620.001
51825
COM
2003

052266

Computational Fluid Flow and Heat Transfer

Second Edition

ASTU022553EIT

Editors
K Muralidhar
T Sundarajan

FOR SALE ONLY IN
India, Pakistan, Bangladesh,
Nepal, Bhutan and Sri Lanka
MARKETING RIGHTS VIOLATION
WILL INVITE LEGAL ACTION

Narosa Publishing House
New Delhi Chennai Mumbai Kolkata

አዳማ ዩኒቨርሲቲ
Adama University
ቤተመፅሐፍት
Library

EDITORS

K. Muralidhar
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur
Kanpur, 208016 India

T. Sundararajan
Department of Mechanical Engineering
Indian Institute of Technology, Madras
Chennai, 600036 India

Copyright © 1995, 2003 Narosa Publishing House Pvt. Ltd.
First Reprint 2004
Second Reprint 2005
Third Reprint 2007

NAROSA PUBLISHING HOUSE PVT. LTD.

22 Darya Ganj, Delhi Medical Association Road, New Delhi 110 002
35-36 Greams Road, Thousand Lights, Chennai 600006
306 Shiv Centre, D.B.C. Sector 17, K.U. Bazar P.O., Navi Mumbai 400705
2F-2G Shivam Chambers, 53 Syed Amir Ali Avenue, Kolkata 700019

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of the publisher.

All export rights for this book vest exclusively with Narosa Publishing House. Unauthorised export is a violation of Copyright law and is subject to legal action.

This book has been printed from the camera ready text provided by the editors.

ISBN 81-7319-522-6

Published by N K Mehra for Narosa Publishing House,
22 Darya Ganj, Delhi Medical Association Road, New Delhi 110 002
and printed at Rajkamal Electric Press, Delhi 110033 (India).

PREFACE TO THE SECOND EDITION

It is heartening to see that the book has been widely used. In the second edition, the authors and editors have strived to eliminate typographical errors in the text. Additional corrections that improve the readability of the text have been incorporated. Where appropriate, new references have been added and additional problems have been included. The finite volume method has emerged as a technique that combines the benefits of finite differences and finite elements. Specifically, it is applicable to three dimensional unsteady flows in complex geometries. It uses structured collocated grids; the grids themselves can be orthogonal or non-orthogonal. Extension of the finite volume technique to compresible fluids as well as turbulent flows is possible. To reflect this development in the subject, a new chapter on the finite volume method has been added to the book. It replaces one on Spectral Methods, whose utlity in engineering applications has been found to be limited. A set of FORTRAN programs accompany the text and can be procured from the publisher at a nominal charge.

Assistance provided by Mr Sushanta Dutta during the final stages of preparation of the text is gratefully acknowledged.

K. Muralidhar
T. Sundararajan

PREFACE TO THE FIRST EDITION

Recent years have seen a rapid increase in the use of computers for engineering problem-solving. This is particularly true for subjects involving fluid flow and heat transfer, which occur in applications related to aerospace, power sector and automobile industry. The reasons for this development can be attributed to several factors. They include growth in the complexity of the unsolved engineering problems, the need for quick solutions of moderate accuracy, the prohibitive costs involved in performing even scaled laboratory experiments, the absence of analytical solutions and the near-exponential growth in the number-crunching abilities and memory of the computers. In contrast to the limitations of experimental and analytical techniques, numerical methods of solving flow and heat transfer equations have rapidly developed over the last decade. These techniques have now reached a level of maturity where they can be applied routinely to solve engineering problems. They are also being employed as powerful research tools, all over the world.

In recognition of the emerging importance of the discipline of computational fluid flow and heat transfer, many engineering and applied mathematics departments in Indian universities and abroad, are now offering an undergraduate or a master's level course on this subject. The available text books, however, do not provide a comprehensive view of the subject and tend to concentrate only on a few selected techniques and a limited class of problems. Further, existing books do not address the common difficulties faced by a beginner during the selection and application of most of the numerical techniques. In the present book special effort has been taken to highlight the strengths and weaknesses of various methods. Moreover a comprehensive coverage of many of the important numerical simulation algorithms has been provided, including significant contemporary developments. Several illustrative examples have been included in order to describe the important details of implementation, for each numerical technique. Although it was desired to include software on some of the important techniques, it was not possible to provide program listings as a part of the book, due to space constraints. However, a set of useful of programs have been provided in a floppy diskette, which could be obtained from the publisher on payment.

The idea of the present book came up at the conclusion of an ISTE sponsored short term course entitled, *Computational Fluid Flow and Heat Transfer* held at IIT Kanpur in May 1992. Around fifty participants from colleges and universities from around the country as well as research organizations attended the course. The audience response was overwhelming. There was unanimity of opinion that the material covered was important and that a book covering these topics would be suitable for the Indian students and research community.

One of the chief goals of the present book is to familiarize the well-developed techniques of computational fluid dynamics (CFD) to senior undergraduate students, post-graduate students, college teachers and researchers working in

this area. The material covered in the book could form the basis of a first level course on CFD. In particular, a one semester UG/PG course could be taught based on Chapters 1-10 of the book. Chapters 11-15 were prepared with the intention of providing reference material to post-graduate students and researchers who employ advanced computational tools in their research work. In all the chapters of the book, sections are arranged in the increasing order of difficulty, with some of the latest techniques discussed towards the end. In a one-semester course, the instructor may employ his discretion to include or omit some of these topics. For the sake of coherence, the book has been divided into four parts. The first part covers the fundamentals of problem formulation, basic numerical techniques and matrix operations, solution of ordinary differential equations and finite difference discretization. Part II covers the techniques for incompressible flow and heat transfer modelling. Methods useful for inviscid, high speed calculations are discussed in Part III. Some of the emerging special areas of research are described in Part IV. The reader is expected to have some familiarity with the subjects of fluid mechanics and heat transfer before reading this book. An advanced reader who is already familiar with the formulation aspects of fluid mechanics and heat transfer problems may skip Chapter 1.

The editors and the authors of the book would like to express their immense appreciation to the sponsor, IIT Kanpur, for bringing out this volume under the IIT-Kanpur series publication programme. Special thanks are due to the Quality Improvement Programme Coordinators, Professors Vijay Gupta, D. Kunzru, G. K. Dubey and Krishna Kumar for their continuous encouragement. The typists Mrs. Debjani Deb and Mrs. Suguna Sathyamurthy deserve special credit for their excellent typing of the manuscript. The authors would like to gratefully acknowledge the help provided by the draughtsmen, Mr. S. S. Kushwaha and Mr. G. K. Shukla for the preparation of the figures in the book. Assistance provided by Dr. P.M.V. Subbarao during the final stages of preparation of the text is acknowledged. Last but not the least, the authors wish to express their deep appreciation to their respective families, without whose love and encouragement this book would not have been possible.

K. Muralidhar
T. Sundararajan

AUTHORS

1. **Arora, N.L.**
   Apartment 2001, Ground Floor, Block II
   Gowri apartments, RMV stage II
   New BEL Road, Bangalore

2. **Biswas, G.**
   Department of Mechanical Engineering
   Indian Institute of Technology, Kanpur

3. **Eswaran, V.**
   Department of Mechanical Engineering
   Indian Institute of Technology, Kanpur

4. **Muralidhar, K.**
   Department of Mechanical Engineering
   Indian Institute of Technology, Kanpur

5. **Sengupta, T.K.**
   Department of Aerospace Engineering
   Indian Institute of Technology, Kanpur

6. **Sharma, A.**
   Department of Mechanical Engineering
   Indian Institute of Technology, Kanpur

7. **Sundararajan, T.**
   Department of Mechanical Engineering
   Indian Institute of Technology, Madras

CONTENTS

Preface to the Second Edition vii
Preface to the First Edition ix

Part I: Preliminaries

Chapter 1 Review of Equations Governing Fluid Flow and Heat Transfer 1
K. Muralidhar

Chapter 2 Applied Numerical Methods 57
K. Muralidhar

Chapter 3 Numerical Solution of Ordinary Differential Equations 107
Vinayak Eswaran

Chapter 4 Finite Differences, Discretization, Consistency, Stability and Fundamentals of Fluid Flow Modelling 135
Gautam Biswas

Part II: Incompressible Flow and Heat Transfer

Chapter 5 Finite Difference Applications in Heat Conduction and Convection 169
T. Sundararajan

Chapter 6 Solution of Viscous Incompressible Flows by the Stream Function -Vorticity Formulation 209
T. Sundararajan

Chapter 7 Solution of Navier-Stokes Equations for Incompressible Flows using MAC and SIMPLE Algorithms 229
Gautam Biswas

Chapter 8 Finite Element Method 275
K. Muralidhar

Part III: Inviscid Flow

Chapter 9	Panel Methods *K. Muralidhar*	**333**
Chapter 10	Three Dimensional Panel Method for Compressible Flow *N. L. Arora*	**367**
Chapter 11	Compressible Flows *T. K. Sengupta*	**403**

Part IV: Special Topics

Chapter 12	The Finite Volume Method *Atul Sharma and Vinayak Eswaran*	**445**
Chapter 13	Automatic Grid Generation for Complex Geometry Problems *T. Sundararajan*	**483**
Chapter 14	Operator-Splitting Algorithm for Advection-Diffusion Equations *K. Muralidhar*	**539**
Chapter 15	Computation of Transitional Flows *T. K. Sengupta*	**573**
	Index	**589**

1
Review of Equations Governing Fluid Flow and Heat Transfer

1.1 Introduction

It is important to know the velocity, pressure and temperature fields in detail in a large number of applications involving fluids, namely, liquids and gases. The performance of devices such as turbomachinery and heat exchangers is determined entirely by the pattern of fluid motion within them. Examples of natural systems are the earth's atmosphere and the oceans. The flow of air and water primarily determines the weather on our planet. A partial list of applications which demand knowledge of flow and heat transfer is given below.

Turbomachinery.
Flow and heat transfer in thermal power plants and nuclear power reactors.
Aerodynamics, computation of lift and drag, flow-induced vibrations and stall.
Combustion chamber design in engines and furnaces.
Material processing, e.g. casting of metals and injection moulding of plastics.
Enhanced oil recovery from rock formations.
Geophysical flows, e.g. atmospheric convection and groundwater movement.
Blood flow in the human body.
Transport of slurries in process industries.

The present chapter derives mathematical equations which govern flow and heat transfer for a wide class of problems. General analytical solutions of these equations do not exist at this time. The goal of computation is to obtain special solutions of these equations for problems that have relevance to engineering .

Physical principles that govern fluid flow and heat transfer are: (1) the law of conservation of mass, (2) Newton's second law of motion and (3) the law

of conservation of energy. For an isothermal fluid the conservation of energy principle reduces to conserving mechanical energy alone. Since the mechanical energy equation is equivalent to Newton's second law of motion, it need not be satisfied explicitly in problems where the temperature field is free of gradients.

The physical laws of motion and energy transfer give rise to equations in terms of the stress components and enthalpy. These are not measurable and one needs to relate them to quantities such as velocity, thermodynamic pressure and temperature. In engineering analysis it is common to relate stresses to the velocity components and enthalpy to temperature through empirical equations called *constitutive relations*. These equations contain undetermined constants that must be found independently from laboratory experiments. A large variety of constitutive relations are possible; however, for commonly encountered fluids such as air, water and oils, the form of these relationships remains the same, the differences arising from the magnitude of the empirical coefficients. Hence the constitutive relations represent a wide classs of fluid behaviour but they are not physical laws. The mathematical form of these equations is restricted by the constraint of the second law of thermodynamics. The second law imposes certain conditions on the empirical coefficients in such a way that there is one-way transfer of mechanical energy that is dissipated in the flow field as thermal energy.

In the mathematical description of fluid flow it is necessary to assume that quantities such as velocity, pressure and temperature vary continuously from one point to another. This is the assumption of *continuum*. In view of the particulate nature of matter it is appropriate to question the validity of this assumption. For most commonly encountered problems in engineering the molecular mean free path is several orders-of-magnitude smaller than the characteristic linear dimension of the problem. Results obtained on the basis of the continuum approximation are also seen to be in good agreement with laboratory experiments. Hence the applicability of the notion of continuum is usually taken for granted.

A convenient starting point for deriving the governing equations for flow and heat transfer is the Reynolds Transport Theorem. Let CV be a control volume fixed in space and CS be its boundary. Let Φ be a transported variable associated with the mass of the fluid in CV and ϕ is its intensive value, i.e. Φ per unit mass. The unit outward drawn normal on CS is denoted as $\mathbf{n}$ (Figure 1.1). Then quantities Φ and ϕ are related as

$$\Phi = \int_{CV} \rho\phi d\mathbf{x}$$

where ρ is the material density and $d\mathbf{x}$ is an infinitesimal volume. Let $\mathbf{u}$ be the velocity vector in the flow domain. The Reynolds Transport Theorem (RTT) can be stated as (Currie, 1993):

$$\frac{d\Phi}{dt} = \int_{CV} \frac{\partial}{\partial t}(\rho\phi)d\mathbf{x} + \int_{CS} \rho\phi(\mathbf{u}\cdot\mathbf{n})ds \tag{1.1}$$

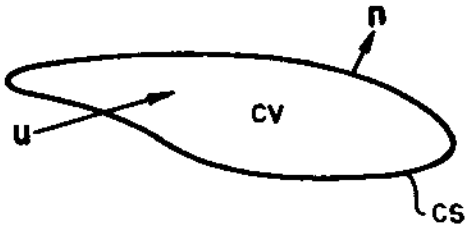


Figure 1.1: Control Volume in a Fluid-filled Region

In words the production rate of the total property Φ is equal to the sum of contributions from changes in ϕ throughout the control volume and the net outflow of ϕ at the control surface. With Φ and ϕ suitably defined, equations governing flow and heat transfer are obtained by substituting the physical laws and the constitutive relations into Equation 1.1.

1.2 Conservation of Mass

Identifying Φ as the total mass of the control volume CV and $\phi = 1$, the conservation of mass principle can be stated as, $d\Phi/dt = 0$. Hence Equation 1.1 reduces to

$$\int_{CV} \frac{\partial \rho}{\partial t} d\mathbf{x} + \int_{CS} \rho \mathbf{u} \cdot \mathbf{n} ds = 0$$

This is an integral form of the mass balance equation and is a useful starting point for a variety of numerical schemes. It is more conventional to recast it in the form of a differential equation. This requires that the velocity field be smooth enough to be differentiable. This condition is usually satisfied in most problems. In problems involving jumps in the dependent variable one applies the differential form of the governing equation on each side of the jump and the discontinuity is separately handled. Using Gauss divergence theorem

$$\int_{CS} \rho \mathbf{u} \cdot \mathbf{n} ds = \int_{CV} \nabla \cdot \rho \mathbf{u} d\mathbf{x}$$

the mass balance equation becomes

$$\int_{CV} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} \right) d\mathbf{x} = 0$$

for arbitrary choices of the control volume CV. Hence the differential form of the mass balance equation is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \tag{1.2}$$

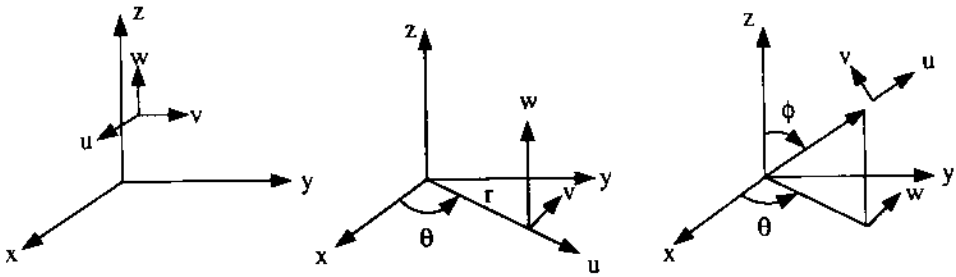


Figure 1.2: Coordinate Systems and Velocity Components

Here ∇ is the gradient operator. Equation 1.2 is in coordinate free form; special forms can be derived by using the appropriate definition of the gradient operator. The mass balance equation is also called the *Continuity Equation*. The expanded version of Equation 1.2 in Cartesian, cylindrical and spherical coordinates is given below:

$$\text{Cartesian}: \quad \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0$$

$$\text{Cylindrical}: \quad \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial r}(\rho u) + \frac{\rho u}{r} + \frac{1}{r}\frac{\partial(\rho v)}{\partial \theta} + \frac{\partial(\rho w)}{\partial z} = 0$$

$$\begin{aligned}\text{Spherical}: \quad \frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 \rho u) \quad &+ \quad \frac{1}{r\sin\phi}\frac{\partial}{\partial \phi}(\rho v \sin\phi) \\ &+ \quad \frac{1}{r\sin\phi}\frac{\partial(\rho w)}{\partial \theta} = 0 \qquad (1.3)\end{aligned}$$

The respective coordinate systems and the velocity components are shown in Figure 1.2.

In a large number of applications changes in pressure and temperature are small enough for density to be nearly a constant in the flow field. The flow is then said to be incompressible. Equation 1.1 reduces to

$$\nabla \cdot \mathbf{u} = 0 \quad (\rho = \text{ constant}) \qquad (1.4)$$

The expanded form of Equation 1.4 in the three coordinate systems will be:

$$\text{Cartesian}: \qquad u_x + v_y + w_z = 0$$

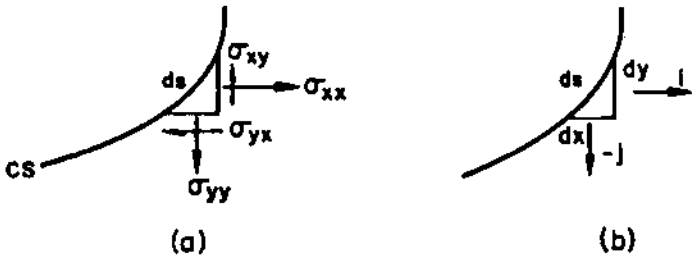


Figure 1.3: Stress Components on a Control Surface

$$\text{Cylindrical}: \qquad u_r + \frac{u}{r} + \frac{1}{r}v_\theta + w_z = 0$$

$$\text{Spherical}: \quad \frac{1}{r}(r^2u)_r + \frac{1}{\sin\phi}(v\sin\phi)_\phi + \frac{1}{\sin\phi}w_\theta = 0 \tag{1.5}$$

The notation used is $u_x = \partial u/\partial x, u_r = \partial u/\partial r, (r^2u)_r = \partial(r^2u)/\partial r$ and so on. Liquids can be frequently assumed to be incompressible. In gaseous flows density variations are limited to 5 percent of the average value for a Mach number M of upto 0.3. Here M is defined as U/c, where U is a characteristic velocity and c is the speed of sound in the medium. Hence air velocities upto about 100 m/s under standard atmospheric conditions can be treated as incompressible.

1.3 Newton's Second Law of Motion

In Reynold's Transport Theorem we choose Φ as momentum of the mass of the fluid contained in CV and hence ϕ is the local velocity vector $\mathbf{u}$. From Newton's second law of motion $d\Phi/dt = \mathbf{F}$, the total external force acting on the control volume. This force is the sum of the body forces acting throughout the control volume and the surface force acting on the boundary of the control volume. Let χ be the body force per unit volume. In applications it is generally a prescribed quantity. For example, body force fields can arise from gravitation and electromagnetism. To calculate the surface forces, consider a part of the control surface and the stresses acting on it (Figure 1.3).

The x and y components of the force acting on the element ds are $(\sigma_{xx}\ dy - \sigma_{yx}\ dx)$ and $(\sigma_{xy}\ dy - \sigma_{yy}\ dx)$. The unit normal acting on the element ds is, $\mathbf{n}ds = dy\ \mathbf{i} - dx\ \mathbf{j}$. Hence the elemental traction vector on the control surface is

$$(dy\ \ -dx)\begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} = \mathbf{n}\cdot\sigma ds$$

where σ is the stress tensor. The total surface force is $\int_{cs}\mathbf{n}\cdot\sigma ds$. Using Gauss divergence theorem, the surface integral can be converted into the volume integral, $\int_{CV}\nabla\cdot\sigma\ \mathbf{dx}$. Hence

$$\frac{d\Phi}{dt} = \int_{CV} (\chi + \nabla \cdot \sigma) d\mathbf{x}$$

With ϕ identified as $\mathbf{u}$, the right hand side of Reynolds Transport Theorem can be written as $\int_{CV}[(\rho\mathbf{u})_t + \nabla \cdot (\rho\ \mathbf{u}\ \mathbf{u})]\ d\mathbf{x}$. The momentum equation that arises from RTT is

$$\int_{CV} (\chi + \nabla \cdot \sigma) d\mathbf{x} = \int_{CV} \left[\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) \right] d\mathbf{x}$$

for arbitrary choices of CV. Hence the integrands on each side can be directly equated to yield

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) = \chi + \nabla \cdot \sigma$$

Subtracting the mass balance equation (1.2) finally results in

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \chi + \nabla \cdot \sigma \tag{1.6}$$

Equation 1.6 is a vector equation valid for each component of $\mathbf{u}$. The quantity $\nabla\mathbf{u}$ is a second order tensor interpreted as the matrix

$$\nabla \mathbf{u} = \begin{pmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{pmatrix}$$

It is worth recalling that the left side of Equation 1.6 is the *Eulerian* derivative of $\mathbf{u}$ and is equivalent to the total (*Lagrangian*) derivative $d\mathbf{u}/dt$ in a coordinate system moving with the fluid particle of velocity $\mathbf{u}$.

Equation 1.6 contains the stress tensor and is not in a useful form since σ is not easily measurable. For a fluid medium, the consequence of applying a stress distribution is the occurrence of a finite strain rate and hence a flow. The rate-of-strain tensor ε can be defined as (Farrashkhalvat and Miles, 1990)

$$\varepsilon = \frac{1}{2}[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^T]$$

It represents all modes of deformation of a fluid element and transforms as a second order tensor. Further, it shares the following properties with the stress tensor σ.

1. Trace(ε) $= u_x + v_y + w_z$ ($= \nabla\cdot\ \mathbf{u}$) is invariant to coordinate rotation. Trace (σ) is also an invariant and is equated to $-3p$ where p is called the static pressure in the fluid.

2. Both ε and σ are symmetric. This condition may not be satisfied in fluids composed of long-chain molecules. Such fluids (called *micropolar*) are excluded from the present discussion.

It now remains to connect the stress tensor to the rate-of-strain tensor. In general, this connection depends on the material being studied. For a large class of fluids the functional form of the stress - strain rate relation remains the same, the differences arising purely from empirical coefficients whose magnitude depends only on the choice of the fluid medium. Commonly encountered fluids such as air, water and oils are homogeneous and isotropic. Laboratory experiments show that their σ-ε relation is linear. Such fluids are called *Newtonian*. The most general linear relationship between two second order tensors representing the external force and the material response for a homogeneous, isotropic fluid can be shown to be of the form (Farrashkhalvat and Miles, 1990):

$$\sigma_{ij} = (-p_d + \lambda\varepsilon_{kk})\,\delta_{ij} + 2\mu\varepsilon_{ij} \tag{1.7}$$

The Cartesian tensor notation has been used here to represent σ and ε. The trace of ϵ is ε_{kk} and is equal to $(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) = \nabla\cdot\mathbf{u}$. Further, λ and μ are fluid properties and δ_{ij} is the Kroncker delta (=1 if $i = j$ and 0 otherwise). In the absence of flow and body forces $\varepsilon = 0$ and the stress field in the fluid region reduces to the thermodynamic pressure p_d, i.e.

$$\text{if } \mathbf{u} = 0 \text{ and } \chi = 0 \text{ then } \sigma = -\begin{bmatrix} p_d & 0 & 0 \\ 0 & p_d & 0 \\ 0 & 0 & p_d \end{bmatrix}$$

While λ and μ are almost independent of σ and ε, experiments show that they are weak functions of pressure and a strong function of temperature. Computing the trace of σ from Equation 1.7 and equating it to $-3p$ yields

$$p_d - p = (\lambda + \frac{2}{3}\mu)\nabla\cdot\mathbf{u}$$

Hence the difference between the thermodynamic and static pressures vanishes either when $\mathbf{u} = 0$ or $\nabla\cdot\,\mathbf{u} = 0$. The latter is the condition of incompressibility of the fluid. An unusual result obtained on the basis of kinetic theory of gases states that for a wide class of fluids, the quantity $(3\lambda + 2\mu)$, the bulk viscosity is itself quite small. This is called *Stokes hypothesis*. Hence the stress-strain rate relation (Equation 1.7) contains only one fluid property, namely λ or μ. The ratio $\nu = \mu/\rho$ is called kinematic viscosity. It is quite common to work with μ; it is called as the dynamic viscosity of the fluid. Equation 1.7 can be rewritten as

$$\sigma_{ij} = -p\delta_{ij} + \mu(2\varepsilon_{ij} - \frac{2}{3}\nabla\cdot\mathbf{u}\delta_{ij}) \tag{1.8}$$

where p $(= -p_d)$ now refers to both hydrostatic and thermodynamic pressures.

The constitutive relationship given by Equation 1.8 must be qualified by the constraint of the second law of thermodynamics. This requires that the rate of dissipation of kinetic energy to thermal energy by internal friction (and hence the entropy generation rate) be a positive quantity. For an incompressible fluid the viscous dissipation per unit volume is $1/2\mu\left(\partial u_i/\partial x_j + \partial u_j/\partial x_i\right)^2$, where x_i and u_i are the components of $\mathbf{x}$ and $\mathbf{u}$ respectively and a repeated index implies summation over the index. Hence the required constraint of the second law of thermodynamics is $\mu > 0$.

The equation of motion in terms of measurable quantities is finally obtained for a Newtonian fluid by substituting Equation 1.8 into Equation 1.6. This leads to

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = \chi - \nabla p + 2\nabla \cdot [\mu(\varepsilon_{ij} - \frac{1}{3}\nabla \cdot \mathbf{u}\delta_{ij})] \tag{1.9}$$

Equation 1.9 is a vector equation for the velocity field $\mathbf{u}$. For historical reasons it goes by the name *Navier-Stokes* equation. The term *momentum* equation is also used in this context. In words the equation reads as

Mass × Accleration = Body force + Pressure force + Viscous force

To expand the system of governing equations in any coordinate system it is simpler to focus attention on a special class of problems where μ and ρ are spatially constant. Constancy of μ requires that the variations in temperature by limited; constancy of ρ requires the fluid to be incompressible and hence $\nabla \cdot \mathbf{u} = 0$. Under these conditions, Equation 1.9 reduces to

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = \chi - \nabla p + \mu \nabla^2 \mathbf{u} \tag{1.10}$$

When $\nabla \cdot \mathbf{u} = 0$ the left hand side of Equation 1.9 can also be written as

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}))$$

It is referred to as the *conservative form* and has several implications from the view point of numerical methods.

1.4 Expanded Forms of Navier-Stokes Equations

Let $\chi = (X, Y, Z)$ be the orthogonal components of the body force field in the Cartesian coordinate system; it is suitably defined in curvilinear coordinates as well. Equation 1.10 in various coordinate systems (Figure 1.2) is given below.

Cartesian coordinates (x, y, z):

$$x: \quad \rho(\frac{\partial u}{\partial t} + uu_x + vu_y + wu_z) = X - p_x + \mu(u_{xx} + u_{yy} + u_{zz})$$

$$y: \quad \rho(\frac{\partial v}{\partial t} + uv_x + vv_y + wv_z) = Y - p_y + \mu(v_{xx} + v_{yy} + v_{zz})$$

$$z: \quad \rho(\frac{\partial w}{\partial t} + uw_x + vw_y + ww_z) = Z - p_z + \mu(w_{xx} + w_{yy} + w_{zz}) \qquad (1.11)$$

Cylindrical coordinates (r, θ, z):

$$r: \quad \rho\left(\frac{du}{dt} - \frac{v^2}{r}\right) = \chi_r - p_r + \mu\left(\nabla^2 u - \frac{2}{r^2}v_\theta - \frac{u}{r^2}\right)$$

$$\theta: \quad \rho\left(\frac{dv}{dt} + \frac{uv}{r}\right) = \chi_\theta - \frac{1}{r}p_\theta + \mu\left(\nabla^2 v + \frac{2}{r^2}u_\theta - \frac{v}{r^2}\right)$$

$$z: \qquad \rho\frac{dw}{dt} = \chi_z - p_z + \mu\nabla^2 w \qquad (1.12)$$

Here

$$\begin{aligned} \frac{du}{dt} &= \frac{\partial u}{\partial t} + uu_r + \frac{v}{r}u_\theta + wu_z \\ \nabla^2 u &= u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} + u_{zz} \end{aligned}$$

Spherical coordinates (r, ϕ, θ):

$$\begin{aligned} r: \quad \rho\left(\frac{du}{dt} - \frac{v^2 + w^2}{r}\right) &= \chi_r - p_r \\ &+ \mu\left(\nabla^2 u - \frac{2}{r^2}\left(u + v_\phi + v\cot\phi + \frac{1}{\sin\phi}w_\theta\right)\right) \end{aligned}$$

$$\begin{aligned} \phi: \quad \rho\left(\frac{dv}{dt} + \frac{uv}{r} - \frac{w^2\cot\phi}{r}\right) &= \chi_\phi - \frac{1}{r}p_\phi \\ &+ \mu\left(\nabla^2 v + \frac{2}{r^2}\left(u_\phi - \frac{1}{2\sin^2\phi}v - \frac{\cot\phi}{\sin\phi}w_\theta\right)\right) \end{aligned}$$

$$\begin{aligned} \theta: \qquad \rho\left(\frac{dw}{dt} + \frac{uw}{r} + \frac{vw\cot\phi}{r}\right) &= \chi_\theta - \frac{1}{r\sin\phi}p_\theta \\ + \quad \mu\left(\nabla^2 w - \frac{2}{r^2}\left(\frac{w}{2\sin^2\phi} - \frac{1}{\sin\phi}u_\theta - \frac{\cot\phi}{\sin\phi}v_\theta\right)\right) \qquad (1.13) \end{aligned}$$

Here

$$\begin{aligned}\frac{du}{dt} &= \frac{\partial u}{\partial t} + uu_r + \frac{v}{r}u_\phi + \frac{w}{r\sin\phi}u_\theta \\ \nabla^2 u &= \frac{1}{r^2}\left((r^2u_r)_r + \frac{1}{\sin\phi}(\sin\phi u_\phi)_\phi + \frac{1}{\sin^2\phi}u_{\theta\theta}\right)\end{aligned}$$

The determination of the solution of the Navier-Stokes equations even by approximate means is quite involved. Depending on the problem being solved it is convenient to work with a simpler form of these equations obtained after dropping terms that are insignificant. Special forms of these equations are given in Section 1.6.

1.5 Conservation of Energy Principle

In many flow problems it is of interest to calculate heat transfer rates from the fluid to a surface or from a hot surface to a cold fluid. To this end, the differential form of the heat transfer equation can be utilized. In the Reynolds Transport Theorem we identify ϕ as the specific total energy $e = h + u^2/2 + gz$. The total energy is taken to be composed of enthalpy h, the kinetic energy $u^2/2$ and potential energy gz arising from gravitation. In the presence of other body force fields the potential energy contribution will have to be suitably modified. The generation rate of the total energy Φ of the matter contained in the control volume ($= d\Phi/dt$) consists of contributions from energy generation throughout the control volume, say due to chemical reactions and energetic interactions in the form of heat and work transferred at the boundary of the control volume. Let Q be the energy released locally per unit volume and per unit time. The first part of $d\Phi/dt$ is $\int_{CV} Q d\mathbf{x}$. The second part of the generation term consists of the following:
(i) Energy lost to the surroundings by mechanisms such as conduction and radiation

$$= -\int_{CS} \mathbf{q}\cdot\mathbf{n}ds$$

The negative sign is consistent with the sign convention of thermodynamics where heat added to the matter contained in CV is positive and the work done by the fluid medium is also positive.
(ii) Mechanical work done on the control surface by the stress field

$$= -\int_{CS} \mathbf{u}\cdot(\sigma\cdot\mathbf{n}ds)$$

The quantity $\sigma\cdot\mathbf{n}ds$ is the traction vector on the control surface and its product with $\mathbf{u}$ represents the rate at which external work is done. Hence

$$\begin{aligned}\frac{d\Phi}{dt} = & \int_{CV} Q d\mathbf{x} - \int_{CS} \mathbf{q}\cdot\mathbf{n} ds + \int_{CS} \mathbf{u}\cdot(\sigma\cdot\mathbf{n}) ds \\ & = \int_{CV} (Q - \nabla\cdot\mathbf{q} + \nabla\cdot(\mathbf{u}\cdot\sigma)) d\mathbf{x}\end{aligned}$$

Using Reynolds Transport Theorem, this is equated to

$$\int_{CV} \left(\frac{\partial(\rho e)}{\partial t} + \nabla\cdot\rho e\mathbf{u} \right) d\mathbf{x}$$

The equality must hold for arbitrary choices of CV. Hence the respective integrands can be equated as

$$\frac{\partial(\rho e)}{\partial t} + \nabla\cdot\rho e\mathbf{u} = Q - \nabla\cdot\mathbf{q} + \nabla(\mathbf{u}\cdot\sigma)$$

Subtracting Equation 1.2 from the above equation, we get

$$\rho\left(\frac{\partial e}{\partial t} + \mathbf{u}\cdot\nabla e\right) = Q - \nabla\cdot\mathbf{q} + \nabla\cdot(\mathbf{u}\cdot\sigma) \tag{1.14}$$

This is known as the total energy equation in terms of e. Subtracting the dot product $\mathbf{u}\cdot$ [Equation 1.10] from Equation 1.14 yields an equation entirely for enthalpy as

$$\rho\left(\frac{\partial h}{\partial t} + \mathbf{u}\cdot\nabla h\right) = Q - \nabla\cdot\mathbf{q} + \nabla\mathbf{u} : \sigma \tag{1.15}$$

This is called the thermal energy equation. Here we have used the tensor identity

$$\nabla\cdot(\mathbf{u}\cdot\sigma) - \mathbf{u}\cdot(\nabla\cdot\sigma) = \nabla\mathbf{u} : \sigma$$

The double dot product is evaluated as the sum $\sum_i \sum_j \ \partial u_j/\partial x_i \ \sigma_{ij}, \ i,j =$ 1,2,3. In nearly parallel flows it gives rise to the mechanical work done of the form pu_x and viscous dissipation term $\mu(u_y)^2$ where x is a predominant flow direction whose velocity component is u and y is a direction in which u changes rapidly. In most incompressible flow applications these terms are quite small and can be ignored.

The thermal energy equation in terms of enthalpy needs to be simplified to a form which contains measurable quantities such as temperature. The energy generation term Q is assumed to be specified independently. Radiation is important in combustion-related problems as in furnaces and IC engines. In the present development, the contribution of radiation to the heat flux $\mathbf{q}$ is assumed to be small so that $\mathbf{q}$ is primarily a conductive heat flux. This is reasonable if temperature levels in the fluid are moderate. The conductive heat flux is modelled by Fourier's law, $\mathbf{q} = -k\nabla T$, where k is the fluid thermal conductivity. This is only a working relationship for $\mathbf{q}$ since k in turn can

depend on temperature T. This dependence is however not strong for many fluids. Enthalpy in general is of the form $h = h(T,p)$ with a strong depedence on temperature T and a weak dependence on pressure p. We define a specific heat $C_p = \partial h/\partial T\ |_p$, evaluated at an average pressure in the flow. Equation 1.15 can now written in terms of temperature as the primary dependent variable in the form:

$$\rho C_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = \nabla \cdot k \nabla T + Q \tag{1.16}$$

Here ρ,C_p and k are fluid properties that must be determined independently from laboratory experiments. If these properties were to be sensibly constant over the temperature range of interest and if $Q = 0$ we get

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \alpha \nabla^2 T \tag{1.17}$$

where α $(= k/\rho C_p)$ is called the thermal diffusivity. The second term on the left side of Equation 1.17 represents advective transport of energy by fluid motion. The right side represents diffusive transport of energy across a temperature gradient. For this reason, Equation 1.17 is often called the *advection-diffusion* equation for thermal energy. The expanded forms of the thermal energy equation in the three coordinate systems are:

$$\text{Cartesian:} \quad \frac{\partial T}{\partial t} + uT_x + vT_y + wT_z = \alpha(T_{xx} + T_{yy} + T_{zz})$$

$$\text{Cylindrical:} \quad \frac{\partial T}{\partial t} + uT_r + \frac{v}{r}T_\theta + wT_z = \alpha(T_{rr} + \frac{1}{r}T_r + \frac{1}{r^2}T_{\theta\theta} + T_{zz})$$

$$\begin{aligned} \text{Spherical:} \quad \frac{\partial T}{\partial t} + uT_r + \frac{v}{r}T_\phi + \frac{w}{r\sin\phi}T_\theta &= \frac{\alpha}{r^2}((r^2T_r)_r \\ + \frac{1}{\sin\phi}(\sin\phi T_\phi)_\phi + \frac{1}{\sin^2\phi}T_{\theta\theta}) & \end{aligned} \tag{1.18}$$

Heat conduction in a solid region involves mechanisms of diffusion and energy generation but no advective transport since $\mathbf{u} = 0$. It is hence governed by the equation

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T + \frac{Q}{\rho C_p} \tag{1.19}$$

Solids can, however, be anisotropic and inhomogeneous and in such cases Fourier's law of heat conduction must be generalized as $\mathbf{q} = -\mathbf{k}(\mathbf{x}) \cdot \nabla T$. Here $\mathbf{k}$ is the thermal conductivity tensor. For a majority of known materials this tensor is symmetric. In two dimensions

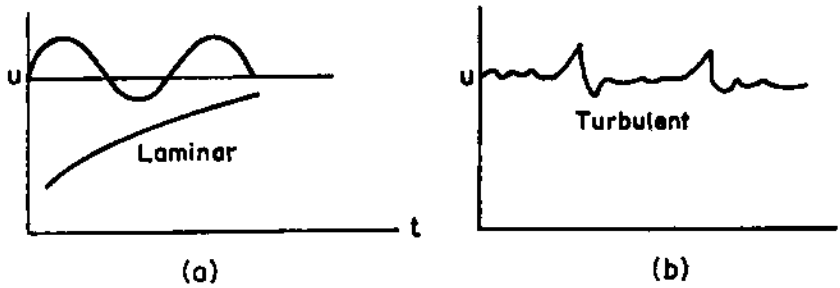


Figure 1.4: Instantaneous Velocity Signals in Laminar and Turbulent Flows

$$\mathbf{k} = \begin{pmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{pmatrix}$$

The heat conduction equation for an anisotropic, inhomogeneous solid in two dimensional Cartesian coordinates is

$$\rho C_p \frac{\partial T}{\partial t} = (k_{11}T_x)_x + (k_{12}T_y)_x + (k_{12}T_x)_y + (k_{22}T_y)_y + Q \tag{1.20}$$

1.6 Special Forms of the Navier-Stokes Equations

The Navier-Stokes equations represent every state of Newtonian fluid flow. However, at large Reynolds numbers (Re>> 1) certain regions of the flow can become unstable and reveal fluctuations in velocity. Under these conditions the governing equations themselves become sensitive to small perturbations. The equations are then said to exhibit a discontinuous dependence on data. A detailed discussion on the transition of steady flow towards instability is beyond the scope of this book.

For the present, it is sufficient to state that experiments reveal two distinct states of flow (Figure 1.4). A well-behaved state of flow is called *laminar*. A velocity measuring probe inserted in such a flow will reveal a steady, periodic or a transient flow pattern (Figure 1.4a). A random chaotic state of flow is said to be *turbulent* (Figure 1.4b).

It is conventional to study each class of flows separately since the approximation and techniques used in each are distinct. Hence the state of flow as laminar or turbulent must be assumed before hand. In this book flow is always considered laminar unless specified explicitly as turbulent. A unified analysis of the Navier-Stokes equations where the state of flow as laminar or turbulent would appear as a solution of the problem remains incomplete at this time.

1.6.1 Steady Flow

When the prime mover sustaining the flow is in steady operation the velocity, pressure and temperature measured at a point will attain a constant value in time. This is true despite the fact that a fluid particle moving through a physical region will acquire different velocities and temperatures at various instants of time. Hence at steady state, the partial derivatives $\partial u/\partial t$, $\partial T/\partial t$ and $\partial \rho/\partial t$ at a point are all zero.

1.6.2 Dimensionless Form of Momentum and Energy Equations

Let $U, \Delta T, L$ and ρ_0 be the characteristic velocity,temperature difference, length and density scales applicable for the problem being studied. Let $\mathbf{u}$, T, p, t and $\mathbf{x}$ be the dimensionless form of velocity, temperature difference, pressure,time and distance scaled by $U, \Delta T$, $\rho_0 u^2$,L/U and L respectively. In coordinate-free form the mass balance, momentum and energy equations for an incompressible flow reduce to:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= \chi - \nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u} \\ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T &= \frac{1}{\mathrm{Pe}} \nabla^2 T + Q \end{aligned} \tag{1.21}$$

In Equation 1.21, χ is the dimensionless body force vector, Q the dimensionless rate of energy generation, Re the Reynolds number $\rho_0 U L/\mu$ and Pe, the Peclet number UL/α.

In problems where the body force term is not important, the Reynolds number appears as a single dimensionless parameter that characterizes flow fields with varying velocity and length scales. Hence universal solutions can be obtained for prescribed flow geometries by varying Re alone. Specific solutions can then be extracted by identifying the scales prevalent in the problem of interest. Reynolds number has a physical interpretation of being the ratio of inertia (I) and viscous (V) forces acting on a fluid element. Hence

$$\mathrm{Re} \gg 1, I \gg V; \;\; \mathrm{Re} \ll 1, I \ll V \;\; \text{and} \;\; \mathrm{Re} \sim 1, I \sim V$$

This result can be used to estimate the orders-of-magnitude of various terms in the Navier-Stokes equations.

When heat generation is unimportant the Peclet number appears as a single dimensionless parameter that describes heat transfer in problems with varying length and velocity scales. It is interpreted as the ratio of the advective thermal energy (A) and the diffusive thermal energy (D) transferred in the fluid medium. Hence we have

$$\mathrm{Pe} \gg 1, A \gg D; \;\; \mathrm{Pe} \ll 1, A \ll D; \;\; \text{and } \mathrm{Pe} \sim 1, A \sim D$$

These estimates can be used to simplify the energy equation where appropriate.

1.6.3 Stokes Equations

If Re< 1, the inertia terms in Navier-Stokes equations can be dropped in comparison to the viscous terms. Then the momentum equation reduces to

$$\rho\frac{\partial \mathbf{u}}{\partial t} = \chi - \nabla p + \mu\nabla^2\mathbf{u} \tag{1.22}$$

These are known as Stokes equations. The Reynolds number can be small when the characteristic velocity or the length scale is small or when the viscosity is large. Problems that involve small Reynolds numbers are called *creeping flows*.

1.6.4 Euler Equations

If Re approaches infinity, the viscous terms can be dropped from the Navier-Stokes equations. The equations resulting from this approximation are called Euler equations and are given as

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}) = \chi - \nabla p \tag{1.23}$$

1.6.5 Conservative Body Force Fields

A force field is said to be *conservative* if it can be expressed as the gradient of a scalar potential. The work done by such a force in a closed path is zero. Typically, frictionless forces arising from gravitation and electromagnetism are conservative. In such cases we write, $\chi = -\nabla\phi$, where ϕ is a scalar potential. For example, in a gravity field $\chi = -\rho g\mathbf{k}$ and ϕ is obtained on integration as $\phi = \rho g z$ upto an additive constant. Here z is the vertical coordinate direction. For conservative body force fields χ can be merged with the pressure gradient as

$$-\nabla p + \chi = -\nabla(p + \phi) = -\nabla\overline{p}$$

where $\overline{p}$ is a modified pressure. Hence the momentum equation becomes

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla\overline{p} + \mu\nabla^2\mathbf{u} \tag{1.24}$$

For most applications studied in succeeding chapters, gravity is the only force field. Further, the extent of z in the vertical direction is usually small and p and $\overline{p}$ are quite close to each other.

1.6.6 Stream Function-Vorticity Formulation

If the flow field is incompressible and two dimensional (or axisymmetric), the system of two momentum equations and one continuity equation can be reduced to two equations in the following manner. Define a stream function ψ as, $u = \psi_y$ and $v = -\psi_x$. Then $\nabla \cdot \mathbf{u} = \psi_{yx} - \psi_{xy} = 0$, identically. The vorticity vector ω is defined as $\nabla \times \mathbf{u}$ and it represents the net rotation of an infinitesimal fluid element. The vorticity component normal to the plane of the flow can be shown to be

$$\omega = (\nabla \times \mathbf{u}) \cdot \mathbf{k} = v_x - u_y$$

The equation relating ψ and ω is obtained from the definition of vorticity as $\omega = -(\psi_{xx} + \psi_{yy}) = -\nabla^2\psi$. The equation governing the transport of vorticity is obtained from the momentum equation by eliminating pressure. Pressure can be eliminated from Equation 1.24 by applying the curl ($\nabla\times$) operator throughout. With $\mathbf{k}$ representing unit normal defining the plane of the flow we get

$$\mathbf{k} \cdot \left[\nabla \times \rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla \overline{p} + \mu \nabla^2 \mathbf{u}\right]$$

that is

$$\rho(\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega) = \mu \nabla^2 \omega \tag{1.25}$$

Equation 1.25 along with the statement, $\omega = -\nabla^2\psi$ provides a complete equivalent description of the flow field. From the definition of vorticity it is clear that ω is small in undisturbed regions of flow and quite large in the neighbourhood of solid walls where large changes in velocity take place. The physical picture that emerges from this is that vorticity produced in the vicinity of solid walls is transported in a manner analogous to energy transfer from a heated surface. The transport coefficient that plays the role of thermal diffusivity is the quantity ν $(= \mu/\rho)$ called the kinematic viscosity. With the knowledge of ψ and ω, the velocity field is constructed by differentiating ψ. Pressure is recovered from Poisson equation of pressure derived by taking the divergence of equation 1.24 as follows:

$$\nabla \cdot \left[\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla \overline{p} + \mu \nabla^2 \mathbf{u}\right]$$

leading to

$$\nabla^2 \overline{p} = -\rho \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \tag{1.26}$$

where the incompressibility condition $\nabla \cdot \mathbf{u} = 0$ has been used.

