-0.0547
50	3.3750	-0.0125
100	3.3750	-0.0090

Table 11.3. Effect of grid on the reattachment location and conservation of mass, $\beta = 100$.

Grid	x_r/H	$\dot{m}_{x=0} - \dot{m}_{x=L}$
50 × 25	3.00000	-0.0053
80 × 40	3.18750	-0.0082
100 × 50	3.30000	-0.0086
150 × 75	3.40000	-0.0089
200 × 100	3.3750	-0.0090

Table 11.4. Effect of grid on the reattachment location and conservation of mass, $\beta = 1$.

Grid	x_r/H	$\dot{m}_{x=0} - \dot{m}_{x=L}$
50 × 25	2.7000	-0.0433
80 × 40	2.8125	-0.0522
100 × 50	2.8500	-0.0537
150 × 75	2.9000	-0.0552
200 × 100	2.9250	-0.0547

Table 11.5. Comparison of Calculated Results [12, 13, 14, 15].

Investigator	$x_r/(hR_h)$	R_h
Present Method	0.135	50
Allen [9]	0.138	50
Kwon et al. [10]	0.138	50
Hung [11]	0.132	46.6
Agarwal [12]	0.136	46.6

and $\beta = 100$. Table 11.4 shows similar results for $\beta = 1$. These results show that, as expected, the reattachment location x_r/H becomes constant with a value equal to 3.3750 as the mesh is refined (grid convergence).

The results shown in Table 11.5 indicate that the reattachment length $x_r/(hR_h)$ is in good agreement with the boundary-layer solutions of Allen [12] and Kwon et al. [13] and the Navier–Stokes solutions of Hung [14] and Agarwal [15].

11.6 Model Problem: Laminar and Turbulent Flat Plate Flow

This section discusses the application of the INS2D code described in the previous section to a laminar and turbulent flow over a flat plate. This is a useful test case to validate a CFD code since analytical solutions for a laminar flow and experimental results for a turbulent flow are available.

Of the three modifications to the INS2D code of the previous section, the first one is the modification of the grid in the y -direction. As discussed in subsection 7.4.2, for laminar flows, it is often sufficient to use a uniform grid in the y -direction. For turbulent flows, however, a uniform grid is not satisfactory because the boundary layer thickness and dimensionless wall shear are much larger in turbulent flows than laminar flows. Since short steps in y must be taken to maintain computational accuracy when the wall shear is large, the steps near the wall in a turbulent flow must be shorter than the corresponding steps in a laminar flow under similar conditions. A convenient grid for this purpose is the grid given by Eq. (7.4.2) in the real coordinate system. The y -convective flux is then modified as

$$\frac{\partial F}{\partial y} \approx \frac{\tilde{F}_{i,j+1/2} - \tilde{F}_{i,j-1/2}}{\Delta y_j} \quad (11.6.1)$$

where

$$\Delta y_j = \frac{(y_{j+1} - y_{j-1})}{2}$$

varies from cell to cell. Similarly, the y -viscous flux is modified as

$$\frac{\partial F_v}{\partial y} = \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y_j} \quad (11.6.2)$$

Calculations for turbulent flow with the first-order upwind scheme used in the INS2D showed the need for higher accuracy. As a result it was replaced by the third-order upwind scheme discussed in Section 5.5. This second modification is straightforward, but care must be taken at the boundary of the computational domain as the computational stencil is larger than in the first-order upwind method. In our case, zero-order extrapolation of the dissipation is used at the boundaries.

A third modification is the introduction of a turbulence model to compute turbulent flows. This is achieved by using the Cebeci-Smith turbulence model described in subsection 3.2.1. The resulting subroutine is named CS_TURB and is called before the fluxes are evaluated.

Once these modifications are made to the solution algorithm, it is necessary to select appropriate boundary conditions consistent with the computational domain. They were chosen as follows:

- The top boundary velocities are specified and the pressure is extrapolated;
- The bottom boundary is specified on the flat plate, with zero pressure-gradient normal to the wall. Symmetric flow is specified upstream of the plate;
- The left boundary conditions are specified, and the normal velocity is extrapolated from the interior;
- The right boundary pressure is specified, and the velocities are extrapolated.

The resulting program is included in Appendix B. Since the changes are minimal, the program is not further discussed here, except that for commonality, the eddy viscosity is computed for both laminar and turbulent flows. In the former, the values are set to zero, whereas in the latter, they are obtained with the Cebeci-Smith turbulence model.

Figure 11.4 shows the results for a laminar flow at a Reynolds number of 10^6 . Figure 11.4a shows the velocity profiles at two x -stations together with the Blasius solution [16], and Fig. 11.4b shows the computed local skin-friction results with the Blasius solution.

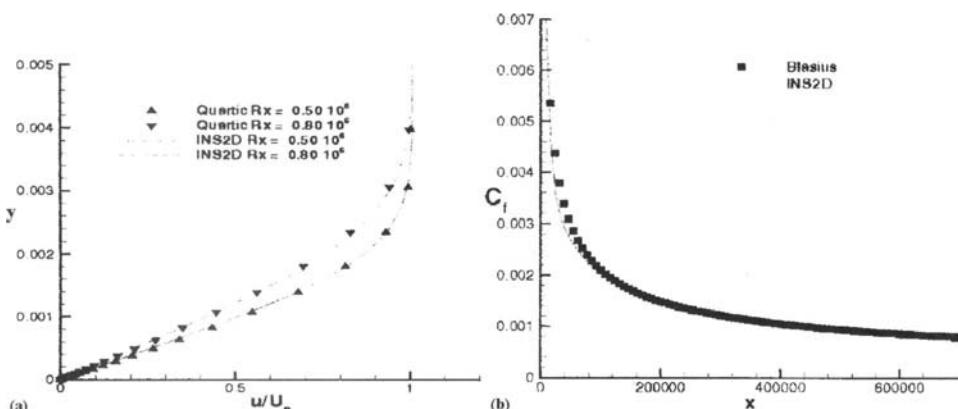


Fig. 11.4. Comparison of computed and Blasius results for a laminar flow at a Reynolds number of 10^6 . (a) velocity profiles and (b) local skin-friction coefficients.

The computed results for a turbulent flow over a flat plate at a Reynolds number of 10^7 are shown in Figs. 11.5 and 11.6 together with the empirical relations based on experimental data [16]. Figure 11.5a shows a comparison between the computed velocity profiles and those based on the $1/7^{\text{th}}$ power-law profile [16] and Fig. 11.5b shows those for the local skin-friction coefficients.

In order to validate the turbulence model implementation, the near-wall region is examined further in Fig. 11.6. As can be seen, the Cebeci-Smith turbulence model duplicates the experimentally observed laminar sublayer and law-of-the wall region.

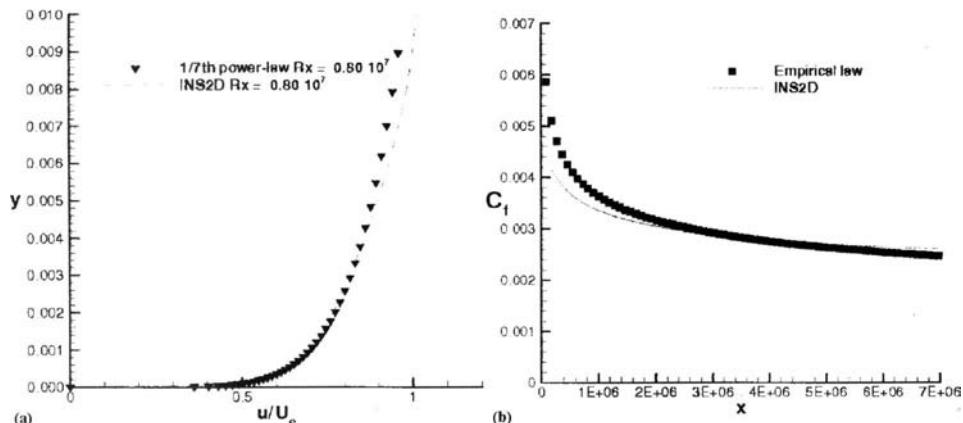


Fig. 11.5. Comparison of computed and experimental results for a turbulent flow over a flat plate at a Reynolds number of 10^7 . (a) velocity profiles at $R_x = 0.80 \times 10^7$ and (b) local skin-friction coefficients.

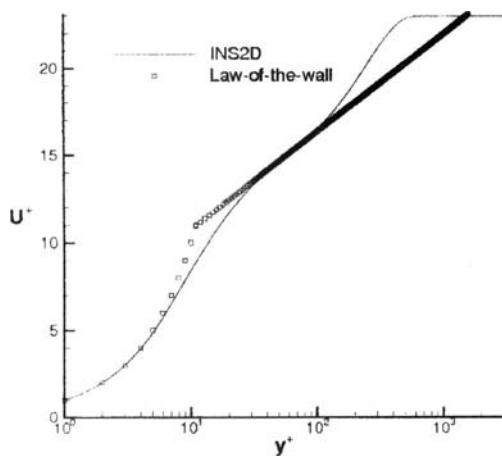


Fig. 11.6. Near-wall region comparison of the INS2D results with the law-of-the-wall.

11.7 Applications of INS2D

The INS method discussed in the previous two sections have been applied to a number of aerodynamic flows, including high lift flows, see for example [17–19]. In this section we present results for multielement airfoils computed with several turbulence models that include the SA model discussed in Section 3.3 the one-equation models of Baldwin-Barth (BB) [20] and Durbin-Mansour (DM) [21], and SST model of Menter [22]. In the calculations an overset Chimera grid approach was used. The performance of each of the turbulence models was evaluated for test cases involving different angles of attack, Reynolds numbers, and flap riggings. The resulting surface pressure coefficients, skin friction, velocity profiles, and lift, drag, and moment coefficients were compared with experi-

mental data. The models produced very similar results in most cases. Excellent agreement between computational and experimental surface pressures was observed, but only moderately good agreement was seen in the velocity profile data. In general, the difference between the predictions of the different models was less than the difference between the computational and experimental data.

The three-element configuration used in the study conducted in [23] is a McDonnell-Douglas airfoil. The experimental measurements include surface pressure, skin friction, and velocity profiles. Two configurations, A and B, were used, each with a 30 degree slat deflection and a 30 degree flap deflection. The geometries differed only in their flap rigging: configuration *A* had a slightly smaller flap gap than configuration *B*. Velocity profiles were measured in the experiment at 9 different survey stations along the top surface of the main element and the flap. Figure 11.7 shows the locations of these stations, as well as the two different flap positions of geometry *A* and geometry *B*.

Figure 11.8 shows the grids used around configuration *A*. Only every other grid line in each direction is shown for clarity. A total of 68,000 grid points and six zones were used: a 121×41 *C*-grid around the slat (top of Fig. 11.8); a 321×101 *C*-grid around the main element (near field shown in middle of Fig. 11.8); a 141×51 *C*-grid around the flap (top of Fig. 11.8); a 41×31 *H*-grid in the wake of the flap (middle of Fig. 11.8); a 131×61 *H*-grid extending from the main elements' flap cove to the downstream far-field (bottom of Fig. 11.8); and a 141×101 embedded grid above the flap (middle of Fig. 11.8). The normal wall spacing for all grids was 2×10^{-6} chords. The overlaid chimera scheme allowed individual grids to be generated for each airfoil element. When the grid for one element intersected another airfoil element, a hole was cut to remove grid points lying inside the element. This created a hole boundary. The fringe-point variables on the hole boundaries were updated by interpolating the value of the dependent variables from interior points of neighboring grids. Similarly, the variables on the outer boundaries of all but the main-element grid were updated using interpolation of dependent variables from neighboring grids.

Figures 11.9 to 11.14 present a sample of results from [19]. Figure 11.9 shows the computed and experimental c_p distributions for geometry *A* and $Re = 9 \times 10^6$. Data is shown for the slat, main element, and flap at $\alpha = 8^\circ$, and 21° . In

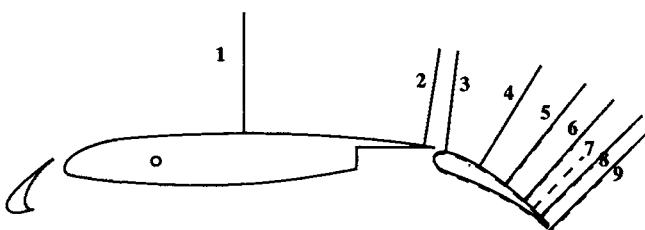


Fig. 11.7. Geometry of the three-element airfoil and velocity survey station locations.

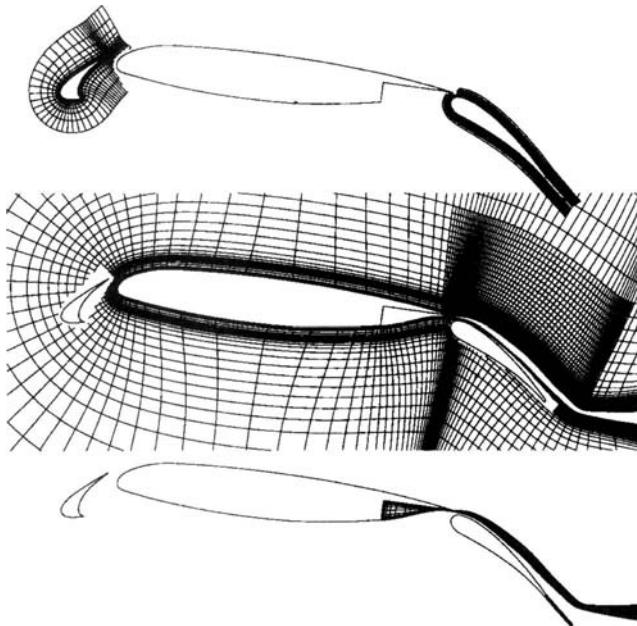


Fig. 11.8. Grids around the three-element airfoil.

general, the agreement between the c_p data of the experiment and all four of the turbulence models is good. The biggest discrepancies occur for the lower angle of attack, particularly on the slat and the flap. This directly correlates to the amount of separation on the flap: the greater the region of separation on the flap, the lower the lift on the flap and slat. The SST results show the greatest evidence of separation on the flap: a flattening of the c_p distribution near the trailing edge. The SST results also show the best agreement in the pressure on the upper surface of the slat. The only other notable aspect of these c_p plots is that the DM model has trouble matching the experimental pressures on the flap at 21° .

The experimental measurements of the skin friction coefficient, are available on only a few points on the upper surface of the main element and flap. These data points are plotted along with the computational results in Fig. 11.10 for the same geometry A and $Re = 9 \times 10^6$ for $\alpha = 8^\circ$ and $\alpha = 21^\circ$. The skin friction is fairly well predicted by all but the DM model. The SST models consistently predict the highest values, giving it the best agreement with the experiment on the flap and the main-element trailing-edge.

Figures 11.11 and 11.12 plot the lift coefficient c_l and the pitching moment coefficient c_m versus angle-of-attack, and the drag coefficient c_d versus c_l , for the computations and the experiment for geometry B and $Re = 9 \times 10^6$. The lift values are all quite close up to 16 degrees, as anticipated by the c_p results. None of the models agree with the experimental value of maximum lift. This is most

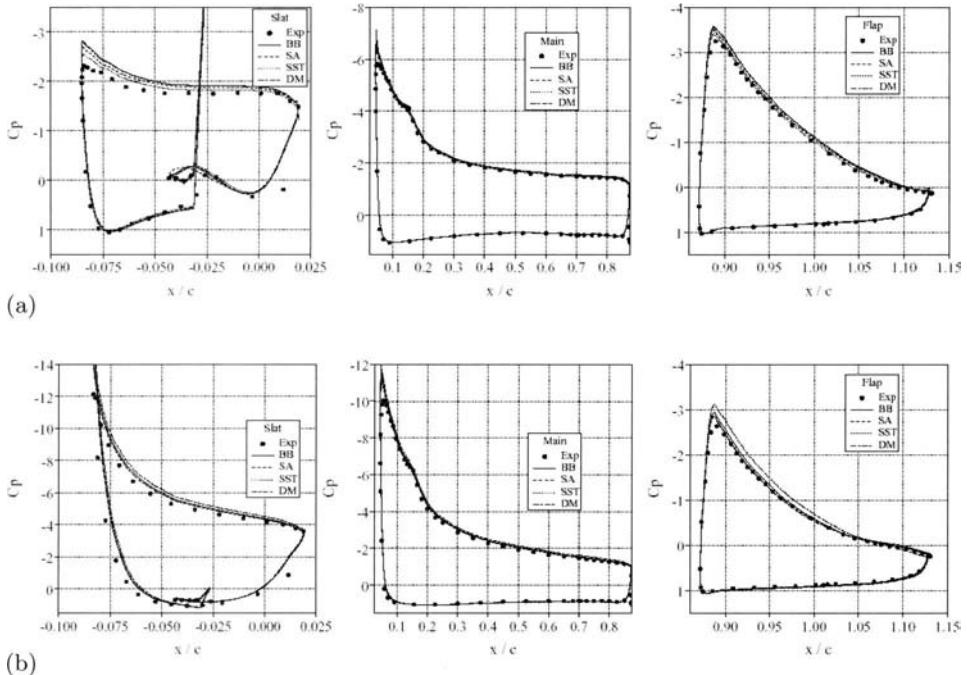


Fig. 11.9. c_p surface data for Geometry A and $Re = 9 \times 10^6$ with (a) $\alpha = 6^\circ$ and (b) $\alpha = 21^\circ$.