1.6.7 Potential Theory

Equations governing two dimensional inviscid, incompressible, irrotational flow past an object can be solved by the method of potentials. Consider the vorticity equation (Equation 1.25) at the limit of $\mu \to 0$:

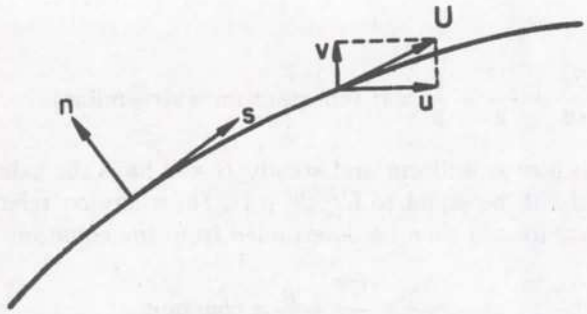


Figure 1.5: Coordinate Directions along a Streamline

$$\rho \frac{d\omega}{dt} = \rho(\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega) = 0$$

The first terms shows that the vorticity of a fluid element as it moves along a streamline is preserved in the absence of viscosity. If the approach flow is uniform, the vorticity is zero at that location. In the inviscid limit this requires that vorticity be zero everywhere including the proximity of the solid object. Hence the flow field satisfies the equation $\omega = 0$; such flows are called *irrotational*. Since $\omega = \nabla \times \mathbf{u}$, the choice of $\mathbf{u} = \nabla \phi$ automatically satisfies the requirement of zero vorticity in the flow field. The function ϕ is called the velocity potential and serves as a new scalar dependent variable for the flow problem. The equation governing ϕ is obtained from the continuity equation as

$$\nabla \cdot \mathbf{u} = \nabla^2 \phi = 0 \tag{1.27}$$

Since the above equation has been derived at the limit of $\mu \to 0$ (Re$\to \infty$), it is equivalent to Euler equations subject to the constraint of incompressibility. Once ϕ is determined the velocity field is obtained by differentiation. The pressure field is determined as follows.

Let (s, n) be a local coordinate system drawn at any point along a streamline (Figure 1.5). With respect to local coordinate system the velocity components are $(U, 0)$ where $U = \sqrt{u^2 + v^2}$ and $\nabla = (1/h_1 \ \partial/\partial s, 1/h_2 \ \partial/\partial n)$ is the gradient operator. Here h_1 and h_2 are metric coefficients and $h_1 \Delta s$ and $h_2 \Delta n$ represent differential arc lengths in the respective directions. Equation 1.23 can now be written as

$$\frac{\rho}{h_1} \frac{\partial}{\partial s} \left(\frac{\partial \phi}{\partial t} + \frac{U^2}{2} + \frac{\bar{p}}{\rho} \right) = 0$$

Integrating along the streamline and assuming gravitational effects to be small we get

አዳማ ዩኒቨርሲቲ
Adama University
ቤተመፅሐፍት
Library

$$\frac{\partial \phi}{\partial t} + \frac{U^2}{2} + \frac{p}{\rho} = B \text{ (constant on a streamline)}$$

If the approach flow is uniform and steady B will have the same value for all streamlines and will be equal to $U_\infty^2/2 + p_\infty/\rho$, where ∞ refers to upstream conditions. Pressure can then be determined from the equation

$$\frac{\partial \phi}{\partial t} + \frac{U^2}{2} + \frac{p}{\rho} = \text{constant} \tag{1.28}$$

Equation 1.28 is known as *Bernoulli's equation.* If the approach flow is uniform but unsteady, the right hand side is replaced by a function of time that is also evaluated at upstream conditions.

1.6.8 Boundary-layer Theory

Even if Re$\gg$ 1, it is possible that inertia (I) and viscous (V) forces are of comparable magnitude at some locations, for example in the neighbourhood of a solid boundary. On a stationary wall the fluid velocity is zero and hence has zero acceleration. Viscous shear stresses are, however, non-zero at the wall and we estimate $I \ll V$ at this location. Far away from the wall, the condition of Re$\gg$ 1 guarantees $I \gg V$. There is an intermediate region above the solid wall where inertia and viscous forces are of comparable magnitude. The size of this region can be estimated from Equation 1.11 by assigning a length scale L to x, δ to y and U to the velocity component u. Using the notation O for an order-of-magnitude estimate we get

$$I \sim V, \quad O(\rho u u_x) = O(\mu u_{yy})$$

and so

$$\frac{\delta}{L} \sim \frac{1}{\sqrt{\text{Re}}}$$

where Re $= \rho U L/\mu$. Since Re is large, δ, the extent over which inertia and viscous forces are of the same order is small in comparison to L. The quantity δ is called the hydrodynamic boundary-layer thickness. When Re $\to \infty$, $\delta \to 0$ and viscous forces are absent everywhere in the flow. Potential theory described in section 1.6.7 is applicable at this limit.

Equations governing flow within a boundary-layer are of intermediate complexity when compared to potential flows on one hand and the full Navier-Stokes equations on the other. They are of fundamental importance in the study of fluid mechanics and heat transfer since they predict non-zero drag on a body placed in flow and onset of separation and turbulence. Boundary-layer equations are derived by using the fact that at high Reynolds numbers, $\delta/L \ll 1$, i.e. the boundary-layer is thin. As a consequence one obtains the following *boundary-layer approximations:*

1. $|u| \gg |v|$, where u is the velocity component in the predominant flow direction aligned with x coordinate and having a length scale L and v is normal to it, aligned with the y - coordinate and having a length scale δ.

2. $\mathbf{u}_{xx} \ll \mathbf{u}_{yy}$ and hence $\nabla^2\mathbf{u}$ can be approximated as $\mathbf{u}_{yy}$.

3. The normal pressure gradient within a boundary-layer ($= \partial p/\partial y$) is small and so pressure can be taken to be uniform in the y -direction. Pressure is then said to be impressed on the boundary-layer.

Since pressure within the boundary-layer depends only on x it can be computed at the edge of the boundary-layer where the flow may be treated as inviscid. For steady incompressible flows $p(x)$ is determined from Bernoulli's equation as

$$\frac{p}{\rho} + \frac{U^2}{2} = \text{constant} \quad \text{and so} \quad -p_x = \rho U U_x$$

where U is the potential velocity for the geometry being studied. Since δ is small U can be computed at the wall itself, namely $y = 0$.

When results 1-3 are applied to the Navier-Stokes equations for a two dimensional steady incompressible boundary-layer problem, the following system of equations is generated:

$$\begin{aligned} u_x + v_y &= 0 \\ \rho(uu_x + vu_y) &= UU_x + \mu u_{yy} \end{aligned} \tag{1.29}$$

The dependent variables in Equation 1.29 are u and v; $U(x)$ is the potential wall velocity and pressure is determined from Bernoulli's equation.

In high Peclet number problems where heating is applied at the wall, a thermal boundary-layer of thickness δ_T is observed. In the free stream, energy is transferred predominantly by advection while at the wall it is by diffusion alone. Within this layer advective and diffusive energy transfer rates are comparable. Using the order-of-magnitude analysis δ_T can be estimated as $\delta_T/L \sim 1/\sqrt{\text{Pe}}$ and hence $\delta_T/L \ll 1$. Hence the thermal boundary-layer is also thin. The ratio $\delta/\delta_T = \sqrt{\nu/\alpha}$ where the ratio ν/α is called the Prandtl number (Pr) and is a fluid property. For air Pr~ 0.7 and hence $\delta \sim \delta_T$; for water Pr~ 5 and $\delta > \delta_T$; for liquid metals Pr ~ 0.01 and $\delta \ll \delta_T$.

The consequence of a large Peclet number is the formation of a thin thermal boundary-layer within which $T_{xx} \ll T_{yy}$. Hence the unsteady energy transport equation (Equation 1.17) reduces to

$$\frac{\partial T}{\partial t} + uT_x + vT_y = \alpha T_{yy} \tag{1.30}$$

Hydrodynamic and thermal boundary-layers over a heated flat plate are sketched in Figure 1.6.

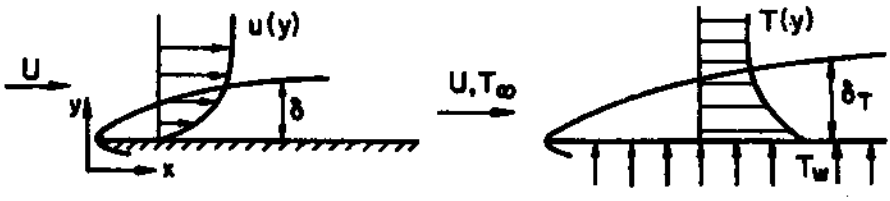


Figure 1.6: Velocity and Temperature Profiles In a Boundary-layer

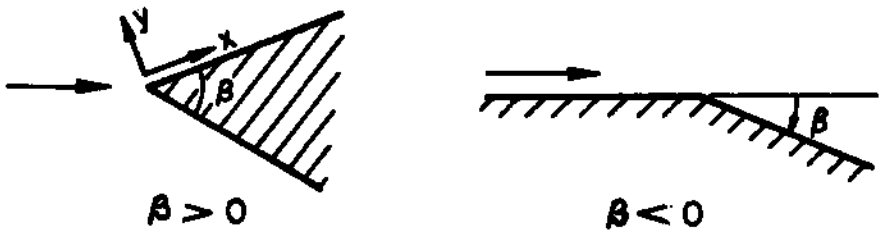


Figure 1.7: Flow Past a Wedge-shaped Surface

For a class of problems called *wedge flows*, the potential velocity at the wall is given as $U \sim x^m$ (Figure 1.7). Here m is related to the wedge angle β as

$$m = \frac{\beta/\pi}{2 - \beta/\pi}.$$

For a flat plate held parallel to the flow $m = \beta = 0$.

Meaningful values of the potential velocity are generated only for $m \leq 1$. For $m = -0.09$ the flow separates from the wall and thin boundary-layer approximations are no longer valid. Hence m is restricted to $1 \geq m \geq -0.09$ and β to, $\pi \geq \beta \geq -0.625$ radians. For this choice of $U(x)$ Equation 1.29 can be reduced to a single third order ordinary differential equation with dimensionless stream function as the dependent variable and a composite quantity $\eta = y\sqrt{U/\nu x}$ as the independent (*similarity*) variable (White, 1991). This equation is

$$f''' + \left(\frac{m+1}{2}\right) ff' + m(1 - (f')^2) = 0 \tag{1.31}$$

Here $f(\eta)$ is the dimensionless stream function constructed in such a way that $f' = df/d\eta = u(x,y)/U(x)$. $f''(\eta)$ is the dimensionless shear stress and is a quantity of great engineering importance. $f''(0) = 0$ represents the point of separation on the object surface. early separation is indicative of a large increase in form drag. When $f'''(\eta) = 0$ for a given value of η the velocity profile is said to develop an inflexion point and this marks the onset of turbulence within the

boundary-layer. Equation 1.31 is solved subject to the no-slip wall conditions and potential far stream conditions, i.e. at $\eta = 0$, $f = f' = 0$ and as $\eta \to \infty$, $f' \to 1$.

Consider the problem of steady flow over a wedge-shaped surface kept at a constant temperature T_w. The fluid approaches the surface with a uniform temperature T_∞. Let $T(\mathbf{x})$ be the temperature distribution in the thermal boundary-layer. Normalizing T as $\theta = (T(\mathbf{x}) - T_\infty)/(T_w - T_\infty)$, it can be shown that $\theta = \theta(\eta)$, that is, a function of the similarity variable alone. In terms of θ, Equation 1.30 reduces to the following ordinary differential equation

$$\theta'' + \frac{\mathrm{Pr}}{2} f\theta' = 0 \tag{1.32}$$

Here $f(\eta)$ is taken to be known from the solution of Equation 1.31. The boundary conditions for θ are $\theta(0) = 1$ and $\theta(\infty) = 0$. The derivative $-\theta'(0)$ is the dimensionless wall heat flux.

Boundary-layer theory is valid only over those portions of the flow where $\delta/x \ll 1$. Clearly this condition is violated when $x \to 0$. Errors from this discrepency near the forward stagnation point of the object are small and are usually neglected. The theory also breaks down beyond the point of flow separation. This is because of a large increase in δ and the inapplicability of the assumption of a thin boundary-layer.

1.6.9 Buoyancy-driven Convection

When a heated surface is immersed in a still fluid, convection currents are set in. The origin of fluid movement in such problems is related to variations in density of the fluid as one moves from the hot surface to the cooler ambient. The flow pattern that is generated by a density field in the presence of gravity is called *buoyancy-driven convection* and more commonly, *natural convection*. Atmospheric flows, platetectonic movement and cooling of electronic equipment are examples where the knowledge of natural convection is important. Variations in density lead to fluid motion through buoyancy forces that develop in the heated region. If $\rho(\mathbf{x})$ is the local density, the body force per unit volume is $\rho g\mathbf{k}$, where $\mathbf{k}$ is the unit vector in the vertical direction. The body force term cannot be absorbed with the pressure gradient since the density ρ is not a constant. However, it is quite common to assume ρ to be a constant (and equal to an average value prevailing in the flow domain) as far as other factors such as acceleration, mass and thermal capacity effects are concerned. This is called *Boussinesq approximation*. For moderate variations in temperature, the functional form relating density and temperature can be taken as linear, i.e.

$$\rho = \rho_\infty(1 - \beta(T - T_\infty))$$

Here β is the volumetric coefficient of expansion defined as $(1/\rho)\partial\rho/\partial T\,|_p$. For ideal gases $\beta = 1/T$(absolute) while for liquids its value must be taken from handbooks. The mass, momentum and energy equations are fully coupled for

Figure 1.8: Buoyancy-driven Flows: a. Internal b. External.

buoyancy-driven flows. In two dimensional Cartesian coordinates, the expanded form of Equations 1.5, 1.11 and 1.18 for a natural convection problem are obtained as follows (Figure 1.8):

$$\begin{aligned}
u_x + v_y &= 0 \\
\rho_\infty(\frac{\partial u}{\partial t} + uu_x + vu_y) &= -\bar{p}_x + \mu(u_{xx} + u_{yy}) \\
\rho_\infty(\frac{\partial v}{\partial t} + uv_x + vv_y) &= -\bar{p}_y + \mu(v_{xx} + v_{yy}) + \rho_\infty g\beta(T - T_\infty) \\
\rho_\infty c_p(\frac{\partial T}{\partial t} + uT_x + vT_y) &= k(T_{xx} + T_{yy}) \qquad (1.33)
\end{aligned}$$

Here u and v are velocity components in x and y directions respectively, T_∞ and ρ_∞ are reference temperature and density respectively and $\bar{p} = p - \rho_\infty gy$. In internal flows (Figure 1.8a) $\bar{p}$ is not a constant and is an additional dependent variable along with u, v and T. In problems where one is interested in wall heat transfer rates it is convenient to eliminate pressure from Equation 1.33 using the stream function (ψ) – vorticity(ω) formulation. Defining $u = \psi_y$, $v = -\psi_x$ and $\omega = v_x - u_y$, Equation 1.33 can be reduced to a system of three coupled equations:

$$\begin{aligned}
\psi_{xx} + \psi_{yy} &= -\omega \\
\rho_\infty(\frac{\partial \omega}{\partial t} + u\omega_x + v\omega_y) &= \mu(\omega_{xx} + \omega_{yy}) + \rho_\infty g\beta T_x \\
\rho_\infty c_p(\frac{\partial T}{\partial t} + uT_x + vT_y) &= k(T_{xx} + T_{yy}) \qquad (1.34)
\end{aligned}$$

Using a length scale L, a velocity scale ν/L, a time scale L^2/ν and a characteristic temperature difference $(T_w - T_\infty)$ (or $T_h - T_c$ in Figure 1.8a), Equations 1.34 can be cast in the following dimensionless form

$$\begin{aligned}
\psi_{xx} + \psi_{yy} &= -\omega \\
\frac{\partial \omega}{\partial t} + u\omega_x + v\omega_y &= \omega_{xx} + \omega_{yy} + \frac{\mathrm{Ra}}{\mathrm{Pr}} T_x
\end{aligned}$$

$$\frac{\partial T}{\partial t} + uT_x + vT_y = \frac{1}{\mathrm{Pr}}(T_{xx} + T_{yy}) \tag{1.35}$$

Here Pr is Prandtl number and Ra, the Rayleigh number is defined as $g\beta(T_w - T_\infty)L^3/\nu\alpha$. The Rayleigh number is interpreted as the ratio of potential energy made available by buoyancy to that dissipated by viscosity. Increasing values of Ra denote a strong buoyancy-driven problem and a vigorous velocity field.

In external flow problems (Figure 1.8b) pressure is normally a known quantity. When a vertical heated plate is immersed in a stationary fluid, boundary-layers are formed starting from its leading edge. The boundary-layers are thin if the Rayleigh number is large ($\gg 1$); this condition is usually met in most applications. The pressure from the ambient is then impressed on the boundary-layer and $p = p_\infty$, a constant everywhere. The dimensionless form of the u-v-T equations governing this problem are:

$$\begin{aligned} u_x + v_y &= 0 \\ \frac{\partial v}{\partial t} + uv_x + vv_y &= v_{xx} + v_{yy} + \frac{\mathrm{Ra}}{\mathrm{Pr}}T \\ \frac{\partial T}{\partial t} + uT_x + vT_y &= \frac{1}{\mathrm{Pr}}(T_{xx} + T_{yy}) \end{aligned} \tag{1.36}$$

1.6.10 Compressible Flows

In many applications such as high-speed flight and nozzle flows associated with turbomachinery the velocity of the fluid is quite high. As a consequence, the accompanying pressure drops are so large that the fluid cannnot be treated as incompressible any longer. Compressibility effects are particularly important if the fluid involved is a gas. In gaseous flows, the gas density becomes a field variable whose value depends on the local pressure and temperature. Secondly, for a gas, adjustments in density in response to a change in pressure must take place simultaneously with temperature. Hence it is no longer appropriate to treat the flow as isothermal even if heat transfer to the external environment is absent.

Equations governing compressible flow are continuity, momentum and the energy. Since velocities encountered here are large, Reynolds numbers are large and viscous forces can be dropped in comparison to inertia forces. The physical extent of the fluid region is usually small in applications and body forces arising from gravity can also be negleted. Similarly diffusive transport can be neglected in comparison to advection of thermal energy, since the Peclet number in gases is of the same order as Reynolds number. It is quite common to assume that the ideal gas law holds at every point in the flow field. Equations governing compressible flow can now be derived and are summarized below:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} &= 0 \\ \rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) &= -\nabla p \end{aligned}$$

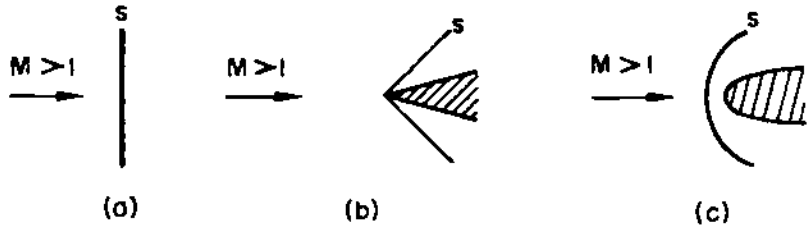


Figure 1.9: (a) Normal, (b) Oblique and (c) Bow Shocks (Shocks marked S)

$$\begin{aligned} \rho(\frac{\partial e}{\partial t} + \mathbf{u} \cdot \nabla e) &= 0 \\ \frac{p}{\rho} &= RT \end{aligned} \tag{1.37}$$

Here e, the specific total energy is $h+ \mid \mathbf{u}^2 \mid /2$ where h is enthalpy. R is the specific gas constant and ρ is a mass-based density. Equations 1.37 solve for the variables $\rho, \mathbf{u}$, T and p respectively.

Small disturbances introduced in a fluid propagate due to collisions between energized molecules of the gas and the undisturbed molecules. Hence it is reasonable to expect that the speed of propagation of these disturbances (called speed of sound and denoted by c) is comparable to the average molecular speed of the gas molecules. Since the molecular speed is only a function of temperature the speed of sound is expected to depend only on the gas temperature. Small perturbation analysis of Equation 1.37 shows that c has the form $\sqrt{\partial p/\partial \rho}\,|_s$ and for an ideal gas, is equal to $\sqrt{\gamma RT}$ where γ is the ratio of specific heats of the gas and s is entropy. The ratio u/c is called the Mach number (M) of flow. It is a measure of the bulk movement of the fluid relative to the random thermal motion of the gas molecules. If M> 1 Equation 1.37 permits a discontinuous solution for the dependent variables. The discontinuity can arise from either the initial or the boundary data. Examples of boundary disturbance include corners, protrusions and objects placed in flow. Planes of discontinuities across which ρ, $\mathbf{u}$, p and T can have a jump in their magnitudes are known as *shocks*. A shock oriented normal to the flow is called a normal shock; oblique and bow shocks are also possible (Figure 1.9).

Flows in which M< 1 are called subsonic, sonic if M= 1 and supersonic if M> 1. The Mach number can vary from point to point in the flow field. It is conceivable that M be less than unity in parts of the physical domain and be greater elsewhere. These are called transonic flows. At the limit of incompressibility, ρ = constant and speed of sound $c \to \infty$. Hence the Mach number is zero for this problem. In practice, the assumption of incompressibility is valid if M< 0.3. Under standard atmospheric conditions this corresponds to an air speed of about 100 m/s.

When viscous and diffusive effects are dropped from analyses, the thermodynamic processes in the flow become fully reversible and entropy changes are zero. However, a shock is a region of exceedingly large velocity and tempera-

ture gradients and is characterized by dominant viscous dissipation and thermal diffusion. The entropy change across a shock is positive and finite. For this reason, Equation 1.37 is commonly applied on each side of the shock and solutions patched up in such a way that mass, momentum and energy equations are satisfied but the change in entropy is non-zero.

Consider a streamline intersecting the shock at right angles. Integrating the steady version of Equations 1.37 along the streamline, the following expressions can be derived (Anderson, 1986):

$$\begin{aligned} \rho u &= \text{constant} \\ p + \rho u^2 &= \text{constant} \\ v &= \text{constant} \\ h + \frac{u^2}{2} &= \text{constant} \end{aligned} \tag{1.38}$$

Here u is the velocity component normal to the shock and v is parallel to it. Equation 1.38 is applied across the shock to obtain equations of the type $\rho_1 u_1 = \rho_2 u_2$. The jump in entropy across the shock must be computed from the global solution of the flow field. When the shock itself moves with a uniform velocity, the normal velocity component is interpreted as the relative velocity between the fluid and the shock, with flow being directed from 1 to 2. Equation 1.38 can then be cast in more useful form as (Currie, 1993):

$$\begin{aligned} \frac{p_2}{p_1} &= 1 + \frac{2\gamma}{\gamma+1}(\mathrm{M}_1^2 - 1) \\ \frac{\rho_2}{\rho_1} &= \frac{u_1}{u_2} = \frac{(\gamma+1)\mathrm{M}_1^2}{2 + (\gamma-1)\mathrm{M}_1^2} \\ \frac{T_2}{T_1} &= \frac{p_2}{p_1}\frac{\rho_1}{\rho_2} \\ \mathrm{M}_2^2 &= \frac{1 + ((\gamma-1)/2)\mathrm{M}_1^2}{\gamma\mathrm{M}_1^2 - (\gamma-1)/2} \\ s_2 - s_1 &= C_p \ln\left[\left[1 + \frac{2\gamma}{\gamma+1}(\mathrm{M}_1^2 - 1)\right]\frac{2 + (\gamma-1)\mathrm{M}_1^2}{(\gamma+1)\mathrm{M}_1^2}\right] \\ &- R \ln\left[1 + \frac{2\gamma}{\gamma+1}(\mathrm{M}_1^2 - 1)\right] \end{aligned} \tag{1.39}$$

These are called Rankine-Hugoniot relations. Since the shock process is irreversible $s_2 - s_1 > 0$. This requires $\mathrm{M}_1 > 1$, $\mathrm{M}_2 < 1$ and $\rho_2/\rho_1 > 1$. Hence, flow approaching a shock is supersonic and the flow leaving it is subsonic. The fluid particles get compressed as they go through the shock region. The ratio M_1 / M_2 is a measure of the shock strength; in the limit of $\mathrm{M}_1 \to 1$, M_2 approaches M_1 and the shock is at its weakest state.

If the flow is steady, Equation 1.37 can be reduced to a single equation in terms of velocity potential ϕ defined as $\mathbf{u} = \nabla\phi$. The existence of ϕ requires that

the vorticity $\omega = \nabla \times \mathbf{u}$ be zero everywhere. As discussed in Section 1.6.7 this requires the approach flow to be uniform. It can be shown that in irrotational compressible flows the entropy production is also zero. Hence, when the velocity potential is used, shock regions cannot be clearly delineated unless the shock strength is close to unity. Despite this drawback the velocity potential is used commonly in transonic flow calculations where shocks formed are in fact quite weak.

The single equation in terms of ϕ arising from Equation 1.37 is called the *full potential equation*. In two dimensional Cartesian coordinates it is given as

$$\left(1 - \frac{u^2}{c^2}\right)\phi_{xx} + \left(1 - \frac{v^2}{c^2}\right)\phi_{yy} - \frac{2uv}{c^2}\phi_{xy} = 0 \tag{1.40}$$

Here c is the local speed of sound; u and v are the velocity components in x and y directions and equal to ϕ_x and ϕ_y respectively. At the limit of incompressibility $c \to \infty$ and Equation 1.29 reduces to Laplace equation, $\nabla^2\phi = \phi_{xx} + \phi_{yy} = 0$. The local speed of sound can be computed from the integrated form of the energy equation

$$\frac{c^2}{\gamma - 1} + \frac{u^2 + v^2}{2} = \text{ constant}$$

Once ϕ is computed the velocity field is obtained as $\nabla\phi$ and other flow quantities such as p and T are obtained from isentropic gas relations (p/ρ^γ = constant).

The full potential equation can be simplified for slender objects placed in uniform flow $\mathbf{u} = (U, 0)$. Let u' be the velocity perturbation due to the object and ϕ the potential associated with it, i.e., $\mathbf{u}' = \nabla\phi$. If $\mid u'/U \mid \ll 1$, the full potential equation can be reduced to

$$\left(\frac{1 - \mathrm{M}^2}{\mathrm{M}^2} - (\gamma + 1)\frac{u'}{U}\right)\mathrm{M}^2\phi_{xx} + \phi_{yy} = 0 \tag{1.41}$$

Here M$= U/c$ and γ is once again the ratio of specific heats of the gas. Equation 1.41 is called the transonic small perturbation equation. If M<1 or M>1 but not near unity, Equation 1.41 reduces to the Prandtl-Glauert equation

$$(1 - \mathrm{M}^2)\phi_{xx} + \phi_{yy} = 0 \tag{1.42}$$

for $u'/U \ll 1$.

1.6.11 Stability and Orr-Sommerfeld Equation

Consider flow over a flat plate at zero angle of incidence. The typical response of a velocity measuring probe inserted in the boundary-layer is shown in Figure 1.10. Experiments show that farther away from the leading edge such as position B the probe response is oscillatory and a chaotic function of time.

Hence even if the incoming flow U is steady in time the boundary-layer develops unsteadiness accompanied by an eddying motion of flow. Vigorous mixing of flow within the boundary-layer produces an almost uniform velocity

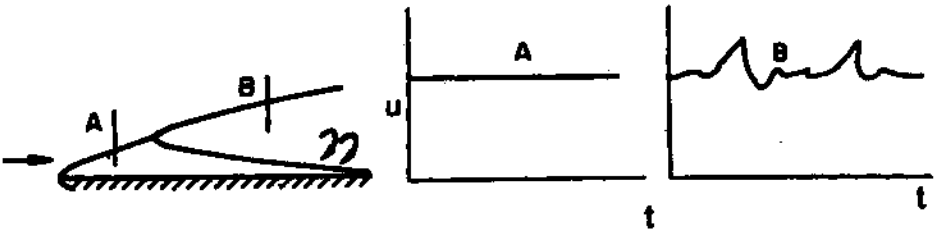


Figure 1.10: Instantaneous Velocity in a Boundary-layer

profile with sharp changes taking place very near the wall. The boundary-layer is then said to be turbulent. The origin of turbulence lies in a region adjacent to the laminar regime (where the probe response is steady) and is called the transition region. Small disturbances introduced in this region tend to grow and for this reason the flow here is said to be unstable. The evolution of laminar flow to turbulence via the transition region can be thought of as a process of instability of the flow system. The consequence of transition can be a state of flow which is still laminar, though disturbed.

Consider a physical system consisting of a viscous fluid and rigid bodies not subjected to any external action. This system will be in a state of rest. As the system departs from equilibrium a parameter R can be conceived, which quantifies external action. As R increases new phenomena successively appear. Hence one observes, (1) a steady motion, (2) a motion whose symmetry pattern is altered, (3) a periodic motion and (4) for sufficiently large values of R the motion becomes irregular and chaotic. Steps (1) to (4) define transition as a flow proceeds towards turbulence.

Specific examples of transition are described here.

1. Reynolds experiment: In this, a head H drives flow Q through a pipe. The apparatus (Figure 1.11) permits H to be varied. It is seen in experiments that for $H < H_c$ a dye introduced in the pipe traces a straight path and is not dispersed.

For $H > H_c$, the dye streak becomes wavy and mixes uniformly with the fluid. The flow is then said to be turbulent. R may be identified as H in this experiment. The transition proceeds from step (1) to step (4) and intermediate stages are not observed.

2. Flow in a curved channel: In a two dimensional channel the flow remains two dimensional for a Dean number De defined as Re $/\sqrt{R_1/h}$ below a critical value (around 50). When this value is exceeded secondary flow sets in giving rise to a swirl in the channel. This is shown in the section marked X-X (Figure 1.12). After this transition the flow continues to remain laminar. At much higher Dean numbers, the flow becomes turbulent.

3. Rayleigh-Benard Convection: Consider a horizontal layer of fluid con-

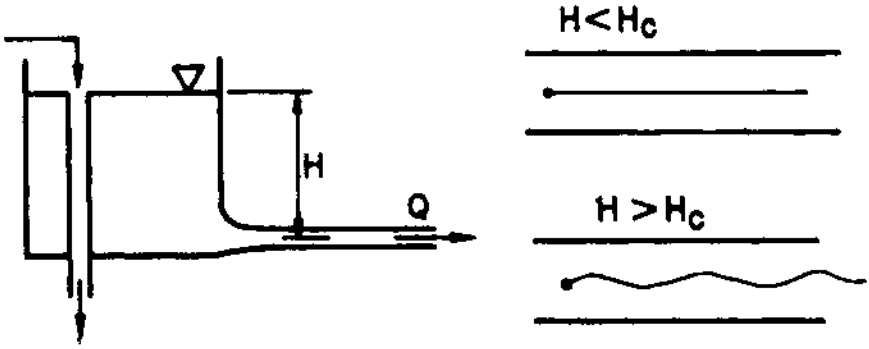


Figure 1.11: Schematic Drawing of Reynolds Experiment on Transition

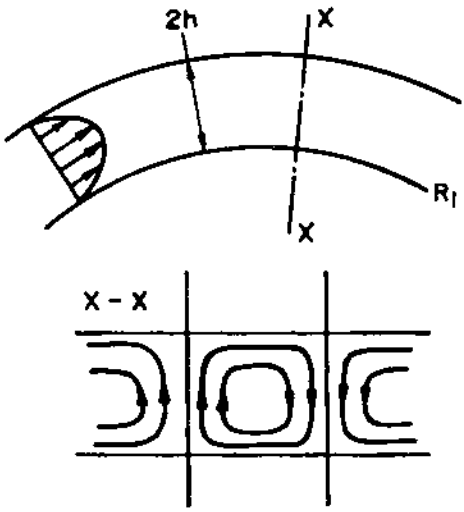


Figure 1.12: Formation of Secondary Rolls in a Curved Channel

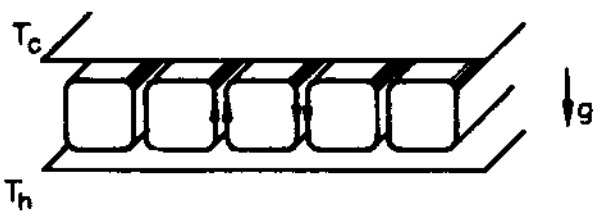


Figure 1.13: Rayleigh-Benard Convection: Longitudinal Rolls in a Differentially Heated Fluid Layer

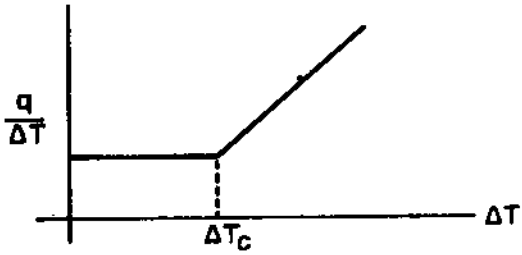


Figure 1.14: Energy Transfer Diagram

fined between two plates (Figure 1.13). The lower plate is maintained at a temperature higher than the upper plate. This gives rise to a stratification in density, with heavier fluid being on top of the lighter one. This situation is potentially unstable. However for $\Delta T < \Delta T_c$, a critical value of the temparature difference, the density difference is not enough to overcome the viscous friction in the fluid. Consequently, the fluid remains motionless and heat transfer between the two plates is by conduction. For $\Delta T > \Delta T_c$ there is incipient fluid movement parallel to the walls and a roll pattern sets in. The onset of fluid movement can be easily located on an energy transfer diagram (Figure 1.14).

As the temparature difference is increased the flow becomes multicellular, and finally turbulent (Busse, 1985).

4. Boundary-layer over a flat plate: A flat plate boundary layer undergoes transition from a laminar state directly to the turbulent state and intermediate laminar regimes are not seen. Turbulence here can be described as a state in which the flow oscillations display a whole spectum of frequencies. The transition occurs over a definite length of the plate, during which the flow becomes increasingly sinuous and chaotic.

Once the flow becomes turbulent the mean flow is accompanied by vigorous mixing and considerably larger shear stresses at the wall. Additional energy is required to sustain turbulent motion. It is of interest to engineers to study both the onset of turbulence and fully turbulent state of flow.

Even though the transition regime of a boundary-layer represents a set of unstable states (during which disturbances are amplified) the smallest distance from the leading edge at which there is a possibility of instability needs to be calculated. This is called the *critical point* of the stability problem. In a more general setting where other forms of instabilities may also appear, critical points R_c do exist for the forcing parameter R. All flows with $R < R_c$ are then called stable with respect to the particular transition. $R > R_c$ represents instability in the sense that a system which may have accidentally strayed into the region $R > R_c$ will be so sensitive to small perturbations that it will inevitably undergo transition. Investigation of stability involves determining the value of R_c.

For flow over a flat plate a critical value of Reynolds number (Re_c) exists below which flow is laminar and above which undergoes transition eventually

to become turbulent. This value is approximately, $\mathrm{Re}_c = Ux/\nu = 60000$. Below this value, all spurious disturbances due to atmospheric turbulence, free stream vortices, wall roughness and vibration will be damped. In the complete absence of these disturbances, the flow may still be laminar above Re_c. However, the smallest disturbance will render such a flow system unstable and revert it to a turbulent state.

As in Section 1.6.2 the dimensionless parameters that are relevent in the absence of body forces are those related to geometry and Reynolds number. For such problems the critical Reynolds number can be determined by perturbing the appropriate laminar equations by a known form of disturbances. The smallest value of Re which fails to damp these disturbances is the critical Reynolds number of the system.

When these disturbances are taken as small in magnitude in comparison to characteristic mean velocity scale, the analysis leads to a *linear stability theory*. Otherwise, the analysis is in general non-linear due to the non-linearity of the Navier-Stokes equations. A fluid system unstable with respect to small disturbances is definitely so with respect to large ones. The contrary is not necessarily valid, i.e., a system may be unstable to large perturbations but be stable for small ones. Hence, linear theory gives a sufficiency condition for instability in the sense that no flow with Re $>\mathrm{Re}_c$ (linear) can be stable. It is possible that for certain large disturbances, a flow with Re $<\mathrm{Re}_c$ (linear) is unstable. This is called subcritical instability and cannot be predicted by linear theory.

When the disturbances imposed on the momentum equations are assumed as periodic in the flow direction and in time, one is led to the *normal-mode* analysis. In quite a few applications, for example boundary-layers and free shear layers, the condition $|u| \gg |v|$ holds, in the sense that the flow is nearly parallel. When normal mode analysis is applied to parallel incompressible flows, the Orr-Sommerfeld equation is obtained. This formulation determines whether disturbances would grow or decay with time.

The derivation of Orr-Sommerfeld equation is considered next. Let $\bar{\mathbf{u}} = \bar{\mathbf{u}}(\mathbf{x})$ and $\bar{p} = \bar{p}(\mathbf{x})$ represent the initial steady state, often called the base flow. They satisfy the steady NS equations

$$\bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} = -\nabla \bar{\mathbf{p}} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}$$

subject to the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$.

Perturbations $\mathbf{u}'$ and p' are introduced in the flow and we would like to calculate the smallest value of Re for which these would be amplified rather than damped. Let $\mathbf{u}$ and p be the instantaneous values of velocity and pressure such that $\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'$, $p = \bar{p} + p' \cdot \mathbf{u}$ and p satisfy the time dependent form of Navier-Stokes equations $\partial \mathbf{u}/\partial t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + (1/\mathrm{Re})\nabla^2 \mathbf{u}$ subject to $\nabla \cdot \mathbf{u} = 0$. Subtracting the base flow out of these equations, we get

$$\nabla \cdot \mathbf{u}' = 0$$

$$\frac{\partial \mathbf{u}'}{\partial t} + [(\bar{\mathbf{u}} + \mathbf{u}') \cdot \nabla(\bar{\mathbf{u}} + \mathbf{u}') - \bar{\mathbf{u}} \cdot \nabla\bar{\mathbf{u}}] = -\nabla p' + \frac{1}{\mathrm{Re}}\nabla^2 \mathbf{u}'$$

Assuming $\mid \mathbf{u}' \mid \ll \mid \bar{\mathbf{u}} \mid$, the term $\mathbf{u}' \cdot \nabla\mathbf{u}'$ can be neglected, in comparison to others so that the equation is linearized with respect to the disturbance velocity $\mathbf{u}'$.

Let ψ be the disturbance stream function, defined as, $\mathbf{u}' = (\psi_y, \ -\psi_x)$. We expand ψ as $\psi = \phi(y)\exp(i\alpha(x - ct))$, a disturbance wave propagated in the x-direction. Here ϕ is the amplitude function of ψ, α is wave number ($= 2\pi/\lambda$, λ being the wavelength), i is the imaginary unit $\sqrt{-1}$ and c is the complex wave speed. One can write $c = c_r + ic_i$, where c_r is the real wave speed and c_i is an amplification factor. Writing $\psi = \phi(y)e^{i\alpha x}e^{i\alpha c_r t}e^{\alpha c_i t}$, it is clear that for $c_i > 0$, ψ will grow exponentially fast for fixed values of x and y. Hence $c_i > 0$ is a criterion of instability. $c_i = 0$ indicates incipient instability and $c_i < 0$ represents a stable system.

In terms of ψ, the continuity equation is identically satisfied. The disturbance equation in Cartesian coordinates is

$$\frac{\partial u'}{\partial t} + \bar{u}u'_x + v'\bar{u}_y = -p'_x + \frac{1}{\mathrm{Re}}\nabla^2 u' \tag{1.43}$$

$$\frac{\partial v'}{\partial t} + \bar{u}v'_x + v'\bar{v}_y = -p'_y + \frac{1}{\mathrm{Re}}\nabla^2 v' \tag{1.44}$$

Using the parallel flow approximation one can drop the term $v'\bar{v}_y$ from Equation (1.44). Eliminating pressure between Equations (1.43) and (1.44) we get

$$(\frac{\partial}{\partial t} + \bar{u}\frac{\partial}{\partial x})(\psi_{xx} + \psi_{yy}) \ - \bar{u}_{yy}\psi_x = \frac{1}{\mathrm{Re}}(\psi_{yyyy} + 2\psi_{xxyy} + \psi_{xxxx})$$

Within the framework of normal mode analysis, the fourth order PDE reduces to an ordinary differential equation in ϕ, namely

$$\frac{i}{\alpha\mathrm{Re}}(D^2 - \alpha^2)^2\phi + (\bar{u} - c)(D^2 - \alpha^2)\phi - \bar{u}_{yy}\phi = 0 \tag{1.45}$$

where $D = d/dy$. This is known as Orr-Sommerfeld equation (OSE). With reference to a flat plate the boundary conditions are $\phi(0) = 0$ (the wall is a streamline), $\phi'(0) = 0$ (no-slip condition), $\phi'(\infty) = 0$ (flow prescribed) and $\phi(\infty) = 0$ (the streamline away from the wall is undisturbed).

The OSE is a linear, homogeneous fourth order ordinary differential equation with homogeneous boundary conditions. Clearly $\phi = 0$ is a possible solution of the problem. We look forward for the right combination of α, c and Re which gives non-trivial values of ϕ. This is an eigenvalue problem and can only be numerically solved for arbitrary forms of the velocity distributions $\bar{u}(y)$. The critical Reynolds number of the problem is defined as $\mathrm{Re}_c = \min_\alpha[\mathrm{Re} : c_i = o_+]$, where we view c as a complex eigenvalue, ϕ as an eigen vector and α and Re as

parameters. The smallest value of Re for all possible values of α which makes c_i just positive, is the critical Reynolds number of the problem. For Re>Re_c, $c_i > 0$ and the disturbances are amplified. The transition process is then initiated in the boundary-layer. ϕ is the mode shape of disturbance stream function and ϕ' the perturbation to the velocity profile in the post stability period.

The disturbance equation for inviscid flow is obtained by taking the limit Re $\rightarrow \infty$. The resulting form is called the Rayleigh equation and is given below.

$$\begin{aligned} (\bar{u} - c)(D^2 - \alpha^2)\phi - \bar{u}_{yy}\phi &= 0 \\ \phi(0) = 0, \quad \phi(\infty) &= 0 \end{aligned} \tag{1.46}$$

The Rayleigh equation is simpler in form compared to OSE since it is a second order ordinary differential equation. OSE gives the range of velocities over which viscosity damps oscillations. However, the Rayleigh equation does not contain Reynolds number and gives only a criterion as to whether a system is stable or unstable. A wide range of results can be obtained by using Rayleigh equation when the effect of viscosity is purely stabilizing. However this is inadequate in the study of stability of boundary-layers (Drazin and Reid, 1981).

1.6.12 Turbulent Flows

Turbulence refers to the random irregular unsteady movement of fluid particles. It is a natural state of fluids in motion and is widely encountered. It is inherently three dimensional. Vigorous motion of the fluid particles improves mixing in internal flows and one obtains a flatter velocity profile compared to the laminar flows. Consequently the wall shear stress and pressure drop increase over and above the values that would have prevailed in the absence of velocity fluctuations. From an engineering point of view this is quite harmful since additional energy must be spent to sustain turbulent flow as compared to laminar flow. Yet turbulent fluctuations can enhance heat transfer from solid surfaces and so can be desirable in many applications. In external flows turbulence can delay the point of separation and can reduce form drag. At the same time the time-dependent velocity fluctuations can generate flow-induced oscillations of objects such as tall buildings and aircraft structures and can lead to premature structural failure. Therefore, prediction and control of turbulence is an important engineering design activity.

The physics of turbulence is problem-dependent. Features such as redistribution of energy among the length scales, development of a two-point correlation and tendency to grow or decay will depend on the presence of body forces, curvature, surface roughness and wall vibration. The smallest length scale that one encounters in turbulent flows is still several orders-of-magnitude larger than the molecular mean free path and consequently the continuum approximation is valid. It is important to emphasize at this point that the time dependent

Navier-Stokes equations do represent turbulence without any approximation. All modelling efforts and simplifications that are employed in the study of turbulence are related to the difficulty in solving either analytically or numerically the full Navier-Stokes equations. The primary difficulty in a numerical calculation is the fineness of spatial and temporal resolution required to represent the smallest length and time scales of velocity and pressure fluctuations in the simplest flow problem. As the grid size is refined the computational effort becomes so large that it is intractable even on the most modern powerful computers. A more serious drawback related to numerical techniques is the build-up of round-off errors in large systems of equations which must be iteratively solved. Beyond a certain level of grid refinement the accumulation of errors can totally destroy the accuracy of the solution unless exceptionally accurate numerical schemes are used. Direct Numerical Simulation (DNS) of turbulent flows is an emerging area of research (Canuto *et al.*, 1987) but is beyond the scope of this book.

An engineering computation of turbulence is concerned with the determination of the time-averaged velocity, pressure and temperature profiles and the effect of time-dependent fluctuations on them. Quantities of engineering importance such as wall shear stress, heat transfer rates and point of separation can be determined from the time-averaged flow properties. This approach is further developed below. It is assumed here that the flow is incompressible, body forces are absent and that the mean flow is steady. Unsteadiness is linked to turbulent fluctuations alone. Analysis of flows where the mean flow is itself time-dependent is not covered here. Hence the instantaneous flow properties can be decomposed into a steady and an unsteady part as

$$\mathbf{u}(\mathbf{x},t) = \bar{\mathbf{u}}(\mathbf{x}) + \mathbf{u}'(\mathbf{x},t),\ p(\mathbf{x},t) = \bar{p}(\mathbf{x}) + p'(\mathbf{x},t) \text{ and } T\ (x,t) = \bar{T}\ (\mathbf{x}) + T'(\mathbf{x},\ t)$$

A schematic drawing of this splitting, called the *Reynolds decomposition* is shown in Figure 1.15. If the mean quantities are independent of time, the global properties such as wall shear stress and heat transfer will be well-defined and independent of time. The definition of a time average of a quantity $u(\mathbf{x},t)$ is

$$\bar{u}(\mathbf{x}) = \lim_{t_0 \to \infty} \frac{1}{t_0} \int_0^{t_0} u(\mathbf{x},t)dt$$

It can be easily shown that for a stationary flow, for which the time-mean is well-defined, the time average of a fluctuation is zero, i.e., $\overline{u'}(\mathbf{x},t) = 0$. Higher order quantities such as $(\overline{u'})^n$ and $\overline{u'v'}$ are non-zero in general. The goal of an engineering calculation is to determine the bar quantities ($\bar{\mathbf{u}}, \bar{p}$ and $\bar{T}$) with the effect of the prime quantities suitably represented in the governing equations. To this end we substitute Reynolds' decomposition into the instantaneous mass balance, Navier-Stokes and energy equations. (Equations 1.4, 1.10 and 1.17) and time average these equations. The resulting system of equations can be

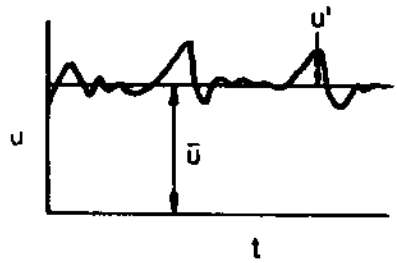


Figure 1.15: Reynolds Decomposition of Instantaneous Velocity into a Time-average and a Fluctuation.

shown to be (White, 1991):

$$\begin{aligned} \nabla \cdot \bar{\mathbf{u}} &= 0 \\ \rho \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} &= -\nabla \bar{p} + (\mu \nabla^2 \mathbf{u} + \nabla \cdot \sigma_T) \\ \rho c_p \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{T}} &= k \nabla^2 \bar{T} + \nabla \cdot \mathbf{q}_T \end{aligned} \tag{1.47}$$

Here σ_T, called the Reynolds stress tensor, is given in expanded form as

$$\sigma_T = -\rho \begin{bmatrix} \overline{(u')^2} & \overline{u'v'} & \overline{u'w'} \\ \overline{u'v'} & \overline{(v')^2} & \overline{v'w'} \\ \overline{u'w'} & \overline{v'w'} & \overline{(w')^2} \end{bmatrix}$$

In index notation $\sigma_T = -\rho \overline{u_{i'} u_{j'}}, i, j = 1, 2, 3$. Reynolds stresses account for the interaction between mean flow and velocity fluctuations. The quantity $\mathbf{q}_T$ called the turbulent heat flux and is given as $\mathbf{q}_T = -\rho C_p(\overline{u'T'}, \overline{v'T'}, \overline{w'T''}) = -\rho C_p \overline{u_i' T'}, i = 1, 2, 3$. Equations 1.47 form a starting point for engineering calculation of mean flow fields in the presence of fluctuations. These however do not form a closed system of equations since the number of equations and number of unknowns are unequal. The additional unknowns are the components of the Reynolds stress tensor and the turbulent heat flux.

Equations that govern the components of σ_T can be generated by multiplying the Navier-Stokes equations by $\mathbf{u}'$ and time averaging each term in them. The differential equation for the turbulent heat flux is obtained by multiplying Navier-Stokes equations by T', adding it to the energy equation multiplied by $\mathbf{u}'$ and time-averaging the sum. These equations are (Launder, 1988):

$$\begin{aligned} \frac{d}{dt}\overline{u_i' u_j'} &= -\{\overline{u_i' u_k'} \frac{\partial \overline{u_j}}{\partial x_k} + \overline{u_j' u_k'} \frac{\partial \overline{u_i}}{\partial x_k}\} \\ &+ \overline{\frac{p'}{\rho}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)} - \frac{2\mu}{\rho}\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}} \\ -\frac{\partial}{\partial x_k}&\left[\overline{u_i' u_j' u_k'} + \frac{\overline{p' u_i'}}{\rho}\delta_{jk} + \frac{\overline{p' u_j'}}{\rho}\delta_{ik} - \nu \frac{\partial}{\partial x_k}\overline{u_i' u_j'}\right] \end{aligned} \tag{1.48}$$

$$\frac{d}{dt}\overline{u_i'T'} = -\{\overline{u_i'u_k'}\frac{\partial\overline{T}}{\partial x_k} + \overline{T'u_k'}\frac{\partial\overline{u}_i}{\partial x_k}\} + \overline{\frac{p'}{\rho}\frac{\partial T'}{\partial x_i}}$$
$$-\frac{\mu}{\rho}\left[\overline{\frac{\partial T'}{\partial x_k}(\frac{\partial u_i'}{\partial x_k} + \frac{\partial u_k'}{\partial x_i})} + \frac{1}{\mathrm{Pr}}\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial T'}{\partial x_k}}\right]$$
$$-\frac{\partial}{\partial x_k}\left[\overline{u_i'T'u_k'} + \overline{\frac{p'T'}{\rho}}\delta_{ik} - \frac{\mu}{\rho}\left(\overline{T'\frac{\partial u_i'}{\partial x_k}} + \overline{T'\frac{\partial u_k'}{\partial x_i}} + \frac{1}{\mathrm{Pr}}\overline{u_i'\frac{\partial T'}{\partial x_k}}\right)\right] \quad (1.49)$$

Here $d/dt = (\bar{\mathbf{u}} \cdot \nabla) = \bar{u}_i\partial/\partial x_i$, repeating indices imply summation over that index and δ is Kronecker-delta. Note that an equation for $\overline{u_i'u_j'}$ contains the triple product $\overline{u_i'u_j'u_k'}$ and an equation for $\overline{u_i'T'}$ contains $\overline{u_i'T'u_k'}$. These are higher order terms and in principle one can generate equations that govern their production, dissipation and transport. However the non-linearity of the flow equations guarantees that an equation for the nth order moment will contain the $(n+1)$th order moment and the system of equations can never be closed. This is often called the *closure* problem of turbulence. It tells us that we necessarily require modelling using experimental data or approximations based on physical reasoning to solve the time-averaged, Reynolds-decomposed Navier-Stokes equations (Wilcox, 1998).

Assuming that modelling is adopted for the third order moments, the number of second order partial differential equations that need to be simultaneously solved for $\bar{\mathbf{u}}$, $\bar{p}$ and $\bar{T}$ is 14 (1-continuity, 3-momentum, 1-energy, 6-Reynolds stresses and 3-turbulent heat fluxes). This by itself is a computationally intensive task. Other complications such as complex geometry, implicit boundary conditions and reactive flows can render this formulation beyond the reach of modern computers. Reynolds stress models based on Equations 1.48 and 1.49 are currently in use for problems with intermediate complexity.

In view of its wide applicability turbulence modeling of wall bounded flows has developed rapidly in recent years. It is described in detail in the succeeding paragraphs. We start with a discussion on turbulence in boundary-layers.

With the growth of boundary-layer on a solid surface the possibility of natural disturbances such as in the free stream and wall vibrations getting amplified increases greatly. In this respect the onset of transition where disturbances are not damped by the action of viscosity is a stability problem. It can be addressed using the formulation described in Section 1.6.11. It has been speculated that instabilities are initiated in a critical layer within the boundary-layer where the local fluid speed and the speed of propagation of the disturbances are equal. Under these conditions the effective stiffness of the fluid is too small to resist the growth of disturbances. Other explanations are based on the momentary formation of inflexion point profiles in the flow, which being unconditionally unstable will repeatedly breakdown to sustain fluctuations in the mean velocity.

Experiments show that the mean flow in the transition region of a boundary-layer is not necessarily stationary. However farther downstream, time-invariant

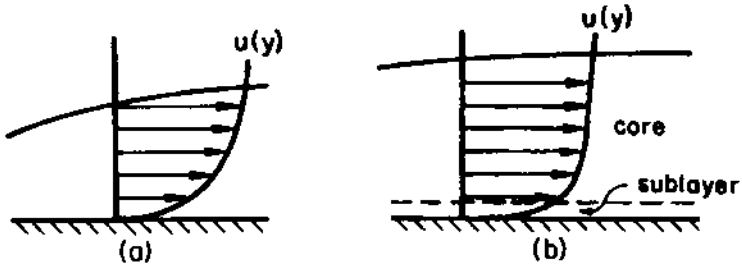


Figure 1.16: Mean Velocity Profile in (a) Laminar and (b) Turbulent Flow

mean velocity, pressure and temperature profiles are obtained with an identifiable pattern of turbulence. For a flat plate this state is reached when $\mathrm{Re}_x(= Ux/\nu)$ exceeds 5×10^5. The mean flow is then said to have reached equilibrium. Here, the boundary-layer thickness changes slowly with distance. Equilibrium conditions are equivalent to fully developed flow conditions observed in internal flows.

Turbulence models of historical importance have been developed for two dimensional equilibrium boundary-layers. Measurements show that a fully developed turbulent boundary-layer is characterized by a thin viscous sublayer near the wall where the flow is itself laminar and an outer region consisting of vigorous eddies and chaotic motion on the small scale superimposed on the mean flow (Figure 1.16).

Though the sublayer is only a small fraction of the boundary-layer thickness its role is far from static. It is believed that the growth of the sublayer thickness followed by its breakdown provides the energy required to sustain fluctuations in the core of the boundary-layer (Kays and Crawford, 1993). The chunk of fluid ejected by the sublayer is called a *burst*; the burst frequency is usually high and changes in the sublayer thickness is not usually discernible. Eddies present in the turbulent core get energized by these bursts. Their size however shrink continuously owing to collisions among them. Eddies of very small size dissipate kinetic energy to the internal energy of the fluid by the action of viscosity. Hence at any given instant eddies of varying length scales are observed. Once the turbulent boundary-layer has evolved long enough in terms of distance, there is equilibrium between the production mechanism associated with the bursts originating at the wall and energy dissipation by viscosity.

Results that can be derived for an equilibrium turbulent boundary-layer over a flat plate play a crucial role in modelling flow in other engineering applications. The flat plate results are derived here. The governing equation for the time-averaged x-component velocity $u(y)$ (written here without a bar) can be easily shown to be

$$(\nu u_y - \overline{u'v'})_y = 0 \tag{1.50}$$

To integrate this equation we need a model to relate $\overline{u'v'}$ with mean flow itself. Of historical importance is the mixing length model derived in analogy

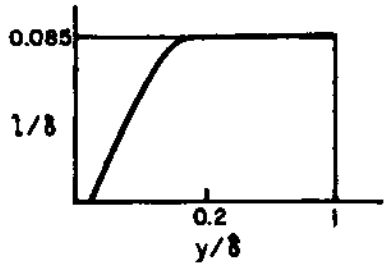


Figure 1.17: Variation of Mixing Length With Distance From the Wall

to the kinetic theory of gases. According to kinetic theory gas molecules are in constant state of motion with an energy that depends on the temperature of the bulk of gas. These molecules move in straight lines till their direction is changed by an elastic collision with another molecule. The average distance travelled between collisions is called the mean free path (λ). At the molecular level the motion is isotropic, i.e., equally probable in all directions. In a flow field having a velocity gradient collisions will lead to momentum transfer across a reference plane normal to the gradient and hence a shear stress acting on it. This stress can be estimated as $\sigma_{xy} \sim \rho v_{rms}\lambda du/dy$, where v_{rms} is the rms molecular speed. Noting that σ_{xy} for unidirectional flow is $\mu du/dy$ (Equation 1.7) one can conclude that the molecular viscosity varies as $\rho v_{rms}\lambda$.

Equation 1.50 shows that $-\rho\overline{u'v'}$ is also a shear stress quantity related to the turbulent fluctuations. It is obviously zero in the sublayer but predominant in the turbulent core. As an empirical extension of kinetic theory of gases we model

$$-\rho\overline{u'v'} = \mu_t \frac{du}{dy} = \rho l^2 (\frac{du}{dy})^2 \tag{1.51}$$

where we have used $\mid u' \mid \sim \mid v' \mid$ and each is modelled as $l du/dy$. Here l is called the mixing length and is analogous in physical terms to the molecular mean free path. The quantity μ_t is called turbulent viscosity and equals $\rho l^2 du/dy$. Clearly it depends on the state of flow and is not a fluid property. The ratio μ_t/ρ $(= \nu_t)$ is called eddy viscosity. It is a measure of momentum transfer by turbulent fluctuations (Schlichting, 1979).

Measurements show that l varies within the boundary-layer (Figure 1.17). It is zero in the sublayer, equal to χy over about 20 percent of δ and proportional to δ elsewhere. Parameter χ, called von Karman constant has a value of 0.4 (± 0.02).

Equation 1.50 can now be integrated to solve for the velocity profile $u(y)$. The results are commonly expressed in terms of the dimensionless '+' coordinates (y^+ and u^+). These are given below in Equation 1.52:

$$\begin{aligned} \text{Sublayer} \quad &: \quad \nu \gg \nu_t,\ y^+ \leq 10.8, \\ u^+ \quad &= \quad y^+ \ \ (l = 0) \end{aligned}$$

$$\begin{aligned} \text{Turbulent core} \quad &: \quad \nu \ll \nu_t, \quad y^+ > 10.8, \\ u^+ &= \frac{1}{\chi}\ln y^+ + 5 \quad (l = \chi y) \end{aligned} \tag{1.52}$$

where $y^+ = y/\nu\sqrt{\tau_w/\rho}$, $u^+ = u/\sqrt{\tau_w/\rho}$ and τ_w is the wall shear stress. This velocity profile satisfies the no-slip condition and is continuous at $y^+ = 10.8$. The velocity profile in the turbulent core is called the logarithmic law of the wall. With suitable adjustments in the magnitude of constants it can be used for boundary-layers with pressure gradient, wall roughness and transpiration. For flows that have not reached global equilibrium the near wall region can still be expected to stabilize rapidly. In this case Equation 1.52 is a convenient form of boundary condition applied not at the wall but in the near-wall region where the log-law is applicable.

The solution $T(y)$ for a fully developed thermal boundary-layer is obtained by solving

$$(\alpha T_y - \overline{v'T'})_y = 0 \tag{1.53}$$

In analogy to the modelling of the Reynolds stress component, the turbulent heat flux is written as, $-\overline{v'T'} = \nu_t/\mathrm{Pr}_t \; dT/dy$. Here Pr_t is called the turbulent Prandtl number. Experiments show that it is nearly a constant for a given kind of flow but may vary from one application to another. For boundary layers $\mathrm{Pr}_t = 0.9$ and for jets and mixing-layers $\mathrm{Pr}_t = 0.5$. Pr_t also depends on the molecular Prandtl number Pr. The values quoted here are for gases ($\mathrm{Pr} \sim 1$). Once Pr_t is specified both flow and heat transfer equations can be solved when the eddy viscosity ν_t is specified. The solution of Equation 1.53 for a constant wall temperature T_w is

$$\begin{aligned} \text{Sublayer} \quad &: \quad \nu \gg \nu_t, \quad y^+ \leq 13.2, \\ T^+ &= \mathrm{Pr}\, y^+ \\ \text{Turbulent core} \quad &: \quad \nu \ll \nu_t, \quad y^+ > 13.2, \\ T^+ &= 2.075 \ln y^+ + 13.2\mathrm{Pr} - 5.34 \quad \text{(air)} \\ T^+ &= 2.075 \ln y^+ + 7.55\mathrm{Pr} - 3.95 \quad \text{(water)} \end{aligned} \tag{1.54}$$

Here $T^+ = (T_w - T)\sqrt{\tau_w/\rho}/(q_w/\rho C_p)$, where q_w is the wall heat flux. The thermal sublayer extending to $y^+ = 13.2$, rather than 10.8, is related to the turbulent Prandtl number being larger than unity in the sublayer. Comparison of Equations 1.52 and 1.54 with laboratory measurements show agreement to an acceptable level of accuracy.

The turbulence model valid for equilibrium boundary-layers have been extended to developing boundary-layers and internal flows with adjustments being made to the model parameters. The starting point in this approach is

generalized Boussinesq hypothesis relating the Reynolds stress tensor σ_T to the time-averaged velocity field. This relation for incompressible flows is

$$\sigma_T = -\rho\overline{u_i'u_j'} = -\frac{2}{3}\rho k\delta_{ij} + \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{1.55}$$

where $i, j = 1, 2, 3$ and k is the kinetic energy of turbulence defined as $0.5(\overline{u_i'u_i'})$. As before, repeated indices refer to summation over all possible values of the respective index. Velocity symbols without a prime represent the mean flow. The turbulent heat flux is expressed as

$$q_{\mathbf{T}} = -\rho C_p\overline{u_i'T'} = \frac{\mu_t}{\mathrm{Pr}_t}C_p\frac{\partial T}{\partial x_i} \tag{1.56}$$

As stated earlier Pr_t is a constant when the nature of flow is prescribed and the modeling effort is directed towards the specification of μ_t and k. Equations 1.47 can now be written as

$$\begin{aligned}
\nabla\cdot\mathbf{u} &= 0\\
\rho\mathbf{u}\cdot\nabla\mathbf{u} &= -\nabla\left(p + \frac{2}{3}k\rho\right)\\
&+ 0.5\nabla\cdot((\mu+\mu_t)(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\\
\mathbf{u}\cdot\nabla T &= \nabla\cdot\left(\alpha + \frac{\mu_t}{\rho\,\mathrm{Pr}_t}\right)\nabla T
\end{aligned} \tag{1.57}$$

In problems where the focus of attention is on the velocity and thermal fields, one can solve for $(p + 2/3k\rho)$ as a composite variable with k, the kinetic energy specified at the same point as pressure. In many applications, k interpreted as turbulence pressure will be small compared to the static pressure and hence can be dropped. In other applications involving fully developed turbulence but not necessarily fully developed mean flows, ∇k is small and can be negleted. Equations 1.57 afford a useful interpretation that turbulence augments molecular mechanisms of momentum and energy transport through the terms $(\mu + \mu_t)$ and $(\alpha + \mu_t/\rho\mathrm{Pr}_t)$. The expanded form of Equations 1.57 in two dimensional Cartesian coordinates is given here for the sake of completeness.

$$\begin{aligned}
u_x + v_y &= 0\\
\rho(uu_x + vu_y) &= -\frac{\partial}{\partial x}\left(p + \frac{2}{3}k\rho\right)\\
&+ \frac{\partial}{\partial x}\left[2(\mu+\mu_t)\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[(\mu+\mu_t)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right]
\end{aligned}$$

$$
\begin{aligned}
\rho(uv_x + vv_y) &= -\frac{\partial}{\partial y}\left(p + \frac{2}{3}k\rho\right) \\
&+ \frac{\partial}{\partial x}\left[(\mu + \mu_t)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[2\,(\mu + \mu_t)\,\frac{\partial v}{\partial y}\right] \\
uT_x + vT_y &= \frac{\partial}{\partial x}\left(\left(\alpha + \frac{\mu_t}{\rho \mathrm{Pr}_t}\right)\frac{\partial T}{\partial x}\right) \\
&+ \frac{\partial}{\partial y}\left(\left(\alpha + \frac{\mu_t}{\rho \mathrm{Pr}_t}\right)\frac{\partial T}{\partial y}\right) \qquad (1.58)
\end{aligned}
$$

With the spatial variation of μ_t established, Equations 1.58 can be solved for the time-averaged velocity, pressure and temperature in turbulent flow. Note that the above equations reduce to the laminar form when μ_t is set equal to zero.

An example of the specification of the turbulent viscosity correlated using extensive amount of experimental data and valid under both equilibrium as well as non-equilibrium conditions is given in the present section (Cebeci and Bradshaw, 1977). The near-wall layer where the mixing length varies with distance is termed the inner region; where it is sensibly constant is the outer region (Figure 1.17).

Inner region: The turbulent viscosity is determined here by the formula

$$\mu_t = \rho l^2 \left|\frac{\partial u}{\partial y}\right| \gamma_{tr}\gamma$$

where the mixing length $l = \chi y(1-\exp(\frac{y}{A}))$, $\chi = 0.4$ and A, a damping constant is specified as $A = 26\frac{\nu}{N}u_*^{-1}$. Here

$$N = \left[p^+/v_w^+(1 - \exp(11.8v_w^+)) + \exp(11.8v_w^+)\right]^{1/2}$$

when $v_w^+ > 0$ and

$$N = (1 - 11.8p^+)^{\frac{1}{2}}$$

if $v_w^+ = 0$. The '+' quantities are defined as, $v_w^+ = v_w/u_*$ and

$$p^+ = \nu\frac{U_e}{u_*^3}\,\frac{dU_e}{dx}$$

u_*, the friction velocity is $\sqrt{\tau_w/\rho}$, U_e is the outer (potential) velocity and x is the main flow direction. The transpiration wall velocity normal to the solid surface is given the symbol v_w. The factor γ_{tr} in the definition of μ_t accounts for intermittency of turbulence in the transition region. It is empirically correlated as

$$\gamma_{tr} = 1 - \exp[-G(x - x_{tr})\int_{x_{tr}}^{x}\frac{dx}{U_e}]$$

where x_{tr} is the location where transition is initiated and G is 8.33×10^{-4} $(U_e^3/\nu^2)(U_e x/\nu)^{-1.34}$. For a flat plate boundary-layer $U_e x_{tr}/\nu \sim 60000$. The other intermittency factor γ accounts for the fact that turbulence is intermittent as the free stream is approached. It is specified as, $\gamma = (1+5.5(y/y_0)^6)^{-1}$ where y_0 is the point at which $u/U_e = 0.995$. In internal flows γ can be set equal to unity.

Outer region: The turbulent viscosity in the outer region is determined from

$$\mu_t = \rho\alpha \int_0^\infty \mid (U_e - u)dy \mid \gamma_{tr}\gamma$$

where the factor $\alpha = 0.0168\frac{1.55}{1+F}$, $F = 0.55(1 - \exp(-0.243z^{\frac{1}{2}} - 0.298z)), z = \mathrm{Re}_\theta/425 - 1$ if $\mathrm{Re}_\theta > 425$ and zero otherwise. Re_θ is the Reynolds number based on the momentum thickness. The inner and outer regions are matched at a level where $\mu_t(\text{inner}) = \mu_t(\text{outer})$.

Equations 1.55 and 1.56 must be used with caution. These resemble the constitutive relations for a Newtonian fluid (Equation 1.7) and Fourier's law of heat conduction (Section 1.5) respectively. The eddy viscosity relations are however far from being physical laws since μ_t depends on the flow field and is not a fluid property. These relations are quite restrictive as well. For example, the expressions for the Reynolds stresses and the turbulent heat flux become zero at points where the mean velocity and temperature gradients vanish respectively. This can occur at symmetry planes or points of maximum velocity as in wall jets and reattachment points of separated zones. Experiments show that turbulence levels are finite and non-zero at these locations and the eddy viscosity approach is inapplicable for these problems. As a second counter-example where Equation 1.55 breaks down, consider parallel shear flow given by $\mathbf{u} = (U(y), 0, 0)$. For this problem we get $\overline{(u')^2} = \overline{(v')^2} = \overline{(w')^2} = 2k/3$ and turbulence is isotropic. Laboratory experiments however show $\overline{(u')^2}$, the turbulence fluctuation in the flow direction to be larger than the other two components. As a rule-of-thumb the eddy viscosity approach is used in problems that do not develop a strong tendency for anisotropy. The near-wall region of a boundary-layer is anisotropic. For a flat plate typical rms values of velocity fluctuations normalized by the free stream velocity are $u' = 10\%$, $v' = 4\%$ and $w' = 6\%$. For this reason the eddy viscosity equations are not applied all the way upto the wall; instead they are matched with equations such as logarithmic law (Equations 1.52 and 1.54) which have been conditioned by laboratory measurements.

The problem of anisotropy can be handled when the original Reynolds stress and turbulent heat flux equations (1.48 and 1.49) are solved. The difficulty of closure of these equations as well as their computational complexity has led to the search for models that are less demanding. The turbulence kinetic energy-dissipation (k -ε) model that is described below has come up in response to this need.

The k-ε model continues to employ the eddy viscosity concept. It however estimates μ_t using principles of energy balance in the turbulent microstructure

rather than algebraic formulas. Since k and ε are both produced, destroyed and transported by flow one obtains partial differential equations governing their values in the flow domain. The problem of anisotropy in the near-wall region is resolved by prescribing the variation of k and ε that are compatible with the logarithmic law. The k-ε equations continue to have model constants whose values are continually updated as improved experimental data becomes available. Some of these values do change from problem to problem. The strength of the k-ε model lies in the fact that the model constants do not vary drastically from one application to another since some of the physics of turbulence has been properly accounted for.

The definition of turbulent kinetic energy k, as given by Equation 1.55 is

$$k = \frac{1}{2}\overline{(((u')^2 + (v')^2 + (w')^2)} = \frac{1}{2}\overline{(u_i')^2}, \quad i = 1, ...3.$$

The total viscous dissipation per unit volume for an incompressible fluid (Section 1.5) is $\Phi = 0.5\mu(\partial u_i/\partial x_j + \partial u_j/\partial x_i)^2$. We split Φ into dissipation in the mean flow $\bar{\phi}$ and that corresponding to the fluctuations ϕ. One can show that $\bar{\phi}$ and ϕ have the same expression as Φ with u_i interpreted as the time-averaged velocity and velocity fluctuation respectively. The assumption of isotropy continues to hold and the k-ε model works with the isotropic portion of the dissipation related to velocity fluctuations. Thus ε is the isotropic part of ϕ that is invariant to the interchange of indices i and j is chosen as $\nu\overline{(\partial u_i'/\partial x_j)(\partial u_j'/\partial x_i)}$. It represents the isotropic dissipation rate of the turbulent kinetic energy k.

The relationship among k, ε and μ_t is developed as follows. Turbulence is produced owing to the instability of the base flow resulting in phenomena such as sublayer bursts in a boundary-layer. Hence the largest length scale L associated with turbulence will scale with the base flow itself, i.e., L will be of the order of δ in boundary-layers and tube radius in pipe flows. Collision among eddies characterized by length L will produce progressively smaller length scales. The smallest length scale l is one for which viscous effects are important and is instantaneously dissipated as thermal energy. The greatest portion of kinetic energy of velocity fluctuations is associated with the large length scales. The lower end of the length scale distribution acts as a sink for the kinetic energy k and one can visualize energy being transferred from the large to the small length scales. If U is the characteristic velocity fluctuation, the limits that will apply are $UL/\nu \gg 1$ and $Ul/\nu \sim 1$.

While the quantity L called the inertial length scale is associated with kinetic energy k and the momentum transfer owing to turbulence, the smallest length scale l is the dissipation scale of the flow. It is clear that the mixing length referred Equation 1.51 (and Figure 1.17) can be identified with L. Hence in analogy to the result $\mu = \rho v_{rms}\lambda$ derived from kinetic theory of gases (Schlichting, 1979) one can write $\mu_t = \rho UL$. A measure of U is $\sqrt{k}$ and so $\mu_t = \rho L\sqrt{k}$.

An important assumption made in the development of k-ε model is that the cascade mechanism of energy transfer is fully evolved at every instant of time even if the mean flow is far from being fully developed. Hence, the production rate of turbulence is balanced by dissipation, i.e. $\varepsilon = dk/dt \sim k/t$. Here t is a characteristic time scale of the large scale eddies. One can estimate t as $L/\sqrt{k}$ and so $\varepsilon \sim k^{3/2}/L$. Since $\mu_t = \rho L\sqrt{k}$ we get

$$\mu_t = \rho \frac{C_\mu k^2}{\varepsilon} \tag{1.59}$$

C_μ being a model constant that must be determined from laboratory experiments.

The equation governing k is obtained by contracting indices ($i = j$) in Equation 1.48. This gives

$$\frac{\partial k}{\partial t} + \mathbf{u} \cdot \nabla k = -\nabla \cdot \overline{(\frac{p'}{\rho} + k)\mathbf{u}'} - \overline{u_i' u_j'} \frac{\partial u_i}{\partial x_j} + \nu \nabla^2 k - \varepsilon$$

Here $\overline{u_i' u_j'}$ is to be replaced by the Boussinesq expression (Equation 1.55). In the absence of any other information the term $\overline{(p'/\rho + k)\,\mathbf{u}'}$ is approximated assuming gradient diffusion as $(\nu_t/\sigma_k)\partial k/\partial x_i$. σ_k, called turbulent Prandtl number of kinetic energy must be obtained from experiments dedicated to model development. Since the mean flow is assumed steady and the turbulence field fully developed, the kinetic energy equation finally simplifies to

$$\rho u_i \frac{\partial k}{\partial x_i} = \mu \frac{\partial}{\partial x_j}\left(1 + \frac{\mu_t}{\mu \sigma_k}\right)\frac{\partial k}{\partial x_j} + \mu_t \frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \varepsilon \tag{1.60}$$

It represents the equality of the advection of kinetic energy with the algebraic sum of diffusion, production and dissipation.

It is quite difficult to develop a transport equation for the isotropic dissipation rate ε. Instead we mimic the kinetic energy equation with proper adjustments for dimensions and write the ε equation as

$$\begin{aligned} \rho u_i \frac{\partial \varepsilon}{\partial x_i} &= \mu \frac{\partial}{\partial x_j}\left(1 + \frac{\mu_t}{\mu \sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j} \\ &+ C_1 \mu_t \frac{\varepsilon}{k}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - C_2 \frac{\varepsilon^2}{k} \end{aligned} \tag{1.61}$$

The model constants $C_\mu, C_1, C_2, \sigma_k$ and σ_ε are commonly chosen as 0.09, 1.44, 1.92, 1.0 and 1.3 respectively (Rodi, 1984). These are based on a wide range of experimental data generated from flows that satisfy the thin shear layer approximation. Their applicability to low Reynolds number and separated flows is a topic of current research (Speziale and Thangam, 1992).

As mentioned earlier the assumption of isotropy restricts the applicability of the k-ε model and the eddy viscosity approach to regions away from the wall. Expressions such as Equations 1.52 and 1.54 that have been independently tested must be used as wall boundary conditions for k and ε. These equations do not hold when flows with pressure gradient, curvature and transpiration are encountered. However equivalent expressions can be derived for these problems with additional complexities. Expressions for k and ε in the near-wall region for an impermeable flat plate are derived here.

Very close to the wall but outside the laminar sublayer the assumption of equilibrium among the turbulent length scales is equivalent to setting production equal to dissipation in Equation 1.60. Hence $\varepsilon = \mu_t \left(\partial u/\partial y\right)^2$. Again, close to the wall the shear stress is a constant and equal to its value at the wall, i.e.

$$\tau_w = (\mu_t + \mu)\partial u/\partial y \sim \mu_t \partial u/\partial y$$

outside the laminar sublayer. Combining these results with Equation 1.59 yields an expression for the near wall kinetic energy k as

$$k = \frac{1}{\sqrt{C_\mu}}\frac{\tau_w}{\rho}, \quad 30 < y^+ < 100$$

Here wall shear stress is taken to be computed from the law of the wall (Equation 1.52). Computing $\partial u/\partial y$ from this equation and using the expression for wall shear stress given above yields the near wall dissipation at a distance y as

$$\varepsilon = \frac{C_\mu k^2}{\chi y \sqrt{\tau_w/\rho}} = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{\chi y} 30 < y^+ < 100$$

If the wall is at a uniform temperature the boundary condition is straightforward. However if a constant heat flux is prescribed the thermal boundary condition for temparature can be written as $T^+ = \mathrm{Pr}_t(u^+ + P_f)$ where

$$P_f = 9.24 C_\mu^{-0.25}\left(\left((\mathrm{Pr}/\mathrm{Pr}_t)^{\frac{3}{4}} - 1\right)\left(1 + 0.28\exp\left(-0.007\mathrm{Pr}/\mathrm{Pr}_t\right)\right)\right)$$

is a correction function whose form is derived on the basis of pipe flow data for a variety of fluids (Launder, 1988). The near wall expressions replace wall boundary conditions in turbulent flow calculations. Inflow, outflow and far-field conditions can be prescribed for k and ε as for other flow variables.

Typical variations of the flow quantities in a flat plate boundary-layer are shown in Figure 1.18. Figures 1.18 (a, b and c) respectively show profiles for the entire boundary-layer, near the wall and near-wall low Reynolds number problems. The turbulence quantities attain a maximum within the boundary-layer and decay to zero as the free-stream is approached. There is some uncertainty in their values near a wall and are strictly defined only outside the sublayer through the wall functions. Since the sublayer is laminar, k, ε and μ_t can be thought of as being zero in this region. In low-Reynolds number as well as

separated flows the equilibrium distributions of turbulence quantities are not fully established and the wall function approach based on a two-layer model is not applicable. In these problems, the k -ε equations must be integrated all the way up to the wall and near-wall behaviour must be built into the governing equation itself. A useful specification is a quadratic variation of k, a linear variation of shear stress near the wall and a constant dissipation rate in the limit $y \to 0$ as in Figure 1.18c (Wilcox, 1998).

The k-ε model formally employs several restrictive assumptions and must be constantly updated using experimental data. It is however an important development in engineering calculations since it has beeen proved to be surprisingly accurate in many high Reynolds number applications. There is some evidence k-ε model does have a theoretical basis and that the empirical constants have analytical expressions which can be determined from direct numerical simulation of Navier-Stokes equations (Moin, 1997).

While the drawback of assuming isotropy is overcome by using Reynolds' stress equations it continues to be a computationally intensive option even on powerful computers. An algebraic stress model that predicts anisotropy without requiring the solution of differential equations for the stress components has evolved as a refinement of the k-ε model (Launder, 1988). The computational steps of this model are given below.

1. Solve Equation 1.47 with assumed distributions of $\sigma_{\mathbf{T}}$ and $q_{\mathbf{T}}$.

2. Solve for k and ε (Equations 1.60 and 1.61).

3. Calculate the Reynolds stress and turbulent heat flux components using the pair of formulas:

$$\left(\overline{u_i'u_j'} - \frac{2}{3}k\delta_{ij}\right) = -C_s \ \frac{k}{\varepsilon}\left(P_{ij} - \frac{1}{3}\delta_{ij}P_{kk}\right)$$
$$\overline{u_i'T'} = -C_\theta \ \frac{k}{\varepsilon}\left(\overline{u_i'u_k'}\frac{\partial T}{\partial x_k} + \overline{u_k'T'}\frac{\partial u_i}{\partial x_k}\right) \tag{1.62}$$

Here $P_{ij} = \overline{u_i'u_k'}\partial u_j/\partial x_k + \overline{u_j'u_k'}\partial u_i/\partial x_k$ with repeated indices, for example k in the expression for P_{ij}, representing summation. C_s and C_θ are empirical coefficients whose values have been experimentally determined to be 0.26 and 0.3 respectively. The algebraic equations given above form a system of 6 simultaneous equations in three dimensions and 3 in two dimensions at each point in the flow domain.

4. With $\sigma_{\mathbf{T}}$ and $q_{\mathbf{T}}$ updated one returns to step 1 for iterating on the time-averaged velocity and temperature fields.

The algebraic Reynolds stress model is based on the hypothesis that the extent of anisotropy/turbulent heat flux is proportional to the product of production rate of anisotropy/turbulent heat flux and a characteristic time scale. The time scale is estimated as k/ε.

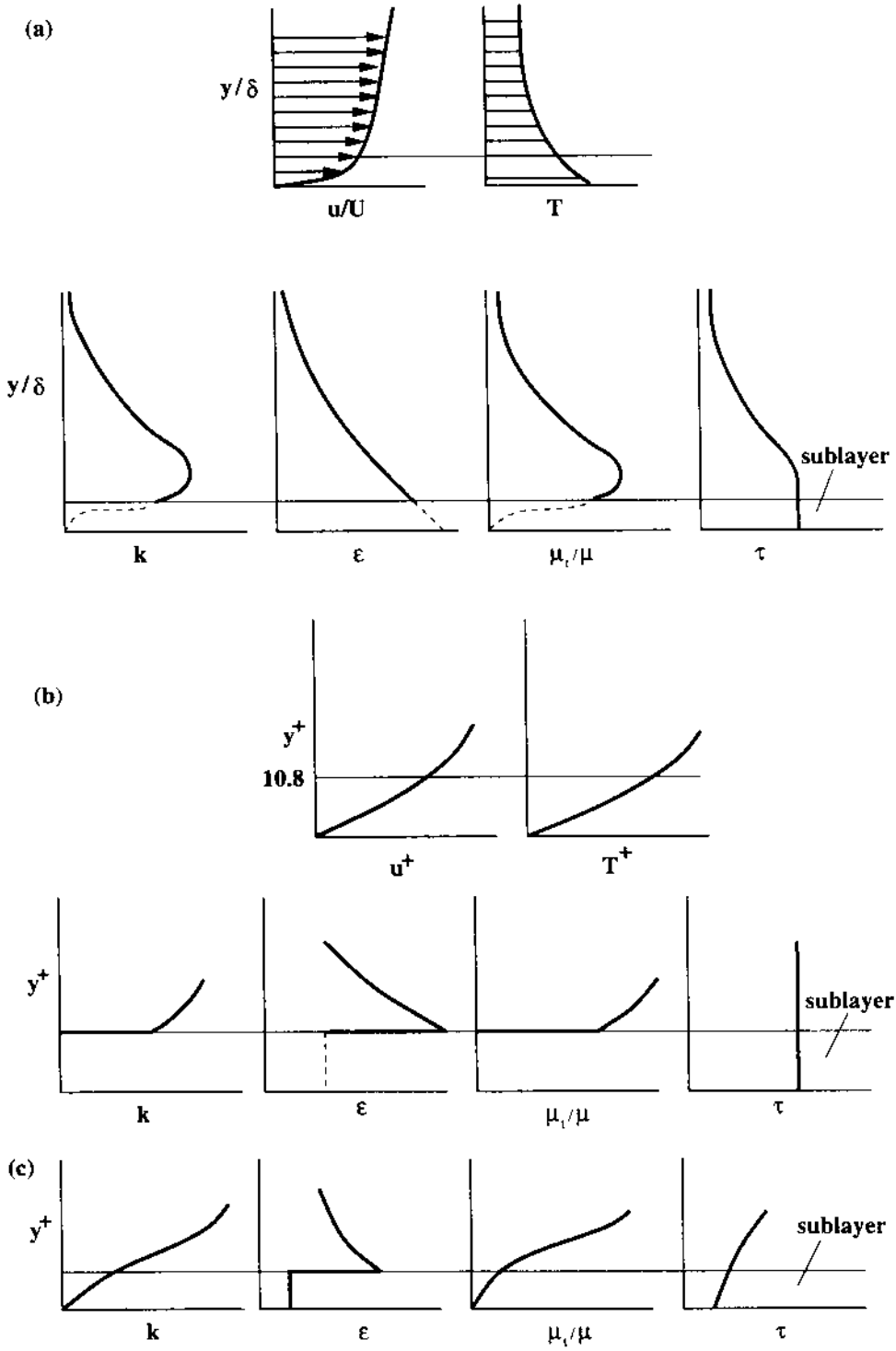


Figure 1.18: (a) Global and (b) Local Profiles in a Turbulent Boundary-layer. (c) Near-wall profiles in a Low Reynolds Number Problem.

1.7 Classification of Second Order Partial Differential Equations

The momentum and energy equations are second order partial differential equations (PDE). The choice of the numerical technique and the number of boundary conditions required to solve these equations depend primarily on their mathematical character. Unfortunately this character is not fixed and it depends on the magnitude of the dimensionless parameters such as Reynolds number and Peclet number.

A brief introduction to the classification of linear second order partial differential equations in two dimensions is given here (Greenberg, 1998; Sneddon, 1986). Consider the equation

$$\begin{aligned} A(x,y)u_{xx} &+ 2B(x,y)u_{xy} + \\ C(x,y)u_{yy} &+ F(x,y,u,u_x,u_y) = 0 \end{aligned}$$

in terms of the dependent variable $u(x,y)$. It is assumed that A, B and C have continuous second order partial derivatives, $B^2 - AC$ is of uniform sign for the range x and y values of interest, and it never vanishes unless it is identically zero and A is non-zero. Equations for which $B^2 - AC > 0$ are called *hyperbolic*. In this case it is possible to define transformations $r(x,y)$ and $s(x,y)$ that are solutions of the form $r =$ constant and $s =$ constant to the equations

$$\frac{dy}{dx} = \frac{-B - \sqrt{B^2 - AC}}{A} \quad \text{and} \quad \frac{dy}{dx} = \frac{-B + \sqrt{B^2 - AC}}{A}$$

In terms of r and s, u satisfies the equation $u_{rs} = f(r,s,u,u_r,u_s)$. This is the canonical form of a second order hyperbolic PDE.

If $B^2 - AC = 0$ the differential equation is classified as *parabolic*. The canonical form of a parabolic PDE is, $u_{ss} = g(r,s,u,u_r,u_s)$ where r is a function as defined above and s is a function for which $(As_x^2 + 2Bs_xs_y + Cs_y^2)$ is never zero.

If $B^2 - AC < 0$ the differential equation is classified as *elliptic*. The functions r and s are distinct and are complex conjugates of each other. Defining a pair of real functions p and q as, $p = (r+s)/2$ and $q = (r-s)/2i$, $i = \sqrt{-1}$, the canonical form of an elliptic PDE can be written as $u_{pp} + u_{qq} = h(p,q,u,u_p,u_q)$.

Though the classification of a PDE is subject to numerous constraints it is a useful starting point for understanding the physical processes being represented by the mathematical problem. Consider the following examples. The steady heat conduction equation in two dimensions $T_{xx} + T_{yy} = 0$ is elliptic. The unsteady heat conduction in one space dimension $T_t = \alpha T_{xx}$ is parabolic. The unsteady one dimensional transport equation with a constant velocity at the limit Pe $\to \infty$ is given as $T_t + UT_x = 0$. Differentiating first with respect to time and next with respect with x and subtracting, it can be written as $T_{tt} - U^2T_{xx} = 0$. This equation can be shown to be hyperbolic. It has a general

solution, $T(x - Ut) =$ constant. On a $x - t$ plane, the line $x - Ut =$ constant is called the characteristic direction. Temperature is a constant along this line and equal to $T(x, 0)$. This is interpreted as information being transferred to any other location $T(x, t)$ on the characteristic curve with a speed U.

In analogy to the examples given above, steady diffusion-dominated problems are classified as elliptic even when they are three dimensional and has temperature-dependent properties. Similarly, evolutionary problems are classified as parabolic. Advection-dominated problems are classified as hyperbolic and are accompanied by the existence of characteristic directions.

An elliptic PDE can be integrated when boundary conditions are prescribed and hence constitutes a boundary-value problem. A parabolic PDE requires an initial condition with respect to the direction in which the physical process undergoes evolution. Hence it is viewed as an initial-boundary value problem. A hyperbolic PDE is also solved as an initial-boundary value problem. However it can be integrated as a purely initial value problem along the characteristic directions.

The momentum and energy equations do not possess a unique character even when restricted to two independent variables. The energy equation is hyperbolic if the Peclet number $\mathrm{Pe} \to \infty$, parabolic if $\mathrm{Pe} \to 0$ and is of mixed mathematical character for finite values of Pe. The steady incompressible Navier-Stokes equations are elliptic for moderate values of the Reynolds number. As Re becomes large, boundary-layers form on solid surfaces. The boundary-layer equations are parabolic since they can be solved as a marching problem upto the point of separation. Unsteady Navier-Stokes equations must be solved as an initial-boundary value problem. For compressible flows the unsteady Euler equations are hyperbolic but the steady equations are hyperbolic only if the Mach number exceeds unity. For linear hyperbolic problems, discontinuities in the physical region can arise only from discontinuities in the initial and boundary data, that propagate into the interior along the characteristic direction. The Euler equations are non-linear and jump discontinuities can appear in the fluid even when the initial and boundary data are completely smooth.

1.8 Initial and Boundary Conditions

For incompressible flows, one can view the momentum equation as determining the velocity components, the continuity equation for pressure and the energy equation for temperature. In compressible flows density appears as an additional variable. Density is then obtained from the continuity equation while pressure is calculated using the equation of state of the fluid being studied. For definiteness, this section is restricted to a discussion on incompressible flows.

While the governing equations must be satisfied in the interior of the fluid, specific solutions can be obtained only by prescribing the constraint of flow geometry as well as the initial state of the flow field. Hence on the boundary of the region velocity, pressure and temperature must be suitably defined to

permit integration of the governing equations. The extent to which this data is required can be established by examining the order of the governing differential equations. For a differential equation of nth order in a dependent variable f, n conditions such as derivatives upto $(n-1)$th order (i.e., $f, f' \ldots\ldots f^{n-1}$) at a particular location may be specified. For partial differential equations these derivatives must be calculated along the local normal direction on the flow boundary. In the absence of information about f itself on a part of the flow domain, unique solutions in general will not exist. In transient problems, the time-derivative is of first order and the value of the dependent variable at $t = 0$ must be given. This is called an *initial* condition. Other conditions prescribed on the physical boundary of the fluid region are called *boundary* conditions.

The Navier-Stokes equations form a system of coupled, non-linear, second order partial differential equations. Non-linearity arises primarily from the acceleration terms $(\mathbf{u} \cdot \nabla \mathbf{u})$ and weakly from the dependence of fluid properties on pressure. The Navier-Stokes equations under steady flow conditions require the velocity vector $\mathbf{u}$ or its normal derivative to be specified on the entire flow boundary. It is enough to specify pressure at one point since only its first derivative appears in the equations. If u and $\partial u/\partial n$ are both known on part of the boundary, integration could be initiated into the flow domain. However, it is very uncommon that both quantities are simultaneously known. For example, consider a solid surface. The fluid is assumed to stick to the wall by the action of viscosity. This is called the no-slip condition and it requires that the solid and the adjacent fluid do not have a velocity relative to each other. The velocity gradient is however not known at the wall and must be obtained as a part of the solution.

For the unsteady NS equations the remarks given above carry over for boundary conditions. However, the solution evolves in time and integration must start with an initial condition at time $t = 0$. Typical initial conditions may correspond to a stationary flow state $(\mathbf{u} = 0)$ after which a definite non-zero flow is prescribed at the boundary. More generally, the initial flow field must be a solution of the steady Navier-Stokes equations.

The steady incompressible Euler equations can be replaced by the potential equation $\nabla^2\phi = 0$ which in turn is elliptic. Euler equations are obtained by dropping the viscous terms and the no-slip condition would give rise to an inconsistency in the solution procedure. It can also be shown that the Euler equations along with $\mathbf{u} = 0$ on a solid surface constitute an ill-posed problem whose solutions are excessive sensitive to the boundary data. The appropriate conditions on a solid boundary is then a zero normal velocity, i.e. $\mathbf{u} \cdot \mathbf{n} = 0$. In terms of ϕ, the impermeability boundary condition can be expressed as $\partial\phi/\partial n = 0$.

When the dependent variable is completely prescribed on the surface of the flow domain, it is called a Dirichlet boundary condition and when the normal gradient is prescribed, Neumann condition is obtained. Meaningful solutions are possible only with a Dirichlet condition, or a combination of Dirichlet and Neumann conditions. Unique solutions are normally not possible by applying

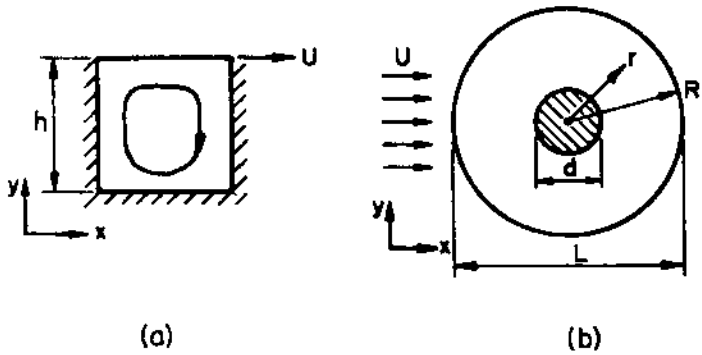


Figure 1.19: Examples of Initial and Boundary Conditions

Neumann conditions alone. For example, if T is a solution, (T + constant) is also a solution of the steady energy equation which satisfies purely Neumann conditions. Specific examples of initial (IC) and boundary conditions(BC) are given below.