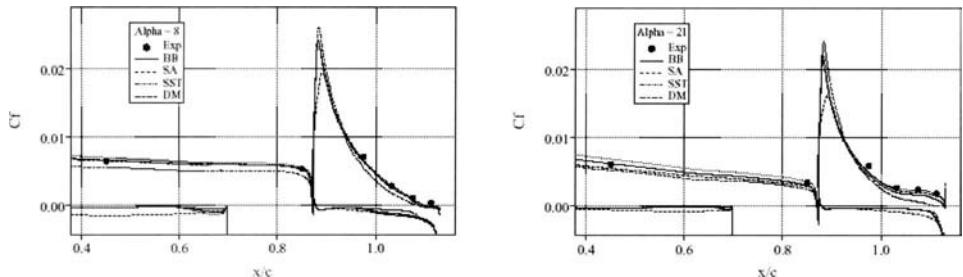


Fig. 11.10. Skin friction coefficient for geometry A and $Re = 9 \times 10^6$ (left $\alpha = 8^\circ$, right $\alpha = 21^\circ$).

likely because the experiment does start to undergo some three-dimensional effects at the very high values of lift. The drag coefficient, c_d is not well predicted by any model. In the computations the drag is computed by directly integrating the pressure and skin friction forces on the surface. This method has been shown to be extremely sensitive to a number of parameters such as normal grid spacing, grid skewness, wall extrapolation, etc. The AIAA 1st and 2nd Drag Prediction Workshops [24] have been studying the accuracy of Navier–Stokes methods for predicting c_d .

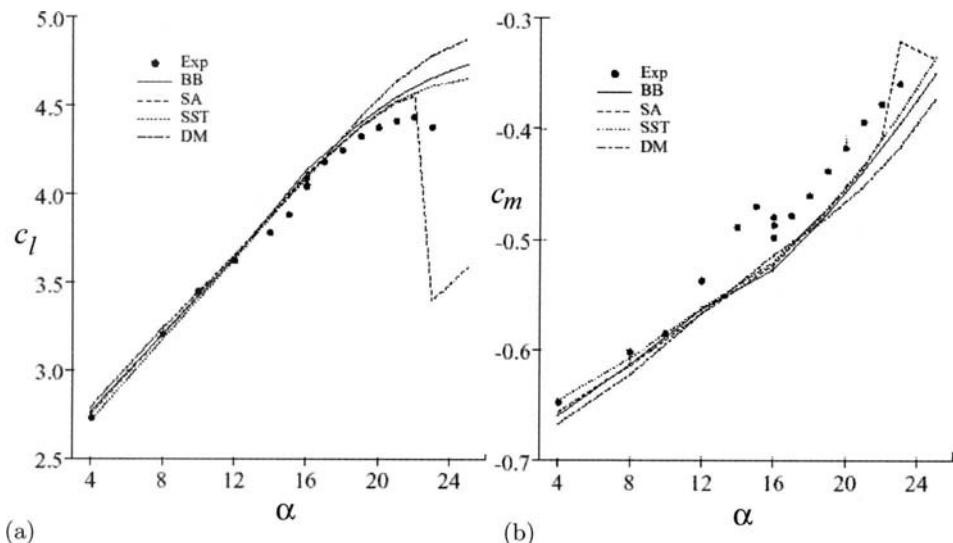


Fig. 11.11. Results for geometry B at $Re = 9 \times 10^6$, variation of (a) c_l and (b) c_m with α .

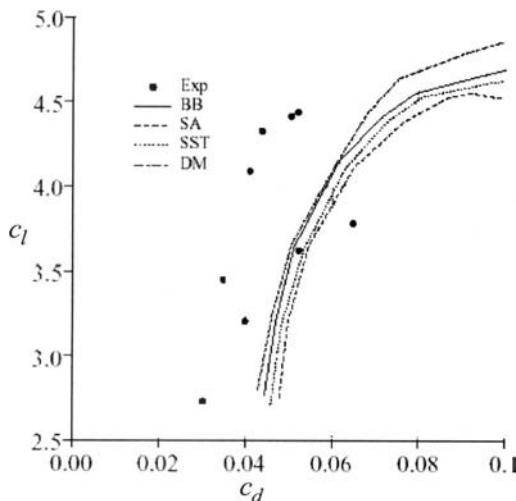


Fig. 11.12. Drag polar for geometry B at $Re = 9 \times 10^6$.

The computing time utilized by INS2D varies somewhat from one turbulence model to the other. The BB and SA models have a single differential equation whereas the SST and DM have two differential equations. The SST model converges best when 10 line-relaxation sweeps are used during one iteration, the others work best when only 2 sweeps are used. Thus the SST model takes more time per iteration. Figure 11.13 shows the maximum residual of the mean flow equations versus iteration number for geometry A at 21° . There are numerous oscillations occurring in the residual. These are due to the behavior at the zonal

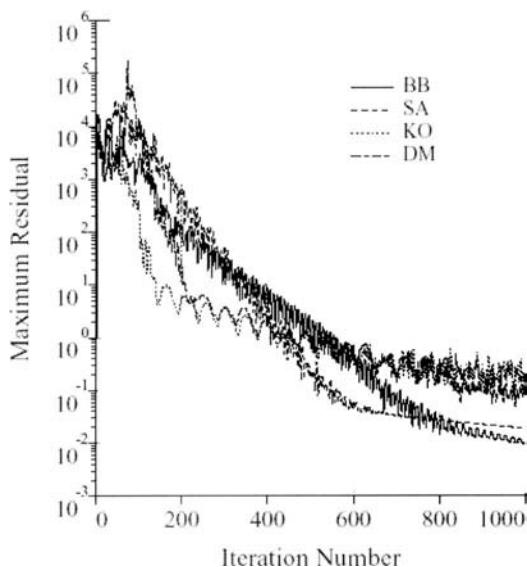


Fig. 11.13. Maximum residual versus iteration number.

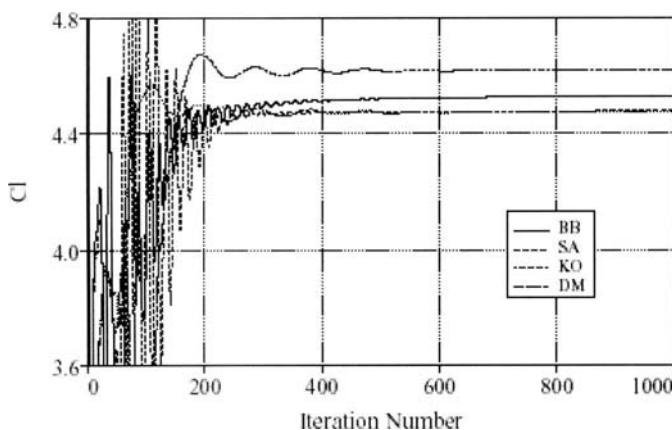


Fig. 11.14. c_l versus iteration number.

interfaces as information travels between zones, as well as the fact that the flow tends to be unsteady at the thick trailing edge of the flap. This unsteadiness causes the high-frequency oscillations in the lift as it converges to a steady-state as can be seen in Fig. 11.14. The solutions are considered converged when the maximum residual has dropped over 5 orders of magnitude and the lift coefficient has converged to 4 significant digits. The BB and SA runs have met this criteria in 600 iterations, which required 16 minutes of CPU time on a Cray C90 (or 27 minutes on a Cray YMP). The SST and DM take longer to damp out the oscillations in the lift coefficient. They each take 900 iterations to reach

convergence in this case. This required 32 minutes on a Cray C90 for the SST model, and the DM calculation used 27 minutes of Cray C90 CPU time.

References

- [1] Harlow, F. H. and Welsh, J. E.: Physics of Fluids, Vol. 8, pp. 2182–2189, 1965.
- [2] Fletcher, C. A. J.: *Computational Techniques for Fluid Dynamics*, Volume 2, Springer-Verlag, 1988.
- [3] Patankar, S. V. and Spalding, D. B.: International Journal of Heat and Mass Tranfer, Vol. 15, pp. 1787–1806, 1972.
- [4] Ferziger, J. H. and Peric, M.: *Computational Methods for Fluid Dynamics*, Springer-Verlag, 1996.
- [5] Chorin, A. J.: A numerical method for solving incompressible viscous flow problems. J. Comput. Phys., Vol. 2, p. 12, 1967.
- [6] Gustafson, B. and Sundstrom, A.: SIAM J. Applied Math., Vol. 35, pp. 343–357, 1978.
- [7] Chang, J. L. and Kwak, D.: On the Method of Pseudo Compressibility for numerically solving incompressible flows, AIAA 22nd Aerospace Sciences meeting, AIAA Paper 84-0252, 1984.
- [8] Roger, S. E. and Kwak, D.: An upwind differencing scheme for the time accurate incompressible Navier–Stokes equations. AIAA J., Vol. 28, No. 2, pp. 253–262, 1990.
- [9] Roger, S. E. and Kwak, D.: An upwind differencing scheme for the incompressible Navier–Stokes equations. AIAA J., Vol. 8, pp. 43–64, 1991.
- [10] Roger, S. E., Kwak, D., Kiris, C.: Steady and unsteady solutions of the incompressible Navier–Stokes equations. AIAA J., Vol. 4, No. 4, pp. 603–610, 1991.
- [11] Hirsh, C.: Numerical Computation of Internal and External Flows, Volume 2, John Wiley and Sons, N.Y., 1988.
- [12] Allen, Timothy J.: Stability-Transition Properties of Sudden Expansion Flows in Ducts, MS Thesis, California State University, Long Beach, 1990.
- [13] Kwon, O. K., Pletcher, R. H. and Lewis, J. P.: Prediction of Sudden Expansion Flows Using the Boundary-Layer Equations, J. of Fluids Engineering, Vol. 106, pp. 285–290, 1984.
- [14] Hung, T. K.: Laminar Flow in Conduit Expansion, Ph.D. Dissertation, University of Iowa, Iowa City, 1966.
- [15] Agarwal, R. K.: A Third-Order-Accurate Upwind Scheme for Navier–Stokes Solutions at High Reynolds Numbers, AIAA Paper 81-0112, 1981.
- [16] Cebeci, T. and Cousteix, J.: Modeling and Computation of Boundary-Layer Flows, 2nd edn. Horizons Pub., Long Beach, Calif. and Springer-Verlag, Heidelberg, 2005.
- [17] Rogers, S. E.: Progress In High-Lift Aerodynamic Calculations, AIAA Paper 93-0194, Jan. 1993.
- [18] Rogers, S. E., Wiltberger, N. L. and Kwak, D.: Efficient Simulation of Incompressible Viscous Flow Over Single and Multi-Element Airfoils, AIAA Paper 92-0405, Jan. 1992. Also appears in J. of Aircraft, Vol. 30, No. 5, pp. 736–743, Sept. 1993.
- [19] Rogers, S. E., Mentor, F., Durbin, P. A. and Mansour, N. N.: A comparison of turbulence models in computing multielement airfoil flows, AIAA Paper No. 94-0291, 1994.
- [20] Baldwin, B. and Barth, T.: A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows, NASA TM 102847, Aug. 1990.
- [21] Durbin, P. A., Mansour, N. N., and Yang, Z.: Eddy viscosity Transport Model for Turbulent Flow, to appear in Physics of Fluids.

- [22] Menter, F. R.: Zonal Two Equation $k-\omega$ Turbulence Models for Aerodynamic Flows, AIAA Paper 93-2906, July 1993.
- [23] Chin, V., Peters, D., Spaid, F., and McGhee, R.: Flowfield Measurements About a Multi-Element Airfoil at High Reynolds Numbers, AIAA Paper 93-3137, July, 1993.
- [24] AIAA CFD Drag Prediction Workshop, www.aiaa.org/tc/apa/dragpredworkshop/dpw.html

Problems

- 11-1.** Show that the eigenvalues of Eq. (11.3.1) are those given by Eq. (11.3.5)
- 11-2.** Show that the Jacobian matrices of the convective flux E and F are given by Eq. (11.4.8).
- 11-3.** Show that the eigenvalues and eigenvectors of matrices Eq. (11.4.8) are given by Eq. (11.4.9).
- 11-4.** Use INS method and the computer program given in Appendix B to solve the driven cavity problem for $R_L = 100$. In this problem, the incompressible viscous flow is driven by constant velocity u shown in Fig. P11.1. See the example problem for other boundary conditions.

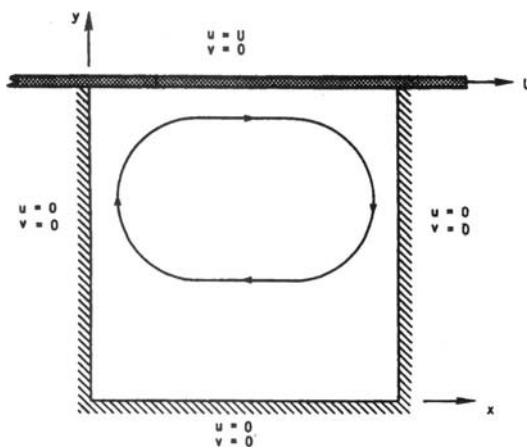


Fig. P11.1. Driven cavity.

12 Compressible Navier–Stokes Equations

12.1 Introduction

The compressible Navier–Stokes equations represent the highest level of mathematical modeling for describing aerodynamic flows commonly used in engineering applications. These include: the aerodynamic flow around aircraft, the internal flow occurring in turbomachinery; the flow developing in ducts in the presence of heat transfer; or a combination of these such as the external aerodynamic flow impinging on a wing leading edge surface heated by internal hot air emanating from the engine compressor stages to avoid ice accumulation.

The methods developed by the research community for the solution of these equations for specific applications have been building on the advances made on solving the simplified forms of the equations, since the compressible Navier–Stokes equations do not introduce fundamentally different flow characteristics. Analysis of these equations shows that their types are a mix of all types studied in the previous chapters. Since the viscous terms transform the inviscid first-order partial differential equations into second-order, the momentum and energy equations are parabolic in time and space, but elliptic in space when steady state conditions are reached. The continuity equation is, however, hyperbolic in space and time. Therefore, since the methods for addressing each of these issues have been discussed in the previous chapters, emphasis is placed on the specific difficulties associated with the engineering applications of the Reynolds-Averaged compressible Navier–Stokes equations in this chapter (Section 12.2) and their methods of solution. In particular, the MacCormack scheme adapted for viscous flows is discussed in Section 12.3, the Beam–Warming method in Section 12.4 and the finite-volume method in Section 12.5. Application on the model problem of Section 11.5, that is the sudden expansion laminar duct flow, is discussed in Section 12.6 along with the detailed description of the computer programs given in Appendix B.

12.2 Compressible Navier–Stokes Equations

The compressible Navier–Stokes equations, which are in fact five scalar conservation equations, Eqs. (2.2.17)–(2.2.22) contain five unknowns: the density, the three components of velocity and the energy. Their solution can be obtained by Direct Numerical Simulation (DNS) which requires that all the physical scales imbedded within the flow under study are captured. Since the smallest scale is the Kolmogorov scale, and that it scales with the Reynolds number of the flow, the grid requirements to capture these scales become so large that the time to obtain converged flow solutions becomes impractical with the current available computer power. One way to reduce the computing requirements is to model the smallest scales and to compute only the larger ones, giving rise to Large Eddy Simulation (LES). Yet the required computing power remains large and most computational techniques use the Reynolds-Averaged form of the compressible Navier–Stokes equations (RANS) with turbulence models (Chapter 3) to represent the Reynolds stress and turbulent heat flux terms with their solutions depending on the accuracy of the models.

12.2.1 Practical Difficulties

Apart from the accuracy of turbulence models, the solution of the RANS equations presents several difficulties. One is that the turbulence model introduces additional equations whose solutions can be rather involved if turbulence models based on transport equations are used rather than the zero-equation models.