Example 1.1. Flow in a lid-driven cavity (Figure 1.19a)

The initial condition can be chosen as, $\mathbf{u}(x, y, 0) = 0$. The boundary conditions are the no-slip conditions on the walls of the cavity, namely, $\mathbf{u}(x, 0, t) = 0$,$\mathbf{u}(0, y, t) = 0$, $\mathbf{u}(h, y, t) = 0$, $u(x, h, t) = U$ and $v(x, h, t) = 0$. It is sufficient to specify the pressure at one point in the flow domain. If the initial pressure field is $p(\mathbf{x}, 0) = 0$, the pressure at an interior point where $\mathbf{u} = (0, 0)$ can be specified as zero. When a stream function-vorticity formulation is used the boundary conditions must be specified in terms of ψ and ω. For a quiescent initial state, $\psi(\mathbf{x}, 0) = \omega(\mathbf{x}, 0) = 0$. For all subsequent times, the cavity boundary remains a streamline where $\psi = 0$. Vorticity is produced at the wall and can be computed as $\omega = -\nabla^2\psi$. On the walls parallel to the x-axis, $\partial v/\partial x$ and therefore $\psi_{xx} = 0$. Hence $\omega = -\psi_{yy}$. Similarly on the walls parallel to y-axis, $\omega = -\psi_{xx}$.

Example 1.2. Potential flow past a cylinder (Figure 1.19b)

A velocity potential formulation can be used here. On the surface of the cylinder the normal velocity component $\partial\phi/\partial n = 0$; the no-slip condition cannot be applied here. Far away from the cylinder the flow is undisturbed. ϕ can be computed by integrating the equation $\nabla\phi = (U, 0)$.

Example 1.3. Real flow past a cylinder (Figure 1.19b).

On the surface of the cylinder both velocity components are zero ($\mathbf{u} = (0, 0)$). On the inflow plane $(x = 0), u = U$ and $v = 0$. The magnitude of velocity is generally unknown on the planes above and below the cylinder as well as on the outflow plane. It is conventional to choose L, the size of the computational domain to be large for the following conditions to hold

$$y = 0,\ L: \quad u = U, \quad \frac{\partial v}{\partial y} = 0;$$
$$x = L: \quad \frac{\partial \mathbf{u}}{\partial x} = 0$$

The gradient condition for velocity on the outflow plane does not arise from any physical criterion except that changes in the magnitude of the appropriate velocity component are small when one moves sufficiently far away from the region of disturbance. This condition can be generalized as, $\nabla \mathbf{u} \cdot \mathbf{n} = 0$. If the flow is steady, pressure can be set equal to a reference value at any one point in the flow domain. Otherwise this point can be moved either to the inflow or the outflow plane.

When a stream function-vorticity formulation is used the appropriate boundary conditions are:

On cylinder surface : $\psi = 0, \quad \omega = \partial^2\psi/\partial r^2$

Far field: $(\psi_y, -\psi_x) = (U, 0), \quad \omega = 0$

The far field equations for ψ can be integrated to produce a Dirichlet condition for the stream function.

In a temperature calculation initial conditions are prescribed as $T(\mathbf{x}, 0) =$ given function of $\mathbf{x}$. Boundary conditions of the following form are possible.

Known temperature : $T = T(\mathbf{x}, t)$

Prescribed heat flux : $-k\partial T/\partial n = q(\mathbf{x}, t)$

Convective heat transfer : $-k\partial T/\partial n + h(T - T_{\text{reference}}) = g(\mathbf{x}, t)$

Radiative heat transfer : $-k\partial T/\partial n + h_r T^4 = f(\mathbf{x}, t)$

In the expressions given above the right hand side is a known quantity. Since the energy equation is a second order PDE the boundary conditions that are permitted are specified temperature and heat flux at a solid wall. Fluid particles come to rest on a wall and so the heat flux consists of purely the diffusion component and can be estimated using Fourier's law of heat conduction. Hence in the expressions given above, k is the fluid thermal conductivity. The diffusive flux may itself be prescribed or can be inferred if ambient conditions are known. This inference can be based on a convective heat transfer coefficient h or a radiation coefficient h_r. On an outflow boundary the boundary condition can be chosen as $\mathbf{u} \cdot \nabla T = 0$. This is equivalent to assuming transport by advection alone. If the field is unsteady, the time derivative of temperature can be added to this expression to yield $T_t + \mathbf{u} \cdot \nabla T = 0$.

1.9 Governing Equations in Generalized Coordinates

Expanded forms of mass, momentum and energy equations in orthogonal coordinate systems are presented in Sections 1.2,1.4 and 1.5. In many applications it is necessary to use coordinate systems that are not necessarily orthogonal.

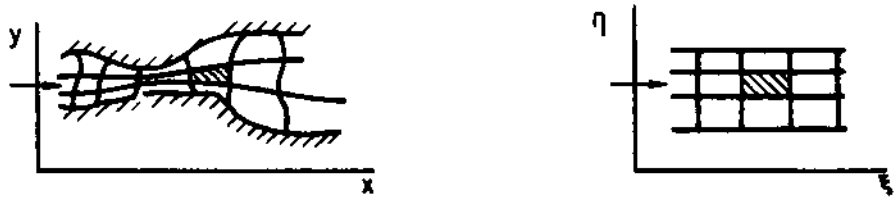


Figure 1.20: Sketch of a Generalized Coordinate System

An example of such an application is flow in a complex geometry. In general a coordinate system viewed through which a complex geometry appears regular will itself be non-orthogonal in the Cartesian space (Figure 1.20). The expanded form of the governing equations in generalized coordinates can be obtained using chain rule of partial differentiation. The final form of these equations for fluids with constant properties is included here.

As before $\mathbf{u} = (u, v, w)$ is a velocity vector defined in Cartesian coordinates and h is enthalpy. Let (ξ, η, ζ) be a time-invariant three-dimensional generalized coordinate system generated for a specific flow geometry. The Cartesian form of the governing equations can be written in vector notation as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = 0 \tag{1.63}$$

Here

$$\begin{aligned}
\mathbf{U} &= (0, \rho u, \rho v, \rho w, h), \\
\mathbf{E} &= (u, \rho u^2 + p - \sigma_{xx}, \rho uv - \sigma_{xy}, \rho uw - \sigma_{xz}, hu + q_x), \\
\mathbf{F} &= (v, \rho uv - \sigma_{xy}, \rho v^2 + p - \sigma_{yy}, \rho vw - \sigma_{yz}, hv + q_y) \\
\mathbf{G} &= (w, \rho uw - \sigma_{xz}, \rho vw - \sigma_{yz}, \rho w^2 + p - \sigma_{zz}, hv + q_z).
\end{aligned}$$

The heat flux terms q_x, q_y and q_z are components of the vector $-k\nabla T$ and σ is the stress tensor specified through the constitutive relation (Equation 1.7). The expanded form of the governing equations in (ξ, η, ζ) coordinates can be expressed in a similar form as

$$\frac{\partial \mathbf{U_1}}{\partial t} + \frac{\partial \mathbf{E_1}}{\partial \xi} + \frac{\partial \mathbf{F_1}}{\partial \eta} + \frac{\partial \mathbf{G_1}}{\partial \zeta} = 0 \tag{1.64}$$

with

$$\begin{aligned}
\mathbf{U_1} &= \frac{1}{J}\mathbf{U}, \\
\mathbf{E_1} &= \frac{1}{J}(\mathbf{E}\xi_x + \mathbf{F}\xi_y + \mathbf{G}\xi_z), \\
\mathbf{F_1} &= \frac{1}{J}(\mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{G}\eta_z),
\end{aligned}$$

$$\mathbf{G_1} = \frac{1}{J}(\mathbf{E}\zeta_x + \mathbf{F}\zeta_y + \mathbf{G}\zeta_z).$$

The Jacobian $\mathbf{J}$ of the transformation is the determinant

$$\mathbf{J} = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix} = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}^{-1}$$

Derivatives such as ξ_x and ξ_y are the metric coefficients of the transformation from the (x, y, z) space to the (ξ, η, ζ) space. Vectors $\mathbf{E}, \mathbf{F}$ and $\mathbf{G}$ (Equation 1.63) contain the Cartesian stress tensor σ and the heat flux vector $\mathbf{q}$. The stress components in Cartesian and transformed coordinates is summarized as:

$$\begin{aligned}
\sigma_{xx} &= -p + 2\mu u_x = -p + 2\mu(\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) \\
\sigma_{yy} &= -p + 2\mu v_y = -p + 2\mu(\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta) \\
\sigma_{zz} &= -p + 2\mu w_z = -p + 2\mu(\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta) \\
\sigma_{xy} &= \mu(v_x + u_y) \\
&= \mu((\xi_x v_\xi + \eta_x v_\eta + \zeta_x v_\zeta) + (\xi_y u_\xi + \eta_y u_\eta + \zeta_y u_\zeta)) \\
\sigma_{xz} &= \mu(w_x + u_z) \\
&= \mu((\xi_x w_\xi + \eta_x w_\eta + \zeta_x w_\zeta) + (\xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\zeta)) \\
\sigma_{yz} &= \mu(w_y + v_z) \\
&= \mu((\xi_y w_\xi + \eta_y w_\eta + \zeta_y w_\zeta) + (\xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta))
\end{aligned} \tag{1.65}$$

The corresponding expressions for the heat flux are

$$\begin{aligned}
q_x &= -kT_x = -k(T_\xi \xi_x + T_\eta \eta_x + T_\zeta \zeta_x) \\
q_y &= -kT_y = -k(T_\xi \xi_y + T_\eta \eta_y + T_\zeta \zeta_y) \\
q_z &= -kT_z = -k(T_\xi \xi_z + T_\eta \eta_z + T_\zeta \zeta_z)
\end{aligned} \tag{1.66}$$

It is often necessary to evaluate the metric derivatives in the transformed plane, for example ξ_x and ξ_y in terms of x_ξ and x_η. The relationship connecting the two sets of derivatives is

$$\begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}^{-1}$$

In Equation 1.64, (u, v, w) continue to represent the Cartesian components of velocity. It is convenient to work with velocity components that are locally orthogonal to the transformed coordinates (Figure 1.21). These are called the contravariant components of velocity and are represented as (U, V, W).

The vector $\mathbf{U_c} = (U, V, W)$ can be computed from $\mathbf{U} = (u, v, w)$ using the following formula: $\mathbf{U_c} = (\mathbf{u} \cdot \nabla)\xi$. For example, $U = u\xi_x + v\xi_y + w\xi_z$. A variety of numerical schemes (for example, upwinding) are stated in terms of $\mathbf{U_c}$.

Figure 1.21: Contravariant Components of Velocity

References

Anderson, Jr, J.D., *Fundamentals of Aerodynamics*, McGraw-Hill, New York, 1986.

Batchelor, G.K., *An Introduction to Fluid Dynamics*, Cambridge University Press, UK, 1993.

Bird, R.B., Stewart, W.E. and Lightfoot, E.N. *Transport Phenomena*, Wiley International, New York, 1960.

Busse F.H., Transition to Turbulence in Rayleigh-Benard Convection in *Topics in Applied Physics, Hydrodynamic Instabilities and Transition to Turbulence*, Edited by Swinney H.L. and Gollub J.P., Springer-Verlag, Berlin, 1985.

Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zhang, T.A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1987.

Cebeci, T. and Bradshaw P., *Momentum Transfer in Boundary - Layers*, Hemisphere, Washington DC,1977.

Currie, I.G., *Fundamental Mechanics of Fluids*, McGraw-Hill, New York, 1993.

Drazin, P.G. and Reid, W.H., *Hydrodynamic Stability*, Cambridge University Press, Cambridge, 1981.

Faber, T.E., *Fluid Dynamics for Physicists*, Cambridge University Press, Cambridge, 1995.

Farrashkhalvat, M. and Miles, J. P., *Tensor Methods for Engineers*, Ellis Horwood, New York, 1990.

Greenberg, M.D., *Advanced Engineering Mathematics*, Prentice-Hall International, New Jersey, 1998.

Hinze, J O, *Turbulence*, McGraw Hill,New York, 1975.

Kays, W.M. and Crawford, M.E., *Convective Heat and Mass Transfer*, McGraw-Hill, New York, 1980 (second edition: 1993).

Lamb, H., *Hydrodynamics*, Dover Publications, New York, 1945.

Launder, B. E., *On the Computation of Heat Transfer in Complex Turbulent Flows, ASME Journal of Heat Transfer*, Vol 110, pp 1112 - 1128, 1988.

Landau, L.D. and Lifshitz, E.M., *Fluid Mechanics*, Pergamon, UK, 1984.

Lesieur, M., *Turbulence in Fluids*, Kluwer Academic Publishers, New York, 1990.

Moin, P., *Progress in Large Eddy Simulation of Turbulent Flows*, AIAA paper 97-0749, 1997.

Panton, R. L., *Incompressible Flow*, John Wiley, New York, 1984.

Rodi, W., *Turbulence Models and Their Applications in Hydraulics*, IAHR, The Netherlands, 1984.

Rodi, W. and Laurence, D., Editors,*Engineering Turbulence Modeling and Experiments 4*, Elsevier, Amsterdam, 1999. Also see proceedings of the symposia 1-3.

Schlichting, H., *Boundary-layer Theory*, McGraw-Hill, New York, 1979 (8th edition: 2000).

Shames, I.H., *Mechanics of Fluids*, McGraw-Hill, New York, 1989.

Slattery, J.C., *Advanced Transport Phenomena*, Cambridge University Press, Cambridge, 1999.

Ockendon, H. and Ockendon, J. R., *Viscous Flow*, Cambridge University Press, London, 1995.

Schlichting, H., *Boundary-Layer Theory*, McGraw Hill, New York, 1979; also see the 8th revised and enlarged edition by H. Schlichting and K. Gersten, 2000.

Siegel, R. and Howell, J.R., *Thermal Radiation Heat Transfer*, Hemisphere Publishing Corporation, New York, 1981.

Modest, M.F., *Radiative Heat Transfer*, McGraw-Hill, New York, 1993.

Sneddon, I., *Elements of Partial Differential Equations*, McGraw Hill, New York,1986.

Speziale, C. G. and Thangam, S., Analysis of an RNG based Turbulence Model for Separated Flows, Int. J. Engg. Science, Vol 30, pp 1379 - 1388, 1992.

Van Dyke, M., *An Album of Fluid Motion*, The Parabolic Press, California, 1982.

Warsi, Z. U. A., *Fluid Dynamics: Theoretical and Computational Approaches*, CRC Press, Boca Raton, 1992.

White, F.M., *Viscous Fluid Flow*, McGraw-Hill, New York, 1991.

Wilcox, D.C., *Turbulence Modeling for CFD*, DCW Industries, California, second edition, 1998.

Problems

1. Derive the continuity, Navier-Stokes and energy equations in Cartesian, cylindrical and spherical coordinates beginning from their respective coordinate-free forms.
2. Derive the Poisson equation of pressure for incompressible flow by performing the curl operation on the coordinate-free form of the Navier-Stokes equations. Expand the resulting equation in Cartesian, cylindrical and spherical coordinates.
3. Starting with the three dimensional incompressible Stokes equations derive the depth-averaged pressure equation

$$\nabla \cdot h^3 \nabla p = 0$$

for flow a thin channel of variable aperture $h(\mathbf{x})$. Here $\nabla = (\partial/\partial x, \partial/\partial y)$ and $\mathbf{x} = (x, y)$ is the plane defining the channel.

4. 3. Consider free surface flow of water over a wavy solid surface. Write the boundary conditions applicable on the free surface. How do these conditions change in the presence of surface tension. In most cases, the coefficient of surface tension is a function of temperature. If the temperature over the free surface varies with position, develop the boundary conditions for velocity. Consider planar as well as non-planar surfaces. Generalize the result for a pair of immiscible fluids.

5. a. Consider the configuration of one flow superimposed on another, the fluids being identical and velocities in each layer being constant but distinct. At the inviscid limit show that the system is unconditionally unstable.

   b. Show that a stationary liquid held in a container is inherently stable by proving the inequality, $dk/dt < 0$ where the kinetic energy of the liquid is

$$k = \int_{\Omega} \frac{1}{2} u^2 d\mathbf{x}$$

   c. Show that a heavier fluid placed above a lighter fluid constitutes an unstable configuration.

6. a. Develop wall functions for k and ε in low Reynolds number flows using non-constant total shear stress near the wall as shown in Figure 1.18c.

   b. Write the expanded form of the Reynolds stress equations when the mean flow is two dimensional.

   c. Repeat part b for the algebraic Reynolds stress model.

7. Consider a glassy medium that is at an elevated temperature with respect to the ambient. The body is transparent to visible radiation, but loses heat to the ambient not only at its surface, but also from the interior. Develop an expression for the radiative flux at any point in the interior of the body. Examine the cases of spectral versus total, specular versus diffuse, and scattering separately. The optically thick and thin approximations are also to be individually studied. For an introduction to the subject of radiative transfer from participating media, see Siegel and Howell (1981) and Modest (1993).

2
Applied Numerical Methods

The present chapter reviews a few of the basic numerical techniques available for integration, root-finding and solution of simultaneous algebraic equations. For the sake of brevity only important methods which are frequently encountered in the solution of fluid flow and heat transfer problems are included. For a comprehensive discussion of the available methods the reader must consult the literature listed in the references.

2.1 Numerical Integration

Consider the evaluation of a definite integral

$$I = \int_a^b f(x)\,dx \tag{2.1}$$

for which the indefinite integral $\int f(x)\,dx$ is not known. We assume that the integral itself exists and has a finite value. Numerical integration refers to determination of I using values of $f(x)$ at distinct points without algebraically specifying the integral $\int f(x)dx$. Since the quantity $1/(b-a)\int_a^b f(x)\,dx$ is the average of the function $f(x)$ one may determine the integral as

$$I = \frac{(b-a)}{N}\sum_{i=1}^{N} f(x_i) \tag{2.2}$$

where x_i are equally spaced points in the interval (a, b) and N is the total number of points. The quantity $\frac{1}{N}\sum_i f(x_i)$ is the arithmetic average of $f(x)$. The notation $f_i = f(x_i)$ is frequently used to represent the value of the function at the ith x-location. Equation 2.2 provides an approximate value of the integral I. It is clear that the formula is exact if f is a constant or a linear function of

x. In all other cases the computed value of I and its analytical value will be separated by an amount called the *error* of numerical integration.

Numerical integration can also be performed by interpreting the integral as magnitude of the area under the curve $f(x)$ between the limits of integration. The interval (a, b) can be sub-divided into m strips of equal thickness ($\Delta x = (b-a)/m$) and the area of each strip estimated as $(f_i + f_{i+1})\,\Delta x/2$, $i = i, \ldots N$ and $N = m + 1$. This is equivalent to approximating the function as a line between points x_i and x_{i+1}. The error in the area computed using the formula given above is $(\Delta x)^3\, f_i''/12$. When all the strips are assembled one obtains the following result:

$$I = \int_a^b f(x)\,dx = \frac{b-a}{2(N-1)}\,\{f_i + f_N + 2(f_2 + f_3, \ldots + f_{N-1})\} \tag{2.3}$$

Equation 2.3 is called Trapezoidal rule. The total error in the value of I is $\Delta x^3 \sum_{i=1}^{N} f_i''/12$ which is approximately $\Delta x^2\,(b-a) f''(\max)/24$. Based on the exponent of Δx, this suggests a second order convergence rate for the integral. Trapezoidal rule is exact for functions that are constant or linear in x since $f'' = 0$ in these cases.

Integration formulas can be generated by fitting polynomials of a higher degree through a set of data points (x_i, f_i) followed by an analytical evaluation of the integral (Carnahan *et al.*, 1969). For example a quadratic fit through points $i, i+1$ and $i+2$ yields the result

$$\delta\, I(i) = \int_i^{i+2} f(x)\,dx = \frac{\Delta x}{3}\,(f_i + 4 f_{i+1} + f_{i+2})$$

with an error of $-\Delta x^5\, f_i''''/90$. Assembling all the part-integrals results in the popular Simpson's rule

$$\begin{aligned} I = \int_a^b f(x)\,dx &= \frac{\Delta x}{3}\,[f_1 + f_N + 4\,(f_2 + f_4 \ldots f_{N-1}) \\ &+ 2\,(f_3 + f_5 \ldots f_{N-2})] \end{aligned} \tag{2.4}$$

Here N, the number of points at which the function is evaluated must be odd; the error in approximation is of the order of $\Delta x^4\,(b-a)\,f''''(\max)/180$. Hence the convergence rate is of the fourth order. Integration is exact if f is a polynomial of degree three or lower since $f'''' = 0$ in such cases.

Both trapezoidal and Simpson's rules can be compactly expressed as

$$I = s_x \sum_{i=1}^{N} W_i\, f_i$$

where s_x is a scale factor and W_i are weights multiplying the corresponding value of the function $f(x)$ at x_i. For example, in Simpson's rule $W_i = 1$ if

$i = 1$ or N, $W_i = 4$ if i is even and $W_i = 2$ if i is odd. The scale factor s_x is $\Delta x/3$. With this notation Simpson's rule can be generalized to the evaluation of multiple integrals as follows. For definiteness we consider a double integral

$$\begin{aligned} I &= \int_c^d \int_{a(y)}^{b(y)} f(x,y)\,dx\,dy \\ &= s_y \sum_{j=1}^{N} W_j \left\{ s_{xj} \sum_{i=1}^{N} W_i\, f(x_i, y_j) \right\} \end{aligned} \tag{2.5}$$

Here $s_y = (d-c)/3(N-1)$ and $s_{xj} = (b(y_j) - a(y_j))/3(N-1)$. Equation 2.5 assumes that both integrals are evaluated using N points, though this is not necessary. If the limits a and b are constants Equation 2.5 simplifies to

$$I = \frac{(b-a)\,(d-c)}{9\,(N-1)^2} \sum_{j=1}^{N} \sum_{i=1}^{N} W_i\, W_j\, f(x_i, y_j)$$

Integration over a complex shape can be accomplished by mapping this region to one with regular boundaries. Mapping transformations can be analytically developed or established numerically using techniques described in Chapter 13. Let (x, y) represent the coordinate system in the physical region and (ξ, η) be the transformed coordinate system. Then

$$\begin{aligned} I &= \int\int f(x,y)\,dx\,dy \\ &= \int\int f(\xi,\eta)\,J(\xi,\eta)\,d\xi\,d\eta \end{aligned}$$

where J, the Jacobian of the transformation is $x_\xi\, y_\eta - x_\eta y_\xi$. Here we use the notation $x_\eta = \partial x/\partial \eta$ and so on. Since the shape of the physical region in (ξ, η) is regular, the second integral can be easily evaluated using Equation 2.5 by interpreting the product $f \times J$ as the new integrand.

Other complications can arise in the evaluation of definite integrals. If $f(x)$ is singular in the interval (a, b), the integral can still exist when evaluated analytically; it will however be meaningless if any of the approximate methods that employ individual values of $f(x)$ are employed. One method of circumventing this difficulty is to split the integral at the singular point (say c) as

$$I = \int_a^b f(x)\,dx = \int_a^{c-\epsilon} + \int_{c-\epsilon}^{c+\epsilon} + \int_{c+\epsilon}^{b} \tag{2.6}$$

Here ϵ is a small number. The first and third integrals can be numerically evaluated. The second integral must be determined analytically. Similarly if one of the limits is infinity we compute I as

$$I = \int_a^\infty f(x)\,dx = \int_a^b + \int_b^\infty \tag{2.7}$$

where the second integral is determined analytically using the limiting form of f for large x.

By a proper choice of the weights as well as the point at which the function is evaluated it is possible to gain accuracy with very few integration points. A class of such methods where integration with N points results in an exact answer for polynomials of degree upto $(2N-1)$ is described below. These methods go by the general name of *Gaussian quadrature*.

Consider the evaluation of the definite integral, $I = \int_{-1}^{1} f(y)\,dy$. The use of $(-1, 1)$ as the limits of integration is not a restriction since any integral of the form given in Equation 2.1 can be transformed using a new dependent variable $y = 2(x-a)/(b-a) - 1$. More specifically

$$I = \int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f(y)\,dy,$$

where y is related to x through the formula given above. The normalized integral is evaluated as $\sum_{i=1}^{N} W_i\, f(y_i)$, where the values of y_i and W_i depend on the choice of N. For $N = 3$ and 5 these values are given in Table 2.1. Double integrals can be evaluated using Equation 2.5. If the region of integration is a straight or a curvilinear triangle the integral can be evaluated using a single summation operator as

$$I = \int\int f(x,y)\,dx\,dy = \sum_{i=1}^{N} W_i\, f(x_i, y_i) \tag{2.8}$$

The values of W_i, x_i and y_i for a normalized right-angled triangle with two edges of unit length are given in Table 2.2. The mapping from a generalized curvilinear triangle to a normalized triangle is easily accomplished using polynomial interpolation (see Chapter 8).

When finite number of digits are used in the weights and the integration points of Gaussian quadrature, the error in the numerical value of the integral is approximately, (remainder)$\times(2N-2)$th derivative of f. This suggests that a large number of digits of the weights and the integration points must be carried when f is a function that cannot be represented as a polynomial.

We illustrate below the application of the results for numerical integration through examples. Consider the evaluation of the integrals $I_1 = \int_0^4 x(1 -$

Table 2.1: Base Points and Weights for Gaussian Quadrature

$N = 3$	y_i	±0.77459 66692	0.0	0.0
	W_i	0.55555...	0.88888...	0.0
$N = 5$	y_i	±0.90617 98459	±0.53846 93101	0.0
	W_i	0.23692 88850	0.47862 86704	0.56888...

Table 2.2: Integration Over a Right Triangle

$N = 3$	(x_i, y_i)	(0,1/2)	(1/2,1/2)	(1/2,0)	
	W_i	1/3	1/3	1/3	
$N = 7$	(x_i, y_i)	(0,0)	(0,1)	(1,0)	(0,1/2)
	W_i	1/40	1/40	1/40	1/15

$N = 7$	(x_i, y_i)	(1/2,0)	(1/2,1/2)	(1/3,1/3)
	W_i	1/15	1/15	9/40

$x)^3\,dx$, $I_2 = \int_0^1 \log x\,dx$ and $I_3 = \int_0^\infty \sin x/x\ dx$. Each of these integrals has a value of $I_1 = -68.8$, $I_2 = -1$ and $I_3 = \pi/2$. The integrand in I_1 is well-behaved and the integral has finite limits. The integrand in I_2 is singular at the lower limit; the integral I_3 is complicated by the fact that the upper limit is at infinity. To use the results discussed in the present section we rewrite

$$\begin{aligned} I_2 = \int_0^1 \log x\,dx &= \int_\epsilon^1 \log x\,dx + \int_0^\epsilon \log x\,dx \\ &= \int_\epsilon^1 \log x\,dx + \epsilon \log \frac{\epsilon}{2}. \end{aligned}$$

A typical value of ϵ is 10^{-4}. The treatment of I_3 comprises splitting the range of integration in such a way that the integrand can be approximated by simpler functions for large values of the independent variable. In the present example the finite interval in x is chosen as periodic with a period of 2π. It can be shown that

$$I_3 = \sum_{i=1}^{M} \int_0^{2L\pi} \frac{\sin x}{x + 2L(i-1)\pi}\,dx + \int_0^\infty \frac{\sin x\,dx}{x + 2LM\pi}$$

The second integral is small for large values of the product LM. Results are given below for $L = M = 2$ and the remainder integral assumed to be zero.

Results presented in Tables 2.3, 2.4, 2.5 and 2.6 are to be interpreted in the following manner. The numerically computed value of the integral, the magnitude of the error e in relation to the exact value of the integral and the ratio $p = e/\Delta x^n$ are listed for each of the methods considered in the present section. In the definition of p, n is chosen as the convergence rate of the integration scheme, namely unity for arithmetic average, 2 for trapezoidal rule and 4 for Simpson's rule. n is undefined for Gaussian quadrature. For functions of polynomial order p is a constant; for other functions p changes with N, the number of integration points indicating the absence of monotone convergence. If p increases with N it shows that the error does not decrease proportionally as the interval Δx is reduced. This is to be expected when the error in numerical

Table 2.3: Integral Evaluation using Arithmetic Average

N	5	11	21	101	501
I_1	−107.2	−83.82	−76.24	−70.27	−69.09
$e\%$	55.81	21.84	10.81	2.14	0.428
p	55.81	54.6	54.07	53.61	53.51
I_2	−1.425	−1.163	−1.075	−1.0109	−0.999
$e\%$	42.5	16.3	7.5	1.09	0.1
p	171.7	165.1	152.36	110.45	18.7
I_3	2.513	1.968	1.758	1.578	1.5406
$e\%$	59.98	25.28	11.91	0.45	1.92
p	19.09	20.12	19.03	3.76	76.44

Table 2.4: Integral Evaluation using Trapezoidal Rule

N	5	11	21	101	501
I_1	−80.0	−70.609	−69.25	−68.81	−68.80
$e\%$	16.27	2.62	0.65	0.01	0
p	16.27	16.44	16.46	16.45	16.45
I_2	−1.198	−1.001	−1.012	−0.997	−0.997
$e\%$	19.8	0.1	1.2	0.3	0.3
p	323.4	236.27	512.74	2328.2	77447.7
I_3	1.5707	1.536	1.532	1.531	1.531
$e\%$	0	2.21	2.46	2.53	2.53
p	1.53×10^{-6}	1.382	6.18	159.69	3997.5

integration is mainly contributed by the error in the analytic estimate of the remainder term, for example $\epsilon \log \epsilon/2$ in I_2.

Gaussian quadrature shows remarkable accuracy with very few integration points and is hence invariably used in applications where integrals must be repeatedly evaluated. For this reason it is of interest to know in detail the method of calculation of the weights W_i and the points y_i at which the function is to be determined. The central ideas involved in computing I in Equation 2.1 are the following. Let $L_i(x) = \Pi_{j=0\, j\neq i}^{M} \frac{x-x_j}{x_i-x_j}$, where Π denotes a product of factors such as $(x-x_j)/(x_i-x_j)$. With the limits of integration set as $(-1,1)$ one can expand the integrand as

$$f(x) = \sum_{i=0}^{M} L_i(x)\, f_i + \left[\Pi_{i=0}^{M}\,(x-x_i)\right] \left[\frac{f^{M+1}(\xi)}{(M+1)!}\right] \tag{2.9}$$

where $-1 < \xi < 1$. The first part of Equation 2.9 is the principal estimate of f using $M+1$ points and the second part is the remainder and a source of error in integration. Note that the index i ranges from 0 to M making L_i a polynomial of degree $M+1$. The reason for this is as follows: Quadrature formulas invariably

Table 2.5: Integral Evaluation using Simpson's Rule

N	5	11	21	101	501
I_1	−69.33	−68.81	−68.80	−68.80	−68.80
$e\%$	0.77	0.014	0	0	0
p	0.775	0.775	0.776	0.776	0.776
I_2	−1.847	−1.318	−1.153	−1.027	−1.003
$e\%$	84.7	31.8	15.3	2.7	0.3
p	2.2×10^4	3.3×10^5	2.5×10^6	2.8×10^8	1.96×10^{10}
I_3	3.1416	2.368	1.949	1.614	1.547
$e\%$	100.0	50.75	24.07	2.75	1.51
p	1.026	20.35	154.88	11261.1	3.65×10^6

Table 2.6: Integral Evaluation using Gaussian Quadrature

N	3	5
I_1	−68.79	−68.79
$e\%$	1.07×10^{-3}	5.23×10^{-3}
I_2	−0.965	−0.989
$e\%$	3.43	1.0869
I_3	2.227	1.5518
$e\%$	41.78	1.209

use orthogonal polynomials which are constants but non-zero when their degree is chosen as zero. Functions $L_i(x)$ are the Lagrange interpolation polynomials of degree i and Equation 2.9 stems directly from the theory of interpolation. The $(M+1)$ points x_i $(i = 0, 1, \ldots, M)$ are not necessarily equally spaced in this development. They must be so chosen as to make the integral of the error equal to zero for functions $f(x)$ that are polynomials of degree $(2M+1)$ or less. Hence we require, the error integral

$$I_E = \int_{-1}^{1} \left[\Pi_{i=0}^{M} (x - x_i)\right] \left[\frac{f^{M+1}(x)}{(M+1)!}\right] dx = 0 \tag{2.10}$$

if $f(x)$ is a polynomial of degree less than $2M+1$. It can be shown that Equation 2.10 is certainly valid if x_i are chosen as the $(M+1)$ roots of $(M+1)$th degree Legendre polynomial $P_{M+1}(x)$. The Legendre polynomials are defined through the recursion formula

$$P_0 = 1, \quad P_1 = x \quad \text{and} \quad P_i(x) = \frac{2i-1}{i}\, x\, P_{i-1}(x) - \frac{i-1}{i}\, P_{i-2}(x), \quad i > 1$$

Legendre polynomials fall in the class of orthonormal polynomials that satisfy the conditions

$$\int_{-1}^{1} P_i(x)\, P_j(x)\, dx \;=\; 0, \quad i \neq j$$

$$= \quad 1 \quad \text{otherwise.}$$

Any arbitrary polynomial $f_m(x)$ of degree m can be represented over the interval $-1 < x < 1$ as a linear combination of $P_i(x)$, $i = 0, \ldots, m$ in the form

$$f_m(x) = \sum_{i=0}^{m} b_i \, P_i(x)$$

where

$$b_i = \int_{-1}^{1} f_m(x) \, P_i(x) \, dx \tag{2.11}$$

The function $\Pi_{i=0}^{M} (x - x_i)$ appearing in Equation 2.10 is a polynomial of degree $M + 1$; assuming the integrand $f(x)$ to be a polynomial of degree $2M + 1$ makes $f^{M+1}(x)$ a polynomial of degree M. Hence both terms in the integral of Equation 2.10 can be expanded using Legendre's polynomials as a basis. Using the orthogonality condition $\int_{-1}^{1} P_i \, P_j \, dx = 0$ $(i \neq j)$ the error integral reduces to

$$I_E = \sum_{i=0}^{M} \int_{-1}^{1} b_i \, c_i \, P_i^2(x) \, dx$$

where we have used the expansions,

$$\Pi_{i=0}^{M} (x - x_i) = \sum_{i=0}^{M+1} b_i \, P_i(x)$$

and

$$\frac{f^{M+1}(x)}{(M+1)!} = \sum_{j=0}^{M} c_j \, P_j(x)$$

The base points x_i are as yet unspecified. If they are chosen as the roots of $P_{M+1}(x)$ the form $\Pi_{i=0}^{M} (x - x_i)$ is simply the factored version of $P_{M+1}(x)$, except for a constant scale factor. Using Equation 2.11 this results in

$$b_i = 0 \quad i = 0, 1, \ldots, M$$

and

$$b_{M+1} = \int_{-1}^{1} \Pi_{i=0}^{M} (x - x_i) \, P_{M+1}(x) \, dx$$

Hence I_E is identically zero, irrespective of the magnitude of c_i.

The Gaussian quadrature using Legendre polynomials (called *Gauss-Legendre quadrature*) can now be developed as follows.

$$I = \int_{-1}^{1} f(x) dx = \sum_{i=0}^{M} W_i \, f(x_i)$$

where

$$W_i = \int_{-1}^{1} \Pi_{j=0,\, j\neq i}^{M} \frac{(x - x_j)}{(x_i - x_j)}\, dx$$

and x_i are roots of the $(M+1)$th degree Legendre polynomial. It must be noted that N appearing in Table 2.1 is not independent of M and $N = M + 1$. This is related to starting the summation and product index from 0 and not 1.

Gaussian quadratures can be developed for a larger class of integrals of the form

$$I = \int_a^b w(x)\, f(x)\, dx \tag{2.12}$$

Here $w(x)$ is a weighting function that is required to be positive (or of constant sign) over the integration interval. The integration formula continues to be

$$I = \sum_{i=0}^{M} W_i\, f(x_i)$$

The central idea that is employed is the use of polynomials that are orthonormal over the interval of (a, b) with respect to the weight function $w(x)$. Two examples of Equation 2.12 are given below.

Gauss-Chebyshev quadrature:

Here the integration formula is

$$I = \int_{-1}^{1} \frac{1}{\sqrt{1 - x^2}}\, f(x)\, dx = \sum_{i=0}^{M} W_i\, f(x_i)$$

where $W_i = \frac{\pi}{M+1}$ (constant) and x_i are the roots of $(M+1)$th degree Chebyshev polynomial. This polynomial is given by the recursion formula

$$\begin{aligned} T_0 &= 1, \quad T_1 = x \quad \text{and} \\ T_i(x) &= 2xT_{i-1}(x) - T_{i-2}(x), \quad i > 1. \end{aligned}$$

The roots of $(M + 1)$th degree Chebyshev polynomial are

$$x_i = \cos \frac{(2i + 1)\, \pi}{2M + 2}$$

Gauss-Laguerre quadrature:

Here the integration formula is

$$I = \int_0^{\infty} e^{-x}\, f(x)\, dx = \sum_{i=0}^{M} W_i\, f(x_i)$$

Table 2.7: Base Points and Weights for Gauss-Laguerre Quadrature

$M = 2$	x_i	0.41577 45567	2.29428 03602	6.28994 50829
	W_i	0.71109 30099	0.27851 77335	0.01038 92565
$M = 4$	x_i	0.26356 03197	1.41340 30591	3.59642 57110
	W_i	0.52175 56105	0.39866 68110	0.07594 24497
	x_i	7.08581 00058	12.64080 08442	
	W_i	0.00361 17587	0.00002 33699	

where

$$W_i = \int_0^\infty e^{-x} \Pi_{j=0,\, j\neq i}^{M} \frac{(x - x_j)}{x_i - x_j}\, dx$$

and x_i are the roots of $(M+1)$th order Laguerre polynomial. This polynomial is defined as

$$L_i(x) = (2i - x - 1)\, L_{i-1}(x) - (i-1)^2\, L_{i-2}(x), \quad i > 1$$

and $L_0 = 1$ and $L_1 = -x + 1$. The roots of L_i and the associated weights for $M = 2$ and 4 (corresponding to $N = 3$ and 5 of Table 2.1) are given in Table 2.7.

The Fourier integral $\int_0^\infty f(x) \sin \omega x\, dx$ cannot be computed using Gaussian quadrature since $\sin \omega x$ is not a function of constant sign. It is best computed using the FFT algorithm described below (Press *et al.*, 1986).

While higher order integration formulas are quite popular, the arithmetic average finds utility in calculation of Fourier integrals of the form, $I(\omega) = \int_{-\infty}^{\infty} f(t) \exp(-j\omega t)\, dt$. Here the independent variable is chosen as time t, ω is frequency and $j = \sqrt{-1}$. Specifically N equally spaced values of f are given at instants of time separated by a time increment Δt. We compute I at each value of $\omega = k/N\Delta t$, $k = 1, 2, \ldots, N$ using Equation 2.2 as

$$I(\omega) = \sum_{i=1}^{N} f_i \ \exp\ (-j\omega(i\Delta t))$$

Clearly a total of N^2 multiplications are required for computing all values of $I(\omega)$. However if N is of the form 2^n for integer n, the number of multiplications can be reduced to $N \log_2 N$ through the use of a clever algorithm called Fast Fourier Transform (FFT). The difference between N^2 and $N \log_2 N$ is significant if N is large, Table 2.8. The FFT algorithm proceeds along the following lines.

The time series f_i, $i = 1, \ldots, N$ is split into two sub-sequences f_{r1} and f_{r2} where $r1 = 1, 3, 5, \ldots$ and $r2 = 2, 4, \ldots$. Since $N = 2^n$ the number of elements in f_{r1} and f_{r2} will be each equal to 2^{n-1}. The discrete Fourier transform of $f(t)$ is related to those of f_{r1} and f_{r2} through the formula,

$$I_k = I\,(\omega = k/N\Delta t) = \frac{1}{2}\left[I_{1k} + W^{k-1}\, I_{2k}\right]$$

$$I_{k+N/2} = I\left(\omega = \frac{k+\frac{N}{2}}{N\Delta t}\right) = \frac{1}{2}\left[I_{1k} - W^{k-1}\, I_{2k}\right]$$

Here $k = 1, 2, \ldots, N/2$, I_{1k} and I_{2k} are Fourier transforms of the subsequences f_{r1} and f_{r2} and $W = \exp\,(-2\pi\, j/N)$. In the FFT algorithm each subsequence is further split into half-sequences till finally only one term $(= 2^o)$ is left. The discrete Fourier transform of the one-term sequence is the term itself. Hence it is possible to reconstruct from N one-term series the Fourier integral of the N-term series of f values. This is illustrated by an example. Let $N = 4$, $f_i = (f_1, \ldots, f_4)$. The FFT algorithm is schematically as shown in Figure 2.1.

2.2 Roots of a Function

It is frequently required to determine the value of x at which a function $f(x)$ is zero. This value is called the root of the function $f(x)$. A commonly used method for determining the root is the *Newton-Raphson* scheme. It estimates the new value of the root by using the local tangent direction of the function evaluated at the assumed value of the root. Improvement in the estimate of the root is iteratively carried out. This is shown in Figure 2.2. Let p be the point at which f is zero. Let x_1 be the initial guess for the root and let $f_1 = f(x_1)$ and $f_1' = df/dx$ at $x = x_1$. The improved value of the root x_2 is obtained from the formula

$$\tan\theta = f_1' = f_1/(x_1 - x_2)$$

and so

$$x_2 = x_1 - (f_1/f_1')$$

In general, the root is updated by $\Delta x = -f_n/f_n'$, that is

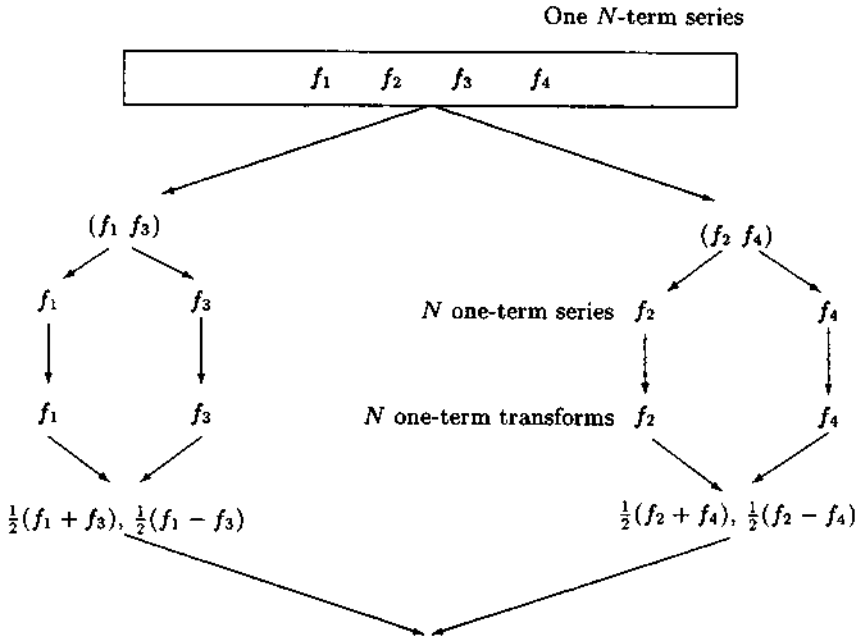
$$x_{n+1} = x_n - f_n/f_n' \tag{2.13}$$

Iterations continue till $\mid x_{n+1} - x_n \mid$ and f_{n+1} are both small. It is quite clear that the iterations will converge only if f' is non-zero.

Newton-Raphson scheme converges rapidly in general as the root is approached. It works best when a reasonable initial guess of $x(P)$ is available.

Table 2.8: Speedup Using FFT

N	N^2	$N \log_2 N$
64	4096	384
128	16384	896
256	65536	2048
572	262144	4608
1024	1048576	10240



$$I_1 = \tfrac{1}{4}(f_1 + f_2 + f_3 + f_4)$$

$$I_2 = \tfrac{1}{2}(f_1 - f_3 - j(f_2 - f_4))$$

$$I_3 = \tfrac{1}{2}(f_1 + f_3 - j(f_2 + f_4))$$

$$I_4 = \tfrac{1}{4}(f_1 - f_3 + j(f_2 - f_4))$$

æ

Figure 2.1: The FFT Algorithm (with $j = \sqrt{-1}$)

This estimate is usually available from the physical problem being studied. In the event of multiple roots the direction of convergence is determined by the initial guess. Again a prior knowledge of the separation of roots is useful in forcing convergence of the iterations to a particular one of them. For example it is known that the roots of the zeroth order Bessel function $J_0(x)$ are separated nearly by π. Alternatively if X_1 is one root of $f(x) = 0$ the search for the second root can be initiated using the equation $f(x)/(x - X_1) = 0$.

The Newton-Raphson scheme of root finding may not converge for certain initial guesses. In fact associated with every root finding method is a 'radius of convergence' centred around the root. Values of initial guesses outside this radius will lead to divergence of the iterations.

It is quite difficult to know beforehand the radius of convergence for any root of a given function. Hence it is not possible to determine in advance if the

initial guess will lead necessarily to convergence. Underrelaxation can stabilize the iterations if the extent of fluctuations in the estimates of the root between successive iterations is too large. Underrelaxation consists of updating the root by an amount $\alpha\Delta x$ where $0 < \alpha < 1$. For the original unrelaxed Newton-Raphson scheme, $\alpha = 1$.

If $f(x)$ is known as an algebraic expression $f'(x)$ can be determined in closed form. If this expression is not available or is too lengthy to be differentiated, the derivative is calculated using finite differences (Chapters 4 and 5) as

$$f'(x) = (f(x+\epsilon) - f(x))/\epsilon$$

where ϵ is a small value, say $\epsilon = 0.001x$.

Example 2.1

Consider the application of the Newton-Raphson scheme to the problem of root finding for the function

$$f(x) = x^3 + 4x^2 - 6x + 1 = 0$$

The iteration number is denoted as n. For this problem f' is $3x^2 + 8x - 6$. Results obtained using different initial estimates for the root are summarized in Table 2.9. Values of f after the root has converged to within 0.001 percent between successive iterations is also given here. Note that the root approached by the scheme depends on the initial guess.

Example 2.2

The following polynomial is encountered in the study of boundary-layer equations using the momentum-integral technique.

$$f(x) = \left(\frac{37}{315} - \frac{1}{945}x - \frac{1}{9072}x^2\right)\left(2 - \frac{116}{315}x + \left(\frac{2}{945} + \frac{1}{120}\right)x^2 + \frac{2}{9072}x^3\right)$$

Clearly it is cumbersome to differentiate f. It is more convenient to use a finite difference representation of f'. The roots of f and its residual values are given in

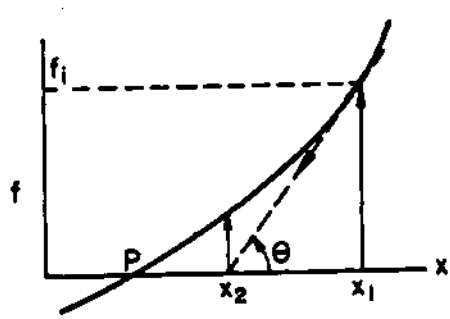


Figure 2.2: Schematic representation of the Newton-Raphson Method

Table 2.9: Results for Example 2.1

n	0	1	2	4	6
x	5	3.203	2.097	1.139	1.0005
x	20	12.91	8.326	3.41	1.53
x	−3	6.41	4.09	1.768	1.0675
x	−4	−6.501	−5.54	−5.193	−5.1925

n	8	10	f (residual)
x	1.0	1.0	0
x	1.028	1.0	0
x	1.00005	1.0	0
x	−5.1925	−5.1925	-7.63×10^{-6}

Table 2.10. The roots of a pair of simultaneous equations $f(x, y) = g(x, y) = 0$ is determined as follows.

Moving along the tangent direction is equivalent to moving along a tangent plane in a many-variable problem. On this plane the tangent is represented by the total differential of the function. Using rules of partial differentiation we get

$$df \simeq \Delta f = f_x\,\Delta x + f_y\,\Delta y \quad \text{and}$$
$$dg \simeq \Delta g = g_x\,\Delta x + g_y\,\Delta y$$

where $f_x = \partial f/\partial x$, $f_y = \partial f/\partial y$ and so on. If x and y are the present estimates of the roots, we need to find the corrections Δx and Δy in such a way as to make the new values of f and g, namely $f(x+\Delta x,\ y+\Delta y)$ and $g(x+\Delta x,\ y+\Delta y)$, equal to zero. Hence $df = f(\text{new}) - f(\text{old}) = -f(\text{old})$ and $dg = g(\text{new}) - g(\text{old}) = -g(\text{old})$ and the equations governing Δx and Δy become

$$f_x\,\Delta x + f_y\,\Delta y = -f(x, y)$$

$$g_x\,\Delta x + g_y\,\Delta y = -g(x, y)$$

Table 2.10: Results for Example 2.2

n	0	1	2	3	4	f (residual)
x	6	6.9629	7.05227	7.05227	7.05227	1.08×10^{-6}
x	8	6.9239	7.04950	7.05232	7.05232	9.65×10^{-8}
x	−4	4.1519	6.521	7.0289	7.0521	2.27×10^{-8}

Solving for the corrections Δx and Δy we get

$$\Delta x = (f g_y - g f_y)/\det$$

$$\Delta y = (g f_x - f g_x)/\det$$

where $\det = g_x f_y - f_x g_y$. These corrections are repeatedly applied till $|\Delta x|$ and $|\Delta y|$ are themselves small and f and g are close to zero. Clearly the scheme will converge only if the quantity det in the denominator is non-zero.

Example 2.3

Consider f and g given as $(x^2 + 3xy + y^3 - 77)$ and $(10x + 4y + \sqrt{y} - 28)$. The iterative improvement in the initial values (x, y) using Newton-Raphson scheme are given in Table 2.11. Calculations are stopped when the roots converge to within 0.001% between successive iterations.

If complex roots are to be found it is convenient to identify the independent variable as z and $z = x + jy$ where $j = \sqrt{-1}$. Finding the complex root of a function now corresponds to finding its real and imaginary parts. These are determined by forming the system of equations.

$$\text{Real } (f(z)) = \text{Imaginary } (f(z)) = 0.$$

(Also see Example 2.4).

Example 2.4

Let us find the complex eigenvalues of the matrix,

$$A = \begin{bmatrix} 5 & 8 & 2 \\ 3 & 3 & -4 \\ 1 & 6 & 2 \end{bmatrix}.$$

All eigenvalues of this matrix will be roots of the equation $\det(A - \lambda I) = 0$. Expanding the determinant we get

$$(5 - \lambda)\ ((3 - \lambda)(2 - \lambda) + 24) - 8\ (3(2 - \lambda) + 4) + 2\,(18 - (3 - \lambda)) = 0$$

Table 2.11: Results for Example 2.3

n	0	1	2	3	4	5	6	Residual
x	3	0.679	0.952	0.998	0.9999			$f = 3.05 \times 10^{-4}$
y	6	4.747	4.1106	4.0029	4.0001			$g = -9.5 \times 10^{-6}$
x	-1	-0.775	0.231	0.781	0.977	0.9996	1.0	$f = 0$
y	12	8.22	5.81	4.508	4.054	4.0008	4.0	$g = 1.9 \times 10^{-6}$

Hence, $f(z) = z^3 - 10z^2 + 29z - 100$ for the present problem. Writing $\lambda = x + jy$ and separating f into real and imaginary parts the system of equations is obtained as

$$\begin{aligned} \text{Real}\,(f(z)) &= x^3 - 3xy^2 - 10(x^2 - y^2) + 29x - 100 = 0 \\ \text{Imaginary}\,(f(z)) &= 3x^2 - y^2 - 20x + 29 = 0 \end{aligned}$$

The roots of this system of equations are (1.033, 3.396) and the complex eigen values of matrix A are $(1.033 \pm 3.396j)$.

The procedure developed above for two coupled equations, can be generalized to a system of n coupled equations. Hence the n roots of the system of equations

$$f_i\,(x_1, \ldots, x_n) = 0 \tag{2.14}$$

can be iteratively improved by solving the matrix equation

$$\left[\frac{\partial f_i}{\partial x_j}\right] \{\delta x_j\} = -\{f_i\} \tag{2.15}$$

starting with an initial guess $(\bar{x}_1, \ldots \bar{x}_n)$. In Equation 2.15 δx_j is the correction to the jth value x_j and $\partial f_i / \partial x_j$ is the entry at the ith row and jth column position of the coefficient matrix. Convergence of the Newton-Raphson scheme now requires the matrix $[\partial f_i / \partial x_j]$ to be non-singular, i.e. the determinant is required to be non-zero.

Equations for improving the estimate of roots of a system of equations can also be set up using the least squares technique. In the context of similarity solutions for boundary-layer flows this method is called Nachtsheim-Swigert scheme. We illustrate the technique for a pair of equations $f(x, y) = g(x, y) = 0$. Let $\bar{f}$ and $\bar{g}$ be the values of these functions with assumed values of x and y. Let the corrections to these guesses be Δx and Δy. Let F and G be the values of the functions with the roots estimated as $x + \Delta x$ and $y + \Delta y$. Using a truncated Taylor's series expansion one can write

$$F = \bar{f} + f_x\,\Delta x + f_y\,\Delta y$$

$$G = \bar{g} + g_x\,\Delta x + g_y\,\Delta y$$

Instead of setting F and G equal to zero we choose Δx and Δy such that $E = F^2 + G^2$ is a minimum. Hence, $(\partial E)/(\partial\,\Delta x) = (\partial E)/(\partial\,\Delta y) = 0$. These equations respectively are

$$\left(f_x^2 + g_x^2\right)\,\Delta x + \left(f_x f_y + g_x g_y\right)\,\Delta y = -\bar{f} f_x - \bar{g} g_x \tag{2.16}$$

$$\left(f_x f_y + g_x g_y\right)\,\Delta x + \left(f_y^2 + g_y^2\right)\,\Delta y = -\bar{f} f_y - \bar{g} g_y \tag{2.17}$$

The corrections Δx and Δy can now be obtained by solving Equations 2.16 and 2.17. Experience with the least squares method shows that it has a smaller

Table 2.12: Results for Example 2.3 using the Method of Least Squares

n	0	1	2	3	4	5	6	Residual
x	3	0.6809	0.9525	0.9987	0.9999			$f = 8.39 \times 10^{-5}$
y	6	4.749	4.1105	4.0029	4.0			$g = 0$
x	-1	-0.7779	0.2330	0.7849	0.97716	0.9997		$f = 2.28 \times 10^{-5}$
y	12	8.2155	5.8059	4.5028	4.0531	4.0007	4.0	$g = 0$

radius of convergence compared to the Newton-Raphson scheme. However it converges faster than the Newton-Raphson method when a good initial guess is available. The method of least squares is also applicable for over-determined problems where the number of equations is larger than the number of unknowns. A system of equations

$$F_i(x_1, x_2, \ldots, x_M) = 0, \quad i = 1, \ldots N \ (N > M)$$

is solved for the roots x_j by setting up an alternative determinate system of equations

$$\frac{\partial E}{\partial \Delta x_j} = 0, \quad (j = 1, \ldots, M) \quad \text{where} \quad E = \sum_{i=1}^{M} F_i^2$$

and

$$F_i = \bar{f}_i + \sum_{j=1}^{M} \frac{\partial f_i}{\partial x_j} \Delta x_j$$

As an example consider the functions f and g of Example 2.3 whose roots must be found. Results obtained using Equations 2.16 and 2.17 are given in Table 2.12.

2.3 Solution of a system of Simultaneous Linear Algebraic Equations

In the present section we are concerned with the solution of the system of equations

$$Ax = b \tag{2.18}$$

where A is a square, non-singular matrix with entries a_{ij}, b is the known right hand side vector and x a vector of unknowns. Since $x = A^{-1}b$, solving for x is equivalent to inverting A. However in many problems A^{-1} is never constructed explicitly. Despite this ambiguity, the solution of simultaneous equations is interpreted to be synonymous to the operation of matrix inversion.

Matrix inversion is usually the most intensive of all calculations. Owing to finite precision with which computers perform arithmetic operations, round-off errors occur in any numerical calculation. In the interest of minimizing computer time as well as to preserve accuracy matrix inversion must be accomplished with as few arithmetic operations as possible. Hence the choice of the matrix inverter is crucial in any numerical calculation pertaining to the solution of algebraic equations.

A large variety of matrix inverters are available as standard library packages on mainframe computers. All of these can be classified as either iterative or direct (Ortega, 1988). Iterative methods are only conditionally stable. However, if they are applicable they solve for the unknown quantities in a very simple manner. They require an initial guess for the solution and the number of iterations required for convergence does depend on the initial guess. The rate of convergence of iterative schemes is generally proportional to $1/N$, where N is the order of the matrix A in Equation 2.18. Hence these perform quite poorly if N is very large. In contrast, matrix inverters of the direct type are non-iterative and are unconditionally stable. A general direct matrix inverter requires $N^3/2$ arithmetic (multiplicative) operations and is hence computationally intensive. Simpler versions of the direct inverter can be formulated for special systems such as tridiagonal matrices and the number of arithmetic operations can be considerably reduced (to N^2). However a large system of equations may require pivoting to control round-off errors and pivoting operations can further increase the cost of computation. Hence the choice of the matrix inverter is quite problem dependent.

2.3.1 Iterative Schemes of Matrix Inversion

We consider iterative methods first. A popular method of matrix inversion is the Gauss-Seidel method. Writing Equation 2.18 as

$$\sum_{j=1}^{N} a_{ij}\, x_j = b_i$$

the solution x is iteratively computed component-wise as

$$x_i = \left[b_i - \sum_{j=1,\, j\neq i}^{N} a_{ij}\, x_j \right] / a_{ii} \tag{2.19}$$

If the system of equations is non-linear, matrix A would contain elements that depend on x. These elements are computed using the latest estimates of the components x_i. The Gauss-Seidel method converges only if the matrix A satisfies the condition

$$|\, a_{ii} \,| \;\geq\; \sum_{j=1,\, j\neq i}^{N} |\, a_{ij} \,|, \quad \text{all}\, i, \quad \text{and}$$

$$> \sum_{j=1,\, j\neq i}^{N} |\, a_{ij} \,|, \quad \text{for at least one } i$$

This is called the *Scarborough* condition of diagonal dominance. It requires that the diagonal term of the matrix A be the most dominant entry of any row. This condition is only a sufficiency condition for convergence in the sense that certain systems not satisfying the condition of diagonal dominance will still converge using Equation 2.19.

Example 2.5

Gauss-Seidel method is illustrated below using a numerical example. Consider the system

$$\begin{bmatrix} 4 & 1 \\ 2 & 7 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 6 \\ 16 \end{Bmatrix}$$

for which the solution is $x = (1, 2)$. With an initial guess of $x = (0, 1)$, application of Equation 2.19 gives the following intermediate results.

Iteration 1

$$x_1 = 5/4, \quad x_2 = (16 - 2(5/4))\,/7 = 13.5/7$$

[Note that the latest value of x_1 is used.]

Iteration 2

$$x_1 = 28.5/28, \quad x_2 = 195.5/98$$

Iterations continue till a prescribed convergence criterion for the error is reached as:

$$e = [\max \mid x_i\,(\text{new}) - x_i\,(\text{old}) \mid] < 10^{-5}$$

$$\text{or} \quad e\% = 100 \times \max_i \left| \frac{x_i\,(\text{new}) - x_i\,(\text{old})}{x_i\,(\text{new})} \right| < 0.01$$

A related method called the Jakobi scheme uses Equation 2.19 except that the x-values on the right hand side are all evaluated at the earlier iteration level. It is symbolically written as

$$x_i^{n+1} = \left(b_i - \sum_{j=1,\, j\neq i}^{N} a_{ij}\, x_i^n \right) / a_{ii} \tag{2.20}$$

where n is the iteration number. The Jakobi scheme is slightly slower than Gauss-Seidel method. However with the advent of parallel distributed computers this scheme has been rediscovered for solving practical problems. This is because Equation 2.20 can be applied to all the N values of x simultaneously on N processors using the old values alone.

The rate of convergence of an iteration scheme can be controlled using relaxation factors. As mentioned earlier, large systems of equations exhibit poor convergence even when supplied with a good initial guess. Convergence rates can be improved by replacing x_i^{n+1} with $\alpha\, x_i^{n+1} + (1-\alpha)\, x_i^n$, where $1 < \alpha < 2$. Here α is called an overrelaxation factor. Under other conditions the nonavailability of a good initial guess for x_i can lead to oscillations in the values of x_i during intermediate iterations. For a linear problem where a_{ij}'s are constants this is of no consequence. However if a_{ij}'s were functions of x_i the fluctuations obtained during intermediate calculations can lead to the loss of diagonal dominance. The iterations will subsequently diverge. To circumvent this problem we limit large changes in the values of x_i by replacing x_i^{n+1} with $\alpha\, x_i^{n+1} + (1-\alpha)\, x_i^n$, where $0 < \alpha < 1$. Factor α now represents underrelaxation. Optimum values of α that lead to convergence in a minimum number of iterations do exist. However these depend on the problem at hand and must be determined from numerical experiments.

Example 2.6

The ideas discussed above are illustrated through an example. Given the matrix equation $Ax = b$ with

$$A = \begin{bmatrix} 5 & 1 & 1 \\ 2 & 8 & 3 \\ 1 & 4 & 10 \end{bmatrix} \quad \text{and} \quad b = (10, 27, 39)$$

we wish to solve for x. The exact solution of this problem is $x = (1, 2, 3)$. Iterative methods can be applied because A does satisfy the Scarborough condition of diagonal dominance. We choose an initial guess of $x^o = (-10, 4, 0)$ and a convergence criterion of 0.01 percent. Results obtained using Gauss-Seidel, Jakobi and Gauss-Seidel with relaxation schemes are summarized in Table 2.13. Note that an indiscriminate use of relaxation factors can delay convergence.

2.3.2 Direct Methods for Matrix Inversion

We now proceed to discuss direct techniques for matrix inversion. Most direct methods employ a form of Gaussian elimination. Here a sequence of row manipulations called forward elimination converts the matrix to an upper triangular form. The solution is then obtained by back substitution. Hence the matrix equation

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{Bmatrix} \tag{2.21}$$

is first reduced to

Table 2.13: Results for Example 2.6; Relaxation factors have been applied to the Gauss-Seidel iterations.

Jakobi

n	2	6	14
x	(0.165, 1.837, 1.43)	(0.948, 1.946, 2.918)	(0.999, 1.999, 2.999)

Gauss-Seidel

n	1	3	5
x	(1.2, 3.075, 2.55)	(0.573, 2.031, 2.989)	(0.999, 2.0, 2.999)

Relaxation, $\alpha = 0.5$

n	3	10	20
x	(-0.35, 2.73, 2.42)	(0.981, 2.026, 2.988)	(0.999, 2.0, 2.999)

Relaxation, $\alpha = 1.1$

n	1	3	4
x	(2.32, 2.982, 2.805)	(1.027, 2.003, 2.999)	(0.996, 2.0, 2.999)

Relaxation, $\alpha = 1.6$

n	3	10	22
x	(4.189, 2.634, 3.139)	(0.844, 1.989, 2.99)	(0.995, 2.0, 3.0)

$$\begin{bmatrix} t_{11} & t_{12} & \dots & t_{1N} \\ 0 & t_{22} & \dots & t_{2N} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & t_{NN} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{Bmatrix} = \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{Bmatrix} \tag{2.22}$$

This requires that elements of a column below the diagonal term be reduced to zero. For the ith diagonal element the $(N-i)$ elements on the column below it can be reduced to zero by multiplying the ith row of the matrix by (a_{ji}/a_{ii}), $(j = i+1, \dots N)$ and subtracting it from the jth row. These manipulations start from the first row $(i = 1)$ itself. Row operations must be applied simultaneously to the matrix A and the right hand side vector b. Obtaining the solution x from Equation 2.22 is straightforward; for example $x_N = c_N/t_{NN}$. More generally

$$x_i = \left(c_i - \sum_{j=i+1}^{N} x_j \, t_{ij} \right) / t_{ii}, \quad i = N, N-1, \dots 1 \tag{2.23}$$

Forward elimination is computationally more intensive compared to backward substitution which requires only N^2 operations. Hence if A is reduced to an upper (or lower) triangular form the solution can be considered to have been found. A method closely related to Gaussian elimination is the Gauss-Jordon method. Here back substitution is effected simultaneously with forward elimination. Unlike Gaussian elimination which works with elements below the current ith row, Gauss-Jordon method manipulates rows both above and below

the ith row. It is computationally equivalent to Gaussian elimination and is not described here.

Example 2.7

Consider the application of Gaussian elimination to the problem,

$$\begin{bmatrix} 2 & 2 & 4 \\ 1 & 3 & 2 \\ 3 & 1 & 3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 18 \\ 13 \\ 14 \end{Bmatrix}$$

The solution of this problem is $x = (1, 2, 3)$. Note that the coefficient matrix does not satisfy the condition of diagonal dominance and iterative methods do not apply. The row operations to matrix A and vector b are summarized as:

row 1	2	2	4	18
row 2 - (a_{21}/a_{11}) row 1	0	2	0	4
row 3 - (a_{31}/a_{11}) row 1	0	-2	-3	-13
[Entries below a_{11} are now zero.]				
row 3 - (a_{32}/a_{22}) row 2	0	0	-3	-9
[Entries below a_{22} are now zero.]				

The matrix equation is now

$$\begin{bmatrix} 2 & 2 & 4 \\ 0 & 2 & 0 \\ 0 & 0 & -3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 18 \\ 4 \\ -9 \end{Bmatrix}$$

The solution obtained by back substitution is

$$x_3 = 3, \quad x_2 = (4 - t_{23}x_3)/t_{22} = 2 \quad \text{and} \quad x_1 = (18 - t_{12}x_2 - t_{13}x_3)/t_{11} = 1$$

Gaussian elimination can be used to determine explicitly the determinant and inverse of matrix A. The product of the diagonal terms obtained during forward elimination gives the value of the determinant of A. This is the product $\Pi_{i=1}^{N} t_{ii}$, where t_{ij} is defined in Equation 2.22. For the example given above det $A = 2 \times 2 \times (-3) = -12$. The inverse of A is obtained by starting with the identity matrix I as the right hand side. Each column of I can be thought of as a unique right side vector, i.e.,

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = [(b_1)(b_2)\dots(b_N)]$$

Let $x_1, \dots, x_N$ be the solutions corresponding to each right hand side vector $b_i (i = 1, \dots, N)$. Then

$$A^{-1} = [(x_1)(x_2)\ldots(x_N)] \tag{2.24}$$

During computations row-wise calculations can be applied to the augmented matrix

$$\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \ldots & a_{1N} & 1 & 0 & \ldots & 0 \\ a_{21} & a_{22} & \ldots & a_{2N} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{NN} & 0 & 0 & \ldots & 1 \end{array}\right]$$

and A^{-1} constructed during back substitution.

Though Gaussian elimination does not require diagonal dominance it cannot handle zeroes on the diagonal of matrix A. This is not a serious limitation because the diagonal entries of the matrix can be made non-zero by suitably swapping rows. Of greater concern however is the buildup of round-off errors. Since the diagonal entries of the matrix to be inverted are repeatedly used, very small values will be affected by round-off errors introduced by a computer. This problem is compounded when large systems of equations are encountered. A partial remedy for this problem is the use of a *pivoting* strategy. The pivoting operation comprises of swapping rows and columns till the diagonal terms are the maximum compared to all other terms present on their respective row and column. When similar matrices are to be repeatedly inverted, the sequence of swapping operations can be stored once-and-for-all. This will greatly reduce the computational cost of pivoting. Swapping columns of a matrix will generate new right hand side vectors and is hence an expensive task. In many problems swapping rows alone can improve the situation and control round-off. This is called *partial pivoting*. When pivoting does not make diagonal entries of the matrix large enough, the question of round-off remains critical and the only solution available is to generate matrices which are themselves not very large. This can be accomplished in practical applications by domain decomposition and in multidimensional problems by reducing them to a sequence of one dimensional problems.

Gaussian elimination falls in a class of techniques based on LU (lower-upper) decomposition. Any non-singular matrix A can be split into the product $G \cdot H$, where G has a lower triangular form and H has an upper triangular form (Duff *et al.*, 1986). Hence,

$$G = \begin{bmatrix} g_{11} & 0 & \ldots & 0 \\ g_{21} & g_{22} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ g_{N1} & g_{N2} & \ldots & g_{NN} \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & h_{12} & \ldots & h_{1N} \\ 0 & 1 & \ldots & h_{2N} \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}$$

The diagonal elements of either G or H can be chosen as unity. If G and H are known, the equation $Ax = b$ is solved as $Hx = C$, where C is the solution of the equation $GC = b$. Both equations involving G and H are easily solved by back substitution since they are in lower and upper triangular forms respectively.

The greatest computational effort is required not for determining C and x but for obtaining the elements of G and H. This is accomplished as follows.

Comparing the product of G and H with A, one can explicitly write the equations

$$g_{11} = a_{11},\ g_{21} = a_{21} \cdots g_{N1} = a_{N1}$$

$$g_{11}h_{12} = a12,\ \ g_{21}h_{12} + g_{22} = a_{22}, \cdots g_{N1}h_{12} + g_{N2} = a_{N2}$$

$$g_{11}h_{13} = a_{13}$$

and so on.

Solving a general matrix equation by LU decomposition is called the Crout's algorithm. It is computationally equivalent to Gaussian elimination. If A is symmetric the determination of elements g_{ij} and h_{ij} is simplified to some extent. If A is symmetric and positive definite it can be decomposed as $A = GG^T$, where G is a lower triangular form

$$G = \begin{bmatrix} g_{11} & 0 & \cdots & 0 \\ g_{21} & g_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ g_{N1} & g_{N2} & \cdots & g_{NN} \end{bmatrix}$$

This is called Cholesky decomposition. The elements of G can be determined by sequentially solving the equations

$$\sum_{i=1}^{j} g_{ji}^2 = a_{jj}, \quad j = 1, \ldots, N \tag{2.25}$$

$$\sum_{k=1}^{i} g_{i,i-k+1}\, g_{j,j-k} = a_{ij}, \quad i = 1, \ldots, j-1, \quad j = 1, \ldots, N$$

For example if A is a 3×3 matrix, the g_{ij}'s are explicitly calculated as

$$g_{11} = \sqrt{a_{11}},\ g_{21} = a_{21}/g_{11},\ g_{31} = a_{31}/g_{11}$$

$$g_{22} = \sqrt{a_{22} - g_{21}^2},\ \ g_{31} = a_{31}/g_{11},\ \ g_{32} = (a_{32} - g_{21}g_{31})/g_{22} \quad \text{and}$$

$$g_{33} = \sqrt{a_{33} - g_{31}^2 - g_{32}^2}.$$

If A is not a symmetric and positive-definite matrix the quantities within the square root sign will not necessarily be positive. The application of Crout's algorithm to two model problems is given below.

Example 2.8

Consider the problem

$$\begin{bmatrix} 5 & 8 & 2 \\ 3 & 3 & -4 \\ 1 & 6 & 2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 19 \\ 13 \\ 11 \end{Bmatrix}$$

whose solution is $x = (1, 2, -1)$. The G and H matrices are

$$G = \begin{bmatrix} 5 & 0 & 0 \\ 3 & -9/5 & 0 \\ 1 & 22/5 & -100/9 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & 8/5 & 2/5 \\ 0 & 1 & 26/9 \\ 0 & 0 & 1 \end{bmatrix}$$

We solve $GC = b$ to compute C as (19/5,−8/5,−1).
Solving $Hx = C$ yields

$$\begin{bmatrix} 1 & 8/5 & 2/5 \\ 0 & 1 & 26/9 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 19/5 \\ -8/9 \\ -1 \end{Bmatrix}$$

Hence $x_3 = -1$, $x_2 = -8/9 + 26/9 = 2$ and $x_3 = 19/5 - 2(8/5) + 2/5 = 1$.

Example 2.9

Consider $A = \begin{bmatrix} 4 & -1 & 2 \\ -1 & 2 & 0 \\ 2 & 0 & 6 \end{bmatrix}$ which is symmetric and positive definite and $b = (20, -2, 40)$.
We decompose A as GG^T and determine

$$G = \begin{bmatrix} 2 & 0 & 0 \\ -1/2 & \sqrt{7}/2 & 0 \\ 1 & 1/\sqrt{7} & \sqrt{34/7} \end{bmatrix}$$

Subsequently $GC = b$ yields $C = (10, 6/\sqrt{7}, 204/\sqrt{238})$ and $G^T x = C$ leads to

$$\begin{bmatrix} 2 & -1/2 & 1 \\ 0 & \sqrt{7}/2 & 1/\sqrt{7} \\ 0 & 0 & \sqrt{34/7} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 10 \\ 6/\sqrt{7} \\ 204/\sqrt{238} \end{Bmatrix}$$

Hence $x_3 = \frac{204}{\sqrt{238}} \sqrt{\frac{7}{34}} = 6$, $x_2 = \left(\frac{6}{\sqrt{7}} - \frac{6}{\sqrt{7}}\right) / \frac{\sqrt{7}}{2} = 0$
and $x_1 = \left(10 + \frac{1}{2}(0) - 6\right) / 2 = 2$.

2.3.3 Direct Methods for Banded Matrices

Special versions of Gaussian elimination can be generated for banded, sparse and symmetric matrices. For tridiagonal matrices the number of non-zero elements below the diagonal is unity. Hence the number of row manipulations associated with any row is unity; it is $(N-i)$ for the ith row for a general matrix. The forward elimination and backward substitution steps can be analytically derived for a tridiagonal matrix as follows. Let

$$\begin{bmatrix} d_1 & a_1 & & & 0 \\ b_2 & d_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & a_{N-1} \\ 0 & & & b_N & d_N \end{bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_N \end{Bmatrix} = \begin{Bmatrix} c_1 \\ \vdots \\ \vdots \\ \vdots \\ c_N \end{Bmatrix}$$

The forward elimination operation is

$$\begin{aligned} d_1(\text{new}) &= d_1(\text{old}) \\ d_j(\text{new}) &= d_j(\text{old}) - \frac{b_j}{d_{j-1}(\text{old})}\, a_{j-1}, \quad j = 2, 3, \ldots, N \\ a_j(\text{new}) &= a_j(\text{old}) \quad j = 1, 2, \ldots N. \\ c_1(\text{new}) &= c_1(\text{old}) \\ c_j(\text{new}) &= c_j(\text{old}) - \frac{b_j}{d_{j-1}(\text{old})}\, c_{j-1}, \quad j = 2, 3, \ldots, N \end{aligned} \tag{2.26}$$

The matrix equation is now in upper triangular form,

$$\begin{bmatrix} d_1 & a_1 & & & 0 \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & a_{N-1} \\ 0 & & & & d_N \end{bmatrix}^{\text{new}} \begin{Bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_N \end{Bmatrix} = \begin{Bmatrix} c_1 \\ \vdots \\ \vdots \\ \vdots \\ c_N \end{Bmatrix}^{\text{new}}$$

The solution can be readily obtained by backward substitution (using new values of a, d and c). Hence

$$x_N = c_N / d_N \tag{2.27}$$

$$x_j = (c_j - a_j\, x_{j+1})\, / d_j, \quad j = N-1, N-2, \ldots, 1$$

The procedure given above is called Thomas's tridiagonal matrix algorithm (TDMA). As in the case of a lower (or upper) triangular form matrix inversion using TDMA is rapid involving only N^2 arithmetic operations. Hence solving a full matrix equation as a sequence of tridiagonal matrix equations is computationally attractive. The ADI procedure applied to multidimensional problems

and described in Chapter 4 is an example of this approach. While LU decomposition reduces a matrix equation $Ax = b$ to a triangular form, Householder's algorithm (Householder, 1964) reduces it to a tridiagonal form. TDMA is a useful matrix inversion tool in the field of computational fluid flow and heat transfer.

Procedures similar to TDMA can be developed for banded matrices with a known bandwidth (Anderson *et al.*, 1984). The algorithm for a matrix with 5 diagonals is as follows. Let

$$\begin{bmatrix} d_1 & a_1 & b_1 & & & 0 \\ c_1 & d_2 & a_2 & b_2 & & \\ e_1 & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & b_{N-2} \\ & & \ddots & \ddots & \ddots & a_{N-1} \\ 0 & & & e_{N-2} & c_{N-1} & d_N \end{bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x_N \end{Bmatrix} = \begin{Bmatrix} f_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_N \end{Bmatrix}$$

The forward elimination steps are:

(a)

$$\text{For} \quad j = 1, \ldots, N-2, \quad b_j(\text{new}) = b_j(\text{old})$$

(b)

$$\begin{aligned} a_1(\text{new}) &= a_1(\text{old}) \\ d_1(\text{new}) &= d_1(\text{old}) \\ d_2(\text{new}) &= d_2(\text{old}) - c_1(\text{old}) \left[\frac{a_1}{d_1}\right](\text{new}) \\ a_2(\text{new}) &= a_2(\text{old}) - c_1(\text{old}) \left[\frac{b_1}{d_1}\right](\text{new}) \end{aligned}$$

(c) Intermediate Step

$$\text{For} \quad j = 3, \ldots N, \quad c_{j-1}(\text{new}) = c_{j-1}(\text{old}) - e_{j-2}(\text{old}) \left[\frac{a_{j-2}}{d_{j-2}}\right](\text{new})$$

(d)

$$\begin{aligned} \text{For} \quad j = 3, \ldots, N, \quad d_j(\text{new}) = d_j(\text{old}) \quad &- \quad e_{j-2}(\text{old}) \left[\frac{b_{j-2}}{d_{j-2}}\right](\text{new}) \\ &- \quad \left[\frac{a_{j-1}\, c_{j-1}}{d_{j-1}}\right](\text{new}) \\ \text{For} \quad j = 3, \ldots, N-1, \quad a_j(\text{new}) = a_j(\text{old}) \quad &- \quad \left[\frac{c_{j-1} b_{j-1}}{d_{j-1}}\right](\text{new}) \end{aligned}$$

(e)

$$\begin{aligned} f_1(\text{new}) &= f_1(\text{old}) \\ f_2(\text{new}) &= f_2(\text{old}) - c_1(\text{old}) \left[\frac{f_1}{d_1}\right](\text{new}) \end{aligned}$$

(f) Intermediate Step

$$\text{For} \quad j = 3, \ldots N, \quad \bar{f}_j = -e_{j-2}(\text{old}) \left[\frac{f_{j-2}}{d_{j-2}}\right](\text{new}) + f_j(\text{old})$$

(g)

$$\text{For} \quad j = 3, \ldots, N, \quad f_j(\text{new}) = \bar{f}_j(\text{old}) - c_{j-1}(\text{old}) \left[\frac{f_{j-1}}{d_{j-1}}\right](\text{new})$$

The solution is obtained by backward substitution using the new values of a, b, d and f as follows

$$x_N = f_N / d_N$$

$$x_{N-1} = (f_{N-1} - a_{N-1}\, x_N) / d_{N-1}$$

$$\text{For} \quad j = N-2, \ldots, 1, \quad x_j = (f_j - x_{j+1}\, a_j - x_{j+2}\, b_j) / d_j$$

Example 2.10

As an example of inversion of a pentadiagonal matrix let $N = 10$, $d_i = 0.2N^2$, $a_i = N$, $c_i = N + 5$, $b_i = N/2$ and $e_i = 0.4N$. The matrix A is clearly unsymmetric. For a solution $x_i = i$, $i = 1, \ldots, 10$ the right hand side can be computed as $f = (955, 105, 159, 213, 267, 321, 375, 429, 428, 367)$. The new values of d, a, b and f are obtained as:

$$\begin{aligned} d &= (20, 12.5, 12.5, 13.408, 13.58, 13.53, 13.52, 13.52, 13.52, 13.52) \\ a &= (10, 6.25, 4.8, 4.8, 4.97, 5, 5, 5, 5) \\ b &= (5, 5, 5, 5, 5, 5, 5, 5, 5) \\ f &= (55, 63.75, 81.7, 107.63, 132.77, 156.23, 179.66, 203.19, 171.69, 135.2). \end{aligned}$$

The solution is readily obtained by back substitution as $x = (1,2,3,4,5,6,7,8,9,10)$. For example,

$$x_{10} = f_{10}/d_{10} = 135.2/13.52 = 10, \quad x_9 = (171.69 - 5(10))/13.52 = 9,$$

and so on.

2.3.4 Conjugate Gradient Algorithm

Matrix inverters are central to any numerical algorithm in fluid mechanics and heat transfer problems. A poorly designed matrix inverter can lead to excessive computer memory requirement, high cost of computing and errors due to round-off in the final solution. These can be compounded in non-linear problems where matrix inversion must be repeatedly carried out.

The two common routes for matrix inversion available to a numerical analyst are iterative solvers and direct solvers, as mentioned in the previous sections. The Gauss-Seidel method falls in the former category while Gaussian elimination and its variants fall in the latter category. Two principal advantages of the Gauss-Seidel method are that the full matrix need not be formed explicitly at any stage and it converges rapidly when provided with a good initial guess. However it is only conditionally stable and requires the matrix to be diagonally dominant. This condition is not necessary when direct solvers are used. However they do construct the full matrix before the matrix inverse calculations are initiated. More importantly they do not use the knowledge of the solution even if it is partially available, except in non-linear problems where a_{ij} depends on the solution x. Direct solvers based on Gaussian elimination can be easily modified to minimize storage requirement for banded and symmetric matrices. These are the best among the alternatives available at this time for tridiagonal and pentadiagonal matrices.

The matrix inversion method that is being discussed here is the conjugate gradient (CG) algorithm (Axelsson and Barker, 1984; Golub and Van Loan, 1996). This method combines features that are common to both iterative as well as direct solvers. The original CG algorithm is defined only for symmetric positive definite matrices. However it can be extended to other problems through the use of preconditioning (Kershaw, 1978). In principle the CG method converges to the exact solution in a finite number of steps in the absence of round-off errors. In practice it converges faster than the Gauss-Seidel method for a given initial guess and convergence criterion. The total number of arithmetic operations is also smaller than that for Gaussian elimination unless the matrix is triangular, tridiagonal or pentadiagonal. The principles underlying the CG method come from linear algebra. This topic is briefly reviewed in Appendix I. In complex problems the matrix to be inverted may neither be symmetric nor positive definite. In such cases the computational effort required for preconditioning can outweigh the advantages of the CG algorithm itself. Development of a versatile preconditioned conjugate gradient method is a topic of current research.

We briefly present the background required to understand the conjugate gradient algorithm. The CG method belongs to the same gradient family as the method of steepest descent. Given a system of N equations $Ax = b$ for which the solution x is required and an approximation x_0 is given, a gradient method seeks to minimize a suitable measure of the residual $r = b - Ax$ by deriving a sequence of estimates x_k with a residual r_k. The measure of the

residual vector is so chosen that its minimum occurs when $r_k = 0$, that is the exact solution. One such measure is $h^2 = r_k^T r_k = \sum_{i=1}^{N} r_i^2$; it forms the basis of the least squares method. For each estimate of x_k a direction p_k is selected and the new estimate x_{k+1} is the point on the line $x_k + \alpha p_k$ which minimizes h^2. The method of steepest descent uses the gradient of h^2 itself as the search direction. It is illustrated below through an example.

Suppose we need to solve the system of equations $f_i(x_j) = 0, \quad i,j = 1,\ldots,N$. The error norm is $E = \sum_{i=1}^{N} f_i^2(x_j)$, x_j^0 is an initial guess and E^0 is the corresponding error. The search direction has components $\partial E^0/\partial x_j$ $(j = 1,\ldots,N)$ evaluated at x_j^0. The new value of the roots x_j is taken as $(x_j^0 + \alpha\partial E^0/\partial x_j)$ such that $E\ \left(x_j^0 + \alpha\ (\partial E^0)/(\partial x_j)\right)$ is a minimum with respect to α.

Using a truncated Taylor's series expansion for E yields

$$E\left(x_j^0 + \alpha\frac{\partial E^0}{\partial x_j}\right) = E^0 \quad + \quad \alpha\sum_{j=1}^{N}\left(\frac{\partial E^0}{\partial x_j}\right)^2 + \frac{\alpha^2}{2}\left[\sum_{j=1}^{N}\frac{\partial^2 E^0}{\partial x_j^2}\left(\frac{\partial E^0}{\partial x_j}\right)^2 \right.$$
$$\left. + \quad 2\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 E^0}{\partial x_i \partial x_j}\frac{\partial E^0}{\partial x_i}\frac{\partial E^0}{\partial x_j}\right]$$

One can solve for α by setting $\partial E/\partial\alpha = 0$. For example, in a one-variable problem

$$\alpha = \left[\frac{E_x^2 + E_y^2}{E_{xx}\,E_x^2 + 2E_{xy}\,E_x E_y + E_{yy}\,E_y^2}\right]^0$$

The conventional steepest descent algorithm requires several derivative calculations and guarantees only asymptotic convergence. It is also computationally intensive. The route followed by the conjugate gradient algorithm results in a reduction in the amount of computation as well as a simplification in the algorithm itself. The error norm chosen by the CG method is $h^2 = r_k^T A^{-1} r_k$, where r (as before) is the residual vector $(b - Ax)$. The norm exists (i.e., $h^2 > 0$) if A (and hence A^{-1}) is symmetric and positive definite. The search direction p_k is chosen as close as possible to the direction of steepest descent with the additional requirement that p_k be orthogonal to all the previous directions of search. Since the vectors p_k form a set of conjugate vectors, a set of N of them will form a basis on an N-dimensional space. Since the error h^2 has been separately minimized along each of the conjugate directions, the CG algorithm will reduce h^2 to a null vector (given exact arithmetic) after N steps.

The conjugate gradient method would be a direct method if exact arithmetic were to be used. However, in practice storage as well as arithmetic operations are with finite precision and the CG method behaves as an iterative scheme. When large matrices are involved, performing N passes would result in an effort larger than that required for direct methods such as Gaussian elimination. For this reason the CG method has not been used in applications as a direct method.

However, if applied a few times it produces accuracy that is entirely satisfactory for practical problems.

The application of the conjugate gradient algorithm to the matrix equation $Ax = b$, where A is symmetric and positive definite is given below. The method is iterative and each pass comprises of the following steps.

1. Assume the starting value of x $(= x^0)$ and set iteration index $k = 0$.
2. Compute the direction vector $p_k = b - Ax_k$ and the residue $r_k = p_k$.
3. Compute the steepest descent parameter $\alpha_k = (r_k, r_k)/(p_k, Ap_k)$.
4. Update $x_{k+1} = x_k + \alpha_k p_k$ and $r_{k+1} = r_k - \alpha_k Ap_k$.
5. Compute $\beta_k = (r_{k+1}, r_{k+1})/(r_k, r_k)$.
6. Compute $p_{k+1} = r_{k+1} + \beta_k p_k$.
7. Update $k = k + 1$. Proceed to Step 3 unless $k > N$ or convergence has been attained in the solution vector x.

In the algorithm given above the notation (a, b) stands for the inner product $\sum_{i=1}^{N} a_i b_i$. Quantities such as (p, Ap) can be determined without forming the matrix A explicitly. The vector Ap can be formed directly in most applications. For example, in finite element calculations (Chapter 8) A is formed at an element level and the complete vector Ap is obtained by assembling element level vectors. Any variation of the CG algorithm that requires matrix A to be formed completely is of no practical use since it would not compare favourably with direct elimination methods. We illustrate the conjugate gradient method through two simple examples.

Example 2.11

The system

$$\begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 9 \\ 12 \end{Bmatrix}$$

has a solution $x = (1, 2)$. The initial guess is taken as (1,1). Recall that p_k is a search direction and r_k the residual vector – both of which should approach zero as the solution is reached. The intermediate calculations in the CG method are tabulated as:

Iteration(k)	x_k	p_k	r_k
0	(1,1)	(2,5)	(2,5)
1	(1.31,1.783)	$(-1.032, 1.711)$	$(-1.135, 0.454)$
2	(1,2)	$\sim 10^{-7}$	$\sim 10^{-7}$

Iteration(k)	α	β
0	0.156	0.0575
1	0.304	9.5×10^{-15}
2	–	–

Example 2.12

The system

$$\begin{bmatrix} 5 & 1 & 2 \\ 1 & 8 & 3 \\ 2 & 3 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 13 \\ 26 \\ 38 \end{Bmatrix}$$

has a solution (1,2,3). The initial guess is taken as $(-10, 4, 0)$. The intermediate calculations are summarized as:

Iteration(k)	x_k	p_k	r_k
0	$(-10, 4, 0)$	(59,4,46)	$(59, 4, 46)$
1	$(-3.52, 4.439, 5.05)$	$(27.19, -20.38, -10.1)$	$(16.07, -21.14, 18.77)$
2	(0.544,1.389,3.538)	$(2.42, 3.25, -2.87)$	$(1.808, 3.72, -2.64)$
3	(1,2,3)	$\sim 10^{-7}$	$\sim 10^{-6}$

Iteration(k)	α	β
0	0.1097	0.188
1	0.1496	2.28×10^{-2}
2	0.187	1.06×10^{-13}
3	–	–

2.3.5 Preconditioning

Consider the solution of the matrix equation $Ax = b$ by iterative methods. Preconditioning is a strategy by which the number of iterations required for convergence is reduced. Specifically, it involves multiplying the governing equation throughout by a suitable matrix P so that the modified problem, $PAx = Pb$ requires fewer iterations compared to the original problem. Some computational effort is required to generate matrix P itself. The success of preconditioning depends on the extent to which the number of iterations for matrix inversion is reduced.

We initiate a discussion below that will motivate the selection of a preconditioner for matrix inversion using the conjugate gradient method. Any general iterative scheme that solves the system of equations $Ax = b$ can be cast in the form

$$M\,x^{k+1} = -Rx^k + b \tag{2.28}$$

Here k is the iteration index, x_k is the sequence $(x^0, x^1, \ldots)$ of solutions obtained after each iteration and M and R are suitably decomposed forms of A such that $A = M + R$ and M is non-singular. For example M can be chosen as that part of

A which is symmetric and positive definite. Since the exact solution x satisfies, $Mx = -Rx + b$, the error vector $e = (x^{k+1} - x)$ satisfies

$$M(x^{k+1} - x) = -R(x^k - x)$$

The right hand side can be simplified recursively, i.e., x^k replaced by elements that depend only on x^{k-1}. This yields, $x^k - x = B^k(x^0 - x)$ where $B = -M^{-1}R$ and $B^k = B \cdot B \ldots k$ times. Matrix B can be thought of as an amplification factor of the initial error and is related to the iteration scheme. For the iterations to converge the norm of matrix B, denoted as $\| B \|$ must be less than unity. If B has a set of linearly independent eigen vectors corresponding to eigenvalues ξ_i, then the norm is bounded below by the magnitude of its largest eigenvalue, i.e.

$$\| B \| \geq \max_{i=1,\ldots,N} | \xi_i | \tag{2.29}$$

The right hand side of Equation 2.29 is the spectral radius $\rho(B)$ of matrix B. Hence convergence of the iteration scheme (Equation 2.28) requires $\rho(B) < 1$. The convergence rate increases as ρ decreases further below unity. This is an important result.

In many problems M is chosen as positive definite and M^{-1} as the preconditioner for the system of equations $Ax = b$. Under these conditions the result $\rho(B) < 1$ can be strengthened further for the system of equations $M^{-1}Ax = M^{-1}b$. Let λ_i be the eigenvalues of $M^{-1}A$. Since $A = M + R$, we get $M^{-1}A = I - B$ and $\lambda_i = 1 - \xi_i \, (i = 1, \ldots, N)$. Since $\rho(B) < 1$, we require $-1 < \xi_i < 1$. Hence $\lambda_i > 0$ and the ratio of the maximum to the minimum eigenvalues of $M^{-1}A$ can be estimated as

$$s(A) = \lambda_N/\lambda_1 = (1 - \xi_N)/(1 - \xi_1) \leq (1 + \rho(B))/(1 - \rho(B)) \tag{2.30}$$

Here $s(A)$ is the spectral condition number of A before preconditioning and $M^{-1}A$ after preconditioning. Since $\rho(B)$ must be as small as possible we require $s(A)$ to be as close to unity as possible. This means that the set of eigenvalues $\lambda_1, \ldots \lambda_N$ must cluster around a certain value. The goal of preconditioning is then to reduce the spectral condition number of the coefficient matrix to as close to unity and hence $\rho(B)$ as small as possible.

Two preconditioners that are frequently used with the conjugate gradient method are:

1. Cast $Ax = b$ in the form $(GG^T)^{-1}Ax = (GG^T)^{-1}b$, where G is the lower triangular form of A obtained using LU decomposition. If A is symmetric and positive definite Cholesky decomposition can be used.