Another problem arises for high-Reynolds number flows. Since compressible flows are often associated with high-speed flows, the boundary-layer region is confined to a very small distance away from the solid surface. In order to accurately compute the strong gradients present inside the boundary layer, several grid points need to be placed inside it. Typically, around 20–30 grid points are needed, and the first one must be within $y^+ < 1$ to capture the laminar sublayer region (y^+ being the normal wall distance in non-dimensional boundary-layer coordinates defined in Section 3.5). If a formula for local skin-friction coefficient c_f for a flat plate [1] is used for a Reynolds number of 10^6 , this requirement translates into the first point being located at a distance of $10^{-6}c$ away from the wall, c being the associated length scale of the surface (Fig. 12.1). Not only is this distance smaller than typical CAD system tolerances, which means that CAD surfaces must be “repaired” to mathematically close them, but it leads to high-aspect ratio cells (up to 10^6) within the boundary-layer region. This introduces additional stiffness in the numerical algorithm, since we are now in presence of waves travelling at dissimilar speeds in the streamwise and normal flow directions. In addition, since most codes have formal second-order accurate discretization on smooth grids, grid stretching and skewness can produce high truncation errors leading to excessive dissipation. This puts severe restrictions

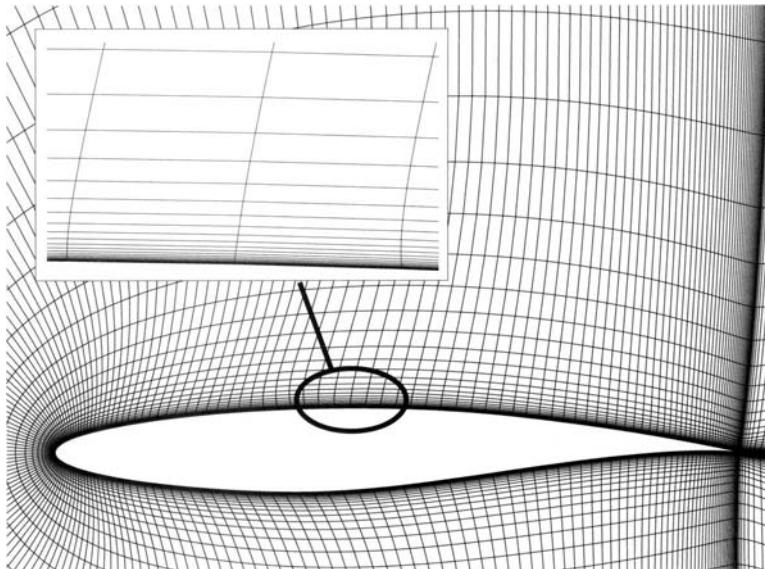


Fig. 12.1. High-Reynolds number grid inside the viscous region.

on the grid quality and typically leads to an increase in discretized mesh points which makes meshes of the order of $1 \sim 10$ million points common for standard aircraft configurations.

12.2.2 Boundary Conditions

The compressible Navier-Stokes equations also require boundary conditions. The system of equations, which contains five variables in three-dimensions (density, velocity field and temperature) needs five variables to be specified along the inflow and outflow boundaries (Table 12.1). In practice and as in incompressible flows, the inviscid compressible flow boundary conditions can be used when the far-field boundaries are placed far enough from the immerse body. Table 12.1 shows the number of physical and numerical boundary conditions to be used for each case, following the analysis of [2]. Three-dimensional Euler

Table 12.1. Physical and numerical boundary conditions for the 3D compressible flow equations.

		Navier–Stokes		Euler	
		$M < 1$	$M > 1$	$M > 1$	$M < 1$
Physical conditions	Inflow	5	5	Inflow	5
	Outflow	4	4	Outflow	0
Numerical conditions	Inflow	0	0	Inflow	0
	Outflow	1	1	Outflow	5

type boundary conditions are often obtained by simple application of the one-dimensional characteristic theory described in subsection 10.11.2 parallel to the flow direction.

When selecting the numerical conditions, it is important to select one which does not introduce any added dissipation, since one objective of a Navier–Stokes solution is to predict the wall shear-forces. A study on the choice of appropriate boundary conditions can be found in [2].

12.3 MacCormack Method

In Section 5.3 the MacCormack method was discussed for a one-dimensional problem and its extension to two-dimensional compressible viscous flows is discussed here. As in Section 11.4, the non-dimensional Navier–Stokes equations are written in Cartesian coordinates,

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = \frac{1}{\text{Re}} \left(\frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \right) \quad (2.2.30)$$

where Q , E , F , E_v and F_v are given by Eqs. (2.2.32) and (2.2.33).

The predictor values at (t^{n+1}, x_i, y_i) are defined by $\bar{Q}_{i,j}^{n+1}$ ($\equiv \bar{Q}_{i,j}$), and the convective flux terms E and F are represented with forward differences followed by a corrector step with backward differences for the same convective flux terms. The viscous flux terms E_v and F_v are represented by central differences, as for the incompressible case (subsection 11.4.3). The predictor step then becomes

$$\begin{aligned} \bar{Q}_{i,j} = Q_{i,j}^n - \Delta t & \left(\frac{E_{i+1,j}^n - E_{i,j}^n}{\Delta x} + \frac{F_{i,j+1}^n - F_{i,j}^n}{\Delta y} \right) \\ & + \frac{\Delta t}{\text{Re}} \left(\frac{(E_v)_{i+1/2,j}^n - (E_v)_{i-1/2,j}^n}{\Delta x} + \frac{(F_v)_{i,j+1/2}^n - (F_v)_{i,j-1/2}^n}{\Delta y} \right) \end{aligned} \quad (12.3.1)$$

and the corrector step

$$\begin{aligned} \bar{\bar{Q}}_{i,j} = Q_{i,j}^n - \Delta t & \left(\frac{\bar{E}_{i,j} - \bar{E}_{i-1,j}}{\Delta x} + \frac{\bar{F}_{i,j} - \bar{F}_{i,j-1}}{\Delta y} \right) \\ & + \frac{\Delta t}{\text{Re}} \left(\frac{(\bar{E}_v)_{i+1/2,j}^n - (\bar{E}_v)_{i-1/2,j}^n}{\Delta x} + \frac{(\bar{F}_v)_{i,j+1/2}^n - (\bar{F}_v)_{i,j-1/2}^n}{\Delta y} \right) \end{aligned} \quad (12.3.2)$$

Updating gives

$$Q_{i,j}^{n+1} = -\frac{1}{2}(\bar{Q}_{i,j}, \bar{\bar{Q}}_{i,j}) \quad (12.3.3)$$

which remains unchanged from the 1D counterpart.

12.4 Beam–Warming Method

The Beam–Warming (B-W) method for the solution of the one-dimensional time-dependent Euler equations, Eq. (5.1.2), was discussed in Section 5.4 and its extension to the two-dimensional compressible viscous flows is discussed now, following [3].

An obvious extension of Eq. (5.4.4) to a two-dimensional flow is,

$$\begin{aligned}\Delta Q^n = & -\frac{\theta \Delta t}{1+\xi} \left[\frac{\partial}{\partial x} (\Delta E^n) + \frac{\partial}{\partial y} (\Delta F^n) \right] + \frac{\theta \Delta t}{1+\xi} \text{Re} \left[\frac{\partial}{\partial x} (\Delta E_v^n) + \frac{\partial}{\partial y} (\Delta F_v^n) \right] \\ & + \frac{\Delta t}{1+\xi} \left[\frac{\partial}{\partial x} \left(-E^n + \frac{E_v^n}{\text{Re}} \right) + \frac{\partial}{\partial y} \left(-F^n + \frac{F_v^n}{\text{Re}} \right) \right] \\ & + \frac{\xi}{1+\xi} \Delta Q^{n-1}\end{aligned}\quad (12.4.1)$$

The first term on the right-hand-side of Eq. (12.4.1) ΔE^n is given by Eq. (5.4.7b) and, similarly, the second term can be written as

$$\Delta F^n = B \Delta Q^n + O(\Delta t^2) \quad (12.4.2)$$

where

$$B^n = \left(\frac{\partial F}{\partial Q} \right)^n \quad (12.4.3)$$

Since E_v^n and F_v^n are functions of Q , Q_x and Q_y , they can be written as

$$E_v = V_1(Q, Q_x) + V_2(Q, Q_y) \quad (12.4.4a)$$

$$F_v = W_1(Q, Q_x) + W_2(Q, Q_y) \quad (12.4.4b)$$

where

$$V_1 = \begin{bmatrix} 0 \\ \frac{4}{3}\mu u_x \\ \mu v_x \\ \mu v v_x + \frac{4}{3}\mu u u_x + \frac{\mu}{(\gamma-1)M_\infty^2 \text{Pr}} T_x \end{bmatrix} \quad V_2 = \begin{bmatrix} 0 \\ -\frac{2}{3}\mu v_y \\ \mu u_y \\ \mu v u_y - \frac{2}{3}\mu u v_y \end{bmatrix} \quad (12.4.5a)$$

$$W_1 = \begin{bmatrix} 0 \\ \mu v_x \\ -\frac{2}{3}\mu u_x \\ \mu u v_x - \frac{2}{3}\mu v u_x \end{bmatrix} \quad W_2 = \begin{bmatrix} 0 \\ \mu u_y \\ \frac{4}{3}\mu v_y \\ \mu u u_y + \frac{4}{3}\mu v v_y + \frac{\mu}{(\gamma-1)M_\infty^2 \text{Pr}} T_y \end{bmatrix} \quad (12.4.5b)$$

Then

$$\frac{\partial}{\partial x} \Delta E_v^n = \frac{\partial}{\partial x} \Delta V_1^n + \frac{\partial}{\partial x} \Delta V_2^n \quad (12.4.6a)$$

$$\frac{\partial}{\partial y} \Delta F_v^n = \frac{\partial}{\partial y} \Delta W_1^n + \frac{\partial}{\partial y} \Delta W_2^n \quad (12.4.6b)$$

As with Eq. (5.4.7a), we can write

$$V_1^{n+1} = V_1^n + \Delta t \frac{\partial}{\partial t} V_1^n + O(\Delta t^2)$$

and, with the chain rule and noting that V_1^n is a function of Q and Q_x ,

$$\begin{aligned} V_1^{n+1} &= V_1^n + \Delta t \left(\frac{\partial V_1^n}{\partial Q} \frac{\partial Q}{\partial t} + \frac{\partial V_1^n}{\partial Q_x} \frac{\partial Q_x}{\partial t} \right) \\ &= V_1^n + \Delta t \left(P \frac{\partial Q}{\partial t} + R \frac{\partial Q_x}{\partial t} \right)^n \end{aligned} \quad (12.4.7)$$

where

$$P^n = \left(\frac{\partial V_1}{\partial Q} \right)^n \quad R^n = \left(\frac{\partial V_1}{\partial Q_x} \right)^n \quad (12.4.8)$$

Analogous to Eq. (5.4.7b), we can write

$$\Delta V_1^n = P^n \Delta Q^n + R^n \Delta Q_x^n + O(\Delta t^2)$$

or

$$\Delta V_1^n = (P^n - R_x^n) \Delta Q^n + (R \Delta Q)_x^n \quad (12.4.9)$$

Similarly, noting that ΔW_2^n is a function of Q and Q_y , we can write

$$\Delta W_2^n = (M^n - N_y^n) \Delta Q^n + (N \Delta Q)_y^n \quad (12.4.10)$$

where

$$M^n = \left(\frac{\partial W_2}{\partial Q} \right)^n \quad N^n = \left(\frac{\partial W_2}{\partial Q_y} \right)^n \quad (12.4.11)$$

Since ΔV_2^n and ΔW_1^n are functions of Q , Q_x and Q_y , the terms

$$\frac{\partial}{\partial y} \Delta V_2^n, \quad \frac{\partial}{\partial x} \Delta W_1^n$$

in Eq. (12.4.6) represent mixed derivatives and are approximated by

$$\frac{\partial}{\partial x} \Delta V_2^n = \frac{\partial}{\partial x} \Delta V_2^{n-1} + O(\Delta t^2) \quad (12.4.12a)$$

$$\frac{\partial}{\partial y} \Delta W_1^n = \frac{\partial}{\partial y} \Delta W_1^{n-1} + O(\Delta t^2) \quad (12.4.12b)$$

Equations (5.4.7a), (12.4.2), (12.4.9), (12.4.10), and (12.4.12) are now substituted into Eq. (12.4.1) to yield

$$\begin{aligned} &\left\{ I + \frac{\theta \Delta t}{1 + \xi} \left[\frac{\partial}{\partial x} \left(A - \frac{P - R_x}{Re} - \frac{1}{Re} \frac{\partial}{\partial x} R \right) \right. \right. \\ &\quad \left. \left. + \frac{\partial}{\partial y} \left(B - \frac{M - N_y}{Re} - \frac{1}{Re} \frac{\partial}{\partial y} N \right) \right] \right\} \Delta Q^n \\ &= \frac{\theta \Delta t}{1 + \xi} \left[\frac{\partial}{\partial x} \left(\frac{\Delta V_2^{n-1}}{Re} \right) + \frac{\partial}{\partial y} \left(\frac{\Delta W_1^{n-1}}{Re} \right) \right] \\ &\quad + \frac{\Delta t}{1 + \xi} \left[\frac{\partial}{\partial x} \left(-E^n + \frac{V_1^n + V_2^n}{Re} \right) + \frac{\partial}{\partial y} \left(-F^n + \frac{W_1^n + W_2^n}{Re} \right) \right] \\ &\quad + \frac{\xi}{1 + \xi} \Delta Q^{n-1} \end{aligned} \quad (12.4.13)$$

where I is the unity matrix given by

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12.4.14)$$

and the Jacobian matrices $A, P, R, R_x, B, M, N, N_y$ are given in Appendix 12. We note that in Eq. (12.4.13)

$$\frac{\partial}{\partial x} \left(A - \frac{P-R_x}{\text{Re}} - \frac{1}{\text{Re}} \frac{\partial}{\partial x} R \right) \Delta Q^n, \quad \frac{\partial}{\partial y} \left(B - \frac{M-N_y}{\text{Re}} - \frac{1}{\text{Re}} \frac{\partial}{\partial y} N \right) \Delta Q^n \quad (12.4.15a)$$

are equivalent to

$$\frac{\partial}{\partial x} \left[\left(A - \frac{P-R_x}{\text{Re}} - \frac{1}{\text{Re}} \frac{\partial}{\partial x} R \right) \Delta Q^n \right], \quad \frac{\partial}{\partial y} \left[\left(B - \frac{M-N_y}{\text{Re}} - \frac{1}{\text{Re}} \frac{\partial}{\partial y} N \right) \Delta Q^n \right] \quad (12.4.15b)$$

The left-hand-side of Eq. (12.4.13) can be factored and expressed in the form

$$\begin{aligned} & \left\{ [I] + \frac{\theta \Delta t}{1+\xi} \left[\frac{\partial}{\partial x} \left([A] - \frac{[P]-[R_x]}{\text{Re}} \right)^n - \frac{1}{\text{Re}} \frac{\partial^2}{\partial x^2} [R]^n \right] \right\} \\ & \left\{ [I] + \frac{\theta \Delta t}{1+\xi} \left[\frac{\partial}{\partial y} \left([B] - \frac{[M]-[N_y]}{\text{Re}} \right)^n - \frac{1}{\text{Re}} \frac{\partial^2}{\partial y^2} [N]^n \right] \right\} \Delta Q^n \\ &= \left\{ \frac{\Delta t}{1+\xi} \left[\frac{\partial}{\partial x} \left(-E + \frac{V_1+V_2}{\text{Re}} \right)^n + \frac{\partial}{\partial y} \left(-F + \frac{W_1+W_2}{\text{Re}} \right)^n \right] \right. \\ & \quad \left. + \frac{\theta \Delta t}{1+\xi} \frac{1}{\text{Re}} \left[\frac{\partial}{\partial x} (\Delta V_2^{n-1}) + \frac{\partial}{\partial y} (\Delta W_1^{n-1}) \right] + \frac{\xi}{1+\xi} (\Delta Q^{n-1}) \right\} \end{aligned} \quad (12.4.16)$$

The solution of Eq. (12.4.16) is obtained in two steps

Step 1:

$$\begin{aligned} & \left\{ [I] + \frac{\theta \Delta t}{1+\xi} \left[\frac{\partial}{\partial x} \left([A] - \frac{[P]-[R_x]}{\text{Re}} \right)^n - \frac{1}{\text{Re}} \frac{\partial^2}{\partial x^2} [R]^n \right] \right\} \Delta Q^{n-1/2} \\ &= \frac{\Delta t}{1+\xi} \left[\frac{\partial}{\partial x} \left(-E + \frac{V_1+V_2}{\text{Re}} \right)^n + \frac{\partial}{\partial y} \left(-F + \frac{W_1+W_2}{\text{Re}} \right)^n \right] \\ & \quad + \frac{\theta \Delta t}{1+\xi} \frac{1}{\text{Re}} \left[\frac{\partial}{\partial x} (\Delta V_2^{n-1}) + \frac{\partial}{\partial y} (\Delta W_1^{n-1}) \right] + \frac{\xi}{1+\xi} (\Delta Q^{n-1}) \end{aligned} \quad (12.4.17)$$

Step 2:

$$\left\{ [I] + \frac{\theta \Delta t}{1+\xi} \left[\frac{\partial}{\partial y} \left([B] - \frac{[M]-[N_y]}{\text{Re}} \right)^n - \frac{1}{\text{Re}} \frac{\partial^2}{\partial y^2} [N]^n \right] \right\} \Delta Q^n = \Delta Q^{n-1/2} \quad (12.4.18)$$

As in Section 5.4, where central differences were used for the spatial derivatives in Eq. (5.4.12), central differences are again used in Eqs. (12.4.17) and (12.4.18), and the resulting system is written in block tridiagonal form and solved with the block elimination method described in Eqs. (4.4.32) through (4.4.34).