2. Cast $Ax = b$ in the form $D^{-1/2}AD^{-1/2}D^{1/2}x = D^{-1/2}b$, i.e., $\tilde{A}\tilde{x} = \tilde{b}$ where $\tilde{A} = D^{-1/2}AD^{-1/2}$, $\tilde{x} = D^{1/2}x$, and $\tilde{b} = D^{-\frac{1}{2}}b$ and $D = \text{diag}(A)$ is given as

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & a_{NN} \end{bmatrix}$$

Further,

$$D^{-\frac{1}{2}} = \begin{bmatrix} 1/\sqrt{a_{11}} & 0 & \dots & 0 \\ 0 & 1/\sqrt{a_{22}} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1/\sqrt{a_{NN}} \end{bmatrix}$$

Both preconditioners can be shown to reduce the spectral condition number of the coefficient matrix towards unity. If A is symmetric and positive definite, GG^T will equal A and preconditioning gives the solution of the problem directly. In general computing G is equivalent to the full LU decomposition and preconditioning alone is equivalent to the direct solution of the matrix equation. In practice G is replaced by an approximate estimate $\bar{G}$ using techniques called *incomplete factorization*. Numerical experiments show that replacing G by $\bar{G}$ in the first preconditioner does improve the spectral condition number and reduces the number of CG iterations. The use of the specific form of the preconditioners 1 and 2 has one additional advantage. Matrix multiplication involved in quantities such as $g = (GG^T)^{-1}r$, for a vector r can be computed without performing an inverse operation by solving $Gf = r$ for f followed by $G^T g = f$ for g. Each system of equations is triangular and can be readily solved by back substitution. Similarly $r = \bar{A}\bar{x}$ is easily obtained by solving the system $D^{\frac{1}{2}}r = Ax$. Here the right hand side is usually known component-wise as a vector, though A itself is not explicitly constructed.

We illustrate the two preconditioning strategies given above through some examples. In particular we focus attention on the improvement in the spectral condition number of the coefficient matrix that must be iteratively inverted.

Example 2.13

Let matrix A be

$$A = \begin{bmatrix} 4 & -1 & 2 \\ -1 & 2 & 0 \\ 2 & 0 & 6 \end{bmatrix}$$

The lower triangular factorization of A gives

$$G = \begin{bmatrix} 2 & 0 & 0 \\ -1/2 & \sqrt{7}/2 & 0 \\ 1 & 1/\sqrt{7} & \sqrt{34/7} \end{bmatrix} \quad \text{such that} \quad GG^T = A.$$

The eigenvalues of A are (1.4157,3.2943,7.2899) giving a spectral condition number $s(A) = 7.2899/1.4157 = 5.149$. The use of a preconditioner $(GG^T)^{-1}$ for

A will reduce the coefficient matrix to an identity matrix which has an ideal condition number of unity. As mentioned earlier complete decomposition of A as GG^T for preconditioning affects adversely the benefits of the CG algorithm. Hence in practice we would like to supply an estimate of G rather than G itself. Let G be given as

$$G = \begin{bmatrix} 2 & 0 & 0 \\ -1/2 & \sqrt{7}/2 & 0 \\ 0 & 1/\sqrt{7} & \sqrt{34/7} \end{bmatrix}$$

This yields

$$\tilde{A} = (GG^T)^{-1}A = \begin{bmatrix} 0.9705 & 0.00735 & 0.48529 \\ -0.11765 & 1.0294 & -0.0588 \\ 0.41176 & -0.10929 & 1.2058 \end{bmatrix}$$

The eigenvalues of $\tilde{A}$ are (0.6376,1.0,1.5682) giving a condition number $s(\tilde{A}) = 2.459$. This is smaller than the value of 5.149 for $s(A)$. If we choose

$$G = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0.5 & 2 \end{bmatrix}$$

$$\tilde{A} = (GG^T)^{-1}A = \begin{bmatrix} 1.4065 & 0.5468 & 0.6562 \\ 0.8125 & 1.5937 & 0.3125 \\ 0.375 & -0.1875 & 1.375 \end{bmatrix}$$

and the eigenvalues of $\tilde{A}$ are (0.6291,1.5122,2.2336). Hence the condition number $s(\tilde{A}) = 3.55$, a value that is again better than 5.149. Note that matrix $\tilde{A}$, the preconditioned form of A is not symmetric. However it has real eigenvalues and the CG algorithm works for the preconditioned system of equations.

Diagonal scaling for the present example leads to $\tilde{A} = D^{-1/2}AD^{-1/2}$ and

$$\tilde{A} = \begin{bmatrix} 1 & -0.3535 & 0.4082 \\ -0.3535 & 1 & 0 \\ 0.4082 & 0 & 1 \end{bmatrix}$$

with eigenvalues (0.4599, 1.0, 1.54). Hence the condition number $s(\tilde{A})$ is 3.348.

Example 2.14

We mention here briefly the application of the preconditioner $(GG^T)^{-1}$ to unsymmetric matrices. Let A be a diagonally dominant matrix given as

$$A = \begin{bmatrix} 3 & 1 & 1 \\ 2 & 6 & -1 \\ -1 & 2 & 4 \end{bmatrix}$$

The eigenvalues of A are not all real and are $(6.34, 3.329 \pm 1.299j)$ where $j = \sqrt{-1}$. The lower triangular form of A is

$$G = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 16/3 & 0 \\ -1 & 7/3 & 81/16 \end{bmatrix}$$

and $\tilde{A} = (GG^T)^{-1}A$ is

$$\tilde{A} = \begin{bmatrix} 0.333 & -0.0138 & 0.2736 \\ 0 & 0.1875 & -0.145 \\ 0 & 0 & 0.197 \end{bmatrix}$$

The eigenvalues of $\tilde{A}$ are (0.1875, 0.197, 0.3333) and are all real and positive. The search directions implied in the CG algorithm are hence defined and mutually orthogonal. Hence this algorithm is applicable for the preconditioned matrix $\tilde{A}$. The condition number $\tilde{A}$ is 1.78. Also note that $\tilde{A}$ is in upper triangular form and can be alternatively solved by back substitution.

Example 2.15

For another example of unsymmetric matrices, consider

$$A = \begin{bmatrix} 5 & 8 & 2 \\ 3 & 3 & -4 \\ 1 & 6 & 2 \end{bmatrix}$$

which does not have diagonal dominance. For this problem

$$G = \begin{bmatrix} 5 & 0 & 0 \\ 3 & -1.8 & 0 \\ 1 & 4.4 & -11.11 \end{bmatrix}$$

and

$$\tilde{A} = (GG^T)^{-1}A = \begin{bmatrix} 0.2 & 0.653 & 1.1929 \\ 0 & -0.555 & -1.825 \\ 0 & 0 & -0.09 \end{bmatrix}.$$

The eigenvalues of $\tilde{A}$ are its diagonal entries namely $(-0.555, -0.09, 0.2)$. The eigenvalues are real and the condition number is 6.16. This is quite large. It is advantageous in this example to solve for x by back substitution.

Example 2.16

Consider the unsymmetric banded matrix as a final example.

$$A = \begin{bmatrix} 5 & 2 & 0 \\ 1 & 3 & 2 \\ 0 & 1 & 2 \end{bmatrix}$$

The eigenvalues of A are (0.822,3.3216,5.855) and $s(A) = 7.13$. We can require the LU decomposition to be banded as well with

$$G = \begin{bmatrix} g_{11} & 0 & 0 \\ g_{21} & g_{22} & 0 \\ 0 & g_{32} & g_{33} \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & h_{12} & 0 \\ 0 & 1 & h_{21} \\ 0 & 0 & 1 \end{bmatrix}$$

This is called *incomplete decomposition.* G can be calculated as

$$G = \begin{bmatrix} 5 & 0 & 0 \\ 1 & 13/5 & 0 \\ 0 & 1 & 4/3 \end{bmatrix}$$

and $\tilde{A} = (GG^T)^{-1}A$ as

$$\tilde{A} = \begin{bmatrix} 0.2 & 0.00307 & -0.00591 \\ 0 & 0.384 & 0.0295 \\ 0 & 0 & 0.692 \end{bmatrix}$$

The eigenvalues of $\tilde{A}$ are (0.2, 0.384, 0.692) which are real and positive. The condition number $s(\tilde{A})$ is 3.46 and is an improvement over the original value of 7.13. As before $\tilde{A}$ is in upper triangular form and can be solved by back substitution. The use of diagonal scaling results in

$$\tilde{A} = D^{-1/2}AD^{-1/2} = \begin{bmatrix} 1 & 0.516 & 0 \\ 0.2581 & 1 & 0.8165 \\ 0 & 0.408 & 1 \end{bmatrix}$$

The eigenvalues of $\tilde{A}$ are (0.3168, 1, 1.683) and the spectral condition number $s(\tilde{A}) = 5.31$. This is marginally better than the original value of 7.13. As a rule diagonal scaling requires fewer arithmetic operations compared to preconditioning based on factorization. However, as can be seen in Examples 2.14 - 2.16 to 2.3.5 the improvement in the condition number using diagonal scaling is usually moderate.

2.3.6 Preconditioned Conjugate Gradient Method

The conjugate gradient (CG) algorithm which includes preconditioning (PCG) using LU decomposition described here has been adapted from the work of Crochet, Davies and Walters (1984).

Algorithm I

Let A be a symmetric and positive definite matrix. We use preconditioning to improve the spectral condition number. The preconditioner is $(GG^T)^{-1}$ where G is a close approximation of the lower triangular form of A. The resulting algorithm is:

1. Set $k = 0$ and choose the initial guess, x_0 for the solution x.

2. Compute $r_k = b - Ax_k$ and $p_k = (GG^T)^{-1} r_k$.

3. Compute

$$\begin{aligned}
\alpha_k &= \left(r_k^T (GG^T)^{-1} r_k\right) / \left(p_k^T A p_k\right). \\
x_{k+1} &= x_k + \alpha_k p_k \\
r_{k+1} &= r_k - \alpha_k A p_k \\
\beta_k &= \left(r_{k+1}^T (GG^T)^{-1} r_{k+1}\right) / \left(r_k^T (GG^T)^{-1} r_k\right). \\
p_{k+1} &= (GG^T)^{-1} r_{k+1} + \beta_k p_k.
\end{aligned}$$

4. Set $k = k + 1$.

5. Repeat steps 3 and 4 to convergence.

As discussed in Section 2.3.5, quantities such as $r_k^T (GG^T)^{-1} r_k$ must be computed by successively solving a sequence of matrix-equations in lower or upper triangular form.

Further special cases arise in practical applications. When A is written as $N \times N$ blocks, i.e.

$$A = \begin{bmatrix} A_{11} & A_{12} & & 0 \\ A_{21} & A_{22} & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & A_{N,N-1} & A_{NN} \end{bmatrix} \tag{2.31}$$

where each diagonal matrix is square and symmetric (i.e. $A_{nn} = A_{nn}^T$ for $1 < n < N$). Also, A is block symmetric (i.e. $A_{n,n+1} = A_{n-1,n}^T$) and G has a form

$$G = \begin{bmatrix} G_{11} & & & 0 \\ G_{21} & G_{22} & & \\ & \ddots & \ddots & \\ 0 & & G_{N,N-1} & G_{NN} \end{bmatrix} \tag{2.32}$$

where the block matrices have the incomplete decompositions

$$\begin{aligned}
&A_{11} = G_{11}\, G_{11}^T \\
&A_{n,n-1} = G_{n,n-1}\, G_{n-1,n-1}^T \\
&A_{nn} = G_{n,n-1}\, G_{n,n-1}^T + G_{nn}\, G_{nn}^T, \quad n = 2, \ldots, N.
\end{aligned} \tag{2.33}$$

Suppose that the block matrices in Equation 2.31 are square, and of equal order $M \times M$, such that each diagonal block is a tridiagonal matrix and each

adjacent codiagonal block is a diagonal matrix, i.e.

$$A_{nn} = \begin{bmatrix} b_1^{(n)} & a_2^{(n)} & & 0 \\ a_2^{(n)} & b_2^{(n)} & a_3^{(n)} & \\ & \ddots & \ddots & \ddots \\ 0 & & a_M^{(n)} & b_M^{(n)} \end{bmatrix}, \quad A_{n,n-1} = \begin{bmatrix} d_1^{(n)} & & & 0 \\ & d_2^{(n)} & & \\ & & \ddots & \\ 0 & & & d_M^{(n)} \end{bmatrix}$$

Then we may write

$$G_{nn} = \begin{bmatrix} e_1^{(n)} & & & 0 \\ f_2^{(n)} & e_2^{(n)} & & \\ & \ddots & \ddots & \\ 0 & & f_M^{(n)} & e_M^{(n)} \end{bmatrix}, \quad G_{n,n-1} = \begin{bmatrix} s_1^{(n)} & & & 0 \\ & s_2^{(n)} & & \\ & & \ddots & \\ 0 & & & s_M^{(n)} \end{bmatrix} \tag{2.34}$$

The PCG algorithm for a symmetric and positive definite matrix can now be summarized as:

1. Let $n = 1$. Set $s_m^{(1)} = d_m^{(N+1)} = 0, \quad m = 1, \ldots M$

2. Compute

$$\begin{aligned} e_1^{(n)} &= \left(b^{(n)} - \left[s_1^{(n)}\right]^2\right)^{1/2}, \\ s_1^{(n+1)} &= d_1^{(n+1)}/e_1(n) \end{aligned}$$

3. For $m = 2, \ldots M$ compute

$$\begin{aligned} f_m^{(n)} &= a_m^{(n)}/e_{m-1}^{(n)}, \\ e_m^{(n)} &= \left(b_m^{(n)} - \left[s_m^{(n)}\right]^2 - \left[f_m^{(n)}\right]^2\right)^{1/2} \\ s_m^{(n)} &= d_m^{(n+1)}/e_m^{(n)} \end{aligned}$$

4. Let $n \to n + 1$

5. If $n \leq N$ (or till convergence) repeat steps 2 to 4, otherwise stop.

It is essential that the diagonal elements e of G be non-zero. If a zero value occurs (due to rounding error) then a non-zero is reassigned in the algorithm. Moreover, square roots can be avoided if the incomplete decomposition GG^T is replaced by GDG^T, where G is a lower triangular matrix of the same structure as the lower part of A but with unit diagonal elements and D is a diagonal matrix.

Algorithm II

Consider the PCG algorithm for an asymmetric banded non-singular system $Ax = b$. The Cholesky decomposition is now replaced by the incomplete LU decomposition $A = GH$ where we take unit diagonal elements in the upper triangular matrix H. Matrices G and H are taken to have the same structure as the lower and upper part of A respectively. This gives rise to the algorithm described below.

1. Let $k = 0$. Choose the initial guess x_0 and compute

$$\begin{aligned} r_0 &= b - Ax_0 \\ p_0 &= (H^T H)^{-1} A^T (GG^T)^{-1} r_0 \end{aligned}$$

2. Compute:

$$\begin{aligned} \alpha_k &= \left(r_k^T (GG^T)^{-1} r_k\right) / \left(p_k^T H^T H p_k\right) \\ x_{k+1} &= x_k + \alpha_k p_k \\ r_{k+1} &= r_k - \alpha_k A p_k \\ \beta_k &= \left(r_{k+1}^T (GG^T)^{-1} r_{k+1}\right) / \left(r_k^T (GG^T)^{-1} r_k\right) \\ p_{k+1} &= (H^T H)^{-1} A^T (GG^T)^{-1} r_{k+1} + \beta_k p_k \end{aligned}$$

3. Let $k = k + 1$

4. Repeat steps 2 to 3 until convergence is reached

When A is given by Equation 2.31, G is again given by Equation 2.32 whereas H has the form

$$H = \begin{bmatrix} H_{11} & H_{12} & & 0 \\ & H_{22} & H_{23} & \\ & & \ddots & \ddots \\ 0 & & & H_{NN} \end{bmatrix}$$

where the block matrices have similar incomplete decomposition to Equation 2.33 and G^T's are replaced by H's.

If the block matrices of A have the tridiagonal and diagonal forms:

$$A_{nn} = \begin{bmatrix} b_1^{(n)} & c_1^{(n)} & & 0 \\ a_2^{(n)} & b_2^{(n)} & c_2^{(n)} & \\ & \ddots & \ddots & \ddots \\ 0 & & a_M^{(n)} & b_M^{(n)} \end{bmatrix}$$

$$A_{n,n-1} = \begin{bmatrix} d_1^{(n)} & & & 0 \\ & d_2^{(n)} & & \\ & & \ddots & \\ 0 & & & d_M^{(n)} \end{bmatrix}$$

$$A_{n-1,n} = \begin{bmatrix} g_1^{(n-1)} & & & 0 \\ & g_2^{(n-1)} & & \\ & & \ddots & \\ 0 & & & g_M^{(n-1)} \end{bmatrix}$$

then in addition to Equation 2.34 we may write

$$H_{nn} = \begin{bmatrix} 1 & h_1^{(n)} & & 0 \\ & 1 & h_2^{(n)} & \\ & & \ddots & \ddots \\ 0 & & & 1 \end{bmatrix}, \; H_{n,n-1} = \begin{bmatrix} p_1^{(n-1)} & & & 0 \\ & p_2^{(n-1)} & & \\ & & \ddots & \\ 0 & & & p_M^{(n-1)} \end{bmatrix} \tag{2.35}$$

Entries of G and H may be found as:

1. Let $n = 1$. Set $s_m^{(1)} = p_m^{(0)} = d_m^{(N+1)} = g_m^{(N)} = 0, \quad m = 1, \ldots, M$
2. Compute:

$$\begin{aligned} e_1^{(n)} &= b_1^{(n)} - s_1^{(n)} p_1^{(n-1)} \\ h_1^{(n)} &= c_1^{(n)} / e_1^{(n)} \\ s_1^{(n+1)} &= d_1^{(n+1)} \\ p_1^{(n)} &= g_1^{(n)} / e_1^{(n)} \end{aligned}$$

3. For $m = 2, \ldots, M-1$ compute

$$\begin{aligned} f_m^{(n)} &= a_m^{(n)} \\ e_m^{(n)} &= b_m^{(n)} - f_m^{(n)} h_{m-1}^{(n)} - s_m^{(n)} p_m^{(n-1)} \\ h_m^{(n)} &= c_m^{(n)} / e_m^{(n)} \\ s_m^{(n+1)} &= d_m^{(n+1)} \\ p_m^{(n)} &= g_m^{(n)} / e_m^{(n)} \end{aligned}$$

4. Compute:

$$\begin{aligned} f_M^{(n)} &= a_M^{(n)} \\ e_M^{(n)} &= b_M^{(n)} - f_M^{(n)} h_{M-1}^{(n)} - s_M^{(n)} p_M^{(n-1)} \\ s_M^{(n+1)} &= d_M^{(n+1)} \\ p_M^{(n)} &= g_M^{(n)} / e_M^{(n)} \end{aligned}$$

5. Let $n \to n+1$
6. If $n \leq N$ repeat steps 2 to 5, otherwise stop

Again, it is important to ensure that the diagonal elements e do not vanish. If they do, they can be reassigned a small value.

References

Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer,* Hemisphere Publishing Corporation, New York, 1984.

Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press Inc, New York, 1984.

Carnahan B., Luther H.A. and Wilkes J.O., *Applied Numerical Methods,* John Wiley Sons, New York, 1969.

Crochet, M.J., Davies A.R. and Walters K., *Numerical Simulation of Non-Newtonian Fluid Flow,* Elsevier, New York, 1984.

Duff I.S., Conjugate Gradient Methods and Similar Techniques, Harwell Report AERE-R 9636, Computer Science and Systems Division, AERE, Harwell, U.K., 1979.

Duff, I.S., Erisman A.M. and Reid, J.K, *Direct Methods for Sparse Matrices,* Clarendon Press, Oxford, 1986.

Golub, G.H. and van Loan, C.F., *Matrix Computations,* John Hopkins University Press, Baltimore, 1996.

Greenberg, M.D., *Advanced Engineering Mathematics,* Prentice-Hall International, New Jersey, 1998.

Hildebrand, F.B., *Introduction to Numerical Analysis,* McGraw Hill, New York, 1956.

Householder, A.S., *The Theory of Matrices in Numerical Analysis,* Blaisdell Publishing Company, New York, 1964.

Kershaw, D.S., The Incomplete Cholesky-conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations, J. Computational Physics, Vol. 26, pp. 43-65, 1978.

Krylov, V.I. and Kruglikova, L.G., *Handbook of Numerical Harmonic Analysis,* Israel Program for Scientific Translations Ltd, Israel, 1969.

Press, W.H., Teukolsky, S.A., Vellerling, W.T. and Flannery, B.P., *Numerical Recipes,* Cambridge University Press, Cambridge, 1986.

Ortega, J.M., *Introduction to Parallel and Vector Solutions of Linear Systems,* Plenum Press, New York, 1988.

Rice, J.R., *Matrix Computations and Mathematical Software,* McGraw-Hill, New York, 1983.

Problems

1. **a** Steady fully developed laminar flow through a tube whose cross section is an equilateral triangle of edge unity has a solution, $u = ah_1h_2h_3$. Here u is the local axial component of velocity, a is a constant, and h_1, h_2 and h_3 are perpendicular distances of point P from the three sides of the triangle. Compute the average velocity in the tube using $a = 5$ units. Use numerical integration.

   b Suggest a method for evaluating the integral $\int_0^\infty \sin x/x\, dx$ by the FFT algorithm.

c Develop quadrature formulas for computing integrals of the form,

$$\int_0^1 \frac{f(x)x^k dx}{\sqrt{1-x}} \quad \text{and}$$

$$\int_0^\infty \frac{f(x)dx}{(1+x)^k}, \quad k = 0, 1, \ldots, n.$$

2. Write a computer program that will determine all the roots of a polynomial of Nth degree using the Newton-Raphson method. Consider all special cases arising from complex and repeating roots. Use this program to determine all eigenvalues of a $N \times N$ matrix.

3. Obtain the least squares solution of the system of equations:

   (a) $x^2 + y^2 = 5$
   $x - 4xy^2 + y^2 = 17$

   (b) $5x + 2y = 5$
   $x^2 + 3y^2 - 2xy = 11$
   $x + \cos \pi y = 5/4$

4. Extend the algorithm given in Section 2.3.3 to a pentadiagonal matrix whose diagonals are not adjacent to one another. Assume that the bandwidth and diagonal positions are specified.

5. Solve the transient heat conduction equation

$$\frac{\partial T}{\partial t} = T_{xx} + T_{yy} + 1$$

with the initial condition $T(x, y, 0) = 0$, and the boundary conditions $T(0, y, t) = T(1, y, t) = T(x, 0, T) = T(x, 1, t) = 0$ using the method of finite differences (Chapter 5). Compare the programming effort and CPU time on a mainframe computer when matrix inversion is accomplished using the following methods: Gauss-Seidel with and without relaxation, Jakobi, TDMA, pentadiagonal matrix solver and conjugate gradient algorithm.

6. Develop an efficient computer program for implementing the PCG algorithm given by Equations 2.31 to 2.35 for unsymmetric matrices. Note that the factored form of matrix A can be constructed but one should not be required to construct A itself. The following model problem approximated by finite diffferences can be considered for analysis:

$$\nabla \cdot [k(T)\nabla T] = 0$$

with T prescribed on the boundaries of the square region and

$$k(T) = k_0 \left(1 - 0.2(T - T_0)\right)$$

7. Return to Problem 5. Develop an analytical solution of the unsteady heat conduction equation. Compute the local, rms and maximum errors with respect to the grid size and time step. Study the effect of the choice of the matrix inverter and the related convergence criterion on the discretization error.

8. Reconsider the square region of Problem 5. Repeat the calculations of Problems 5 and 7 when the square domain is a tube carrying a fluid. Assume the flow to be fully-developed. The heat generation term may be set to zero in this calculation, while the left wall is at a temperature of unity. Consider thermally developed and developing cases separately. Study the effect of Peclet number on the discretization errors. Study the effect of the choice of the matrix inverter on the discretization error.

Appendix I

Review of Linear Algebra

This Appendix is intended to be supplementary reading material for Chapter 2. We are concerned here with the properties of the matrix A arising from the system of equations $Ax = b$. The matrix of interest here is a square array of real elements a_{ij}, $i, j = 1, \ldots, N$ and is symbolically written as

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \vdots & \vdots & \ldots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{bmatrix}$$

The transpose of a matrix A is denoted as A^T and is equal to $[a_{ji}]$. It is obtained by interchanging symmetrically placed elements of A about the diagonal. A matrix is said to be symmetric if $A = A^T$; it is said to be skew-symmetric if $A = -A^T$. For a skew symmetric matrix $a_{ii} = 0$, for each i. An identity matrix I has elements δ_{ij}, the Kronecker-delta defined as $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Hence

$$I = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}$$

A diagonal matrix has non-zero entries on the diagonal and is zero elsewhere. A tridiagonal matrix has non-zero entries only on the diagonal and positions just above and below the main diagonal. It is of the form

$$A = \begin{bmatrix} d_1 & a_1 & & & 0 \\ b_2 & d_2 & a_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & a_{N-1} \\ 0 & & & b_N & d_N \end{bmatrix}$$

A triangular matrix has zero elements either above or below the main diagonal and has non-zero entries otherwise. Examples of upper (U) and lower (L) triangular matrices are

$$U = \begin{bmatrix} u_{11} & u_{12} & \ldots & u_{1N} \\ 0 & u_{22} & \ldots & u_{2N} \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & u_{NN} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} l_{11} & 0 & \ldots & 0 \\ l_{21} & l_{22} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ l_{N1} & l_{N2} & \ldots & l_{NN} \end{bmatrix}$$

A matrix is said to be banded if all its non-zero entries are within a few diagonals above and below the main diagonal. The total number of these diagonal which

span the non-zero elements is called the *bandwidth*. If the bandwidth is small in comparison to the size of the matrix, it is called *sparse*. A tridiagonal matrix is a sparse banded matrix with a bandwidth of 3. A pentadiagonal matrix is one which has 5 diagonals with non-zero entries.

Associated with matrix A is the determinant det A. The determinant of a 2×2 matrix is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

Determinants of all other matrices are recursively computed using the formula

$$\det A = \sum_{k=1}^{N} a_{jk} A_{jk} \quad \text{for any } j \text{ between 1 and } N \text{ and}$$

$$A_{jk} = (-1)^{j+k} M_{jk}$$

Here M_{jk} is the minor determinant of the $(N-1) \times (N-1)$ matrix that remains when the jth row and kth column are eliminated. M_{jk}'s in turn must be computed by removing further rows and columns till one is left with a 2×2 array of numbers.

Using the idea of determinants, the solution of the system of equations $Ax = b$ can be explicitly written using Cramer's rule as

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{Bmatrix} = \frac{1}{\det A} \begin{bmatrix} A_{11} & A_{21} & \dots & A_{N1} \\ A_{12} & A_{22} & \dots & A_{N2} \\ \vdots & \vdots & \dots & \vdots \\ A_{1N} & A_{2N} & \dots & A_{NN} \end{bmatrix} \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{Bmatrix}$$

where the elements A_{jk} are as defined above. Symbolically the solution is written as $x = A^{-1}b$, where A^{-1} is called the inverse of matrix A. A^{-1} is also a square matrix of size $N \times N$. The (i,j)th element of A^{-1} is $A_{ji}/\det A$. If $\det A = 0$, A^{-1} does not exist. The system of equations $Ax = b$ is then said to be *singular*. Cramer's rule is computationally inefficient and is seldom used in practical calculations. In problems where b is a zero vector, x is also zero unless det A is zero. In this case x is non-zero but indeterminate. As a rule the equation $Ax = 0$ can have non-trivial solutions only if $\det A = 0$.

The determinant of matrix A can become zero for a variety of reasons. These include elements of a row or a column becoming very small and more generally rows (or columns) becoming linearly dependent. While the classical inverse does not exist for this problem it is possible to define a unique generalized inverse of A. From an operational point-of-view the singularity arising from det A becoming zero is seen as making the inverse non-unique. If A has a pair of equations that are linearly dependent we have $(N-1)$ equations for N unknowns. The $(N-1)$ unknowns can be determined upto an arbitrary parameter that fixes the Nth unknown. This results in an infinite number of inverses of matrix A. The generalized inverse is identified as that part of the inverse mapping that corresponds to the set of linearly independent equations.

In numerical calculations a system of equations can become singular when det A is small and not necessarily zero. For example, the system of equations $100x - 50y = 50$ and $99x - 49y = 50$ have an exact solution of $x = y = 1$. However the solution $x = 1.2$ and $y = 1.402$ satisfies both equations to within 0.102, i.e., within 0.2 percent. If the second equation is changed to $99x - 50y = 50$ the solution changes drastically to $x = 0$ and $y = -1$. Such systems are said to be *ill-conditioned;* they are characterized by a large change in the solution for small changes in the matrix entries. Ill-conditioned equations can be detected by comparing the magnitude of the determinant itself with one of the products arising from determinant calculation. In the above example det $A = -50$ while the products involved are 4900 and 4950. In practice the possibility of ill-conditioning must be anticipated during modelling of the physical problem and formulation of the mathematical equations.

The inner product between two vectors a and b with N components is denoted as (a, b) and is evaluated as $\sum_{i=1}^{N} a_i b_i$. For a matrix A and non-zero vectors x and y, the equation

$$(Ax, y) = (x, A^* y)$$

defines the adjoint matrix A^*. If $A = A^*$, A is called *self-adjoint.* It can be readily verified by direct substitution that for matrices with real entries, $A^* = A^T$. Hence a symmetric matrix is also self-adjoint.

If the inner product $(Ax, x) > 0$, for non-zero vectors x, A is called *positive definite* and if $(Ax, x) \geq 0$, A is called *positive semi-definite.* Similar definitions can be given for negative definite and negative semi-definite matrices. From an operational point-of-view, matrices that satisfy Scarborough's condition of diagonal dominance will also be positive definite.

The consequence of applying a matrix A to a vector x is to produce a new vector b. This effect can be geometrically interpreted as rotation of x followed by amplification or attenuation. The latter effect is quantified through a measure called the norm of A and is defined as

$$\text{norm}\, A = \text{least upper bound}_{|x|>0} \frac{(Ax, Ax)}{(x, x)} = \max_{|x|=1} (Ax, Ax)$$

The right hand side is evaluated for all possible vectors x and the largest value thus obtained is the norm of A.

As an example consider

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

For x with components (x_1, x_2), $Ax = (x_1 + x_2, x_2)$ and $(Ax, Ax) = (x_1 + x_2)^2 + x_2^2$. The norm of A can be computed by solving the following constrained maximization problem:

$$(x_1 + x_2)^2 + x_2^2 = \text{maximum}$$

subjected to

$$(x, x) = x_1^2 + x_2^2 = 1$$

This gives

$$x_1 = 2/\sqrt{10 + 2\sqrt{5}} \quad \text{and} \quad x_2 = (1 + \sqrt{5})/\sqrt{10 + 2\sqrt{5}}$$

$$\text{and norm } A = (x_1 + x_2)^2 + x_2^2 \approx 2.617$$

Problem $Ax = \lambda x$ admits the trivial solution $x = 0$ for a non-singular matrix A and an arbitrary scalar value of λ. However for special choices of λ, it is possible to discover non-zero vectors x which satisfy the equation $Ax = \lambda x$. Then, λ is called an *eigenvalue* and the corresponding vector x is called an *eigen vector*. The geometric meaning of the eigen pair is as follows. x is that special vector which when operated by A is amplified (but not rotated) by a scale factor λ.

Eigenvalues and eigen vectors contain a considerable amount of information about matrix A from which they are generated. They are frequently required in the analysis of iteration schemes and may have to be computed as intermediate quantities in many calculations. In particular the largest and the smallest eigenvalues of matrix A are of considerable importance. We discuss here the determination of the eigenvalues and eigen vectors of a matrix A.

The eigenvalue problem can be written as, $(A - \lambda I)x = 0$. Since we look for non-zero solutions for x we require det $(A - \lambda I) = 0$. This determinant is a polynomial of Nth degree and the eigenvalues will be the N roots of this polynomial. For example, if $N = 3$ the characteristic equation that determines the eigenvalues is

$$\begin{aligned}(a_{11} - \lambda)\ [(a_{22} - \lambda)(a_{33} - \lambda) - a_{23}a_{32}] - a_{12}\ [a_{21}(a_{33} - \lambda) - a_{31}a_{23}] \\ + a_{13}\ [a_{21}a_{32} - a_{33}(a_{22} - \lambda)] = 0\end{aligned}$$

Corresponding to each eigenvalue λ_i thus determined the eigen vectors are determined by solving the system of equations

$$Ax = \lambda_i x \quad \text{and} \quad \sum_{i=1}^{N} x_i^2 = 1$$

The second equation is required to complete the under-determined system of equations represented by $Ax = \lambda x$. Eigen vectors are given a symbol e_i and (λ_i, e_i) form an eigen pair. Eigenvalues can be real or complex.

Consider the determination of the eigen pair for the matrix $A = \begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix}$

The characteristic equation is $(5 - \lambda)^2 - 4 = 0$ and so, $\lambda_i = (3, 7)$. The eigen vectors are obtained by solving the following problems.

(a) $\lambda_1 = 3$
$5x_1 + 2x_2 = 3x_1$
$x_1^2 + x_2^2 = 1$

(b) $\lambda_2 = 7$
$5x_1 + 2x_2 = 7x_1$
$x_1^2 + x_2^2 = 1$

The eigen vectors corresponding to $\lambda_1 = 3$ and $\lambda_2 = 7$ are hence given as, $e_1 = (1/\sqrt{2}, 1/\sqrt{2})$ and $e_2 = (1/\sqrt{2}, -1/\sqrt{2})$.

There are special properties of the eigen pairs which are useful in numerical computation. These are

1. The trace of matrix A is equal to the sum of the eigenvalues, i.e., $\sum_{i=1}^{N} a_{ii} = \sum_{i=1}^{N} \lambda_i$. The product of the eigenvalues is equal to the determinant of A. Hence for a non-singular matrix none of its eigenvalues will be zero.

2. If matrix A is self-adjoint (and hence symmetric) then its eigenvalues are all real. Eigen vectors corresponding to each of these eigenvalues are mutually orthogonal, i.e.

$$(e_i, e_j) = 0 \quad \text{if} i \neq j \quad \text{and} \quad = 1 \quad \text{if} \quad i = j$$

It is possible that some eigenvalues of a self-adjoint matrix repeat. However for an eigenvalue of multiplicity k it is possible to construct k mutually orthogonal eigen vectors. Hence if A is self-adjoint its eigen vectors form a basis in an N-dimensional domain. Any vector of length N can then be resolved into each of the directions represented by the eigen vectors as $b = \sum_{i=1}^{N} b_i e_i$, where the coefficients b_i can be easily determined as (b, e_i). This suggests a method for solving the matrix equation $Ax = b$ using the eigen pairs of A. Expanding $x = \sum x_i e_i$ and $b = \sum b_i e_i$ and noting that $Ae_i = \lambda_i e_i$ we get the unknowns x_i as b_i/λ_i. For the matrix $A = \begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix}$ considered above and $b = (9, 12)$, the solution of the equation $Ax = b$ proceeds as follows.

We compute $b_i = (b, e_i)$ as $21/\sqrt{2}$ and $-3/\sqrt{2}$ and $x_i = b_i/\lambda_i$ as $3/\sqrt{2}$ and $-1/\sqrt{2}$. Hence $x = \sum x_i e_i = (1, 2)$.

3. A matrix A and its inverse A^{-1} have eigenvalues λ_i and $1/\lambda_i$ respectively. The matrix $B = P^{-1}AP$, for any non-singular matrix P is said to be similar to matrix A. Similar matrices have identical eigenvalues. The matrices and A and A^T are similar. Matrix products AB and BA are also similar.

4. If A is symmetric as well as positive definite, its eigenvalues will be real and positive.

5. The largest of the absolute values of the eigenvalue of a matrix A is called the *spectral radius* $\rho(A)$; hence

$$\rho(A) = \max_i \mid \lambda_i \mid$$

It can be shown that $\rho(A) \leq$ norm A. The norm of a matrix is defined earlier in this appendix. It can be simply estimated however as,

$$\text{norm}\, A = \max_i \sum_{j=1}^{n} \mid a_{ij} \mid$$

and is called the row-sum norm. This can be used to get the upper bound for the spectral radius. The ratio of the largest to the smallest eigenvalue of matrix is called its *spectral condition number*.

6. Consider a self-adjoint matrix A. The matrix Q formed by placing eigen vectors of unit magnitude columnwise adjacent to one-another is called the modal matrix of A. Hence $Q = [(e_1)\ldots(e_N)]$. It can be shown that Q is orthogonal, i.e., $Q^{-1} = Q^T$ with a norm of unity. The matrix $Q^{-1}AQ$ $(= Q^TAQ)$ can be shown to be diagonal and

$$Q^{-1}AQ = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & \lambda_N \end{bmatrix}$$

where λ_i's are eigenvalues of A. $Q^{-1}AQ$ is called the *canonical* form of matrix A.

7. If A is not self-adjoint it can still have N distinct eigenvalues and the corresponding eigen vectors will be linearly independent but will not form an orthogonal basis. Matrix Q can still be constructed and det Q will be non-zero. $Q^{-1}AQ$ will once again be a diagonal matrix with eigenvalues of A on its diagonal.

If A is not self-adjoint and the eigenvalues are not distinct then the repeating eigenvalues cannot be associated with a set of orthogonal eigen vectors. However it is possible to find a set of eigen vectors for the repeating eigenvalues of A which are at least linearly independent. Specifically this is done as follows. If the eigenvalue λ_i repeats k times, the k eigen vectors e_i^k are $e_i^l\,(l = 1, \ldots, k)$. These are found by obtaining non-trivial solutions of the problems

$$(A - \lambda_i I)\, e_i^1 = 0, \quad (A - \lambda_i I)^2\, e_i^2 = 0 \ldots\ (A - \lambda_i I)^k\, e_i^k = 0.$$

Vectors e_i^l are linearly independent and are called generalized eigen vectors. Once again the modal matrix Q can be constructed using the generalized eigen vectors. Q is however not orthogonal $(Q^T \neq Q^{-1})$ though det $Q \neq 0$. The matrix $Q^{-1}AQ$ is not diagonal; instead it is in the following upper triangular form

$$Q^{-1}AQ = \begin{bmatrix} \lambda_1 & 0 & \ldots & \ldots & 0 \\ 0 & \ldots & 1 & 0 & 0 \\ \vdots & \ldots & \lambda_k & 1 & \vdots \\ \vdots & \ldots & \ldots & \lambda_k & 0 \\ 0 & 0 & \ldots & \ldots & \lambda_N \end{bmatrix}$$

where the elements above the repeating eigenvalues are unity. This is called the Jordon-canonical form of A.

8. Some of the powerful results for eigen pairs given above hold only if matrix A is symmetric. It is thus tempting to construct for an unsymmetric system of equations $Ax = b$, an equivalent problem $A^TAx = A^Tb$ since A^TA is symmetric. This approach is seldom used since A^TA can be an ill-conditioned matrix. In particular the spectral condition number of A^TA is the square of

that of A. As explained in Chapter 2, a large condition number can adversely affect iterative schemes of matrix inversion.

We now return to the discussion on solving the matrix equation $Ax = b$. If det A is small, A is ill-conditioned and the solution is prone to large errors. Ill-conditioning arises from excessive inter-dependence of row (or column) vectors on one-another and can be treated in the following manner. The motivation for this approach comes from the notion of the modal matrix Q of a matrix A for which Q^{-1} is simply equal to Q^T. This special property is linked to the rows (or columns) of Q being strictly orthogonal to one another. Let Q_i represent the rows of Q. Orthogonality then implies, $(Q_i, Q_j) = 0, \quad i \neq j$ and $= 1$ otherwise. Normalizing Q_i^2 to unity is helpful in practice in controlling round-off errors. In summary, having a matrix A as close to Q as possible is most helpful from the view point of matrix inversion. This can be accomplished as follows:

The rows (or columns) of a given matrix can be orthogonalized (and more specifically orthonormalized) by a procedure known as Gram-Schmidt orthonormalization. The problem is explicitly stated as: Given vectors $a_1, \ldots, a_N$ (as rows or columns of A) find an equivalent set $v_1, \ldots v_N$ which are orthonormal. To do this we start with a_1 and obtain v_1 as $a_1 / \mid a_1 \mid$, where $\mid a_1 \mid^2 = \sum_{i=1}^{N} a_{1i}^2$. We now subtract off the component of a_2 in the direction of v_1 and the remainder is orthogonal to v_1. This can now be normalized. The component of a_2 in the direction of v_1 is (a_2, v_1) and so this step involves the calculations

$$\begin{aligned} a_2' &= a_2 - (a_2, v_1)\, v_1 \quad \text{and} \\ v_2 &= a_2' / \mid a_2' \mid \end{aligned}$$

This process is continued by subtracting components of a_3 in v_1 and v_2 directions and normalizing the remainder.

These steps can be generalized as:
Compute

$$v_1 = a_1 / \mid a_1 \mid$$

For $k = 2$ to N

$$\begin{aligned} a_k' &= a_k - \sum_{j=1}^{k-1} (a_k, v_j)\, v_j \\ v_k &= a_k' / \mid a_k' \mid \end{aligned}$$

end of k-loop.

In practice Gram-Schmidt orthonormalization is not performed after matrix A is constructed. Instead it is combined with the matrix assembly operations so that when A is constructed it is orthonormal. Again, programming considerations dictate that the new and the old matrices have the same name and hence occupy identical memory locations in a computer. Orthonormalization can also be carried out with a few rows of the matrix A when ill-conditioning is known to be localized.

3
Numerical Solution of Ordinary Differential Equations

Ordinary differential equations (ODEs) often appear in the engineering analysis of heat transfer and fluid flow. They are found, for example, in lumped mass analysis of heating and cooling processes, similarity solutions to heat-conduction and boundary-layer type equations and in stability analyses of flows to evaluate their susceptibility to turbulence. Further examples can be given from hydraulics and heat-exchanger design. Therefore, for those interested in fluid mechanics and heat-transfer, some familiarity with ODEs and the methods of their solution is essential. Furthermore, as most of the ODEs in these fields are not amenable to analytical solutions, some knowledge of numerical techniques is also required.

This chapter is a short survey of the basic numerical methods employed in solving ODEs. We consider Runge-Kutta and Predictor-Corrector methods applied to Initial Value Problems, Implicit and Semi-Implicit methods for Stiff equations and the Shooting and Relaxation methods applied to two-point Boundary Value Problems.

3.1 Introduction

An Ordinary Differential Equation (ODE) is one in which a function $y(x)$, say, of one independent variable x, appears with its differentials, e.g.

$$\frac{d^3y}{dx^3} + g\,\frac{d^2y}{dx^2} + f\,\frac{dy}{dx} + hy = r \tag{3.1}$$

where g, f, h, and r are known and can be functions of x and y. The order of an ODE is the order of the highest derivative that appears in the equation; for example, Equation 3.1 is of third-order.

In the standard ODE form shown above, the coefficients f, g, h, and the right hand side r can be either functions of x alone or functions of x, y and some or all derivatives of y up to and including the $(n-1)$th order (where n is the order of the ODE). Linear ODEs are those in which no products of the dependent variables (y and its derivatives) appear. Thus, if the coefficients g, f, h, and r, are either functions of x alone, or are constants, then the ODE is linear. If however even one of these is a function of y or its derivatives then the ODE is non-linear.

3.2 Initial Value and Boundary Value Problems

The ODE shown in (3.1) cannot have a unique solution unless more conditions are imposed. These conditions typically specify the values of the dependent variable and its derivatives at one or two points. The number of such conditions is equal to the order of the ODE. If all these conditions are given at a single point x_0 , then the conditions are called *initial conditions* (I.C.) and the problem (ODE + I.C.) is an initial value problem (IVP). If, however, the additional conditions are given at two points (x_0 and x_L) then the problem is called a two-point boundary value problem (BVP) and the conditions are called *boundary conditions* (B.C.).

For example, if the ODE Equation 3.1 has the conditions of, say, $y(0) = 0, y'(0) = 1, y''(0) = 1/2$, where the primes indicate derivatives of y, then the problem is an IVP, since all the conditions are given at $x_0 = 0$. If however the conditions are, say, $y(0) = 0, y'(1) = 1$ and $y''(0) = 1/2$ then it is a BVP because the conditions are given at two points $x_0 = 0$ and $x_L = 1$. Although, the two problems seem similar, considerably more effort is usually required to solve a BVP numerically than to solve an IVP for the same ODE.

3.3 Analytical and Numerical Solutions

It is important to understand the relative advantages of analytical (or *closed form*) solutions over numerical solutions. For example the IVP

$$\frac{d^2y}{dx^2} + \omega^2 y = 0, \qquad \begin{aligned} y(0) &= A \\ y'(0) &= B \end{aligned} \tag{3.2}$$

has the analytical solution

$$y = A\ \cos\omega x + \frac{B}{\omega}\ \sin\omega x \tag{3.3}$$

which is valid for any arbitrary values of A, B and ω. The particular solution for a given set of values of A, B and ω can be obtained by substituting these

values into Equation 3.3. However, in the case of numerical solutions, for each set of values of A, B, ω, the entire calculation will have to be re-done.

Secondly, the analytical solution gives the solution over the entire domain of x, while the numerical solution is confined to a few discrete points of this domain. Therefore the solution values for, say, $x = \pi$ can be obtained directly from Equation 3.3. In a numerical solution we would have to get the solution value by interpolation, unless $x = \pi$ happened to be at a grid point.

It can be seen that the analytical solution has considerable advantages over the numerical solution. So why do we need numerical solutions? Because it is not always possible to get analytical solutions! Most nonlinear equations encountered do not have known analytical solutions. Furthermore analytical solutions for general boundary conditions are even more rare and so, we often have to resort to numerics to find answers to practical problems.

3.4 IVP of First Order

The general form of this problem is

$$\frac{dy}{dx} = f\ (x, y), \quad y(x_0) = y_0 \tag{3.4}$$

The Euler method for solving this IVP can be obtained by substituting Equation 3.4 in the Taylor series expansion of y

$$y(x + \Delta x) = y(x) + \frac{dy}{dx}\ \Delta x + \frac{d^2y}{dx^2}\ \frac{\Delta x^2}{2} ... \tag{3.5}$$

(where all the terms on the right are evaluated at location x), substituting Equation 3.4 for dy/dx, and by further neglecting the higher order terms to get the discretized equation:

$$y(x + \Delta x) \cong y(x) + \Delta x\ f(x, y) \tag{3.6}$$

The Euler method is based on this approximation for small values of Δx. If y at x_0 is known to be y_0, we can find estimates of the values of y at $x_0 + \Delta x, x_0 + 2\Delta x, x_0 + 3\Delta x$ and so on, by repeated application of Equation 3.6.