As discussed in Section 5.7, in some implicit methods for hyperbolic equations, damping terms are necessary to suppress high frequency oscillations in the solution. This can be accomplished by adding constant coefficient implicit and explicit dissipation terms to the solution algorithm and, in the Beam–Warming method, this is accomplished by adding an explicit fourth-order dissipation term to the right-hand side of Eq. (12.4.17). If the steady-state is of interest, implicit second order dissipation terms are added to the left-hand sides of Eqs. (12.4.17) and (12.4.18) and the resulting equations become

Step 1:

$$\begin{aligned} & \left\{ [I] + \frac{\theta \Delta t}{1 + \xi} \left[\frac{\partial}{\partial x} \left([A] - \frac{[P] - [R_x]}{\text{Re}} \right)^n - \frac{1}{\text{Re}} \frac{\partial^2}{\partial x^2} [R]^n - \varepsilon_i \delta_x^2 [I] \right] \right\} \Delta Q^{n-1/2} \\ &= \frac{\Delta t}{1 + \xi} \left[\frac{\partial}{\partial x} \left(-E + \frac{V_1 + V_2}{\text{Re}} \right)^n + \frac{\partial}{\partial y} \left(-F + \frac{W_1 + W_2}{\text{Re}} \right)^n \right] \\ &+ \frac{\theta \Delta t}{1 + \xi} \frac{1}{\text{Re}} \left[\frac{\partial}{\partial x} (\Delta V_2^{n-1}) + \frac{\partial}{\partial y} (\Delta W_1^{n-1}) \right] \\ &+ \frac{\xi}{1 + \xi} (\Delta Q^{n-1}) - \varepsilon_e (\delta_x^4 + \delta_y^4) Q^n \end{aligned} \quad (12.4.19)$$

Step 2:

$$\begin{aligned} & \left\{ [I] + \frac{\theta \Delta t}{1 + \xi} \left[\frac{\partial}{\partial y} \left([B] - \frac{[M] - [N_y]}{\text{Re}} \right)^n - \frac{1}{\text{Re}} \frac{\partial^2}{\partial y^2} [N]^n - \varepsilon_i \delta_y^2 [I] \right] \right\} \Delta Q^n \\ &= \Delta Q^{n-1/2} \end{aligned} \quad (12.4.20)$$

Here δ^2 and δ^4 denote central difference operators defined by

$$\delta_x^2 u = u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \quad (12.4.21)$$

and

$$\delta_x^4 u = u_{i+2,j} - 4u_{i+1,j} + 6u_{i,j} - 4u_{i-1,j} + u_{i-2,j} \quad (12.4.22)$$

On mesh points adjacent to the boundaries, Eq. (12.4.22) is approximated by second-order four point finite differences rather than five point finite differences to avoid requesting information from outside the computational domain. For example, near the bottom boundary, Eq. (12.4.22) becomes

$$\delta_y^4 u \approx u_{i,4} - 4u_{i,3} + 5u_{i,2} - 2u_{i,1} \quad (12.4.23)$$

The coefficients of the damping terms ε_i and ε_e are usually selected according to

$$0 \leq \varepsilon_e \leq \frac{1 + 2\xi}{8(1 + \xi)} \quad (12.4.24a)$$

and

$$\varepsilon_i = 2\varepsilon_e \quad (12.4.24b)$$

to ensure stability and optimal convergence rate [2].

12.5 Finite Volume Method

In this section, we discuss the application of the finite volume of Section 5.6 to two dimensional compressible flows given by Eq. (2.2.30)

$$\frac{\partial Q}{\partial t} = R(Q) = -\frac{\partial E}{\partial x} - \frac{\partial F}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \right) \quad (12.5.1)$$

with Q , E , F , E_v and F_v given by Eqs. (2.2.32) and (2.2.33). The method has been extensively developed by Jameson [4].

We first integrate (12.5.1) over rectangular cell $\Omega_{i,j}$ (see Fig. 12.2).

The left-hand-side becomes

$$\iint_{\Omega_{i,j}} \frac{\partial Q}{\partial t} dx dy = \frac{\partial Q_{i,j}}{\partial t} \Delta x \Delta y \quad (12.5.2a)$$

while the right-hand-side is expanded as

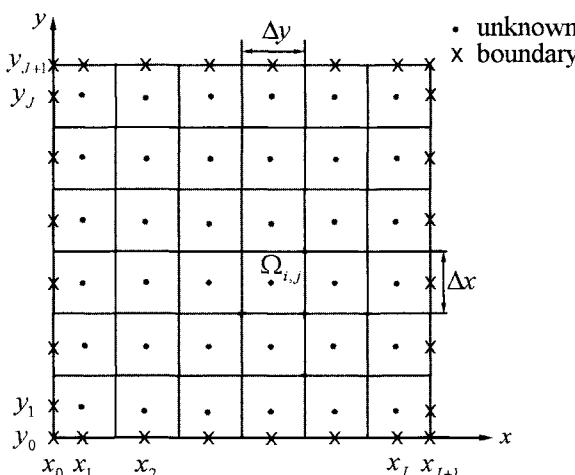


Fig. 12.2. Subdomains $\Omega_{i,j}$, interior (unknown) and boundary points.

$$\begin{aligned}
& \iint_{\Omega_{i,j}} \left[-\frac{\partial E}{\partial x} - \frac{\partial F}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \right) \right] dx dy \\
&= \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \left[-\frac{\partial E}{\partial x} - \frac{\partial F}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \right) \right] dx dy \\
&= - \int_{y_{j-1/2}}^{y_{j+1/2}} (E_{i+1/2} - E_{i-1/2}) dy - \int_{x_{i-1/2}}^{x_{i+1/2}} (F_{j+1/2} - F_{j-1/2}) dx \\
&\quad + \frac{1}{\text{Re}} \left(\int_{y_{j-1/2}}^{y_{j+1/2}} (E_{vi+1/2} - E_{vi-1/2}) dy + \int_{x_{i-1/2}}^{x_{i+1/2}} (F_{vj+1/2} - F_{vj-1/2}) dx \right) \\
&= -(E_{i+1/2,j} - E_{i-1/2,j}) \Delta y - (F_{i,j+1/2} - F_{i,j-1/2}) \Delta x \\
&\quad + \frac{1}{\text{Re}} \left[(E_{vi+1/2,j} - E_{vi-1/2,j}) \Delta y + (F_{vi,j+1/2} - F_{vi,j-1/2}) \Delta x \right]
\end{aligned} \tag{12.5.2b}$$

We then obtain

$$\begin{aligned}
\frac{\partial Q_{i,j}}{\partial t} \Delta x \Delta y &= -(E_{i+1/2,j} - E_{i-1/2,j}) \Delta y - (F_{i,j+1/2} - F_{i,j-1/2}) \Delta x \\
&\quad + \frac{1}{\text{Re}} \left[(E_{vi+1/2,j} - E_{vi-1/2,j}) \Delta y + (F_{vi,j+1/2} - F_{vi,j-1/2}) \Delta x \right]
\end{aligned} \tag{12.5.3}$$

Dividing both sides of Eq. (12.5.3) by $\Delta x \Delta y$ results in a system of ordinary differential equations, applicable to the interior region $\Omega_{i,j}$ including the region close to the boundaries.

$$\begin{aligned}
\frac{\partial Q_{i,j}}{\partial t} &= -\frac{E_{i+1/2,j} - E_{i-1/2,j}}{\Delta x} - \frac{F_{i,j+1/2} - F_{i,j-1/2}}{\Delta y} \\
&\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{i+1/2,j} - (E_v)_{i-1/2,j}}{\Delta x} + \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y} \right]
\end{aligned} \tag{12.5.4}$$

We first discuss the solution for the interior region $\Omega_{i,j}$ and approximate Eq. (12.5.4) by

$$\begin{aligned}
\frac{\partial Q_{i,j}}{\partial t} &= -\frac{E_{i+1,j} - E_{i-1,j}}{2 \Delta x} - \frac{F_{i,j+1} - F_{i,j-1}}{2 \Delta y} \\
&\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{i+1/2,j} - (E_v)_{i-1/2,j}}{\Delta x} + \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y} \right]
\end{aligned} \tag{12.5.5}$$

The elements in the dissipation terms can be approximated by the following central differences.

$$(u_x)_{i+1/2,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \quad (u_x)_{i-1/2,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

$$\begin{aligned}
(u_y)_{i+1/2,j} &= \frac{u_{i+1/2,j+1} - u_{i+1/2,j-1}}{2\Delta y} = \frac{u_{i+1,j+1} + u_{i,j+1} - u_{i+1,j-1} - u_{i,j-1}}{4\Delta y} \\
(u_y)_{i-1/2,j} &= \frac{u_{i-1/2,j+1} - u_{i-1/2,j-1}}{2\Delta y} = \frac{u_{i,j+1} + u_{i-1,j+1} - u_{i,j-1} - u_{i-1,j-1}}{4\Delta y} \\
(u_y)_{i,j+1/2} &= \frac{u_{i,j+1} - u_{i,j}}{\Delta y} \quad (u_y)_{i,j-1/2} = \frac{u_{i,j} - u_{i,j-1}}{\Delta x} \\
(u_x)_{i,j+1/2} &= \frac{u_{i+1,j+1/2} - u_{i-1,j+1/2}}{2\Delta x} = \frac{u_{i+1,j+1} + u_{i+1,j} - u_{i-1,j+1} - u_{i-1,j}}{4\Delta x} \\
(u_x)_{i,j-1/2} &= \frac{u_{i+1,j-1/2} - u_{i-1,j-1/2}}{2\Delta x} = \frac{u_{i+1,j} + u_{i+1,j-1} - u_{i-1,j} - u_{i-1,j-1}}{4\Delta x}
\end{aligned}$$

There are several methods to solve Eq. (12.5.4), and here we use a fourth-order Runge–Kutta scheme. For a time dependent problem

$$\frac{dQ}{dt} = R(Q) \quad (12.5.6)$$

the *fourth-order Runge–Kutta scheme* is given by the following four steps

$$\begin{aligned}
Q^{(1)} &= Q^n \\
Q^{(2)} &= Q^n + \frac{1}{2}\Delta t R^{(1)} = Q^n + \frac{1}{2}\Delta t R(Q^{(1)}) \\
Q^{(3)} &= Q^n + \frac{1}{2}\Delta t R^{(2)} = Q^n + \frac{1}{2}\Delta t R(Q^{(2)}) \\
Q^{(4)} &= Q^n + \Delta t R^{(3)} = Q^n + \Delta t R(Q^{(3)}) \\
Q^{n+1} &= Q^n + \frac{\Delta t}{6}(R^{(1)} + 2R^{(2)} + 2R^{(3)} + R^{(4)}) \\
&= Q^n + \frac{\Delta t}{6} [R(Q^{(1)}) + 2R(Q^{(2)}) + 2R(Q^{(3)}) + R(Q^{(4)})]
\end{aligned} \quad (12.5.7)$$

It proves necessary to augment the finite volume scheme by the addition of artificial dissipative terms. Therefore, Eq. (12.5.5) is replaced by the equation

$$\begin{aligned}
\frac{\partial Q_{i,j}}{\partial t} &= -\frac{E_{i+1/2,j} - E_{i-1/2,j}}{\Delta x} - \frac{F_{i,j+1/2} - F_{i,j-1/2}}{\Delta y} \\
&+ \frac{1}{\text{Re}} \left[\frac{(E_v)_{i+1/2,j} - (E_v)_{i-1/2,j}}{\Delta x} + \frac{(F_v)_{i,j+1/2} - (F_v)_{i,j-1/2}}{\Delta y} \right] \quad (12.5.8) \\
&+ DQ_{i,j} \quad (12.5.9)
\end{aligned}$$

where

$$DQ_{i,j} = D_x Q_{i,j} + D_y Q_{i,j} \quad (12.5.10)$$

Here $D_x Q_{i,j}$ and $D_y Q_{i,j}$ are defined by

$$D_x Q_{i,j} = d_{i+1/2,j} - d_{i-1/2,j}, \quad D_y Q_{i,j} = d_{i,j+1/2} - d_{i,j-1/2} \quad (12.5.11)$$

The terms on the right all have a similar form: for example

$$d_{i+1/2,j} = \frac{h_v}{\Delta t} [\varepsilon_{i+1/2,j}^{(2)} (Q_{i+1,j} - Q_{i,j}) - \varepsilon_{i+1/2,j}^{(4)} (Q_{i+2,j} - 3Q_{i+1,j} + 3Q_{i,j} - Q_{i-1,j})] \quad (12.5.12)$$

where h_v is the area of the cell. For subsonic flows, only the 4th order dissipation coefficient is needed while for supersonic flows, a 2nd order coefficient is added to reduce dispersion effects around shock waves. Define

$$\nu_{i,j} = \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{|p_{i+1,j}| + 2|p_{i,j}| + |p_{i-1,j}|} \quad (12.5.13a)$$

which activates when large pressure gradient occur. Then

$$\varepsilon_{i+1/2,j}^{(2)} = \varpi^{(2)} \max(\nu_{i+1,j}, \nu_{i,j}), \quad \varepsilon_{i+1/2,j}^{(4)} = \max(0, \varpi^{(4)} - \varepsilon_{i+1/2,j}^{(2)}) \quad (12.5.13b)$$

where typical values of the constants $\varpi^{(2)}$ and $\varpi^{(4)}$ are

$$\varpi^{(2)} = \frac{1}{4}, \quad \varpi^{(4)} = \frac{1}{256} \quad (12.5.13c)$$

Next, a residual smoother is introduced not only to add additional implicit characters, similar to the MacCormack method, but also to smooth the high frequency variations of the residual. The high frequency residual smoothing concept is used as a smoother in the multi-grid method and can speed-up the calculations. Here, we discuss explicit smoothing and implicit smoothing. The explicit smoothing residual \bar{R} is defined by

$$\begin{aligned} R_{i,j} &= \bar{R}_{i,j} + \varepsilon(\delta_x^2 + \delta_y^2)\bar{R}_{i,j} \\ &= \varepsilon(\bar{R}_{i-1,j} + \bar{R}_{i+1,j} + \bar{R}_{i,j-1} + \bar{R}_{i,j+1}) + (1 - 4\varepsilon)\bar{R}_{i,j} \end{aligned} \quad (12.5.14)$$

The implicit average is defined by

$$[1 - \varepsilon(\delta_x^2 + \delta_y^2)]R_{i,j} = \bar{R}_{i,j} \quad (12.5.15)$$

which can be approximated using ADI factorization

$$(1 - \varepsilon\delta_x^2)(1 - \varepsilon\delta_y^2)R_{i,j} = \bar{R}_{i,j}. \quad (12.5.16)$$

The solution of Eq. (12.5.16) can be obtained by solving the following two equations

$$(1 - \varepsilon\delta_x^2)R_{i,j}^* = \bar{R}_{i,j} \quad (12.5.17a)$$

and

$$(1 - \varepsilon\delta_y^2)R_{i,j}^{\mp}R_{i,j}^* = \bar{R}_{i,j}. \quad (12.5.17b)$$

These equations form the following block tridiagonal systems

$$-\varepsilon R_{i-1,j}^* + (1 + 2\varepsilon)R_{i,j}^* - \varepsilon R_{i+1,j}^* = \bar{R}_{i,j} \quad (12.5.18a)$$

and

$$-\varepsilon R_{i,j-1}^+ (1 + 2\varepsilon) R_{i,j}^- \varepsilon R_{i+1,j}^- R_{i,j}^* \quad (12.5.18b)$$

which can be solved by using the block elimination method, see subsection 4.4.3.

The region close to the boundaries are treated separately from the procedure used for Eq. (12.5.5) and is discussed in Appendix 12B.

12.6 Model Problem: Sudden Expansion Laminar Duct Flow

To demonstrate the solution of the Navier–Stokes equation for a two-dimensional compressible flow with the methods discussed in Sections 12.3 to 12.5, we again use the model problem discussed in Section 11.5 for the INS method. Clearly the boundary and initial conditions for the velocity profiles remain the same. The initial and boundary conditions for temperature T and pressure p must be specified. We use the following procedure for a given Prandtl number Pr , Mach number M_∞ and ratio of specific heats γ , with the dynamic viscosity μ calculated from,

$$\mu = (T)^{3/2} \frac{1 + C_2/T_\infty}{T + C_2/T_\infty} \quad (12.6.1)$$

where

$$C_2 = 198.6 \quad T_\infty = 540^\circ \quad (12.6.2)$$

12.6.1 Initial Conditions

We assume

$$\varrho = 1.0, \quad T = 1.0 \quad (12.6.3)$$

and compute p and E_t from

$$p = \frac{T\varrho}{\gamma M_\infty^2}, \quad E_t = \frac{P}{\gamma - 1} + \frac{\varrho}{2}(u^2 + v^2) \quad (12.6.4)$$

12.6.2 Boundary Conditions

Along the wall ($y = 0$), the wall pressure p_w can be calculated from

$$\frac{\partial p}{\partial n} = 0, \quad (12.6.5)$$

and approximated by

$$p_w = p_{i,1} - (p_{i,2} - p_{i,1})/2 \quad (12.6.6)$$

with the known interior values. Similarly, the wall temperature can be calculated from the adiabatic relation

$$\frac{\partial T}{\partial n} = 0, \quad (12.6.7)$$

and approximated by

$$T_w = T_{i,1} - (T_{i,2} - T_{i,1})/2 \quad (12.6.8)$$

with the known interior values. The density ϱ_w and $(E_t)_w$ can be calculated from Eq. (12.6.4).

$$\varrho_w = \frac{\gamma p_w M_\infty^2}{T_w}, \quad (E_t)_w = \frac{p_w}{\gamma - 1} \quad (12.6.9)$$

The wall temperature T_w can also be computed from

$$c_p T_w = c_p T_1 + \frac{V_1^2}{2T_\infty} \equiv \text{const.} \quad (12.6.10)$$

where

$$c_p = \frac{\gamma R}{\gamma - 1}, \quad p = \varrho RT \quad (12.6.11a)$$

$$V^2 = u^2 + v^2, \quad M^2 = \frac{V^2}{a^2}, \quad a^2 = \gamma RT \quad (12.6.11b)$$

Since Eq. (12.6.9) is in dimensional form, we express it in dimensionless form,

$$\begin{aligned} \frac{T_w}{T_\infty} &= T_w^* = \frac{T_1}{T_\infty} + \frac{V_1^2}{2T_\infty c_p} = \frac{T_1}{T_\infty} \left(1 + \frac{V_1^2}{2T_1 c_p} \right) \\ &= \frac{T_1}{T_\infty} \left[1 + \frac{(\gamma - 1)}{2} \left(\frac{V_1}{a_1} \right)^2 \right] = T_1^* \left[1 + \frac{(\gamma - 1)}{2} M_1^2 \right] \end{aligned}$$

or

$$T_w = T_1 \left[1 + \frac{(\gamma - 1)}{2} M_1^2 \right] \quad (12.6.12)$$

Our studies for this particular problem show that both choices (Eqs. (12.6.8) and (12.6.12)) of computing T_w are the same.

On the line of symmetry, $y = H$, we set

$$\frac{\partial \varrho}{\partial y} = 0, \quad \frac{\partial(\varrho u)}{\partial y} = v = 0, \quad \frac{\partial E_t}{\partial y} = 0, \quad (12.6.13)$$

The outflow boundary conditions at $i = I$, $0 \leq j \leq J$ are again obtained by using second-order accurate extrapolation,

$$\begin{aligned} \varrho_{I,j} &= 2\varrho_{I-1,j} - \varrho_{I-2,j} \\ (\varrho u)_{I,j} &= 2(\varrho u)_{I-1,j} - (\varrho u)_{I-2,j} \\ (\varrho v)_{I,j} &= 2(\varrho v)_{I-1,j} - (\varrho v)_{I-2,j} \\ (E_t)_{I,j} &= 2(E_t)_{I-1,j} - (E_t)_{I-2,j} \end{aligned} \quad (12.6.14)$$

Table 12.2. Comparison of reattachment locations according to four methods.

Method	$x_r/(hR_h)$
Finite Volume	0.126
B-W	0.132
MacCormack	0.128
INS	0.138

Table 12.3. Effect of dissipation parameters on the reattachment locations.

ε_i	$x_r/(hR_h)$
1/16	0.132
1/8	0.138

12.6.3 Solution Procedure and Sample Calculations

The solution procedure and sample calculations are now discussed separately for the three methods discussed in Sections 12.3 to 12.5 and comparisons are made with those obtained from the INS method of Section 11.5. The computer program for each method is given in Appendix B.

Table 12.2 shows a comparison between the predictions of the four methods for the reattachment locations. Except for the INS method, all calculations are for $M_\infty = 0.15$, a value low enough for the flow to be considered incompressible.

Table 12.3 shows the effect of dissipation parameters on the reattachment locations according to the B-W method.

Figures 12.3 and 12.4 show the effect of the Reynolds number on the reattachment locations and velocity profiles according to the B-W method.

Appendix 12A

Jacobian Matrices of Convection and Diffusion Terms E , F , E_v and F_v

To derive the Jacobian matrices of the convective flux terms E and F , we first express E and F by the components of Q :

$$Q = \begin{bmatrix} \varrho \\ \varrho u \\ \varrho v \\ E_t \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

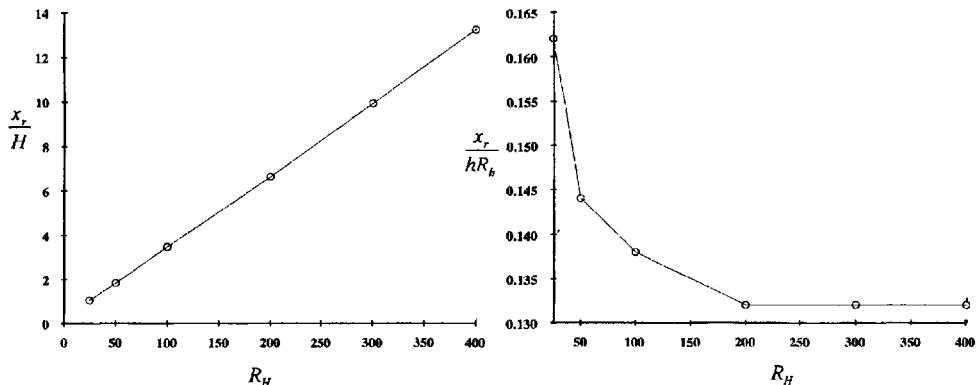


Fig. 12.3. Effects of Reynolds number on reattachment locations.

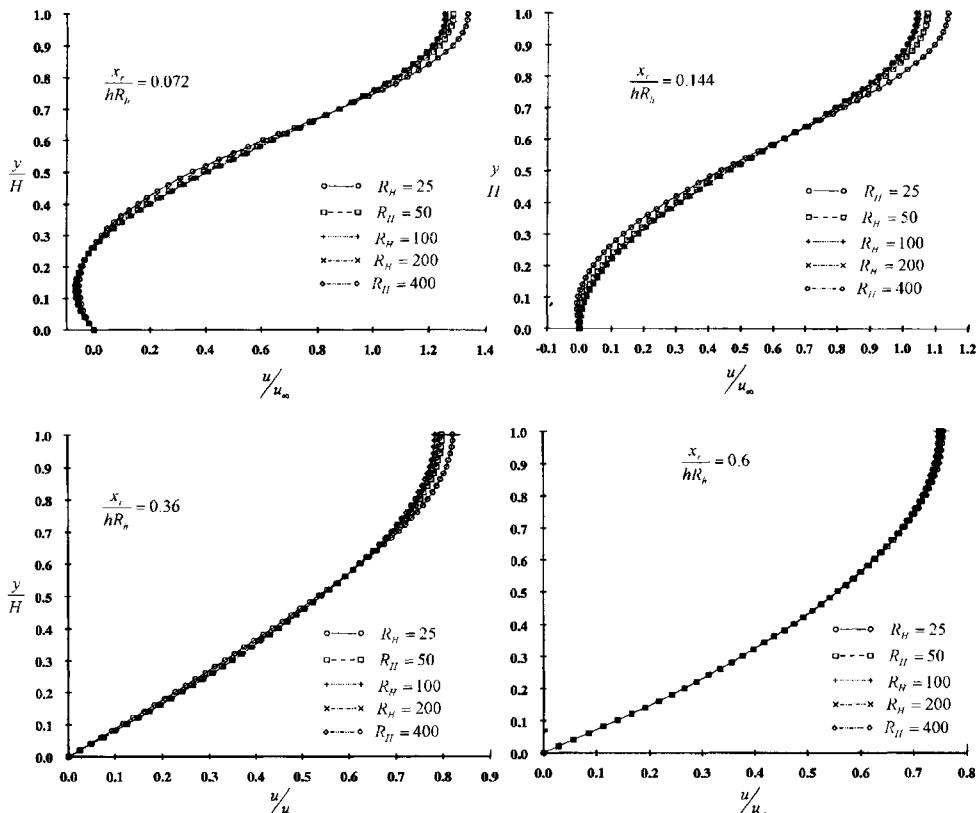


Fig. 12.4. Effects of Reynolds number on velocity profiles.

$$E = \begin{bmatrix} \varrho u \\ \varrho u^2 + p \\ \varrho uv \\ (E_t + p)u \end{bmatrix} = \begin{bmatrix} q_2 \\ q_2^2/q_1 + (\gamma - 1)[q_4 - \frac{1}{2}(q_2^2/q_1 + q_3^2/q_1)] \\ q_2 q_3/q_1 \\ \{q_4 + (\gamma - 1)[q_4 - \frac{1}{2}(q_2^2/q_1 + q_3^2/q_1)]\} q_2/q_1 \end{bmatrix} \quad (12A.1)$$

$$F = \begin{bmatrix} \varrho v \\ \varrho uv \\ \varrho v^2 + p \\ (E_t + p)v \end{bmatrix} = \begin{bmatrix} q_3 \\ q_2 q_3/q_1 \\ q_3^2/q_1 + (\gamma - 1)[q_4 - \frac{1}{2}(q_2^2/q_1 + q_3^2/q_1)] \\ \{q_4 + (\gamma - 1)[q_4 - \frac{1}{2}(q_2^2/q_1 + q_3^2/q_1)]\} q_3/q_1 \end{bmatrix}$$

The Jacobian matrices of convective terms are obtained by differentiating E and F respect to Q .

$$A = \frac{\partial E}{\partial Q} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}u^2 + \frac{\gamma-1}{2}v^2 & (3-\gamma)u & (1-\gamma)v & \gamma-1 \\ -uv & v & u & 0 \\ -\gamma\frac{E_t u}{\varrho} + (\gamma-1)u(u^2+v^2) & \gamma\frac{E_t}{\varrho} - \frac{(\gamma-1)}{2}(3u^2+v^2) & -(\gamma-1)uv & \gamma u \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \frac{\partial F}{\partial Q} = \begin{bmatrix} -uv & v & u & 0 \\ \frac{\gamma-1}{2}u^2 + \frac{\gamma-3}{2}v^2 & -(\gamma-1)u & -(\gamma-3)v & \gamma-1 \\ -\gamma\frac{E_t v}{\varrho} + (\gamma-1)v(u^2+v^2) & -(\gamma-1)uv & \gamma\frac{E_t}{\varrho} - \frac{(\gamma-1)}{2}(u^2+3v^2) & \gamma v \end{bmatrix} \quad (12A.2)$$

Similarly, the Jacobian matrices P , R , R_x , M , N and N_y are obtained from

$$-[P] + [R_x] = -\frac{\partial V_1}{\partial Q} + \frac{\partial}{\partial x} \left[\frac{\partial V_1}{\partial Q_x} \right]$$

$$-[P] + [R_x] = \frac{\mu_x}{\varrho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}u & \frac{4}{3} & 0 & 0 \\ -v & 0 & 1 & 0 \\ -(\frac{4}{3} - \frac{\gamma}{Pr})u^2 - (1 - \frac{\gamma}{Pr})v^2 - \frac{\gamma}{Pr}\frac{E_t}{\varrho} & (\frac{4}{3} - \frac{\gamma}{Pr})u & (1 - \frac{\gamma}{Pr})v & \frac{\gamma}{Pr} \end{bmatrix} \quad (12A.3)$$

$$[R] = \frac{\mu}{\varrho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}u & \frac{4}{3} & 0 & 0 \\ -v & 0 & 1 & 0 \\ -(\frac{4}{3} - \frac{\gamma}{Pr})u^2 - (1 - \frac{\gamma}{Pr})v^2 - \frac{\gamma}{Pr}\frac{E_t}{\varrho} & (\frac{4}{3} - \frac{\gamma}{Pr})u & (1 - \frac{\gamma}{Pr})v & \frac{\gamma}{Pr} \end{bmatrix} \quad (12A.4)$$

$$-[M] + [N_y] = -\frac{\partial W_2}{\partial Q} + \frac{\partial}{\partial y} \left[\frac{\partial N}{\partial Q_y} \right]$$

$$-[M] + [N_y] = \frac{\mu_y}{\varrho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 \\ -\frac{4}{3}v & 0 & \frac{4}{3} & 0 \\ -(\frac{4}{3} - \frac{\gamma}{Pr})v^2 - (1 - \frac{\gamma}{Pr})u^2 - \frac{\gamma}{Pr} \frac{E_t}{\varrho} & (1 - \frac{\gamma}{Pr})u & (\frac{4}{3} - \frac{\gamma}{Pr})v & \frac{\gamma}{Pr} \end{bmatrix} \quad (12A.5)$$

$$[N] = \frac{\partial W_2}{\partial Q_y}$$

$$[N] = \frac{\mu}{\varrho} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 \\ -\frac{4}{3}v & 0 & \frac{4}{3} & 0 \\ -(\frac{4}{3} - \frac{\gamma}{Pr})v^2 - (1 - \frac{\gamma}{Pr})u^2 - \frac{\gamma}{Pr} \frac{E_t}{\varrho} & (1 - \frac{\gamma}{Pr})u & (\frac{4}{3} - \frac{\gamma}{Pr})v & \frac{\gamma}{Pr} \end{bmatrix} \quad (12A.6)$$

Appendix 12B Treatment of the Region Close to the Boundaries for Eq. (12.5.4)

For the region which has at least one boundary side, Eq. (12.5.4) is approximated differently from Eq. (12.5.5),

On the bottom subdomains $\Omega_{i,1}$, close to the wall $j = 1$

$$\frac{\partial Q_{i,1}}{\partial t} = -\frac{E_{i+1/2,1} - E_{i-1/2,1}}{\Delta x} - \frac{F_{i,1+1/2} - F_{i,1-1/2}}{\Delta y}$$

$$+ \frac{1}{Re} \left[\frac{(E_v)_{i+1/2,1} - (E_v)_{i-1/2,1}}{\Delta x} + \frac{(F_v)_{i,1+1/2} - (F_v)_{i,1-1/2}}{\Delta y} \right]$$

Note that $1 - 1/2 \Rightarrow 0$

$$\frac{\partial Q_{i,j}}{\partial t} = -\frac{E_{i+1,1} - E_{i-1,1}}{2\Delta x} - \frac{\frac{F_{i,1} + F_{i,2}}{2} - F_{i,0}}{\Delta y}$$

$$+ \frac{1}{Re} \left[\frac{(E_v)_{i+1/2,1} - (E_v)_{i-1/2,1}}{\Delta x} + \frac{(F_v)_{i,1+1/2} - (F_v)_{i,1-1/2}}{\Delta y} \right]$$

$$(u_y)_{i+1/2,1} = \frac{u_{i+1/2,1+1/2} - u_{i+1/2,1-1/2}}{\Delta y}$$

$$= \frac{(u_{i+1,1+1} + u_{i,1+1} + u_{i+1,1} + u_{i,1})/4 - (u_{i+1,1-1/2} + u_{i,1-1/2})/2}{\Delta y}$$

$$= \frac{(u_{i+1,1+1} + u_{i,1+1} + u_{i+1,1} + u_{i,1})/4 - (u_{i+1,0} + u_{i,0})/2}{\Delta y}$$

$$\begin{aligned}
(u_y)_{i-1/2,1} &= \frac{u_{i-1/2,1+1/2} - u_{i-1/2,1-1/2}}{\Delta y} \\
&= \frac{(u_{i,1+1} + u_{i-1,1+1} + u_{i,1} + u_{i-1,1})/4 - (u_{i,1-1/2} + u_{i-1,1-1/2})/2}{\Delta y} \\
&= \frac{(u_{i,1+1} + u_{i-1,1+1} + u_{i,1} + u_{i-1,1})/4 - (u_{i,0} + u_{i-1,0})/2}{\Delta y} \\
(u_y)_{i,1-1/2} &= \frac{u_{i,1} - u_{i,1-1/2}}{\Delta y/2} = 2 \frac{u_{i,1} - u_{i,0}}{\Delta y}
\end{aligned}$$

On the top subdomains $\Omega_{i,J}$, close to the line of symmetry, $j = J$

$$\begin{aligned}
\frac{\partial Q_{i,J}}{\partial t} &= -\frac{E_{i+1/2,J} - E_{i-1/2,J}}{\Delta x} - \frac{F_{i,J+1/2} - F_{i,J-1/2}}{\Delta y} \\
&\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{i+1/2,J} - (E_v)_{i-1/2,J}}{\Delta x} + \frac{(F_v)_{i,J+1/2} - (F_v)_{i,J-1/2}}{\Delta y} \right]
\end{aligned}$$

The line of symmetry is at $J + 1/2$. Assume the ghost points are at $J + 1^*$ and $J + 2^*$. Then, from symmetry conditions, we can write

$$\begin{aligned}
\varrho_{i,J+1*} &= \varrho_{i,J}, \quad \varrho_{i,J+2*} = \varrho_{i,J-1} \\
u_{i,J+1*} &= u_{i,J}, \quad u_{i,J+2*} = u_{i,J-1} \\
v_{i,J+1*} &= -v_{i,J}, \quad v_{i,J+2*} = -v_{i,J-1} \\
p_{i,J+1*} &= p_{i,J}, \quad p_{i,J+2*} = p_{i,J-1}
\end{aligned}$$

Note that $J + 1/2 \Rightarrow J + 1$

$$\begin{aligned}
\frac{\partial Q_{i,J}}{\partial t} &= -\frac{E_{i+1,J} - E_{i-1,J}}{2\Delta x} - \frac{F_{i,J+1} - \frac{F_{i,J-1} + F_{i,J}}{2}}{\Delta y} \\
&\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{i+1/2,J} - (E_v)_{i-1/2,J}}{\Delta x} + \frac{(F_v)_{i,J+1} - (F_v)_{i,J-1/2}}{\Delta y} \right]
\end{aligned}$$

$$(u_y)_{i+1/2,J} = \frac{u_{i+1/2,J+1*} - u_{i+1/2,J-1}}{2\Delta y} = \frac{(u_{i+1,J} + u_{i,J}) - (u_{i+1,J-1} + u_{i,J-1})}{4\Delta y}$$

$$(u_y)_{i,J+1/2} = \frac{u_{i,J+1*} - u_{i,J}}{\Delta y} = 0$$

$$(v_y)_{i+1/2,J} = \frac{v_{i+1/2,J+1*} - v_{i+1/2,J-1}}{2\Delta y} = \frac{-(v_{i+1,J} + u_{i,J}) - (u_{i+1,J-1} + u_{i,J-1})}{4\Delta y}$$