This stepping algorithm can be written as

$$\begin{aligned}
y(0) &= y_0 \\
x(0) &= x_0 \\
Do\ i &= 0, N-1 \\
y(i+1) &= y(i) + \Delta x \times f(x(i), y(i)) \\
x(i+1) &= x(i) + \Delta x \\
\text{enddo}
\end{aligned}$$

Here $y(i)$ is the discrete solution at the grid point $x(i) \equiv x_0 + i\Delta x$, $i = 0, 1, ..N$

Upon comparing Equations 3.5 and 3.6, we can see that in the Euler method $y(x + \Delta x)$ is approximated using only the first two terms of the Taylor series, and that the other terms constitute the error in the approximation. When $\Delta x \to 0$, each succeeding term is smaller than the previous one (specifically, $\Delta x^2 \gg \Delta x^3 \gg \Delta x^4$), we denote the error by the lowest order (and most significant) term. We say, therefore, that the error per step in the Euler method is $O(\Delta x^2)$ (where the symbol $O(.)$ denotes order of magnitude). The error in successive steps can be expected to be additive. Therefore, the order of magnitude of the error in the solution at $x_0 + L$ will be $O(\Delta x^2)$ multiplied by O(number of steps) i.e. $O(\Delta x^2) \times O(\frac{L}{\Delta x})$, which is $O(\Delta x)$.

The above ideas can be generalised to define the order of the method of solution (which should not be confused with the order of the ODE being solved). The order q of a method has the following significance: The error per step of an qth order method will be of $O(\Delta x^{q+1})$ and the error in the solution will be of $O(\Delta x^q)$ some distance away from the initial point. From this definition, it follows that the Euler method is a first-order method.

3.5 Runge-Kutta (RK) Methods

The most common IVP solvers are from a class called Runge-Kutta (RK) methods. The general RK method with m function evaluations per step can be applied to Equation 3.4 with the following stepping procedure

$$\begin{aligned}
K_1 &= \Delta x f(x_i, y_i), \\
K_2 &= \Delta x f(x_i + \theta_2 \Delta x, y_i + \beta_{21}\ K_1), \\
K_3 &= \Delta x f(x_i + \theta_3 \Delta x, y_i + \beta_{31} K_1 + \beta_{32} K_2), \\
&\ \ \cdots \\
K_m &= \Delta x\ f(x_i + \theta_m \Delta x, y_i + \beta_{m1} K_1 \\
&+ \ \ldots \beta_{m(m-1)} K_{m-1}),
\end{aligned}$$

and finally

$$y_{i+1} = y_i + (\alpha_1 K_1 + \alpha_2 K_2 + \ldots + \alpha_m K_m) \tag{3.7}$$

where y_i and y_{i+1} are the solution values at the point x_i and $x_{i+1} (= x_i + \Delta x)$ respectively, and the parameters α, β and θ are weights, with θ being in the range [0,1]. The Runge-Kutta methods fall in the class of single-step methods because the solution y_{i+1} is explicitly written in terms of the last value i.e., y_i, and not in terms of the preceding ones y_{i-1}, y_{i-2} etc., in Equation 3.7.

The weights in Equation 3.7 are chosen so as to get the highest order accuracy for the m function evaluations. The weights are usually not fixed uniquely by this condition alone, and more than one common RK method may exist for a

given order.The computational effort per step is proportional to the number of function evaluations of $f(x, y)$. For an qth order RK method, typically, $m = q$ for $q \leq 4$, $m = q + 1$ for $5 \leq q \leq 8$, and $m = q + 2$ for $q \geq 9$.

The Euler method is a first order RK procedure. Some higher order RK algorithms in the form of pseudo-computer programs are given below.

Second Order RK (*Mid-point*):

$$\begin{aligned}
y(0) &= y_0 \\
x(0) &= x_0 \\
\text{Do } i &= 0, N-1 \\
K1 &= \Delta x \times f(x(i), y(i)) \\
K2 &= \Delta x \times f(x(i) + \Delta x/2, y(i) + K1/2) \\
y(i+1) &= y(i) + K2 \\
\text{enddo}
\end{aligned}$$

Third Order RK (*Classical*):

$$\begin{aligned}
y(0) &= y_0 \\
x(0) &= x_0 \\
\text{Do } i &= 0, N-1 \\
K1 &= \Delta x \times f(x(i), y(i)) \\
K2 &= \Delta x \times f(x(i) + \Delta x/2, y(i) + K1/2) \\
K3 &= \Delta x \times f(x(i) + \Delta x, y(i) - K1 + 2 * K2) \\
y(i+1) &= y(i) + (K1 + 4 \times K2 + K3)/6 \\
x(i+1) &= x(i) + \Delta x \\
\text{enddo}
\end{aligned}$$

Fourth Order RK (*Classical*):

This is the most commonly used RK method which seems to offer an optimum compromise between accuracy and computational effort.

$$\begin{aligned}
y(0) &= y_0 \\
x(0) &= x_0 \\
\text{Do } i &= 0, N-1 \\
K1 &= \Delta x \times f(x(i), y(i)) \\
K2 &= \Delta x \times f(x(i) + \Delta x/2, y(i) + K1/2) \\
K3 &= \Delta x \times f(x(i) + \Delta x/2, y(i) + K2/2) \\
K4 &= \Delta x \times f(x(i) + \Delta x, y(i) + K3)
\end{aligned}$$

$$
\begin{aligned}
y(i+1) &= y(i) + (K1 + 2 \times K2 + 2 \times K3 + K4)/6 \\
x(i+1) &= x(i) + \Delta x \\
&\text{enddo}
\end{aligned}
$$

As the order of accuracy increases, so does the amount of computation per step. The computational effort increases essentially linearly with the order of the method, while the logarithm of the error decreases linearly with the order.

3.6 Linear Multi-step and Predictor-corrector Methods

We will now consider the so-called multi-step methods. In such methods the solution value at a grid-point is explicitly related to a number of previous solution values (and not just the last one, as in the single-step methods). An m-step linear multi-step method for Equation 3.4 is based on the presumption of a linear relationship between the grid-point values of y and $f(x, y)$ of the type:

$$
\sum_{j=0}^{m} \alpha_j \; y_{i+j} = \Delta x \sum_{j=0}^{m} \beta_j \; f_{i+j} \qquad (\alpha_m = 1) \tag{3.8}
$$

where $y_k \equiv y(x_k)$ and $f_k \equiv f(x_k, y_k)$.

By using the fact that $f(x, y)$ is the derivative of y at x, we can express the left and right sides of the above equation in terms of the Taylor series of y and the derivatives of y respectively, around the point x_i . We can express this result as

$$
\begin{aligned}
\sum_{j=0}^{m} (\alpha_j \; y_{i+j} \; - \; \Delta x \; \beta_j \; f_{i+j}) \; &= \; C_0 y(x_i) \\
&+ \; \Delta x \; C_1 y'(x_i) + \ldots . \\
&+ \; C_p \; \Delta x^p \; y^{(p)} \; (x_i) . \ldots
\end{aligned} \tag{3.9}
$$

where

$$
\begin{aligned}
C_0 &= \alpha_0 + \alpha_1 \; + \; \ldots \; + \alpha_m \\
C_1 &= \alpha_1 + 2\alpha_2 \; + \; \ldots \; + m\alpha_m - (\beta_0 + \beta_1 + \ldots \; + \beta_m) \\
&\ldots \\
C_p &= \frac{1}{p!} \; (\alpha_1 + 2^p \alpha_2 + \ldots + m^p \alpha_m) \; - \; \frac{1}{(p-1)!} \; (\beta_1 + 2^{p-1} \; \beta_2 \\
&+ \; \ldots \; + \; m^{p-1} \; \beta_m), \quad p = 2, 3, \ldots
\end{aligned} \tag{3.10}
$$

When the multi-step method indicated by Equation 3.8 is used to solve the IVP (Equation 3.4), it is clear that the terms on the right side of Equation 3.9 can be interpreted as the error series of the discretisation. Thus if the multi-step method is to be of qth order, the first $q+1$ coefficients of the error series should be zero i.e. $C_0 = C_1 = ... = C_q = 0$ and $C_{q+1} \neq 0$.

From Equation 3.8 it can be seen that a general linear m-step method has $2m+1$ adjustable parameters associated with it, i.e. $\alpha_0, \alpha_1, ..., \alpha_{m-1}$, and $\beta_0, \beta_1, ..., \beta_m$. The coefficient α_m is always chosen as unity, for convenience. The appearance of adjustable parameters would lead one to conclude that they could be chosen such that the first $2m+1$ coefficients of the error series are zero, that is, an m-step method could be made $2m$th order accurate. However, such a method would not be convergent, i.e. it would not give solutions that would converge to the correct solution as $\Delta x \to 0$. In fact, no linear m-step method can be convergent if it has an order exceeding $m+1$ when m is odd, or $m+2$ when m is even. Therefore, typically, most of the α's are chosen as per convenience, and the highest order convergent scheme is generated by choosing the β's such that the appropriate number ($m+2$ or $m+3$) of coefficients in the error series, Equation 3.10 are zero. Depending on the choice of α's, a number of such *optimally accurate* schemes, i.e. convergent schemes with maximum possible order of accuracy, may be generated for an m-step method.

Usually, however, the accuracy is chosen less than optimal to get schemes with better *numerical stability* properties. Schemes which allow errors to grow in an unrestrained manner are called *unstable* schemes while schemes that supress errors are called *stable*. Often stability is related to the step-size Δx, the scheme being stable for all Δx below a critical value, and unstable beyond. This is called *conditional* stability. Schemes which are stable or unstable for all Δx are called unconditionally stable and unstable, respectively. Unconditionally unstable schemes are useless, while unconditionally stable schemes, other things being the same, are obviously the best. However, the latter are often difficult or computationally expensive to implement and so conditionally stable schemes are most often used, with the ones which allow larger step-sizes being considered to be better schemes (Gear, 1971).

The Adams-Bashforth schemes are a common class of schemes that are widely used. For these methods, $\alpha_m = 1$, $\alpha_{m-1} = -1$, and the other α's and β_m are chosen as zero before-hand (with C_0 =0), and β_0 to β_{m-1} are chosen such that the next m coefficients of the error series C_1 to C_m are zero. This results in a scheme of order m. In another class, the Adams-Moulton schemes have α_m =1, α_{m-1} =-1, and the other α's are chosen as zero (with C_0 =0), and β_0 to β_m are chosen such that the next $m+1$ coefficients of the error series C_1 to C_{m+1} are zero. This results in a scheme of order $m+1$.

The β coefficients for the Adams-Bashforth and the Adams-Moulton schemes are presented in Table 3.1.

Returning to Equation 3.8, let us assume that at the ith step, the values of y_i through y_{i+m-1} (and thus f_i through f_{i+m-1}) are known, then Equation 3.8 can be used to calculate y_{i+m} . If β_m =0, then y_{i+m} will be the only unknown

Table 3.1 Values for β for Multi-step schemes

m	β_0	β_1	β_2	β_3	β_4	β_5
Adams-Bashforth ($\alpha_m = 1$, α_{m-1} =-1, other α's = 0) order of accuracy = m						
1	1	0				
2	1/2	3/2	0			
3	5/12	−16/12	23/12	0		
4	−9/24	37/24	−59/24	55/24	0	
5	251/720	−1274/720	2616/720	−2774/720	1901/720	0
Adams-Bashforth ($\alpha_m = 1$, α_{m-1} =-1, other α's = 0) order of accuracy = $m+1$						
1	1/2	1/2				
2	−1/12	8/12	5/12			
3	1/24	−5/24	19/24	9/24		
4	−19/720	106/720	−264/720	646/720	251/720	
5	27/1440	−173/1440	482/1440	−798/1440	1427/1440	475/1440

and can be solved *explicitly*. If however, $\beta_m \neq 0$ then $f_{i+m}(\equiv f(x_{i+m}, y_{i+m}))$ is also an unknown, and Equation 3.8 becomes an *implicit* equation for y_{i+m} and needs to be solved by iteration. Adams-Bashforth schemes are explicit (as $\beta_m = 0$) while Adams-Moulton schemes are implicit (as $\beta_m \neq 0$). Whatever means are used to solve them, implicit methods are computationally expensive and require many function evaluations of $f(x, y)$ per step. However they offer better numerical stability than explicit methods.

Predictor-Corrector methods offer a compromise between these choices. They use a two-step process, the first step (P) uses an explicit scheme to get a *predicted* value y^P_{i+m} and the second step (C) uses an *implicit* scheme to obtain a *corrected* value y^C_{i+m}. However, in the second step, f_{i+m} is evaluated using the known value y^P_{i+m} and the equation becomes an explicit one for the single unknown y^C_{i+m}. The second step can be repeated with the new values, i.e. PECECEC...., where E indicates the evaluation of f_{i+m} using the last computed estimate of y_{i+m}. If this is continued until the value of y_{i+m} converges, it will amount to the iterative solution of the implicit equation, but this will require too many function evaluations and thus is not usually done. The algorithm commonly used has a general form which can be written as P(EC)$^\mu$ E. This indicates a predictor-step followed by μ corrector steps, and a last evaluation of f_{i+m} , with the final solution y_{i+m}, for use in future steps. This final evaluation E is recommended. The last corrected value will be taken as the value of the solution y_{i+m}. The function evaluations in this strategy will number μ+1 for each step. As the formal order of accuracy of the method does not change with μ, a choice of μ=1 is generally as good as any other (and will save computation as well!).

A popular Predictor-Corrector method is the fourth order Adams-Bashforth-Moulton scheme:

$$\text{Predictor:} \quad y_{i+4}^{P} = y_{i+3} + \frac{\Delta x}{24}\,(55\, f_{i+3} \;-\; 59\, f_{i+2} \;+\; 37\, f_{i+1} \;-\; 9\, f_i)$$

$$\text{Corrector:} \quad y_{i+4}^{C} = y_{i+3} + \frac{\Delta x}{24}\,(9\, f_{i+4} \;+\; 19\, f_{i+3} \;-\; 5\, f_{i+2} \;+\; f_{n+1})$$

which has an estimated error of $-19/270\ (y_{i+4}^{C} - y_{i+4}^{P})\Delta x^5$ per step.

What is the advantage of the Predictor-Corrector method? For the above method, with μ=1, only two function evaluations are performed per step (obviously with previous evaluations being stored for further use). Contrast this with the typically four function evaluations required for an 4th order Runge-Kutta scheme and it is clear that the Predictor-Corrector methods require less computations for the same formal accuracy. This difference in computational effort becomes more significant as the function $f(x, y)$ in Equation 3.4 increases in complexity.

However, the Predictor-Corrector methods are not always to be preferred over Runge-Kutta. The order of accuracy of a method does not necessarily translate into actual accuracy, unless the solution is smooth (in the sense that it can be well-represented locally by polynomials). For solutions that show marked and sudden changes in trend, Runge-Kutta methods may be expected to perform better than Predictor-Corrector methods, which will be led astray by their reliance on the trend of previous steps. Furthermore Predictor-Corrector methods have *starting* problems: for an m-step method, at least m-1 previous values of the solution have to be known before the method can be implemented. This means a few steps have to be performed by Runge-Kutta, or by other means, before the Predictor-Corrector algorithm can be implemented. Yet another disadvantage of the Predictor-Corrector methods is that adaptive stepping algorithms (see Section 3.9) for these methods are more difficult to implement. In view of all these considerations, Runge-Kutta methods are more robust, easier to implement, and are in general preferable over Predictor-Corrector unless computational effort *really* needs to be minimised.

3.7 Simultaneous First-order IVP's

We now consider the case of several dependent variables $u(x), v(x), w(x)$, which are each governed by a first-order ODE:

$$\begin{aligned}
\frac{du}{dx} &= f_u\,(x,\ u,\ v,\ w), \quad u(x_0) = u_0 \\
\frac{dv}{dx} &= f_v\,(x,\ u,\ v,\ w), \quad v(x_0) = v_0 \\
\frac{dw}{dx} &= f_w\,(x,\ u,\ v,\ w), \quad w(x_0) = w_0
\end{aligned} \tag{3.11}$$

The right-hand sides of these equations are functions of u, v and w, in each case. While solving the equation for u, say, we also need the values of v, and w, to evaluate the function f_u. Thus, such equations are *coupled* and they have to be solved simultaneously.

The set (3.11) can be solved by the same methods as were used for the single dependent variable problem (3.4), except that in the stepping algorithm the values of all the solutions u, v, and w, will be found for each level ($i = 1, 2, ...$) before proceeding to the next level. For example, if the fourth-order Runge-Kutta methods were used, then the relevant portion of the program to solve Equation 3.11 will appear as given below.

$$
\begin{aligned}
u(0) &= u_0 \\
v(0) &= v_0 \\
w(0) &= w_0 \\
\\
\text{Do } i &= 0, N-1 \\
a1 &= \Delta x \times f_u\ (x(i), u(i), v(i), w(i)) \\
b1 &= \Delta x \times f_v\ (x(i), u(i), v(i), w(i)) \\
c1 &= \Delta x \times f_w\ (x(i), u(i), v(i), w(i)) \\
\\
a2 &= \Delta x \times f_u\ (x(i) + \Delta x/2, u(i) + a1/2, v(i) + b1/2, w(i) + c1/2) \\
b2 &= \Delta x \times f_v\ (x(i) + \Delta x/2, u(i) + a1/2, v(i) + b1/2, w(i) + c1/2) \\
c2 &= \Delta x \times f_w\ (x(i) + \Delta x/2, u(i) + a1/2, v(i) + b1/2, w(i) + c1/2) \\
\\
a3 &= \Delta x \times f_u\ (x(i) + \Delta x/2, u(i) + a2/2, v(i) + b2/2, w(i) + c2/2) \\
b3 &= \Delta x \times f_v\ (x(i) + \Delta x/2, u(i) + a2/2, v(i) + b2/2, w(i) + c2/2) \\
c3 &= \Delta x \times f_w\ (x(i) + \Delta x/2, u(i) + a2/2, v(i) + b2/2, w(i) + c2/2) \\
\\
a4 &= \Delta x \times f_u\ (x(i) + \Delta x, u(i) + a3, v(i) + b3, w(i) + c3) \\
b4 &= \Delta x \times f_v\ (x(i) + \Delta x, u(i) + a3, v(i) + b3, w(i) + c3) \\
c4 &= \Delta x \times f_w\ (x(i) + \Delta x, u(i) + a3, v(i) + b3, w(i) + c3) \\
\\
x(i+1) &= x(i) + \Delta x \\
u(i+1) &= u(i) + (a1 + 2 \times a2 + 2 \times a3 + a4)/6 \\
v(i+1) &= v(i) + (b1 + 2 \times b2 + 2 \times b3 + b4)/6 \\
w(i+1) &= w(i) + (c1 + 2 \times c2 + 2 \times c3 + c4)/6
\end{aligned}
$$

enddo

3.8 Higher order Initial Value Problems

Consider the IVP which consists of an nth order ODE with initial conditions specifying the function and its first $(n-1)$ derivatives as given below.

$$\frac{d^n y}{dx^n} + g_{n-1}\frac{d^{n-1}y}{dx^{n-1}} + g_{n-2}\frac{d^{n-2}y}{dx^{n-2}} + \dots g_0 y = g(x, y, y', \dots y^{(n-1)}) \quad (3.12)$$

where the g_i 's can be functions of x alone (for linear ODEs) or functions of x and of y and its derivatives (nonlinear ODEs). The variables y', $y''\dots y^{n-1}$ represent the first, second,... $(n-1)th$ derivatives of y. The initial conditions for this problem will be of the type:

$$y(x_0) = y_{0,0}\ ;\ y'(x_0) = y_{0,1}\ ;\ \dots y^{(n-1)}(x_0) = y_{0,n-1}. \quad (3.13)$$

This IVP can be reduced to n simultaneous first-order IVP's for y and the $n-1$ derivatives of y as follows.

$$\begin{aligned} \frac{dy}{dx} &= y', \qquad y(x_0) = y_{0,0} \\ \frac{dy'}{dx} &= y'', \qquad y'(x_0) = y_{0,0} \end{aligned}$$

and finally

$$\frac{dy^{n-1}}{dx} = g(x, y) - g_{n-1}\, y^{n-1} - g_{n-2}\, y^{n-2} \dots - g_0 y \quad (3.14)$$

with

$$y^{(n-1)}(x_0) = y_{0,n-1}$$

Once the nth order IVP has been thus transformed to n simultaneous first order IVP's, these simultaneous IVP's can be solved by the methods outlined in the preceding sections.

For example, let us consider the IVP

$$\frac{d^2y}{dx} + 3\,\frac{dy}{dx} + y = \sin x$$

with the initial conditions

$$y(0) = 0 \text{ and } \left.\frac{dy}{dx}\right|_{x=0} = 1$$

To solve this equation, we first define the auxiliary functions y' and y''

$$y' \equiv \frac{dy}{dx}$$
$$y'' \equiv \frac{dy'}{dx}$$

and rewrite the equations as

$$\begin{aligned} \frac{dy}{dx} = y', \qquad y(0) &= 0 \\ \frac{dy'}{dx} = \sin x - y - 3y', \qquad y'(0) &= 1 \end{aligned}$$

We can solve this system of first order IVP's by, say, the Euler method:

$$\begin{aligned} y(0) &= 0 \\ y'(0) &= 1 \\ \text{Do } i &= 0, N-1 \\ y(i+1) &= y(i) + \triangle x \times y'(i) \\ y'(i+1) &= y'(i) + \triangle x \times (\sin(x(i)) - y(i) - 3 \times y'(i)) \\ x(i+1) &= x(i) + \triangle x \\ \text{enddo} & \end{aligned}$$

(The Euler method is not the best algorithm to solve ODEs! It is used here for illustration only because of its simplicity.)

3.9 Adaptive Stepping

We now consider the relationship between the step-size $\triangle x$ and the accuracy of the solution. It is clear that for an ODE solver of any order, the smaller the step-size, within limits, the smaller will be the error in the solution. There is, however, an upper limit on accuracy, which is imposed by the machine precision of the computer. Generally for single precision computation the round-off error is of the order of 10^{-7} or 10^{-8}, while for fully double precision it is of the order of 10^{-14} - 10^{-16}. Therefore for single precision computations we cannot hope to get better than around 10^{-7} accuracy. Hence decreasing $\triangle x$ beyond an optimum value would just waste computer time without improving accuracy.

Improved accuracy is not without cost. The computer time increases in direct proportion to $1/\triangle x$. Therefore, an intelligent user will optimize the computer time with respect to the requirements in accuracy and choose the step size accordingly. However, rapidly varying portions of the solutions need smaller steps to maintain a given level of accuracy. This problem led to the development

of adaptive-step algorithms by which the step-sizes are automatically adjusted to maintain a specified level of accuracy. These methods are required to do two things: first, estimate the level of accuracy in the solution for a given step-size Δx, and second, estimate the appropriate step-size that would conform to a pre-set standard of accuracy.

How can one gauge the level of accuracy in a solution? If we use a Runge-Kutta method of known order then this is quite easy. For example let $y(x)$ be the computed solution at a point x. If we compute $y(x+\Delta x)$ by using one step of Δx with a fourth order RK method we get a solution value y_1, where

$$y_1 = y_{\text{exact}}\ (x+\Delta x) + \alpha\ (\Delta x^5) + O\ (\Delta x^6) \tag{3.15}$$

where $y_{\text{exact}}(x+\Delta x)$ is the value that would have been obtained by an evaluation of the complete Taylor's series expansion (3.5). However, if we compute $y(x+\Delta x)$ by taking two steps of size $\Delta x/2$, we get y_2 , where

$$y_2 = y_{\text{exact}}\ (x+\Delta x) + 2\ \alpha\ (\frac{\Delta x}{2})^5) + O\ (\Delta x^6) \tag{3.16}$$

where each step gives an error of the order $\alpha(\frac{\Delta x}{2})^5$ and final error is the error of the two steps added together (this assumption is valid for small intervals).

From the above equations we get

$$\alpha = \frac{16}{15}\ \frac{(y_1 - y_2)}{\Delta x^5} + O\ (\Delta x) \tag{3.17}$$

Therefore, the error in each step of size Δx is approximately $\frac{16}{15}\ |y_1 - y_2|$. Thus, with y_1 and y_2 computed, the error in the solution can be estimated and α determined.

The standards of accuracy are typically set by specifying that the error per step should be approximately equal to $\varepsilon f(x,y)\Delta x$, where the parameter ε is assigned a small value, typically in the range 10^{-3} - 10^{-6} . As $f(x,y)\Delta x$ is the approximate change in y per step, we are thus specifying that the error per step should be ε times the change in y.

If the current step size is Δx, what should be the new step-size Δx_{new} which would satisfy the accuracy criterion? We obtain this by imposing the condition:

$$|\alpha|\ \Delta x^5_{\text{new}} = \varepsilon\ f(x,y)\ \Delta x_{\text{new}} \tag{3.18}$$

Upon substitution of Equation 3.17 (neglecting the factor of 16/15) we get

$$\Delta x_{\text{new}} = \left[\frac{\varepsilon\ f(x,y)\ \Delta x}{|y_1 - y_2|}\right]^{1/4}\ \Delta x \tag{3.19}$$

This equation need not be implemented directly, nor at every step. The full-step/half-step procedure to compute y_1 and y_2 may be done every few steps,

and furthermore the step-sizes obtained from Equation 3.19 can be *rounded-off* to keep the step-sizes rational. Here we have considered the adaptive step algorithm for a fourth order Runge-Kutta method; this procedure can be easily extended to any Runge-Kutta scheme of known order (Constantinides, 1987).

3.10 Extrapolation to Arbitrary Accuracy

We now introduce a method which has certain unique properties that can be exploited to create schemes of any desired accuracy. This scheme is the modified mid-point method. When used to solve the IVP Equation 3.4 in the region $x_0 \leq x \leq x_0 + L$, this method is implemented as follows:

$$
\begin{aligned}
\Delta x &= L/N \\
y(0) &= y_0 \\
x(0) &= x_0 \\
x(1) &= x(0) + \Delta x \\
y(1) &= y_0 + f(x_0, y_0) \times \Delta x \qquad \text{! Euler first step} \\
Do\ i &= 2, N \\
x(i) &= x(i-1) + \Delta x \\
y(i) &= y(i-2) + 2 \times \Delta x \times f(x(i-1), y(i-1)) \qquad \text{! Leap-frog scheme} \\
\text{enddo} & \\
y(N) &= 0.5 \times [y(N-1) + y(N) + f(x(N), y(N)) \times \Delta x] \\
& \qquad \text{! last value is modified}
\end{aligned}
$$

The extraordinary feature of the modified midpoint method is that the error in computing the solution at $y(x_0 + L)$ can be expressed in a power series comprising only of *even* powers of Δx,

$$y_{\text{exact}}\ (x_0 + L) = y(N) + \sum_{m=1}^{\infty} a_{2m}\ \Delta x^{2m} \tag{3.20}$$

This property may be used to our advantage as follows: Consider the same problem being solved by the same means but with the step-size $\Delta x/2$, i.e. half the previous value. The grid-point corresponding to $x_0 + L$ will now be $i = 2N$, and if the new solution is represented by $y^*(i)$, Equation 3.20 will now become

$$y_{\text{exact}}\ (x_0 + L) = y^*(2N) + \sum_{m=1}^{\infty} a_{2m}\ \left[\frac{\Delta x}{2}\right]^{2m} \tag{3.21}$$

By multiplying Equation 3.21 by 4 and then subtracting Equation 3.20 we get

$$y_{\text{exact}} = \frac{4y^*\ (2N) - \ y(N)}{3} \ + \ O(\Delta x^4) \tag{3.22}$$

If the first term on the right side is used as the new estimate of $y(x_0 + L)$, it will be fourth-order accurate, while $y(N)$ and $y^*(2N)$ were both only second-order accurate. In the same way we can recompute $y(x_0 + L)$ using different step-sizes and use these values to eliminate more error terms. The advantage of using the modified mid-point method is that for *each* error term eliminated, the accuracy will increase by *two* orders, as only even terms occur in the error series.

The above procedure can be generalised and, when combined with rational function extrapolation to extrapolate the solution values to the limit $\Delta x \longrightarrow 0$, produces the extremely accurate Bulirsch-Stoer method. A more elaborate discussion of this method is outside the scope of this book (see Press *et al.*, 1992).

3.11 Implicit and Semi-implicit Methods

Most of the methods discussed in the previous sections are explicit methods, i.e. each unknown value of the discretized solution is expressed directly in terms of the known values at the previous steps. This is, of course, convenient, because the solution can be obtained by straight-forward evaluation. However, explicit methods are more susceptible to numerical instability, caused by the unrestrained growth of errors. If this happens, the numerical solution, if indeed it can be obtained, is rendered meaningless by the presence of large errors. Such instabilities commonly impose constraints of impractically small step-sizes if explicit methods are used. This is especially for *stiff* equations that contain vastly different scales in the solutions, i.e., some parts of the solutions change rapidly while other parts are comparatively slow-varying. For example, consider the IVP's

$$\begin{aligned} \frac{du}{dx} &= -5.5u - 4.5v, \quad u(0) = 2 \\ \frac{dv}{dx} &= -4.5u - 5.5v, \quad v(0) = 0, \quad x > 0 \end{aligned} \tag{3.23}$$

the solution of which is

$$u = e^{-x} + e^{-10x} \quad \text{and } v = -e^{-x} + e^{-10x}$$

If these equations were to be solved numerically, the e^{-10x} part of the solution will quickly become negligible, *but* its presence will make the step-size Δx in a stable explicit method *ten* times smaller than needed otherwise. This is due to the tendency of explicit methods to become unstable.

Implicit methods are less susceptible to numerical instability and therefore allow larger step-sizes. In these methods, each discretized equation will contain, apart from the solution, more than one unknown quantity. Such sets of

equations need to be solved simultaneously and iteration, or other means, will have to be used to solve them. Typically, the implicit methods generally used are of comparatively lower order. This is because higher order methods, first, require a lot more computational effort, and secondly, show a deterioration in stability characteristics.

A simple first-order implicit method for Equation 3.4 would yield the discretized equation:

$$y_{n+1} = y_n + \Delta x \; f(x_{n+1}, y_{n+1})$$

or after rearrangement

$$y_{n+1} - \Delta x \; f(x_{n+1}, y_{n+1}) = y_n \tag{3.24}$$

Obtaining the solution of (3.24) may or may not be easy, depending on the complexity of $f(x, y)$. Newton-Raphson is a likely method to use, if $f(x, y)$ is non-linear in y.

A second-order implicit method could be:

$$y_{n+1} = y_n + \frac{\Delta x}{2} \; f(x_n, y_n) + \frac{\Delta x}{2} \; f(x_{n+1}, y_{n+1})$$

or after rearrangement

$$y_{n+1} - \frac{\Delta x}{2} \; f(x_{n+1}, y_{n+1}) = y_n + \frac{\Delta x}{2} \; f(x_n, y_n) \tag{3.25}$$

in which, $f(x_n, y_n)$ is known from the previous step. Any solution method for Equation 3.24 can be used for Equation 3.25 as well.

Both schemes (Equations 3.24 and 3.25) are unconditionally stable, i.e. stable for all Δx, for linear ODEs. For non-linear ODEs, stability is not guaranteed, but in general, the schemes will be stable for practical values of Δx (large values of Δx will anyway cause deterioration of accuracy).

However, iteration is usually computationally expensive. It may be avoided by *semi-implicit* methods, which are also generally stable. For example, to generate a first-order semi-implicit method we perform a Taylor expansion, up to first-order, of $f(x_{n+1}, y_{n+1})$ in (3.24) to get

$$y_{n+1} = y_n + \Delta x \; f(x_{n+1}, y_n) + \Delta x \; f_y(x_{n+1}, y_n)(y_{n+1} - y_n) \tag{3.26}$$

where f_y is the partial derivative of f with respect to y, which is assumed to be obtainable in analytical form. Equation 3.26 can be re-written as

$$\begin{aligned} y_{n+1}\left[1-\Delta x\ f_y(x_{n+1},y_n)\right] &= y_n\left[1-\Delta x\ f_y(x_{n+1},y_n)\right] \\ &+ \Delta x\ f(x_{n+1},y_n) \end{aligned} \tag{3.27}$$

where y_{n+1} is the only unknown and it can be obtained easily. A similar operation on equation (3.25) yields the second-order semi implicit method:

$$\begin{aligned} y_{n+1}\left[1-\frac{\Delta x}{2}\ f_y(x_{n+1},y_n)\right] &= y_n\left[1-\frac{\Delta x}{2}\ f_y(x_{n+1},y_n)\right] \\ +\frac{\Delta x}{2}\left[f(x_n,y_n)\right. &+ \left. f\left(x_{n+1},y_n\right)\right] \end{aligned} \tag{3.28}$$

These semi-implicit schemes can be easily generalised to a system of ODEs of the type

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, ...y_M), \qquad i = 1, 2...M \tag{3.29}$$

The first and second order semi-implicit methods for this system can be written as

$$[\ [I] - \Delta x[J]\ ]\,\underline{y}^{n+1} = \left[\ [I] - \Delta x[J]\ \underline{y}^n\right] + \Delta x\underline{f''} \tag{3.30}$$

and

$$\left[\ [I] - \frac{\Delta x}{2}[J]\right]\underline{y}^{n+1} = \left[\ [I] - \frac{\Delta x}{2}[J]\right]\underline{y}^n + \frac{\Delta x}{2}\underline{f^q} + \frac{\Delta x}{2}\underline{f^q} \tag{3.31}$$

respectively, where the underlined quantities are column vectors and those in square brackets are $M \times M$ matrices; $[I]$ is the identity matrix; $\underline{y}^{n+1}$ and $\underline{y}^n$ contain the elements $y_1, y_2 ... y_M$, evaluated at the $(n+1)$th and the nth time-step, respectively; $\underline{f^p}$ and $\underline{f^q}$ contain the elements $f_1, f_2 ... f_M$, evaluated using $(x_n, \underline{y}^n)$, and $(x_{n+1}, \underline{y}^n)$, respectively. The elements of the Jacobian matrix $[J]$, are defined as

$$J_{ij} \equiv \frac{\partial f_i}{\partial y_j} \text{ evaluated at } (x_{n+1}, \underline{y}^n) \tag{3.32}$$

Equations 3.30 and 3.31 can be solved using matrix techniques. This allows one to use a direct method instead of the iterative methods generally used for the fully implicit equations.

3.12 Two-point Boundary Value Problems

In this type of problem, we are not given all the boundary conditions for the ODE at a single point x_0 , rather the boundary conditions are given at two points x_0 and x_L and the solution is to be obtained for $x_0 \leq x \leq x_L$. For example, consider the following problem:

$$\frac{d^2y}{dx^2} + f(x,y)\frac{dy}{dx} + h(x,y)y = 0 \tag{3.33}$$

with boundary conditions

$$\begin{aligned} y(x_0) &= y_0 \\ \left.\frac{dy}{dx}\right|_{x_L} &= y_{1,1} \end{aligned} \tag{3.34}$$

If we make the usual transformations, we get the two coupled first-order ODEs

$$\begin{aligned} \frac{dy}{dx} &= y', \qquad y(x_0) = y_0 \\ \frac{dy'}{dx} &= -f(x,y)y' - h(x,y)y, \qquad y'(x_L) = y_{1,1} \end{aligned} \tag{3.35}$$

However we will not be able to solve these two equations by a straightforward application of the methods discussed in the preceding sections; for if we begin stepping at x_0, we will have no starting value for y' (for y' is specified at $x = x_L$ and not at x_0), and similarly, if we begin stepping at x_L then we will not have a starting value for y, which has its boundary condition specified at x_0 .

Under these circumstances we can do either of the following:
(i) If only one boundary condition is given at x_L we may use the *shooting* method.
(ii) Treat the ODE as a *finite-difference* problem: that is, develop a matrix equation for the solution values at the grid points and solve this matrix equation to obtain the solution values. This method, which is called *relaxation*, is not usually adopted (unless no other method works) because of its algorithmic complexity (see Problem 7).

3.13 The Shooting Method

Consider the case of an nth order BVP for which $(n-1)$ boundary conditions are given at x_0 and one is given at x_L. For an nth order ODE we need n

conditions. For an IVP all n conditions should be specified at one point (x_0). In the shooting method we solve the BVP as an IVP by guessing the missing condition at x_0. The correctness of the guess is judged by seeing how closely the solution satisfies the final condition at x_L. Obviously, for an arbitrary guess, the boundary condition at x_L will never be satisfied. Thus a root finding algorithm to converge to the correct guess is used.

We shall illustrate this method by applying it to an ODE of interest in fluid mechanics - the Falkner-Skan equation for wedge flows.

Example 3.1 Falkner-Skan Similarity Solution to the Boundary Layer

The two-dimensional steady boundary-layer (B.L.) equations, under certain conditions, have *similarity* solutions: The tangential velocity $u(x,y)$ can be expressed as a function of one independent *similarity*-variable η instead of the two coordinates (x,y) which are generally needed. In these circumstances, the coupled partial differential equations for the boundary layer reduce to a single ODE (see White, 1991). Two well-known ODEs of this type are the Blasius and the Falkner-Skan equations. The latter is given below. The tangential velocity $u(x,y)$ (normalised by $U(x)$, the outer flow velocity) is expressed as

$$f'(\eta) \equiv \frac{u(x,y)}{U(x)} \tag{3.36}$$

where f' is the derivative of some function $f(\eta)$ which is governed by the Falkner-Skan equation

$$f''' + ff'' + \beta(1 - f'^2) = 0 \tag{3.37}$$

In the above equation f'' and f''' are the second and third derivatives of f, and β is some parameter dependent on the outer flow.. The boundary conditions for (3.37) are

$$f(0) = f'(0) = 0 \text{ and } f'(\infty) = 1. \tag{3.38}$$

Notice that this problem is a BVP with two boundary conditions at $\eta = 0$ and another at $\eta = \infty$.

To solve this BVP by the shooting method we adopt the following strategy:

(i) We develop an IVP solver for the same ODE for the initial conditions $f(0) = f'(0) = 0$ and any choice of $f''(0)$.

(ii) We then use the *Bisection method* to converge to the correct $f''(0)$ value which yields $f'(\infty) = 1$ in the solution.

The bisection method works as follows. Using the IVP solver we find two values of $f''(0)$, say S_1 and S_2 which yield values of $f'(\infty)$ as r_1 and r_2 respectively such that r_1 and r_2 are respectively below and above the correct value,

i.e. $r_1 < 1$ and $r_2 > 1$. Now as $f'(\infty)$ depends directly on the chosen value of $f''(0)$ (the other boundary conditions being fixed) we expect that the $f'(\infty) = 1$ condition will be satisfied by some value of $f''(0)$ which lies between S_1 and S_2. So we guess a new value of $f''(0)$ as $S = (S_1 + S_2)/2$ which yields, after the IVP solution an $f'(\infty)$ value of, say, r. If $r < 1$ then we replace the old value of S_1 by S (and r_1 by r); otherwise if $r > 1$ the old value of S_2 is replaced by S (and r_1 by r). Once again S_1 and S_2 will be such that the correct value of $f''(0)$ will lie between them (but the interval between S_1 and S_2 will be reduced by half!). This process is repeated and at each step the interval between S_1 and S_2 is reduced by half and the correct $f''(0)$ is squeezed into this interval. When $f'(\infty)$ is close enough to 1, for any S value, the process is stopped. Alternative approaches such as the Newton-Raphson scheme (Chapter 2) or the method of least squares (Nachtsheim and Swigert, 1965) can also be used.

In the discussion above, we have repeatedly mentioned the condition related to $f'(\infty)$. Of course if we had actually to solve the ODE for $0 \leq \eta \leq \infty$ it would not be possible to do so *numerically* as infinite amount of computation and storage would be required. Fortunately, however, the solution is such that f' becomes essentially constant (up to six decimal places) for η greater than or equal to 6. Therefore $\eta \longrightarrow \infty$ can be replaced by $\eta = 6$ in the solver and the computation needs to be done only for $0 \leq \eta \leq 6$! The numerical solution of (3.36) for various values of β is summarized in Table 3.2.

3.14 Relaxation methods

The shooting method becomes more difficult to use successfully when more than one boundary condition is prescribed at the far boundary. This requires that we use a multi-dimensional search operation to find the right values for the initial conditions at the wall boundary. Although this can be, and is done, multi-dimensional searches for a root do not have guaranteed convergence and need good initial guesses to succeed. Under these circumstances relaxation methods become a viable alternate route to the numerical solutions of boundary value problems.

In relaxation methods, the ODEs are expressed in terms of finite-difference equations and are converted to a system of linear simultaneous equations and solved by matrix techniques. If the ODEs are non-linear, then the resulting system of simultaneous equations are also non-linear, but are linearized by using previous estimates of the solution while the solution is iterated (*relaxed*) until it converges. The details of this procedure are explained below.

In a relaxation method, an nth order ODE is expressed in terms of n first-order ODEs. Hence, the equation

$$\frac{d^n y}{dx^n} + g_{n-1}\frac{d^{n-1}y}{dx^{n-1}} + g_{n-2}\frac{d^{n-2}y}{dx^{n-2}} + \ldots + g_0 y = g(x, y, y', \ldots y^{(n-1)}) \quad (3.39)$$

can be reduced to n simultaneous first-order ODEs:

Table 3.2 : Numerical Values of the Streamwise Velocity $f'(\eta)$ for Falkner-Skan Similarity Flows (from *Viscous Fluid Flow*, White, 1991).

β	−0.19884	−0.18	0.0	0.3	1.0	2.0	10.0
f_0''	0.0	0.12864	0.46960	0.77476	1.23259	1.68722	3.67523
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.1	0.00099	0.01376	0.04696	0.07597	0.11826	0.15876	0.31843
0.2	0.00398	0.02933	0.09391	0.14894	0.22661	0.29794	0.54730
0.3	0.00895	0.04668	0.14081	0.21886	0.32524	0.41854	0.70496
0.4	0.01591	0.06582	0.18761	0.28569	0.41446	0.52190	0.81043
0.5	0.02485	0.08673	0.23423	0.34938	0.49465	0.60964	0.87954
0.6	0.03578	0.10937	0.28058	0.40988	0.56628	0.68343	0.92414
0.7	0.04868	0.13373	0.32653	0.46713	0.62986	0.74496	0.95259
0.8	0.06355	0.15975	0.37196	0.52107	0.68594	0.79587	0.97057
0.9	0.08038	0.18737	0.41672	0.57167	0.73508	0.83767	0.98185
1.0	0.09913	0.21651	0.46063	0.61890	0.77787	0.87172	0.98888
1.2	0.14232	0.27899	0.54525	0.70322	0.84667	0.92142	0.99591
1.4	0.19274	0.34622	0.62439	0.77425	0.89681	0.95308	0.99856
1.6	0.24982	0.41691	0.69670	0.83254	0.93235	0.97269	0.99957
1.8	0.31271	0.48946	0.76106	0.87906	0.95683	0.98452	0.99998
2.0	0.38026	0.56205	0.81669	0.91509	0.97322	0.99146	0.99999
2.2	0.45097	0.63269	0.86330	0.94211	0.98385	0.99542	
2.4	0.52308	0.69942	0.90107	0.96173	0.99055	0.99761	
2.6	0.59460	0.76048	0.93060	0.97549	0.99463	0.99879	
2.8	0.66348	0.81449	0.95288	0.98480	0.99705	0.99940	
3.0	0.72776	0.86081	0.96905	0.99088	0.99842	0.99972	
3.2	0.78578	0.89853	0.98037	0.99471	0.99919	0.99987	
3.4	0.83635	0.92854	0.98797	0.99704	0.99959	0.99995	
3.6	0.87892	0.95138	0.99289	0.99840	0.99980	0.99998	
3.8	0.91315	0.96805	0.99594	0.99916	0.99991	0.99999	
4.0	0.93982	0.97975	0.99777	0.99958	0.99996		
4.5	0.97940	0.99449	0.99957	0.99994	0.99999		
5.0	0.99439	0.99997	0.99994	0.99999			

$$\begin{aligned}\frac{dy}{dx} &= y' \\ \frac{dy'}{dx} &= y'' \quad \ldots\end{aligned}$$

and finally

$$\begin{aligned}\frac{dy^{(n-1)}}{dx} &= g(x, y \ldots y^{(n-1)}) \; - \; g_{n-1} \; y^{(n-1)} \\ &\quad \ldots g_{n-2} \; y^{(n-2)} \ldots - g_0 \; y \qquad (3.40)\end{aligned}$$

where the g_j's can be functions of x alone (in linear ODEs) or also functions of y and its derivatives (non-linear ODEs).

The decomposition above is similar to the one in Section 3.8, except that here the boundary conditions are not all at x_0 but are distributed between the two boundary points x_0 and x_L. These boundary conditions should specify a total of n boundary values for y and its derivatives.

The domain between x_0 and x_L is discretized in intervals of $\Delta x \equiv (x_L - x_0)/N$. The grid points are $x_i \equiv x_0 + i\Delta x$, where $i = 0, 1, \ldots N$. The corresponding values of y and its derivatives at x_i are represented by $y_i, y_i', y_i'' \ldots y_i^{(n-1)}$. The latter are the unknown variables of the discrete solution. There are a total of $n(N+1)$ such unknown variables and we require a corresponding number of equations.

Many different discretizations may be adopted to transform Equations (3.40) into finite-difference form. Here we will adopt a second-order scheme for these equations:

$$\begin{aligned}y_i - y_{i+1} + \frac{\Delta x}{2}\,(y_i' + y_{i+1}') &= 0 \\ y_i' - y_{i+1}' + \frac{\Delta x}{2}\,(y_i'' + y_{i+1}'') &= 0 \quad \ldots\end{aligned}$$

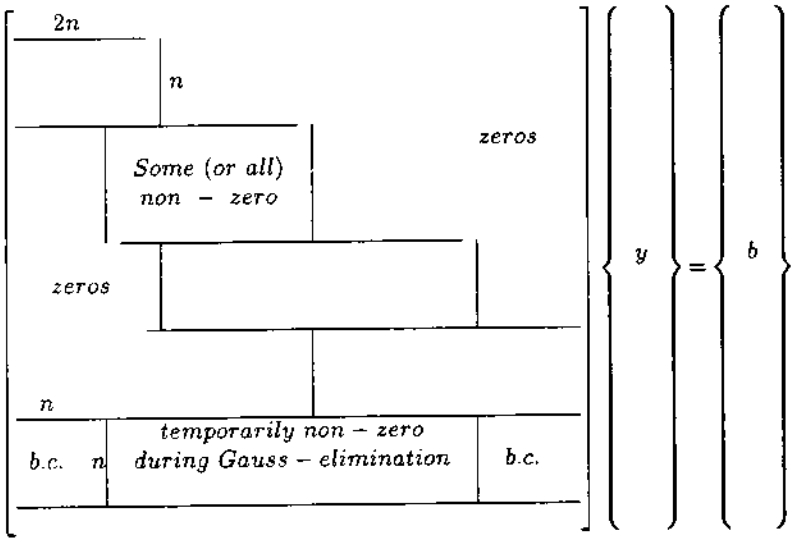
and finally

$$\begin{aligned}y_i^{(n-1)} - y_{i+1}^{(n-1)} \; &+ \; \frac{\Delta x}{2}\,\{(g_i + g_{i+1}) \; - \; (g_{n-1,i}\; y_i + g_{n-1,i+1}\; y_{i+1}) \ldots\} \\ &= \; 0 \qquad (3.41)\end{aligned}$$

where terms such as g_i and $g_{n-1,i}$ are $g(x_i)$ and g_{n-1} respectively, evaluated at x_i. Equation 3.40 holds for $i = 0, 1, \ldots N-1$. This will yield nN equations. The extra n equations will be provided by the boundary conditions.