On the left boundary subdomains $\Omega_{1,j}$, close to inflow and wall boundary, $i = 1$

$$\begin{aligned}\frac{\partial Q_{1,j}}{\partial t} &= -\left(\frac{E_{1+1/2,j} - E_{1-1/2,j}}{\Delta x}\right) - \left(\frac{F_{1,j+1/2} - F_{1,j-1/2}}{\Delta y}\right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{1+1/2,j} - (E_v)_{1-1/2,j}}{\Delta x} + \frac{(F_v)_{1,j+1/2} - (F_v)_{1,j-1/2}}{\Delta y} \right]\end{aligned}$$

Note that $1 - 1/2 \Rightarrow 0$

$$\begin{aligned}\frac{\partial Q_{1,j}}{\partial t} &= -\left(\frac{\frac{E_{1+1,j}+E_{1,j}}{2} - E_{0,j}}{\Delta x}\right) - \left(\frac{F_{1,j+1} - F_{1,j-1}}{2\Delta y}\right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{1+1/2,j} - (E_v)_{0,j}}{\Delta x} + \frac{(F_v)_{1,j+1/2} - (F_v)_{1,j-1/2}}{\Delta y} \right] \\ (u_x)_{1-1/2,j} &= \frac{u_{1,j} - u_{1-1/2,j}}{\Delta x/2} = 2 \frac{u_{1,j} - u_{0,j}}{\Delta x}\end{aligned}$$

$$\begin{aligned}(u_x)_{1,j+1/2} &= \frac{u_{1+1/2,j+1/2} - u_{1-1/2,j+1/2}}{\Delta x} \\ &= \frac{(u_{1+1,j+1} + u_{1+1,j} + u_{1,j+1} + u_{1,j})/4 - (u_{0,j+1} + u_{0,j})/2}{\Delta x}\end{aligned}$$

$$\begin{aligned}(u_x)_{1,j-1/2} &= \frac{u_{1+1/2,j-1/2} - u_{1-1/2,j-1/2}}{\Delta x} \\ &= \frac{(u_{1+1,j} + u_{1+1,j-1} + u_{1,j} + u_{1,j-1})/4 - (u_{0,j} + u_{0,j-1})/2}{\Delta x}\end{aligned}$$

On the right boundary subdomains $\Omega_{I,j}$, close to right outflow boundary, $i = I$

$$\begin{aligned}\frac{\partial Q_{I,j}}{\partial t} &= -\left(\frac{E_{I+1/2,j} - E_{I-1/2,j}}{\Delta x}\right) - \left(\frac{F_{I,j+1/2} - F_{I,j-1/2}}{\Delta y}\right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{I+1/2,j} - (E_v)_{I-1/2,j}}{\Delta x} + \frac{(F_v)_{I,j+1/2} - (F_v)_{I,j-1/2}}{\Delta y} \right]\end{aligned}$$

Note that $I + 1/2 \Rightarrow I + 1$

$$\begin{aligned}\frac{\partial Q_{I,j}}{\partial t} &= -\frac{E_{I+1,j} - \frac{E_{I,j}+E_{I-1,j}}{2}}{\Delta x} - \frac{F_{I,j+1} - F_{I,j-1}}{2\Delta y} \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{I+1,j} - (E_v)_{I-1/2,j}}{\Delta x} + \frac{(F_v)_{I,j+1/2} - (F_v)_{I,j-1/2}}{\Delta y} \right]\end{aligned}$$

$$(u_x)_{I+1/2,j} = \frac{u_{I+1/2,j} - u_{I,j}}{\Delta x/2} = 2 \frac{u_{I+1,j} - u_{I,j}}{\Delta x}$$

$$\begin{aligned} (u_x)_{I,j+1/2} &= \frac{u_{I+1/2,j+1/2} - u_{I-1/2,j+1/2}}{\Delta x} \\ &= \frac{(u_{I+1,j+1} + u_{I+1,j})/2 - (u_{I,j+1} + u_{I,j} + u_{I-1,j+1} + u_{I-1,j})/4}{\Delta x} \end{aligned}$$

$$\begin{aligned} (u_x)_{I,j-1/2} &= \frac{u_{I+1/2,j-1/2} - u_{I-1/2,j-1/2}}{\Delta x} \\ &= \frac{(u_{I+1,j} + u_{I+1,j-1})/2 - (u_{I,j} + u_{I,j-1} + u_{I-1,j} + u_{I-1,j-1})/4}{\Delta x} \end{aligned}$$

On the four corners

Left-bottom corner $\Omega_{1,1}$, close to both walls, $i = 1, j = 1$

$$\begin{aligned} \frac{\partial Q_{1,1}}{\partial t} &= - \left(\frac{E_{1+1/2,1} - E_{1-1/2,1}}{\Delta x} - \frac{F_{1,1+1/2} - F_{1,1-1/2}}{\Delta y} \right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{1+1/2,1} - (E_v)_{1-1/2,1}}{\Delta x} + \frac{(F_v)_{1,1+1/2} - (F_v)_{1,1-1/2}}{\Delta y} \right] \end{aligned}$$

Note that $1 - 1/2 \Rightarrow 0$ and $1 - 1/2 \Rightarrow 0$

$$\begin{aligned} \frac{\partial Q_{1,1}}{\partial t} &= - \left(\frac{\frac{E_{1+1,1} + E_{1,1}}{2} - E_{0,1}}{\Delta x} - \frac{\frac{F_{1,1+1} + F_{1,1}}{2} - F_{1,0}}{\Delta y} \right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{1+1/2,1} - (E_v)_{0,1}}{\Delta x} + \frac{(F_v)_{1,1+1/2} - (F_v)_{1,0}}{\Delta y} \right] \end{aligned}$$

Right-bottom corner $\Omega_{I,1}$, close to wall and outflow, $i = I, j = 1$

$$\begin{aligned} \frac{\partial Q_{I,1}}{\partial t} &= - \left(\frac{E_{I+1/2,1} - E_{I-1/2,1}}{\Delta x} - \frac{F_{I,1+1/2} - F_{I,1-1/2}}{\Delta y} \right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{I+1/2,1} - (E_v)_{I-1/2,1}}{\Delta x} + \frac{(F_v)_{I,1+1/2} - (F_v)_{I,1-1/2}}{\Delta y} \right] \end{aligned}$$

Note that $I + 1/2 \Rightarrow I + 1$ and $1 - 1/2 \Rightarrow 0$

$$\begin{aligned} \frac{\partial Q_{I,1}}{\partial t} &= - \left(\frac{E_{I+1,1} - \frac{E_{I,1} + E_{I-1,1}}{2}}{\Delta x} - \frac{\frac{F_{I,1+1} + F_{I,1}}{2} - F_{I,0}}{\Delta y} \right) \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{I+1,1} - (E_v)_{I-1/2,1}}{\Delta x} + \frac{(F_v)_{I,1+1/2} - (F_v)_{I,0}}{\Delta y} \right] \end{aligned}$$

Left-top corner $\Omega_{1,J}$, close to inflow and the line of symmetry, $i = 1$, $j = J$

$$\begin{aligned}\frac{\partial Q_{1,J}}{\partial t} &= -\frac{E_{1+1/2,J} - E_{1-1/2,J}}{\Delta x} - \frac{F_{1,J+1/2} - F_{1,J-1/2}}{\Delta y} \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{1+1/2,J} - (E_v)_{1-1/2,J}}{\Delta x} + \frac{(F_v)_{1,J+1/2} - (F_v)_{1,J-1/2}}{\Delta y} \right]\end{aligned}$$

Note that $1 - 1/2 \Rightarrow 0$ and $J + 1/2 \Rightarrow J + 1$

$$\begin{aligned}\frac{\partial Q_{1,J}}{\partial t} &= -\frac{\frac{E_{1+1,J} + E_{1,J}}{2} - E_{0,J}}{\Delta x} - \frac{F_{1,J+1} - \frac{F_{1,J} + F_{1,J-1}}{2}}{\Delta y} \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{1+1/2,J} - (E_v)_{0,J}}{\Delta x} + \frac{(F_v)_{1,J+1} - (F_v)_{1,J-1/2}}{\Delta y} \right]\end{aligned}$$

Right-top corner $\Omega_{I,J}$, close to the line of symmetry and outflow, $i = I$, $j = J$

$$\begin{aligned}\frac{\partial Q_{I,J}}{\partial t} &= -\frac{E_{I+1/2,J} - E_{I-1/2,J}}{\Delta x} - \frac{F_{I,J+1/2} - F_{I,J-1/2}}{\Delta y} \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{I+1/2,J} - (E_v)_{I-1/2,J}}{\Delta x} + \frac{(F_v)_{I,J+1/2} - (F_v)_{I,J-1/2}}{\Delta y} \right]\end{aligned}$$

Note that $I + 1/2 \Rightarrow I + 1$ and $J + 1/2 \Rightarrow J + 1$

$$\begin{aligned}\frac{\partial Q_{I,J}}{\partial t} &= -\frac{\frac{E_{I,J} + E_{I-1,J}}{2} - E_{I-1,J}}{\Delta x} - \frac{F_{I,J+1} - \frac{F_{I,J} + F_{I,J-1}}{2}}{\Delta y} \\ &\quad + \frac{1}{\text{Re}} \left[\frac{(E_v)_{I+1,J} - (E_v)_{I-1/2,J}}{\Delta x} + \frac{(F_v)_{I,J+1} - (F_v)_{I,J-1/2}}{\Delta y} \right]\end{aligned}$$

References

- [1] Cebeci, T. and Cousteix, J.: Modeling and Computation of Boundary-Layer Flows, Horizons Pub., Long Beach, Calif. and Springer, Heidelberg, 1998.
- [2] Hirsh, C.: Numerical Computation of Internal and External Flows, Volume 2, John Wiley and Sons, N.Y., 1988.
- [3] Beam, R. M. and Warming, R. F.: An implicit Factor Scheme for the compressible Navier-Stokes Equations, AIAA Journal, Vol. 16, No. 4, 1978, pp. 393–402.
- [4] Jameson, A., Schmidt, W. and Tukel, E.: Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, AIAA 14th Fluid and Plasma Dynamics Conference, June 23–25, 1981.

Problems

- 12-1.** Derive the Jacobian matrices A and B in Eq. (12A.2).
- 12-2.** Derive the Jacobian matrices P , R , R_x in Eqs. (12A.3–4).
- 12-3.** Derive the Jacobian matrices M , N , N_y in Eqs. (12A.5–6).
- 12-4.** Use the MacCormack method to solve the compressible laminar flow over a flat plate. Use Eq. (2.2.30). Take Mach number $M_\infty = 2$, $L = 2$, $H = 0.002$, $R_L = 10^6$, $u_\infty = 160$, $\rho_\infty = 1$ and $T_\infty = 1$

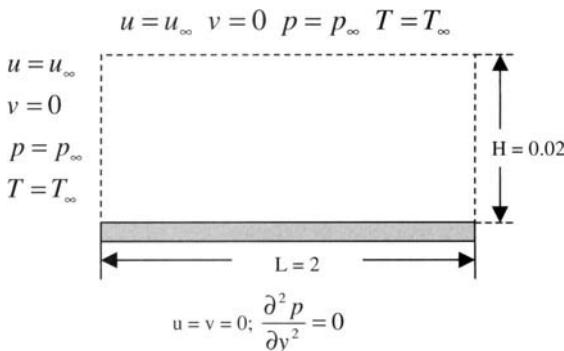


Fig. P12.1. Boundary conditions on the flow over flat plate.

- 12-5.** Repeat Problem 12-4 with $M_\infty = 4$.
- 12-6.** Apply the Beam-Warming method to P12-4.
- 12-7.** Apply the finite volume method to P12-4.
- 12-8.** Apply the MacCormack method to the driven cavity problem (described in P11-4).
- 12-9.** Apply the Beam-Warming method to the driven cavity problem.
- 12-10.** Apply the finite volume method to the driven cavity problem.

Appendix A

Computer Programs

on the Accompanying CD-ROM

The CD-ROM accompanying this book contains both source and executable computer programs and test cases. They are listed below.

Prior to running the test cases, it is necessary to save the executable files and input data on the hard drive in the same directory. The reader can then double click the executable files and run the program with instructions given on the screen.

Chapter 4

Numerical Methods for Parabolic and Elliptic Equations

4.4.1 Explicit Methods for Parabolic Equations

Example 4.1:

Compile `example4_01.f` and generate PC executable file `example4_01.exe`
Input data: `example4_01inp.txt`, output file: `example4_01out.txt`

Example 4.2a:

Central differences for the boundary conditions

Compile `example4_02a.f` and generate PC executable file `example4_02a.exe`
Input data: `example4_02ainp.txt`, output file: `example4_02a0ut.txt`

Example 4.2b:

Forward differences for the boundary conditions

Compile `example4_02b.f` and generate PC executable file `example4_02b.exe`
Input data: `example4_02binp.txt`, output file: `example4_02b0ut.txt`

4.4.2 Implicit Methods: Crank–Nicholson for Parabolic Equations

Example 4.3:

Compile `example4_03.f` and generate PC executable file `example4_03.exe`
Input data: `table4_03inp.txt`, output file: `table4_03out.txt`

4.4.3 Implicit Methods: Keller’s Box Method for Parabolic Equations

Example 4.4:

Compile `example4_04.f` and generate PC executable file `example4_04.exe`
Output file: `example4_04out.txt`

4.5.1 Direct Method for Elliptic Equations

Example 4.5:

Compile `example4_05.f` and generate PC executable file `example4_05.exe`
Input data: `table4_05inp.txt`, output file: `table4_05out.txt`

4.5.2 Iterative Methods for Elliptic Equations

Compile Gauss–Seidel and SOR methods: `SOR.f` and generate PC executable
file: `SOR.exe`

Compile ADI method: `ADI.f` and generate PC executable file: `ADI.exe`

Example 4.6:

Perform calculations by executing `SOR.exe` and `ADI.exe` to get the results.

4.5.3 Multigrid Method for Elliptic Equations

Example 4.7:

Two-Grid Method

Compile `TwoGrid.f` and generate PC executable file: `TwoGrid.exe`
Output: `TG_Out.txt` and Error: `TG_Error.txt`

Chapter 5

Numerical Methods for Hyperbolic Equations

Example 5.2:

Two-step Lax–Wendroff method `LaxWendroff.f`

Compile two-step Lax–Wendroff method `LaxWendroff.f` and generate PC ex-
ecutable file: `LaxWendroff.exe`

Perform calculations by executing **LaxWendroff.exe** to obtain results
LaxWendroff_Error.txt and **LaxWendroff_Out.txt**

Example 5.3:

Explicit MacCormack Method **MacCormack.f** for One-Dimensional Euler Equation

Compile **MacCormack.f** and generate PC executable file: **MacCormack.exe**
Output files: **MacCormackOut.txt** and **MacCormackError.txt**

Example 5.4:

Implicit Beam-Warming method **BeamWarmingBurger.f** for Burger's Equation

Compile **BeamWarmingBurger.f** and generate PC executable file:

BeamWarmingBurger.exe

Output files: **BeamWarmingBurgerOut.txt** and **BeamWarmingBurgerSum.txt**

Example 5.5:

Finite Volume Method

(a) Central difference method **FiniteVolume.f**

Compile **FiniteVolume.f** and generate PC executable file:

FiniteVolume.exe

Output: **FiniteVolumeOut.txt**

(b) Upwind method **FiniteVolumeUpwind.f**

Compile **FiniteVolumeUpwind.f** and generate PC executable file:

FiniteVolumeUpwind.exe

Output: **FiniteVolumeUpwindOut.txt**

Appendix B

Computer Programs

Available from the First Author

The CD-ROM contains both source and executable computer programs and test cases. They are listed below.

Prior to running the test cases, it is necessary to save the executable files and input data on the hard drive in the same directory. The reader can then double click the executable files and run the program with instructions given on the screen.

Chapter 6

Inviscid-Flow Equations for Incompressible Flows

Appendix 6A. Finite Difference Program for a Circular Cylinder

Compile `FDCircular.f` (Finite Difference Program) and generate PC executable file `FDCircular.exe`

Test case: $N_x = 100$, $N_y = 100$. Output file: `FDcircular_Out.txt`

Appendix 6B. Hess-Smith Panel Method (HSPM)

Compile `HSPM.f` (Hess-Smith Panel Method) and generate PC executable file `HSPM.exe`

1. $\alpha = 0$ degree angle of attack. NACA 0012 airfoil

Input file: `deg00_Inp.txt`

Output file: `deg00_Out.txt`

2. $\alpha = 8$ degree angle of attack.

Input file: `deg08_Inp.txt`

Output file: `deg08_Out.txt`

3. $\alpha = 16$ degree angle of attack.

Input file: `deg16_Inp.txt`

Output file: `deg16_Out.txt`

Appendix 6C. Panel Program for Multielement Airfoils

Compile `panelm.f` (Hess-Smith Panel Method) and generate PC executable file `panelm.exe`

Input variable explanations

Geometry of a multielement airfoils for $\alpha = 12$ degree angle of attack for

Input file: `panelm_inp.txt`

Output file: `panelm_out.txt`

Chapter 7

Boundary-Layer Equations

Differential Method with CS Model: Two-Dimensional Flows

Compile `b1p2d.for` (2-dimensional Differential Boundary-Layer Program) and generate PC executable file `b1p2d.exe`. Click for the user's manual.

Test Case NACA 0012 Airfoil with $\alpha = 4$ degree

Input data: `b1p2dIn20.txt`

Output data: `b1p2dOut20.txt`

Chapter 8

Stability and Transition

Transition Prediction with the e^n -Method

Compile `stp2d.f` (Stability Program) and generate PC executable file `stp2d.exe`.

Test Case NACA 0012 Airfoil at $\alpha = 4$ degree

Input data: `stp2dIn3.txt`

Output data: `stp2dOut.txt`

Chapter 9

Grid Generation

9.4 Algebraic and Elliptic Methods (`GRID.f`)

9.6 Conformal Mapping Methods (`Conform.f`)

9.6.2 Wind Tunnel Mapping Methods (`windtunnel.f`)

Chapter 10

Inviscid-Flow Equations for Compressible Flow

10.6 Solution of Full-Potential Equation (tsd.c)

Generate executable `tsd` on unix by using `cc -o tsd tsd.c -lm`

Test Cases with PC executable `tsdPC.exe`

$NX = 101$, $NY = 41$, $Minf = 0.7$, $Relax_factor = 1.5$

1. $K = 2.5$, Output: `tsdK2p50ut.txt` and `tsd2p50utRes.txt`
2. $K = 2.1$, Output: `tsdK2p10ut.txt` and `tsd2p10utRes.txt`
3. $K = 1.8$, Output: `tsdK1p80ut.txt` and `tsd1p80utRes.txt`
4. $K = 1.15$, Output: `tsdK1p15ut.txt` and `tsd1p150utRes.txt`

10.10 Model Problem for the MacCormack Method: Unsteady Shock Tube (tube_mac.c)

Generate executable `tube_mac` on unix by using
`cc -o tube_mac tube_mac.c -lm`

Test Cases with PC executable `tube_mac.exe`

1. $CFL = 0.5$, Output: `tubeMac0p50ut.txt`
2. $CFL = 0.9$, Output: `tubeMac0p90ut.txt`
3. $CFL = 1.1$, Output: `tubeMac1p10ut.txt`

10.11 Model Problem for the MacCormack Method: Quasi-1D Nozzle (nozzle_mac.c)

Generate executable `nozzle_mac` on unix by using
`cc -o nozzle_mac nozzle_mac.c -lm`

Test Cases with PC executable `nozzleMacPC.exe`

1. $CFL = 0.5$, Output: `nozzleMac0p50ut.txt` and `nozzleMac0p50ut_d.txt`
2. $CFL = 1.0$, Output: `nozzleMac1p00ut.txt` and `nozzleMac1p00ut_d.txt`
3. $CFL = 1.1$, Output: `nozzleMac1p10ut.txt` and `nozzleMac1p10ut_d.txt`

10.13 Model Problem for the Implicit Method: Unsteady Shock Tube (tube_implicit.c)

Generate executable `tube_implicit` on unix by using
`cc -o tube_implicit tube_implicit.c blocktri.c -lm`

Test Cases with PC executable `tubeImplicitPC.exe` and explicit artificial viscosity coefficient $\epsilon_e = 0.1$

1. $CFL = 0.5$, Output: `tubeImplicit0p50ut.txt`
2. $CFL = 0.9$, Output: `tubeImplicit0p90ut.txt`
3. $CFL = 2.0$, Output: `tubeImplicit2p00ut.txt`
4. $CFL = 5.0$, Output: `tubeImplicit5p00ut.txt`

10.14 Model Problem for the Implicit Method: Quasi-1D Nozzle (nozzle_imp.c)

Generate executable nozzle_imp on unix by using
`cc -o nozzle_imp nozzle_imp.c blocktri.c -lm`

Test Cases with PC executable nozzleImpPC.exe and NMAX = 100

1. CFL = 0.5, ϵ_e = 1, Output: CFL0p5results.txt and CFL0p5residuals.txt
2. CFL = 1.0, ϵ_e = 1, Output: CFL1results.txt and CFL1residuals.txt
3. CFL = 3.0, ϵ_e = 1, Output: CFL3results.txt and CFL3residuals.txt
4. CFL = 1.0, ϵ_e = 10, Output: CFL1Ee10results.txt and
`CFL1Ee10residuals.txt`

Chapter 11

Incompressible Navier–Stokes Equations

11.2. INS2D Method for Incompressible Navier–Stokes Equations

`ins2d.f` contains main program which reads input data and controls the iterations and computations.

`Residual.f` contains subroutines `flux_de`, `phi_ih`, `residualx`, `flux_df`, `phi_jh` and `residualy`.

Subroutines `flux_dE` and `flux_dF` are for numerical flux computations, see Eq. (11.2.15).

Subroutines `phi_ih` and `phi_jh` are defined according to Eq. (11.2.17).

Subroutines `residualx` and `residualy` calculates righthandside according to Eq. (11.2.22).

`solver.f` contains subroutines `solver`, `invmatrix3`, `tribandsysolver` and `thomas` which use the block elimination method to solve block-tridiagonal system.

Test cases for sudden expansion laminar duct flow with various values of β , grid size and Reynolds number.

Test Cases for Sudden Expansion Laminar Duct Flow:

1. $\beta = 1$, grid size nx = 200 and ny = 100,

Input: `b1g200X100Inp.txt`

Output: `b1g200X1000ut_g.txt`; `b1g200X1000ut_Mass.txt`;
`b1g200X1000ut_p.txt`; `b1g200X1000ut_u.txt`; `b1g200X1000ut_v.txt`

2. $\beta = 50$, grid size nx = 200 and ny = 100,

Input: `b50g200X100Inp.txt`

Output: `b50g200X1000ut_g.txt`; `b50g200X1000ut_Mass.txt`;
`b50g200X1000ut_p.txt`; `b50g200X1000ut_u.txt`; `b50g200X1000ut_v.txt`

3. $\beta = 100$, grid size nx = 200 and ny = 100,

Input: `b100g200X100Inp.txt`

- Output: b100g200X1000ut_g.txt; b100g200X1000ut_Mass.txt;
 b100g200X1000ut_p.txt; b100g200X1000ut_u.txt;
 b100g200X1000ut_v.txt
4. $\beta = 100$, grid size $nx = 150$ and $ny = 75$,
 Input: b100g150X75Inp.txt
 Output: b100g150X750ut_g.txt; b100g150X750ut_Mass.txt;
 b100g150X750ut_p.txt; b100g150X750ut_u.txt; b100g150X750ut_v.txt
5. $\beta = 100$, grid size $nx = 100$ and $ny = 50$,
 Input: b100g100X50Inp.txt
 Output: b100g100X500ut_g.txt; b100g100X500ut_Mass.txt;
 b100g100X500ut_p.txt; b100g100X500ut_u.txt; b100g100X500ut_v.txt
6. $\beta = 100$, grid size $nx = 80$ and $ny = 40$,
 Input: b100g80X40Inp.txt
 Output: b100g80X400ut_g.txt; b100g80X400ut_Mass.txt;
 b100g80X400ut_p.txt; b100g80X400ut_u.txt; b100g80X400ut_v.txt
7. $\beta = 100$, grid size $nx = 50$ and $ny = 25$,
 Input: b100g50X25Inp.txt
 Output: b100g50X250ut_g.txt; b100g50X250ut_Mass.txt;
 b100g50X250ut_p.txt; b100g50X250ut_u.txt; b100g50X250ut_v.txt
8. $\beta = 1$, grid size $nx = 150$ and $ny = 75$,
 Input: b1g150X75Inp.txt
 Output: b1g150X750ut_g.txt; b1g150X750ut_Mass.txt;
 b1g150X750ut_p.txt; b1g150X750ut_u.txt; b1g150X750ut_v.txt
9. $\beta = 1$, grid size $nx = 100$ and $ny = 50$,
 Input: b1g100X50Inp.txt
 Output: b1g100X500ut_g.txt; b1g100X500ut_Mass.txt;
 b1g100X500ut_p.txt; b1g100X500ut_u.txt; b1g100X500ut_v.txt
10. $\beta = 1$, grid size $nx = 80$ and $ny = 40$,
 Input: b1g80X40Inp.txt
 Output: b1g80X400ut_g.txt; b1g80X400ut_Mass.txt;
 b1g80X400ut_p.txt; b1g80X400ut_u.txt; b1g80X400ut_v.txt
11. $\beta = 1$, grid size $nx = 50$ and $ny = 25$,
 Input: b1g50X25Inp.txt
 Output: b1g50X250ut_g.txt; b1g50X250ut_Mass.txt;
 b1g50X250ut_p.txt; b1g50X250ut_u.txt; b1g50X250ut_v.txt

11.6 INS2d for Laminar and Turbulent Flow over Flat Plate

11.6.1 INS2d for Laminar Flow over Flat Plate

Computer programs: ins2dLaminar.f cs.f Residual.f solver.f

PC executable file: ins2dLaminar.exe

Input file: lamInput.txt

Output files: lamOutG.txt lamOut_conv.txt lamOut_vel_pro.txt
 lamOutSkin_friction.txt lamOutMass.txt lamOutp.txt lamOutU.txt
 lamOutV.txt

11.6.2 INS2d for Turbulent Flow over Flat Plate

Computer programs: ins2dTurb.f Residual.f solver.f cs.f
 PC executable file: ins2dTurb.exe
 Input file: turbInp.txt
 Output files: turbOut_conv.txt turbOut_G.txt turbOut_vel_pro.txt
 turbOutSkin_friction.txt turbOut_Mass.txt turbOut_p.txt
 turbOut_U.txt turbOut_V.txt

Chapter 12. Compressible Navier–Stokes Equations

12.6.3.1 MacCormack Method for Compressible Navier–Stokes Equations

MacCormack.f contains the main program which reads input data and controls the iterations and computations. The PC executable file is **MacCormack.exe**

Initial_MC.f contains two subroutines *initialization* and *restart*

Subroutine *initialization* sets up initial conditions according to subsection 12.6.2

Subroutine *restart* sets up restarting calculations.

predictor.f contains subroutine *predictor* which perform prediction calculation according to Eq. (12.3.1)

corrector.f contains subroutine *corrector* which calculates corrections according to Eq. (12.3.2)

UpdateMC.f has subroutine *update* which updates the values according to Eq. (12.3.3)

Output_MC.f contains subroutines *output_g* and *output4* and saves the results into files.

1. Test cases for sudden expansion laminar duct flow with various values of β , grid size and Reynolds number.

RL = 100, L = 30 a grid of 300 x 50 Mach = 0.15

Input file: R100INPUT.TXT

Output files:

General: R100outG.txt

Mass Integration: R100outMass.txt

Velocity in the x-direction: R100outU.txt

Velocity in the y-direction: R100outV.txt

Density: R100outRho.txt

Pressure: R100outP.txt
 Temperature: R100outT.txt
 Energy Et: R100outEt.txt
 Reattachment location: R100reAttach.txt
 Restart file: R100restart.txt

12.6.3.2 Beam–Warming for Compressible Navier–Stokes Equations

BeamWarming.f contains the main program which reads input data and controls the iterations and computations. The PC executable file is **BeamWarming.exe**
Initial.f contains two subroutines initialization and restart
 Subroutine initialization sets up initial conditions according to subsection 12.6.2
 Subroutine restart sets up restarting calculations.

Residual.f contains subroutine *righthandside* which calculates the right hand-side according to Eq. (12.4.19).

solverb4x4.f has subroutines *solver*, *invmatrix4*, *atimeb*, *tribandsyssolver*, *thomas* and *gauss*

Subroutine *solver* controls calculation of linear system described by Eqs.(12.4.19) and (12.4.20)

Subroutine *invmatrix4* computes the inverse of a 4 by 4 matrix.

Subroutine *atimeb* calculates 4 by 4 matrix multiplication. $C = AB$

Subroutine *tribandsyssolver* solves block tridiagonal system

Subroutines *thomas* and *gauss* solve tridiagonal system described in Chapter 4

ainvrs.f contains subroutine *ainvrs* which calculates the inverse of a general matrix

FormMatrix.f contains subroutines *x_directm* and *y_directm*

Subroutines *x_directm* and *y_directm* form a block tridiagonal system according to Eqs. (12.4.19) and (12.4.20), respectively.

Update.f contains subroutine *update* which updates the values interior points and boundary points according to the boundary conditions discussed in subsection 12.6.1

Test cases for sudden expansion laminar duct flow with various values of Reynolds number, Mach number, grid size and dissipation parameters.

Test Cases for Sudden Expansion Laminar Duct Flow:

Reynolds Number Effect

1. $RL = 25$, $L = 20$ a grid of 200×50 $Mach = 0.15$

Input file: R25Mp15Ep5G200X50Inp.txt

Output files:

General: R25Mp15Ep5G200X50outG.txt

Mass Integration: R25Mp15Ep5G200X50outMass.txt

- Velocity in x-direction: R25Mp15Ep5G200X50outU.txt
 Velocity in y-direction: R25Mp15Ep5G200X50outV.txt
 Density: R25Mp15Ep5G200X50outRho.txt
 Pressure: R25Mp15Ep5G200X50outP.txt
 Temperature: R25Mp15Ep5G200X50outT.txt
 Energy Et: R25Mp15Ep5G200X50outEt.txt
2. RL = 50, L = 20 a grid of 200 x 50 Mach = 0.15.
 Input file: R50Mp15Ep5G200X50Inp.txt
 3. RL = 100, L = 30 a grid of 200 x 50 Mach = 0.15.
 Input file: R100Mp15G200X50Inp.txt
 4. RL = 400, L = 40 a grid of 200 x 50 Mach = 0.15.
 Input file: R400Mp15G200X50Inp.txt

Mach Number Effect

1. Mach = 0.15, RL = 50, L = 20 a grid of 300 x 50.
 Input file: R50Mp15Ep5G200X50Inp.txt
2. Mach = 0.7, RL = 50, L = 20 a grid of 300 x 50.
 Input file: R50Mp7Ep5G300X50Inp.txt
3. Mach = 1.0, RL = 50, L = 20 a grid of 300 x 50.
 Input file: R50M1Ep5G300X50Inp.txt

12.6.3.3 Finite-Volume Method for Compressible Navier–Stokes Equations

`FiniteVolume.f` contains the main program which reads the input data and controls the iterations and computations.

The PC executable file is `FiniteVolume.exe`

`InitialFV.f` contains two subroutines *initialization* and *restart*

Subroutine *initialization* sets up initial conditions according to subsection 12.6.2

Subroutine *restart* sets up restarting calculations.

`ResidualFV.f` contains subroutine *righthandside* which performs the integral calculation according to Section 12.5

`RKFV.f` contains subroutines *RKStep*, *updating* and *BoundaryUpdate*

Subroutine *RKStep* performs calculations according to the fourth-order Runge–Kutta method, see Eq. (12.5.7)

Subroutines *updating* and *BoundaryUpdate* perform boundary calculations according to boundary condition and update the solution.

`OutputFV.f` contains subroutines *output_g* and *output4* and saves the results into files.

Test Case for sudden expansion laminar duct flow:

RL = 100, L = 30 a grid of 300 x 50 Mach = 0.15

Input file: `R100input.txt`

Output files:

General: R100outG.txt

Mass Integration: R100outMass.txt

velocity in the x-direction: R100outU.txt

velocity in the y-direction: R100outV.txt

Density: R100outRho.txt

Pressure: R100outP.txt

Temperature: R100outT.txt

Energy Et: R100outEt.txt

Subject Index

- Accuracy 28–29, 34, 37, 189, 223, 237–238, 299, 316, 320
 - order of accuracy 101, 102, 107, 109, 132–134, 149–150, 177, 197, 238, 334
- ADI method 125–127, 336, 339, 364
- Aircraft icing 23–34
 - computing ice accretion 26
 - ice protection 25
 - icing envelopes 23
 - icing program 25
 - LEWICE code 27–28
 - performance degradation due to icing 23
 - prediction of ice shapes 26–28
- Aircraft design and power-plant integration 19–23
- Aerodynamics of ground-based vehicles 34–38
 - application of CFD to automobiles 36–38
- Artificial dissipation (viscosity) 170–173, 296, 309, 312, 320, 321, 325, 363
- Artificial compressibility method 328–350
- Amplification factor 167–169
- Backward-difference 98
- Banded matrix 335, 339
- Beam and Warming method 149–152, 320–325, 353–365
- Bernoulli equation 60
- Block iteration 124–127
- Blowing velocity 214–215, 231
- BLP2 (boundary-layer program) 210–237
- BLP2D 222–229
- Boundary conditions 70–72, 89, 100, 103–107, 116–117, 183–186, 247–249, 302–305, 309–311, 314, 317–318, 329–330, 337–338, 355–356, 365–366
- Burger's equation 97, 151–152, 176
- Central-difference 98
- Characteristic curves 68–69
- Characteristic equation 77
- Characteristic lines 143–146, 153–154, 309–310
- Characteristic slope 169–170
- Circulation 180–182, 192, 199
- Closure problem 56
- Coles formula 86, 91
 - law-of-the-wake 91
- Compatibility equations 144
- Compressible Bernoulli equation 60
- Compressible flow 45–47, 55, 61, 295–325
- Compressible Euler equations 312–313, 316, 320
- Conformal mapping methods 282–287
- Conservation equations 41–72, 159
- Conservation form 47–49, 54, 74, 301
- Conservative fluxes 298, 300
- Conservative TSD 301
- Contact discontinuity 314
- Continuity equation 43, 45, 48, 60–61, 73, 179, 267, 268
- Control volume 42–50, 157, 161, 298
- Convection equation 96, 141, 168
 - diffusion equation 97, 162
 - diffusion term 367–370
- Convergence and stability 165–170
- Convergence rate of the Gauss–Seidel method 123–124
- Courant–Friedrichs–Lowy (CFL) 146, 168, 311
- Crank–Nicolson 105–108, 150, 151
- Cubic equation 142
- Cubic polynomials 270
- Damping parameter 83, 320, 360–361
 - Cebeci–Smith algebraic eddy viscosity 83–85
- Diffusion 57, 97, 162, 173, 334, 367–370
- Diffusion operator 68
- Dimensionless form 50–51, 65–66
- Direct method 115–121
- Direct numerical simulation (DNS) 37, 55
- Dirichlet tessellation 289

- Discontinuity 102, 143, 148, 296–301, 313–314
 Discretization
 – of derivatives 98–99, 331–336
 – of the boundary conditions 337
 Discretized equation 303, 335–336
 Dispersion 170–173
 Dissipation 153, 155–157, 170–173, 320
 Dissipation equation 88
 Dissipation function 46
 Dissipation rate 93
 Dissipative terms 360–365
 Divergence form 46, 297
 Domain of dependence 68–69, 169–170
 Drag reduction 2–10
- Eddy viscosity 59, 81–89, 211, 229
 Eigenvalue procedure 249–253
 Elliptic equations 67–70, 97, 113–132, 184, 278, 301–303
 Energy equation 44–50, 52, 55–56, 62–63, 73–74, 88, 96, 318
 Energy integral 63–64
 e^n -method 253–261
 Enthalpy 56, 61
 Entropy 296–297, 317
 Equation of state 46, 78, 328
 Equivalent first-order system 247
 Error 138–139, 166, 171, 238
 Error of conservation 340
 Error of order 98–99
 Euler 329, 331
 Euler equation 60, 141–164, 297–299, 309–316, 320
 Euler explicit scheme 150, 168
 Euler implicit scheme 150
 Euler method 287
 Euler fluxes 314, 321
 Expansion waves 297, 313, 322
 Expansion shocks 297
 Explicit methods 100–105, 146–148, 312
 Explicit numerical dissipation 296
 Explicit approximation 165
 Explicit artificial dissipation 309, 320, 321, 360
- Factorization 364
 Falkner–Skan equation 217
 – dimensionless pressure-gradient parameter 217
 – similarity variable 222
 Falkner–Skan transformation 216
 Finite difference 98–132, 182–189, 202–204, 218–221, 246–249, 360
 Finite-volume 157–164, 361–365
- Five point finite differences 360
 Flux 48–50, 56–57, 63, 71, 141, 150, 155, 157, 159, 297–300, 339
 – convective 156, 312, 332–336, 342, 356, 363, 367
 – diffusive 328, 367
 – variable 314, 318, 321
 – vector 141–142, 160, 318
 – viscous 54, 334, 342, 356
 Flux-vector-splitting 153–157
 Flux-difference-splitting 153, 157
 Forward-difference 98
 Fourier series method 166–167
 Friction velocity 90
 Full weighting 128
- Gas law 46, 51
 Gauss' elimination method 119–120, 195, 196
 Gauss–Seidel 123–127
- Heat balance 27–28, 37
 Heat 45, 46, 50–51, 56, 96, 310
 Heat-conduction equation 68, 96–97, 121, 151
 Heat conduction vector 50
 Heat flux terms 56–57, 63, 71
 Heat transfer 45, 57, 58, 63, 68, 96
 Hermite interpolation 272–273
 Hess–Smith panel method (HSPM) 189–201
 – applications 197
 – NACA 0012 airfoil 197
 – circular cylinder 198
 – multielement airfoils 201
 Hilbert integral 214
 Hyperbolic equation (problem) 67–69, 96–97, 141–173, 309, 329–330, 356
 Hyperbolic differencing 303
 Hyperbolic operator 304
 Hyperbolic flow region 308
 Hyperbolic tangent function 273–274
- Implicit methods 105–112, 149–151, 321–325
- Incompressible flow 60–63, 96, 179–204
 – three-dimensional 43–45, 55–57
 – two-dimensional 70–72
 – one-dimensional Navier–Stokes equation 172
 Incompressible irrotational flow 179–182
 Incompressible laminar and turbulent flow 62, 83
 Incompressible Navier–Stokes equations 327–347
 Incompressible stability theory 66

- Initial conditions (values) 90–93, 100–101, 104, 252, 314, 316, 338–339, 365
 Integral equation 63–64, 93, 157, 161–164
 Integral form 48–50, 157–158, 214
 Interaction problem 212–215
 – blowing or suction velocity 214, 215
 – displacement surface 214–215
 – interactive boundary-layer scheme 215
 – Kutta condition 192–195, 215
 Interactive boundary-layer theory 214, 215
 Interior points 272–274, 278
 Intermittency factor 84, 86
 Internal energy 46, 48
 Internal flows 36–37, 72
 Interpolation 127–129, 224, 271–277
 – Lagrange 224, 271–272
 – Hermite 272–273
 – spline 273
 – transfinite 271–277
 Inviscid Burger's equation 97, 151–152
 Inviscid compressible flow 295–325
 Inviscid flow 60–62, 97, 175, 179–201
 Inviscid flow equations for incompressible flows 179–204
 Inviscid-pressure distribution 212–213
 Inviscid velocity distribution 7, 214, 239
 Irrotational flow 60–62, 179–181, 191
 Isentropic compressible flow 60–62
 Isentropic flow 61, 298–299
 Isentropic jump relations 297
 Iterative method 121–132, 339
- Jacobian matrix 142–145, 154, 175, 332–333, 367–370
 Jacobian 278–279
 Jacobian determinant 53
 Jacobi iteration method 137–139
- $k-\varepsilon$ model 88, 89
 $k-\omega$ model 88
- Lagrange interpolation 224, 271–272
 Lagrange polynomials 272, 276
 Laminar flow 56–59, 70–71, 212–213, 216–217, 230
 Laminar flow control (LFC) 3–10
 – adjustment of pressure gradient by shaping 3
 – natural laminar flow (NLF) 3
 – hybrid laminar flow control 6–10
 Laplacian operator 44
 Laplace equation 60–61, 78, 79, 97, 114, 179–183, 185, 191
 Laplace difference equation 114–119, 121, 123
 Law of the wall 90
 Lax method (scheme) 168–169, 311, 320
 Lax–Wendroff method 146–148, 312
 Leapfrog scheme 150
 Linearized form 62
 Linear convection equation 96, 141, 153, 168
 Linear equation (system) 113–115, 166, 220, 221, 249, 250
 Linear form 272
 Linear function 270
 Linear interpolation 127, 281
 Linear Lagrange polynomials 276
 Linear multistep method 149
 Linear stability theory 66–67, 243, 253–255
 – e^n -method 253–255
 – disturbance stream function 66
 – linear stability equations 66
 – parallel flow approximation 65
 – radian (circular) frequency 66
 – separation of variables 65
 – small-disturbance theory 64, 66, 299
 – two-dimensional disturbance 64, 65
 – wave number 66
 – wavelength 66
 Linear wave equation 299–300
 Linearization 150, 220, 222, 225
 Linearized Burger's equation 176
 Linearized momentum equation 220
 Local skin-friction coefficient 63, 236, 237, 343
 Local speed of sound 143, 152, 308
 Logarithmic law of the wall 90
 – buffer zone 90
 – linear sublayer 90, 344
- MacCormack 148–149, 312–318, 356
 Mach number 305, 308, 310, 365
 Mass 46, 48–50, 79, 90, 91, 96, 214, 225, 229–231, 340
 Matrix 53–54, 107, 111–121, 122–123, 136–138, 142–146, 151, 154, 187, 194, 221, 226–228, 248, 320, 330–336
 – unity matrix 320, 359
 Mesh 52, 99, 158–165, 218, 263–293, 301–303, 360
 Michel's method 234
 Mixing length 81
 Mixing-length formulations 83
 Model equation 96–97, 141–170
 – turbulence models 81–92
 Modified equation 172
 Momentum 88, 299

- Momentum equation 43–49, 55–56, 62–64, 66, 70, 88–89, 97, 161–162, 180, 211–212, 240, 328–329
 Momentum integral integration 63, 79, 93
 Momentum thickness 63
 Multigrid 127–132
 Multistep time integration 149
 Multistep methods 149, 230
 Navier–Stokes equations 42–63
 – continuity equation 43, 45, 48–49
 – incompressible 43–45, 55–56, 179–204, 327–346
 – compressible 45–47, 60, 295–325, 353–374
 – differential and integral forms 41
 – momentum equation 43–47
 Newton’s method 220–221, 249–253, 258, 318
 Newtonian fluid 44, 56
 Nonconservative form (equation) 47, 75, 301
 Nondimensional form 50–51, 68
 Nonlinear equation 61, 97, 141–142, 144, 220–221, 274, 299–300
 Nonlinear Euler equation 141, 148, 152, 154
 Nonlinear wave equation 300, 312
 Non-uniform 110, 223, 247, 264, 267
 Normal viscous stress 44, 56, 331
 Normal velocity 191, 192, 193, 212, 222, 339
 No-slip surface 310, 338
 Nozzle 268–269, 296, 315–319, 322–325
 Numerical boundary condition 147–148, 151, 310–311, 329, 355
 Numerical dissipation 153, 155, 156, 170–173
 Numerical domain of dependence 170
 Numerical flux 155, 157, 333
 Numerical solution of the boundary-layer equations 216–229
 – block-elimination method 112, 119, 120, 226, 258
 – Falkner–Skan transformation 216–217
 – finite-difference approximation 218–219
 – linear system 220–221
 – Newton’s method 220
 – numerical formulation 218
 – similarity variables 216
 Numerical solution of the Orr–Sommerfeld equation 246–252
 – boundary conditions 243, 247–249
 – box scheme 246–249
 – eigenfunctions 244, 248
 – eigenvalue problem 244, 249, 251
 – eigenvalue procedure 245, 249–253
 – eigenvalues 244, 248
 – Newton’s method 248–249, 251–253
 – numerical scheme 247
 – variational equations 250, 251
 Order-of-magnitude analysis 57–58, 65
 Order of errors 98–99, 101, 107, 238
 Ordinary differential equations 63–64, 66, 110, 218, 240, 362
 Orr–Sommerfeld equation 66, 243
 – amplification rates 7–10, 244, 253–260
 – boundary conditions 243
 – critical Reynolds number 244–245
 – neutral stability curve 244
 – stability equations 64–67, 243
 Outer layer 83–90
 Over-relaxation 124, 304
 Panel method 189–204
 Parabolic equation 67–70, 96, 97, 100–113
 Parabolic mapping function 283–285
 Parabolized Navier–Stokes equations 57–58
 Peaceman and Rachford method 125
 Perfect gas law 46, 79, 142, 175
 Phase angle 66, 168–169
 Physical boundary conditions 68, 310, 329, 351, 355
 Poisson equation 97, 113–132, 205, 278–282
 Potential equation 60–62, 75, 179–204, 297–299, 301–309
 Prandtl number 57, 59
 Prediction of aerodynamic performance characteristics 28
 – interactive boundary-layer method 29
 – panel method 29
 Prediction of transition 243, 261
 – e^n -method 253
 – empirical correlations 234
 Predictor-corrector scheme 148, 312, 316, 356
 Pressure coefficient 188–189, 196
 Pressure difference rule 12–17
 Pressure distribution 212, 213, 305–306
 Pressure force 43, 57
 Pressure gradient 71, 217, 224, 245
 Propagation of information 142–143, 153
 Propagation of error 166
 Propagation velocity 243
 Pseudocompressibility method 78, 330
 Pseudo-sonic speed 330
 Pseudo-time 78, 330, 331
 Quasi-1D nozzle 315–318, 322–325

- Rankine–Hugoniot relations 296–299
 Reduced forms of the Navier–Stokes equations 57–64
 – order-of-magnitude analysis 57, 65
 – parabolized Navier–Stokes equations 57–58
 – thin-layer Navier–Stokes approximation 58, 62
 Relaxation parameter 124–125, 214, 304–308
 Residual 126–127, 129, 318, 335, 364
 Residual smoother 364
 Restriction operators 128–129
 Reynolds-averaged Navier–Stokes equations 41, 55–56
 Reynolds number 50–51, 57, 84, 90, 198, 206, 225, 232, 244, 245, 246, 256–257, 339
 Reynolds shear stress 56, 63, 64, 74, 84, 350
 Richardson extrapolation 238–239
 Richardson number 57
 Runge–Kutta scheme 363
- Separation 214, 233
 Separation point 213
 Shape factor 63
 Shear stress 43–44, 56, 84, 93, 331
 Shock 296–301, 313–314, 321–322, 323
 Shock wave 296–299
 Similar laminar flows 230–231
 Singularity at the separation point 213
 Singularity of boundary layer equations 212, 213, 233
 Skin-friction drag reduction 2–10
 Solution of the Orr–Sommerfeld equation 246–252
 – boundary conditions 247–249
 Space discretization 161, 171
 Specific heat, ratio 45–46, 51, 96, 298, 310, 365
 Speed of sound 50–51, 61, 142, 170, 308, 310
 Sudden expansion laminar duct flow 336–349, 365–367
 Stability 165–170, 311–312, 320, 357
 Stability and transition 243–261
 Stability diagrams 245
 – critical Reynolds number 244, 249
 – neutral stability curve 244, 253–254, 259
 – critical Reynolds number 253, 256
 Stability equations 64–67
 Stability-transition program (STP) 256–258
 – onset of transition 84, 234
- Stagnation flow 230
 Stagnation points 197
 Steady flows 57–58, 60–61, 63–64, 70, 162, 204, 267, 297, 301, 331
 Steady convection and diffusion equation 162
 Stiffness 311, 354
 Stokes flow 62
 Streamlines 42, 60, 66, 180–182, 216, 283, 302, 308
 Subsonic flow 69, 215, 301–302, 309–311, 316–317, 364
 Substantial derivative 43
 Successive over-relaxation (SOR) 124
 Supercritical airfoil 3, 208
 Supersonic flow 69, 295, 301, 302, 307, 309–310, 316–318
- Taylor series expansion 98, 155, 171, 252, 255, 331
 Thermal conductivity 45, 59, 96
 Thin layer approximation 58, 62
 Thomas algorithm 107–108
 Three-level Beam and Warming scheme 151
 Time integration 149
 Tollmien–Schlichting waves 66
 Total energy 45–46, 48, 50,
 Transition prediction 84, 243–261
 Transonic small disturbance (TSD) 75, 296, 299, 301–307
 Transonic flow 208, 295, 301, 305
 Transport of turbulence 82, 83
 Transport equation 87, 89, 90
 Trapezoid 270
 Trapezoidal scheme 106, 150, 151
 Tridiagonal matrix (form, structure, system) 107, 112, 115, 119, 125, 136, 151, 164, 221, 364
 Truncation error 171, 315, 320–321
 Turbulence models 81–90
 – algebraic mixing length 81–86
 – eddy viscosity models 81–83, 86
 – one-equation models 87
 – transport equations 87, 89
 – two-equation models 88–89
 – zero-equation models 83–86
 Turbulent flow 3–4, 55–56, 59, 62–64, 70, 85, 211–212, 229, 243
 Turbulent heat-flux gradients 56, 350
 Turbulent stresses 74
 Two-step Lax–Wendroff method 146–147
 Two-step predictor–corrector scheme 148–149
- Unconditional stable 105–107, 175

- stability 165–170
- Unstable 153, 156, 166, 168, 175, 244
- Upwind method 152–157, 163–164, 301, 305, 308, 309, 332–335
- Upwind direction 153
- Upwind propagation 153
- Velocity-defect law 90
- Viscosity coefficient 59, 81–89, 172, 211, 229, 320, 365
- Viscous equations 214
- Viscous flow 212, 267
- Viscous flux 54, 334, 342, 356
- Viscous region 70, 297, 355
- Viscous stresses 43–45, 47, 49, 51, 56, 57, 328, 331
- Variational equations 250–251, 258
- von Neumann analysis 165–168, 175
- Vortex 181–182, 190
- Vorticity 60, 66, 86, 87, 191–193, 196, 201
- Wave number 66, 136–137, 139, 166
- Wavelength 66
- Wave equation 68, 169–170, 171, 299–300, 330
- Weak solutions 297
- Weak instability 311
- Weighted Jacobi iteration method 137–139
- Well-posed problem 147
- Zero-equation models 83–86
 - mixing length 81, 83