We define the solution vector $\{y\}$ such that it consists of $n(N+1)$ elements of which the first n elements are $y_0, y_0', y_0'' \ldots y_0^{(n-1)}$, the next n elements are $y_i, y_i', y_i'' \ldots y_i^{(n-1)}$ and so on. We can write these equations in matrix form, putting the equations in an order such that successive sets of n equations correspond to Equation 3.40 applied to $i = 0, 1, 2, \ldots N-1$, and the final set correspond

to the boundary conditions. As Equation 3.40 comprises sets of n equations involving only the variables at neighbouring grid-points, the matrix will have a block structure. The final n boundary conditions will involve only the variables at $i = 0$ and $i = N$. The resultant coefficient matrix will have the so-called *singly bordered block diagonal* form shown below:



(3.42)

Here $\{b\}$ contains the right-hand sides of the equations. The coefficient matrix structure is a standard form of sparse matrix and the Gauss-elimination procedure can easily be modified to solve the matrix equation efficiently. The procedure must include partial pivoting (row interchanges) but this needs be confined only to equations within a block.

What if the ODEs are non-linear? In this case the finite difference equations can still be written in the matrix equation form except that the coefficient matrix will be a function of the discrete values of y and its derivatives as well as x. We can adopt a strategy of *lagging* the coefficient matrix, i.e. computing it using previous estimate for y and its derivatives. Thus we begin with an initial estimate $\{y\}_0$; use this to compute the coefficient matrix and $\{b\}$; then solve the matrix equation to obtain the solution $\{y\}_1$ which is the next estimate. We recompute the coefficient matrix and $\{b\}$ using $\{y\}_1$ and solve to get $\{y\}_2$ and so on. The procedure is continued until the changes between successive solutions cease to be significant. The additional burden of iteration is the main complication introduced by the non-linearity of the ODEs. For the linear case the matrix equation needs to be solved only once.

Example 3.2 Falkner-Skan Equation revisited

Let us consider how the above procedure could be applied to the boundary value problem (Equations (3.36 - 3.38)) in the Section 3.13 (Example 3.1). On

reducing the third-order ODE to three first-order ODEs we get

$$\begin{aligned}
\frac{df}{d\eta} &= f', \\
\frac{df'}{d\eta} &= f'', \\
\frac{df''}{d\eta} &= -ff'' - \beta(1 - f'^2),
\end{aligned} \tag{3.43}$$

with the boundary conditions of $f(0) = f'(0) = 1$ and $f'(\infty) = 1$.

Once again we impose the last boundary condition at a finite but large $\eta_{\max}$ (equal to, say, 6 for this case) and we discretize the domain $0 \leq \eta \leq \eta_{\max}$ with the grid points $\eta_i \equiv i\Delta\eta (i = 0, 1, ...N)$ where $\Delta\eta = \eta_{\max}/N$. Upon applying Equation 3.41 we get

$$\begin{aligned}
-f_i - \frac{\Delta x}{2} f_i' &+ f_{i+1} - \frac{\Delta x}{2} f_{i+1}' = 0 \\
-f_i' - \frac{\Delta x}{2} f_i'' &+ f_{i+1}' - \frac{\Delta x}{2} f_{i+1}'' = 0 \\
-(\frac{\beta\Delta x}{2} f_i')\, f_i' + (1 + \frac{\Delta x}{2} f_i)\, f_i'' &- (\frac{\beta\Delta x}{2} f_{i+1}')\, f_{i+1}' \\
&+ (1 + \frac{\Delta x}{2} f_{i+1})\, f_{i+1}'' = -\beta\Delta x, \\
&\text{for } i = 0, 1, 2...N-1
\end{aligned} \tag{3.44}$$

and the boundary conditions of $f_0 = 0, f_0' = 0$ and $f_N' = 1$. In the above equation, the coefficients in brackets ( ) are evaluated using values of f_i, f_i' etc. from the previous iteration (otherwise the third equation of Equation 3.44 will be non-linear). Note that we could also have linearized (for example) the ff'' term in the ODE by including the f'' in the coefficient and leaving f as the unknown; but we follow the usual practice of treating the lower derivative as the coefficient and the higher derivative as unknown.

If Equations 3.43 are written in matrix form we get

$$\begin{aligned}
[A] \quad &\{ f_0 \quad f_0' \quad f_0'' \quad f_1 \quad f_1' \quad f_1'' \quad \ldots \quad \ldots \quad f_N \quad f_N' \quad f_N'' \}^T \\
&= \{ 0 \quad 0 \quad -\beta\Delta x \quad 0 \quad 0 \quad -\beta\Delta x \quad \ldots \quad \ldots \quad 0 \quad 0 \quad 1 \}^T
\end{aligned} \tag{3.45}$$

where $[A]$ is the matrix

$$
\left[\begin{array}{cccccc|c}
-1 & -\Delta x/2 & 0 & 1 & -\Delta x/2 & 0 & \\
0 & -1 & -\Delta x/2 & 0 & 1 & \Delta x/2 & (i=0) \\
0 & -\beta\Delta x f_0'/2 & 1+\Delta x f_o/2 & 0 & -\beta\Delta x f_1'/2 & 1+\Delta x f_1/2 & \\
\hline
 & & & & \textit{block diagonal} & & (i=1) \\
 & & & & & & \textit{etc} \\
\hline
1 & 0 & 0 & & & & 0\ 0\ 0 \\
0 & 1 & 0 & & \textit{b.c.} & & 0\ 0\ 0 \\
0 & 0 & 0 & & & & 0\ 1\ 0
\end{array}\right]
$$

To obtain the ODE solution, we begin with an initial guess, say $f_i = f_i' = 0$ for $i = 0, 1..N$; compute the coefficient matrix; and solve the matrix equation to get the next values of f, f' and f'' for all grid-points. Using these values we again compute the coefficient matrix and solve the matrix equation to get the next guess and so on, until the solutions converge.

3.15 Closure

The present chapter has introduced numerical techniques for solving linear and non-linear ordinary differential equations. Initial as well as boundary value problems have been considered. Explicit and implicit techniques, error control and issues of accuracy and stability have been discussed. Systems of ordinary differential equations occur frequently in the lumped parameter models of engineering systems, where the independent variable is time; steady one dimensional models are also frequently encountered (Hodge, 1990). The solvers presented in this chapter are ideally suited for numerical integration of these systems. Related examples are included in the collection of problems below.

References

Constantinides, A., *Applied Numerical Methods with Personal Computers*, McGraw-Hill, New York, 1987.

Gear, C.W., *Numerical Initial Value Problems in Ordinary Differential equations*, Prentice Hall, New Jersey, 1971.

Henrici, P., *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York, 1962.

Hodge, B.K., *Analysis and Design of Energy Systems*, Prentice Hall, New Jersey, 1990.

Jain, M.K., *Numerical Solutions of Differential Equations*, Wiley Eastern, New Delhi, 1979.

Nachtsheim, P.R. and Swigert, P., Satisfaction of Asymptotic Conditions in Numerical Solution of Systems of Nonlinear Equations of Boundary-layer Type, NASA-TN-D 3004, 1965.

Press, W,H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recipes in Fortran 77: The art of scientific computing*, Second edition, Cambridge University Press, UK, 1992; also see Vol. 2, *Numerical Recipes in Fortran 90: The art of parallel scientific computing*, 1996.

White, F.M., *Viscous Fluid Flow*, McGraw-Hill, New York, 1991.

Problems

1. A parallel-flow heat exchanger consists of two co-axial pipes, the inner pipe carrying hot fluid, the outer one cold fluid. Heat transfer between the two fluids across the tube wall is modelled by the differential equations:

$$\begin{aligned} \dot{m}_h\ C_h\ \frac{dT_h}{dx} + H_i\ P_i\ (T_h - T_c) &= 0; \quad 0 \le x \le L \\ \dot{m}_c\ C_c\ \frac{dT_c}{dx} + H_0\ P_0\ (T_c - T_0) + H_i\ P_i\ (T_c - T_h) &= 0 \end{aligned}$$

with boundary conditions

$$T_h(0) = T_{h0}, \text{ and } T_c(0) = T_{c0}$$

where $T_h, \dot{m}_h, C_h$ is the temperature, mass flowrate, and the specific heat of the hot fluid; $T_c, \dot{m}_c, C_C$, are the corresponding quantities for the cold fluid; H_i and H_0 are the convective heat-tansfer coefficients at the inner and outer pipe surfaces; P_i and P_0 are the perimeters of the inner and outer pipes, respectively; and T_0 is the ambient outer temperature. Write a programme to solve these equations by fourth-order Runge-Kutta for the following given set of parameters: $L, \dot{m}_h, C_h, \dot{m}_c, C_c, H_i, H_0, P_i, P_0, T_0, T_{h0}$ and T_{co}.

2. The free-flight of a projectile through the air can be modelled by the two differential equations

$$\begin{aligned} \frac{d^2x}{dt^2} + \kappa V \frac{dx}{dt} &= 0 \\ \frac{d^2y}{dt^2} + \kappa V \frac{dy}{dt} + g &= 0 \\ \text{where} \qquad V &= \left[\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2\right]^{1/2} \end{aligned}$$

with the boundary conditions: $x(0) = 0$, $y(0) = 0$,

$$\frac{dx}{dt}(0) = V_0 \cos\theta, \text{ and } \frac{dy}{dt}(0) = V_0 \sin\theta$$

where x and y are the horizontal and vertical distance from the firing point, V_0 is the initial velocity, θ is the inital firing angle from the horizontal, g is the gravitational acceleration and κ is a drag parameter defined by

$$\kappa = \frac{1}{8m}\, \rho C_D\, \pi\, D^2$$

where ρ is the density of air, C_D is the drag-coefficient, m is the mass of the projectile, D is the diameter of its cross-section (assumed to be circular). Write a program to solve the two differential equations, given the parameters of the problem, using fourth-order Runge-Kutta method.

3. A counter-flow heat exchanger consists of two co-axial pipes, the inner pipe carrying hot fluid, the outer one cold fluid, the flow velocities being oppositely oriented in the pipes. Heat transfer between the fluid phases can be modelled by the differential equations:

$$\dot{m}_h\; C_h\, \frac{dT_h}{dx} \;+\; H_i\, P_i\, (T_h - T_c) \;=\; 0; \quad 0 \le x \le L$$

and

$$\dot{m}_c\; C_c\, \frac{dT_c}{dx} \;+\; H_0\, P_0\, (T_c - T_0) \;+\; H_i\, P_i\, (T_c - T_h) \;=\; 0;$$

with boundary conditions

$$T_h(0) \;=\; T_{h0} \text{ and } T_c(L) = T_{c0}$$

where the variables and parameters are the same as in problem I. Note however, that the cold fluid enters the pipe at $x = L$, in this case. Use the shooting method, combined with fourth-order Runge-Kutta, to write a program to solve this Boundary value problem, given the parameter values.

4. Write a program which combines fourth-order Runge-Kutta with the shooting method to solve the Falkner-Skan equation, Equations 3.36-3.38 in the text, for any given value of the parameter β. Compare your results with those in Table 3.2.

5. Extend the code developed for Problem 4 to determine the value of β for flow separates from the solid boundary; this is equivalent to requiring that the wall shear stress ($f''(0)$) is zero.

6. Stokes' first problem is that of a infinite flat-plate suddenly accelerated from rest to a tangential velocity U_0 , in a viscous fluid of viscosity ν . The velocity of the fluid tangential to the plate surface, u, is a function of time (t), and normal distance from the plate surface (y). The governing equation with initial and boundary conditions is

$$\begin{aligned}
\frac{\partial u}{\partial t} = \nu\, \frac{\partial^2 u}{\partial y^2} \quad (i)\; u &= 0,\; y > 0,\; \text{at}\;\; t = 0, u = 0, y > 0,\; \text{at} t = 0, \\
(ii)\; u &= U, \text{at } y = 0,\; t > 0, \\
(iii)\; u &= 0 \;\; \text{as } y \to \infty,\; t > 0,
\end{aligned}$$

A similarity solution for this problem can be obtained of the form

$$u = U\ f(\eta) \quad \text{where } \eta = \frac{y}{\sqrt{4\nu t}}$$

and the above problem reduces to the BVP:

$$f'' + 2\eta f' = 0, \text{ with (i) } f = 1 \text{ at } \eta = 0, \text{(ii) } f = 0 \text{ at } \eta \to \infty$$

where f' and f'' are first and second derivatives of f with respect to η. The solution of this is $f(\eta) = \text{erfc}(\eta)$, where $\text{erfc}(\eta)$ is the complementry error function defined by:

$$\text{erfc}(\eta) = 1 - \frac{2}{\sqrt{\pi}} \int_0^\eta \exp(-\eta^2)\ d\eta$$

Write a program to solve the BVP using a second-order relaxation method. (You can impose the boundary condition at *infinity* at $\eta = 3$.) Compare the results of your computation with tabulated values of the error function.

7. Solve the Falkner-Skan equation as a second order ODE by the relaxation method of Section 3.14. For this purpose, the governing differential equation can be rewritten in terms of $y \equiv f'$ as

$$y'' = -fy' - \beta(1 - y^2);\ y(0) = 0 \text{ and } y(\infty) = 1 \tag{3.46}$$

and

$$f(\eta) = f(0) + \int_0^\eta y(\eta')d\eta';\quad f(0) = 0. \tag{3.47}$$

Using a finite-difference approximation directly for the second derivative (see Chapter 4, Table 4.1)

$$y_i'' = \frac{y_{i-1} - 2y_i + y_{i+1}}{\Delta\eta^2} + O(\Delta\eta^2) \tag{3.48}$$

$$y_i' = \frac{y_{i+1} - y_{i-1}}{2\Delta\eta} + O(\Delta\eta^2) \tag{3.49}$$

We can write the ODE in terms of $y(\eta)$ over a grid of N points as

$$\begin{aligned} y_0 &= 0 \\ (1 - \frac{\Delta\eta}{2}\ f_i)y_{i-1} - (2 + \beta\ \Delta\eta^2\ y_i)\ y_i &+ \\ (1 + \frac{\Delta\eta}{2}\ f_{i+1})\ y_{i+1} &= -\beta\Delta\eta^2, i = 1, ...N-1 \\ \text{and} \qquad y_N &= 1 \end{aligned}$$

The trapezoidal rule of numerical integration can be used to determine $f(\eta)$ from y. Note that the matrix arising from the discretization given above is tridiagonal, but nonlinear. The powerful TDMA algorithm can be used here for inversion, though iterations will be required for the treatment of nonlinearity. Repeat the calculation by combining TDMA with the Newton-Raphson scheme to handle the nonlinear terms.

4
Finite Differences, Discretization, Consistency, Stability and Fundamentals of Fluid Flow Modelling

4.1 Introduction

Analytical solutions of partial differential equations provide us with closed-form expressions which depict the variation of the dependent variables in the domain. The numerical solutions, based on finite differences, provide us with values of the dependent variables at discrete points in the domain which are known as grid points. Consider Figure 4.1, which shows a domain of calculation in the x-y plane. Let us assume that the spacing of the grid points in the x-direction is uniform, and given by Δx. Likewise, the spacing of the points in the y-direction is also uniform, and given by Δy. It is not necessary that Δx or Δy be uniform. We could imagine unequal spacing in both directions, where different values of Δx between each successive pairs of grid points are used. The same could be presumed for Δy as well. However, often, problems are solved on a grid which involves uniform spacing in each direction, because this simplifies the programming, and often results in higher accuracy. In some class of problems, the numerical calculations are performed on a transformed computational plane which has uniform spacing in the transformed-independent-variables, but non-uniform spacing in the physical plane. These typical aspects will be discussed later in Chapter 13 on grid generation. In the present chapter we shall consider uniform spacing in each coordinate direction. According to our consideration, Δx and Δy are constants, but it is not mandatory that Δx be equal to Δy.

Let us once again refer to Figure 4.1. The grid points are identified by an index i which increases in the positive x-direction, and an index j which increases in the positive y-direction. If (i, j) is the index of point P in Figure 4.1,

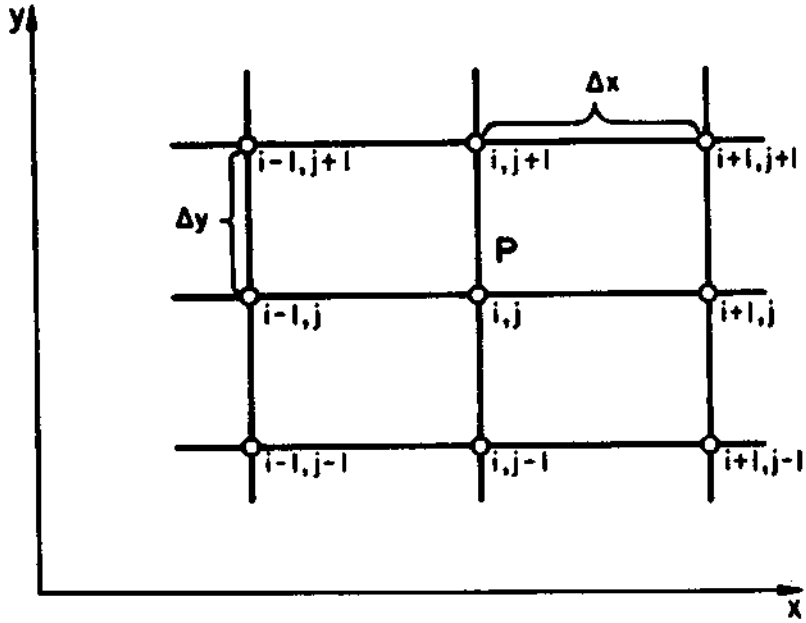


Figure 4.1: Discrete Grid Points around $P(i, j)$

then the point immediately to the right of P is designated as $(i+1, j)$ and the point immediately to the left is $(i-1, j)$. Similarly the point directly above is $(i, j+1)$, and the point below is $(i, j-1)$. **The basic philosophy of finite difference methods is to replace the derivatives of the governing equations with algebraic difference quotients. This will result in a system of algebraic equations which can be solved for the dependent variables at the discrete grid points in the flow field.** Let us now look at some of the common algebraic difference quotients in order to be acquainted with the methods related to discretization of the partial differential equations.

4.2 Elementary Finite Difference Quotients

Finite difference representations of derivatives are derived from Taylor series expansions. For example, if $u_{i,j}$ is the x-component of velocity at point (i, j), then the velocity $u_{i+1,j}$ at point $(i+1, j)$ can be expressed in terms of Taylor series expansion about point (i, j) as

$$u_{i+1,j} = u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{2} + \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{(\Delta x)^3}{6} + \ldots \quad (4.1)$$

Mathematically Equation 4.1 is an exact expression for $u_{i+1,j}$ if the series converges. In practice, Δx is small and any higher order term of Δx is smaller than Δx. Hence for any function $u(x)$, Equation 4.1 can be truncated after a

finite number of terms. For example, if terms of magnitude $(\Delta x)^3$ and higher order are neglected, Equation 4.1 becomes

$$u_{i+1,j} \approx u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{2} \ldots \tag{4.2}$$

Equation 4.2 is second-order accurate, because terms of order $(\Delta x)^3$ and higher have been neglected. If terms of order $(\Delta x)^2$ and higher are neglected, Equation 4.1 is reduced to

$$u_{i+1,j} \approx u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x \tag{4.3}$$

Equation 4.3 is first-order accurate. In Equations 4.2 and 4.3 the neglected higher-order terms represent the truncation error. Therefore, the truncation errors for Equation 4.2 and Equation 4.3 are

$$\sum_{n=3}^{\infty} \left(\frac{\partial^n u}{\partial x^n}\right)_{i,j} \frac{(\Delta x)^n}{n!}$$

and

$$\sum_{n=2}^{\infty} \left(\frac{\partial^n u}{\partial x^n}\right)_{i,j} \frac{(\Delta x)^n}{n!}$$

It is now obvious that the truncation error can be reduced by retaining more terms in the Taylor series expansion of the corresponding derivative and reducing the magnitude of Δx.

Let us once again return to Equation 4.1, and solve for $(\partial u/\partial x)_{i,j}$ as:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} - \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{\Delta x}{2} - \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{(\Delta x^2)}{6} + \ldots$$

or

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x) \tag{4.4}$$

In Equation 4.4, the symbol $O(\Delta x)$ is a formal mathematical nomenclature that means "terms of order of Δx". In Equation 4.4 the order of magnitude of the truncation error is denoted by O. The first-order-accurate difference representation for the derivative $(\partial u/\partial x)_{i,j}$ expressed by Equation 4.4 can be identified as a *first-order forward difference* formula.

We now consider a Taylor series expansion for $u_{i-1,j}$, about $u_{i,j}$:

$$u_{i-1,j} = u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} (-\Delta x) + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(-\Delta x)^2}{2} + \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{(-\Delta x)^3}{6} + \ldots$$

or

$$u_{i-1,j} = u_{i,j} - \left(\frac{\partial u}{\partial x}\right)_{i,j} (\Delta x) + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{2} - \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{(\Delta x)^3}{6} + \ldots \quad (4.5)$$

Solving for $(\partial u/\partial x)_{i,j}$, we obtain

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + O(\Delta x) \quad (4.6)$$

Equation 4.6 is a *first-order backward* or *rearward* difference expression for the derivative $(\partial u/\partial x)$ at grid point (i, j).

Subtracting Equation 4.5 from 4.1

$$u_{i+1,j} - u_{i-1,j} = 2\left(\frac{\partial u}{\partial x}\right)_{i,j} (\Delta x) + \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{(\Delta x)^3}{3} + \ldots \quad (4.7)$$

and solving for $(\partial u/\partial x)_{i,j}$ from Equation 4.7, we obtain

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + O(\Delta x)^2 \quad (4.8)$$

Equation 4.8 is a *second-order central difference* expression for the derivative $(\partial u/\partial x)$ at grid point (i, j).

In order to obtain a finite-difference expression for the second order partial derivative $(\partial^2 u/\partial x^2)_{i,j}$, add Equations 4.1 and 4.5. This produces

$$u_{i+1,j} + u_{i-1,j} = 2u_{i,j} + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} (\Delta x)^2 + \left(\frac{\partial^4 u}{\partial x^4}\right)_{i,j} \frac{(\Delta x)^4}{12} + \ldots \quad (4.9)$$

Solving Equation 4.9 for $(\partial^2 u/\partial x^2)_{i,j}$, we obtain

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 \quad (4.10)$$

Equation 4.10 is a *second-order central difference* form for the derivative $(\partial^2 u/\partial x^2)$ at grid point (i, j).

Difference quotients for the y-derivatives are obtained in exactly the similar way. The results are analogous to the expressions for the x-derivatives.

$$\left(\frac{\partial u}{\partial y}\right)_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{\Delta y} + O(\Delta y) \quad \text{[Forward difference]}$$

$$\left(\frac{\partial u}{\partial y}\right)_{i,j} = \frac{u_{i,j} - u_{i,j-1}}{\Delta y} + O(\Delta y) \quad \text{[Backward difference]}$$

$$\left(\frac{\partial u}{\partial y}\right)_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} + O(\Delta y)^2 \quad \text{[Central difference]}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} + O(\Delta y)^2$$

[Central difference of second derivative]

It is interesting to note that the central difference given by Equation 4.10 can be interpreted as a forward difference of the first order derivatives, with backward differences in terms of dependent variables for the first order derivatives. This is because

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \left[\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)\right]_{i,j} = \frac{\left(\frac{\partial u}{\partial x}\right)_{i+1,j} - \left(\frac{\partial u}{\partial x}\right)_{i,j}}{\Delta x}$$

or

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \left[\left(\frac{u_{i+1,j} - u_{i,j}}{\Delta x}\right) - \left(\frac{u_{i,j} - u_{i-1,j}}{\Delta x}\right)\right]\frac{1}{\Delta x}$$

or

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

The same approach can be made to generate a finite difference quotient for the mixed derivative $(\partial^2 u/\partial x\,\partial y)$ at grid point (i, j). For example,

$$\frac{\partial^2 u}{\partial x\,\partial y} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}\right) \tag{4.11}$$

In Equation 4.11, if we write the x-derivative as a central difference of the y-derivatives, and further make use of central differences to find out the y-derivatives, we obtain

$$\frac{\partial^2 u}{\partial x\,\partial y} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}\right) = \frac{\left(\frac{\partial u}{\partial y}\right)_{i+1,j} - \left(\frac{\partial u}{\partial y}\right)_{i-1,j}}{2\,(\Delta x)}$$

$$\left(\frac{\partial^2 u}{\partial x\,\partial y}\right) = \left[\left(\frac{u_{i+1,j+1} - u_{i+1,j-1}}{2(\Delta y)}\right) - \left(\frac{u_{i-1,j+1} - u_{i-1,j-1}}{2(\Delta y)}\right)\right]\frac{1}{2(\Delta x)}$$

$$\begin{aligned}\left(\frac{\partial^2 u}{\partial x\,\partial y}\right) = & \quad \frac{1}{4\,\Delta x\,\Delta y}\left(u_{i+1,j+1} + u_{i-1,j-1} - u_{i+1,j-1} - u_{i-1,j+1}\right) \\ & + \quad O\left[(\Delta x)^2, (\Delta y)^2\right]\end{aligned} \tag{4.12}$$

Combinations of such finite difference quotients for partial derivatives form finite difference expressions for the partial differential equations. For example, the Laplace equation

$$\nabla^2 T = (\partial^2 T/\partial x^2) + (\partial^2 T/\partial y^2) = 0$$

simplifies to

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

or

$$T_{i+1,j} + T_{i-1,j} + \beta^2(T_{i,j+1} + T_{i,j-1}) - 2(1+\beta^2)\,T_{i,j} = 0 \tag{4.13}$$

where β is the mesh aspect ratio $(\Delta x/\Delta y)$. The value of $T_{i,j}$ can be expressed as

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + \beta^2\,(T_{i,j+1} + T_{i,j-1})}{2(1+\beta^2)} \tag{4.14}$$

Many other forms of difference approximations can be obtained for the derivatives that appear in the governing equations for fluid flow and heat transfer. The basic procedure, however, remains the same. In order to appreciate some more finite difference representations see Tables 4.1 and 4.2. Interested readers are referred to Anderson, Tannehill and Pletcher (1984) for more insight into different kind of discretization methods.

4.3 Implementation Aspects of Finite-Difference Equations

Here we shall look into some of the basic aspects of difference equations. Consider the following one dimensional unsteady state heat conduction equation.

$$\frac{\partial u}{\partial t} = a\,\frac{\partial^2 u}{\partial x^2} \tag{4.15}$$

The dependent variable u (say, temperature) is a function of x and t (time) and a is a constant, the thermal diffusivity of the material, in most instances. It is to be noted that Equation 4.15 is classified as a parabolic partial differential equation.

If we substitute the time derivative in Equation 4.15 with a forward difference, and the spatial derivative with a central difference (usually called FTCS: "Forward Time Central Space" method of discretization), we obtain

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = a\left[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}\right] \tag{4.16}$$

In Equation 4.16, the index for time appears as a superscript, where n denotes conditions at time t, $(n+1)$ denotes conditons at time $(t+\Delta t)$, and so on. The subscript denotes the grid point in the spatial dimension.

However, there must be a truncation error for the equation because each one of the finite-difference quotients has been taken from a truncated series. Considering Equations 4.15 and 4.16, and looking at the truncation errors associated with the difference quotients we can write

$$\begin{aligned}\frac{\partial u}{\partial t} - a\,\frac{\partial^2 u}{\partial x^2} &= \frac{u_i^{n+1} - u_i^n}{(\Delta t)} - a\,\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \\ &+ \left[-\left(\frac{\partial^2 u}{\partial t^2}\right)_i^n \frac{(\Delta t)}{2} + a\left(\frac{\partial^4 u}{\partial x^4}\right)_i^n \frac{(\Delta x)^2}{12} + \ldots\right]\end{aligned} \tag{4.17}$$

Table 4.1: Difference Approximations for Derivatives

$\left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j}$	$=$	$\frac{u_{i+2,j}-2u_{i+1,j}+2u_{i-1,j}-u_{i-2,j}}{2h^3}+O\left(h^2\right)$
$\left(\frac{\partial^4 u}{\partial x^4}\right)_{i,j}$	$=$	$\frac{u_{i+2,j}-4u_{i+1,j}+6u_{i,j}-4u_{i-1,j}+u_{i-2,j}}{h^4}+O\left(h^2\right)$
$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j}$	$=$	$\frac{-u_{i+3,j}+4u_{i+2,j}-5u_{i+1,j}+2u_{i,j}}{h^2}+O\left(h^2\right)$
$\left(\frac{\partial u}{\partial x}\right)_{i,j}$	$=$	$\frac{-u_{i+2,j}+8u_{i+1,j}-8u_{i-1,j}+u_{i-2,j}}{12h}+O\left(h^4\right)$
$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j}$	$=$	$\frac{-u_{i+2,j}+16u_{i+1,j}-30u_{i,j}+16u_{i-1,j}-u_{i-2,j}}{12h^2}+O\left(h^4\right)$
h	$\equiv$	grid spacing in x-direction

Table 4.2: Difference Approximations for Mixed Partial Derivatives

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

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

$$\left(\frac{\partial^2 u}{\partial x\,\partial y}\right)_{i,j} = \frac{1}{\Delta x}\left(\frac{u_{i+1,j+1}-u_{i+1,j-1}}{2\Delta y}-\frac{u_{i,j+1}-u_{i,j-1}}{2\Delta y}\right)+O\left[(\Delta x),\,(\Delta y)^2\right]$$

$$\left(\frac{\partial^2 u}{\partial x\,\partial y}\right)_{i,j} = \frac{1}{2\Delta x}\left(\frac{u_{i+1,j+1}-u_{i+1,j-1}}{2\Delta y}-\frac{u_{i-1,j+1}-u_{i-1,j-1}}{2\Delta y}\right)+O\left[(\Delta x)^2,\,(\Delta y)^2\right]$$

$$\left(\frac{\partial^2 u}{\partial x\,\partial y}\right)_{i,j} = \frac{1}{2\Delta x}\left(\frac{u_{i+1,j}-u_{i+1,j-1}}{\Delta y}-\frac{u_{i-1,j}-u_{i-1,j-1}}{\Delta y}\right)+O\left[(\Delta x)^2,\,(\Delta y)\right]$$

In Equation 4.17, the terms in the square brackets represent truncation error for the complete equation. It is evident that the truncation error (TE) for this representation is $O\left[\Delta t,\ (\Delta x)^2\right]$.

With respect to Equation 4.17, it can be said that as $\Delta x \to 0$ and $\Delta t \to 0$, the truncation error approaches zero. Hence, in the limiting case, the difference equation also approaches the original differential equation. Under such circumstances, the finite-difference representation of the partial differential equation is said to be **consistent**.

4.3.1 Consistency

A finite difference representation of a partial differential equation (PDE) is said to be consistent if we can show that the difference between the PDE and its finite difference (FDE) representation vanishes as the mesh is refined, i.e.,

$$\lim_{\text{mesh}\to 0} (\text{PDE} - \text{FDE}) = \lim_{\text{mesh}\to 0} (\text{TE}) = 0$$

A questionable scheme would be one for which truncation error is $O\left(\Delta t/\Delta x\right)$ and not explicitly $O(\Delta t)$ or $O\left(\Delta x\right)$ or higher orders. In such cases the scheme would not be formally consistent unless the mesh were refined in a manner such that $(\Delta t/\Delta x) \to 0$. Let us take Equation 4.15 and use DuFort-Frankel (1953) differencing scheme. The FDE is

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = a\,\frac{(u_{i+1}^n - u_i^{n+1} - u_i^{n-1} + u_{i-1}^n)}{(\Delta x)^2} \tag{4.18}$$

Now, the leading terms of truncated series form the truncation error for the complete equation:

$$\frac{a}{12}\,\frac{\partial^4 u_i^n}{\partial x^4}\,(\Delta x)^2 - a\,\frac{\partial^2 u_i^n}{\partial t^2}\left(\frac{\Delta t}{\Delta x}\right)^2 - \frac{1}{6}\,\frac{\partial^3 u_i^n}{\partial t^3}\,(\Delta t)^2$$

The above expression for truncation error is meaningful if $(\Delta t/\Delta x) \to 0$ together with $\Delta x \to 0$ and $\Delta t \to 0$. However, (Δt) and (Δx) may individually approach zero in such a way that $(\Delta t/\Delta x) = \beta$.

Then if we reconstitute the PDE from FDE and TE, we obtain

$$\lim_{\Delta t,\,\Delta x\to 0} (\text{PDE} - \text{FDE}) = \lim_{\Delta t,\,\Delta x\to 0} (\text{TE}) = -a\beta^2\left(\partial^2 u/\partial t^2\right)$$

and hence the PDE becomes

$$\frac{\partial u}{\partial t} + a\beta^2\,\frac{\partial^2 u}{\partial t^2} = a\,\frac{\partial^2 u}{\partial x^2}$$

We started with a parabolic equation and ended up with a hyperbolic one!

It is clear that the DuFort-Frankel scheme is not consistent for 1D unsteady state heat conduction equation unless $(\Delta t/\Delta x) \to 0$, together with $\Delta t \to 0$ and $\Delta x \to 0$.

4.3.2 Explicit and Implicit Methods

The solution of Equation 4.16 takes the form of a "marching" procedure (or scheme) in steps of time. We know the dependent variable at all x at a time level from given initial conditions. Examining Equation 4.16 we see that it contains one unknown, namely u_i^{n+1}. Thus, the dependent variable at time $(t + \Delta t)$ can be obtained explicitly from the known results at time t, so to speak, u_i^{n+1} is obtained directly from the known values of u_{i+1}^n, u_i^n and u_{i-1}^n. This is a typical example of an explicit finite-difference method.

Let us now attempt a different discretization of the original partial differential equation given by Equation 4.15. Here we express the spatial differences on the right-hand side in terms of averages between n and $(n+1)$ time level

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{a}{2}\left[\frac{u_{i+1}^{n+1} + u_{i+1}^n - 2u_i^{n+1} - 2u_i^n + u_{i-1}^{n+1} + u_{i-1}^n}{(\Delta x)^2}\right] \tag{4.19}$$

The differencing shown in Equation 4.19 is known as the Crank-Nicolson implicit scheme. The unknown u_i^{n+1} is not only expressed in terms of the known quantities at time level n, but also in terms of unknown quantities at time level $n+1$. Hence, Equation 4.19 at a given grid point i cannot itself result in a solution for u_i^{n+1}. Equation 4.19 has to be written at all grid points, resulting in a system of algebraic equations from which the unknowns u_i^{n+1} for all i can be solved simultaneously. This is a typical example of an implicit finite-difference solution (Figure 4.2). Since they deal with the solution of large systems of simultaneous linear algebraic equations, implicit methods usually require the handling of large matrices.

Generally, the following steps are followed in order to obtain a solution: Equation 4.19 can be rewritten as

$$u_i^{n+1} - u_i^n = \frac{r}{2}\left[u_{i+1}^{n+1} + u_{i+1}^n - 2u_i^{n+1} - 2u_i^n + u_{i-1}^{n+1} + u_{i-1}^n\right] \tag{4.20}$$

where $r = a\,(\Delta t)/(\Delta x)^2$

or

$$-ru_{i-1}^{n+1} + (2+2r)\,u_i^{n+1} - ru_{i+1}^{n+1} = ru_{i-1}^n + (2-2r)\,u_i^n + ru_{i+1}^n$$

or

$$-u_{i-1}^{n+1} + \left(\frac{2+2r}{r}\right)u_i^{n+1} - u_{i+1}^{n+1} = u_{i-1}^n + \left(\frac{2-2r}{r}\right)u_i^n + u_{i+1}^n \tag{4.21}$$

Equation 4.21 has to be applied at all grid points, i.e., from $i = 1$ to $i = k+1$. A system of algebraic equations will result (refer to Figure 4.2).

$$\begin{aligned} &\text{at} \quad i = 2, \qquad -A + B(1)\,u_2^{n+1} - u_3^{n+1} = C(1) \\ &\text{at} \quad i = 3, \qquad -u_2^{n+1} + B(2)\,u_3^{n+1} - u_4^{n+1} = C(2) \end{aligned}$$

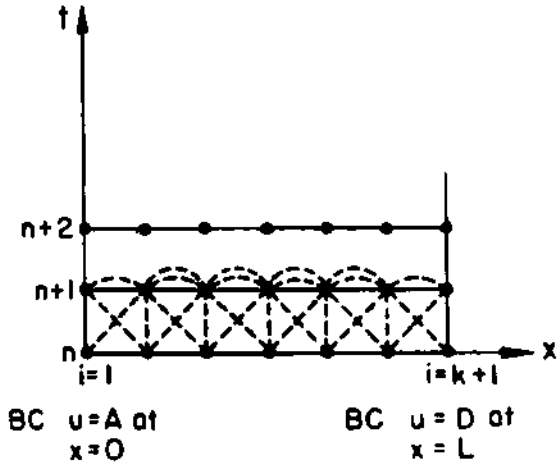


Figure 4.2: Crank Nicolson Implicitness

$$\begin{array}{lll} \text{at} & i=4, & -u_3^{n+1}+B(3)\,u_4^{n+1}-u_5^{n+1}=C(3) \\ & \vdots & \vdots \\ \text{at} & i=k, & -u_{k-1}^{n+1}+B(k-1)\,u_k^{n+1}-D=C(k-1) \end{array}$$

Collecting the algebraic equations in matrix form, we get

$$\begin{bmatrix} B(1) & -1 & 0 & 0 & \dots & 0 \\ -1 & B(2) & -1 & 0 & \dots & 0 \\ 0 & -1 & B(3) & -1 & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & -1 & B(k-1) \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_k \end{Bmatrix}^{(n+1)} =$$

$$\begin{Bmatrix} C(1)+A \\ C(2) \\ C(3) \\ \vdots \\ C(k-1)+D \end{Bmatrix}^{(n)}$$

Here, we express the system of equations in the form of $Ax = B$ where B is the right hand side column vector (known), A the tridiagonal coefficient matrix (known) and x the solution vector (to be determined). Note that the boundary values at $i = 1$ and $i = k + 1$ are transferred to the known right hand side.

For such a tridiagonal system, different solution procedures are available. In order to derive advantage of the zeros in the coefficient-matrix, the well known Thomas algorithm (1949) discussed in Chapter 2, can be used.

The major advantages and disadvantages of explicit and implicit methods are summarized as follows:

Explicit:

- Advantage: The solution algorithm is simple to set up.
- Disadvantage: For a given Δx, Δt must be less than a specific limit imposed by stability constraints. This requires many time steps to carry out the calculations over a given interval of t.

Implicit:

- Advantage: Stability can be maintained over much larger values of Δt. Fewer time steps are needed to carry out the calculations over a given time interval.
- Disadvantages:
  - More involved procedure is needed for setting up the solution algorithm than that for explicit method.
  - Since matrix manipulations are usually required at each time step, the computer time per time step is larger than that of the explicit approach.
  - Since large Δt can be taken, the truncation error is often large, and the exact transients (time variations of the dependent variable for unsteady flow simulation) may not be captured accurately by the implicit scheme as compared to an explicit scheme.

Apparently finite-difference solutions seem to be straightforward. The overall procedure is to replace the partial derivatives in the governing equations with finite difference approximations and then finding out the numerical value of the dependent variables at each grid point. However, this impression is indeed incorrect! For any given application, there is no assurance that such calculations will be accurate or even stable!

Let us now discuss accuracy and stability of finite difference schemes.

4.4 Errors and Stability Analysis

4.4.1 Introduction

There is a formal way of examining the accuracy and stability of linear equations, and this idea provides guidance for the understanding of the behavior of the more complex nonlinear equations which are the governing equations for flow fields.

Consider a partial differential equation, such as Equation 4.15. The numerical solution of this equation is influenced by the following two sources of error.

Discretization:

This is the difference between the exact analytical solution of the partial differential Equation 4.15 and the exact (round-off free) solution of the corresponding finite difference equation (for example, Equation 4.16). The discretization error is simply the truncation error for the finite difference equation plus any errors introduced by the numerical treatment of the boundary conditions.

Round-off:

This is the numerical error introduced for a repetitive number of calculations in which the computer is constantly rounding the numbers to some decimal points.
If A = analytical solution of the partial differtial equation
D = exact solution of the finite difference equation
N = numerical solution from a real computer with finite accuracy
then,
Discretization error = $A - D$ = Truncation error + Error introduced due to treatment of boundary condition
Round-off Error = $\epsilon = N - D$
or

$$N = D + \epsilon \tag{4.22}$$

where, ϵ is the round-off error, which henceforth will be called *error* for convenience. The numerical solution N must satisfy the finite difference equation. Hence from Equation 4.16

$$\frac{D_i^{n+1} + \epsilon_i^{n+1} - D_i^n - \epsilon_i^n}{\Delta t} = a\left[\frac{D_{i+1}^n + \epsilon_{i+1}^n - 2D_i^n - 2\epsilon_i^n + D_{i-1}^n + \epsilon_{i-1}^n}{(\Delta x)^2}\right] \tag{4.23}$$

By definition, D is the exact solution of the finite difference equation, hence it exactly satisfies

$$\frac{D_i^{n+1} - D_i^n}{\Delta t} = a\left[\frac{D_{i+1}^n - 2D_i^n + D_{i-1}^n}{(\Delta x)^2}\right] \tag{4.24}$$

Subtracting Equation 4.24 from 4.23

$$\frac{\epsilon_i^{n+1} - \epsilon_i^n}{\Delta t} = a\left[\frac{\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n}{(\Delta x)^2}\right] \tag{4.25}$$

From Equation 4.25, we see that the error ϵ also satisfies the difference equation.

If errors ϵ_i are already present at some stage of the solution of this equation, then the solution will be stable if the ϵ_i's shrink, or at least stay the same, as the solution progresses in the marching direction, i.e., from step n to $n+1$. If the ϵ_i's grow larger during the progression of the solution from step n to $n+1$,

then the solution is unstable. Hence it stands to reason that for a solution to be stable, the mandatory condition is

$$\left| \frac{\epsilon_i^{n+1}}{\epsilon_i^n} \right| \leq 1 \tag{4.26}$$

For Equation 4.16, let us examine under what circumstances Equation 4.26 holds good.

Assume that the distribution of errors along the x-axis is given by a Fourier series in x, and the time-wise distribution is exponential in t, i.e.

$$\epsilon\,(x,t) = e^{at} \sum_m e^{ik_m x} \tag{4.27}$$

where i is the unit complex number and k the wave number[1]. Since the difference equation is linear, when Equation 4.27 is substituted into Equation 4.25 the behaviour of each term of the series is the same as the series itself. Hence, let us deal with just one term of the series, and write

$$\epsilon_m\,(x,t) = e^{at}\, e^{ik_m x} \tag{4.28}$$

Substitute Equation 4.28 into 4.25 to get

$$\begin{aligned} & \frac{e^{a(t+\Delta t)}\, e^{ik_m x} - e^{at}\, e^{ik_m x}}{\Delta t} \\ = \; & a \left[\frac{e^{at}\, e^{ik_m (x+\Delta x)} - 2e^{at}\, e^{ik_m x} + e^{at}\, e^{ik_m (x-\Delta x)}}{(\Delta x^2)} \right] \end{aligned} \tag{4.29}$$

Divide Equation 4.29 by $e^{at}\, e^{ik_m x}$

$$\frac{e^{a\Delta t} - 1}{\Delta t} = a \left[\frac{e^{ik_m \Delta x} - 2 + e^{-ik_m \Delta x}}{(\Delta x)^2} \right]$$

or

$$e^{a\Delta t} = 1 + \frac{a\,(\Delta t)}{(\Delta x)^2} \left(e^{ik_m \Delta x} + e^{-ik_m \Delta x} - 2 \right) \tag{4.30}$$

Recalling the identity

$$\cos\,(k_m\, \Delta x) = \frac{e^{ik_m\, \Delta x} + e^{-ik_m\, \Delta x}}{2}$$

Equation 4.30 can be written as

$$e^{a\Delta t} = 1 + \frac{a\,(2\Delta t)}{(\Delta x)^2} \left[\cos\,(k_m\, \Delta x) - 1 \right]$$

[1]Let a wave travel with a velocity v. The symbol a is not be confused with thermal diffusivity of Equation 4.15. The time period T is the time required for the wave to travel a distance of one wavelenth λ., so that $\lambda = vT$. Wave number k is defined as $k = 2\pi/\lambda$.

or

$$e^{a\Delta t} = 1 - \frac{4a\,(\Delta t)}{(\Delta x)^2}\,\sin^2[(k_m\,\Delta x)/2] \tag{4.31}$$

From Equation 4.28, we can write

$$\frac{\epsilon_i^{n+1}}{\epsilon_i^n} = \frac{e^{a\,(t+\Delta t)}\,e^{ik_m x}}{e^{at}\,e^{ik_m x}} = e^{a\Delta t} \tag{4.32}$$

Combining Equations 4.32, 4.31 and 4.26, we have

$$\left|\frac{\epsilon_i^{n+1}}{e_i^n}\right| = |\,e^{a\,\Delta t}\,| = \left|1 - \frac{4a\,(\Delta t)}{(\Delta x)^2}\,\sin^2\left[\frac{(k_m\,\Delta x)}{2}\right]\right| \leq 1 \tag{4.33}$$

Equation 4.33 must be satisfied to have a stable solution. In Equation 4.33 the factor

$$\left|1 - \frac{4(a\,\Delta t)}{(\Delta x)^2}\,\sin^2\left[\frac{(k_m\,\Delta x)}{2}\right]\right|$$

is called the *amplification factor* and is denoted by G.

Evaluating the inequality in Equation 4.33, the two possible situations which must hold simultaneously are

(*i*)
$$1 - \frac{4a\,\Delta t}{(\Delta x)^2}\,\sin^2\left[\frac{(k_m\,\Delta x)}{2}\right] \leq 1$$

Thus

$$\frac{4a\,\Delta t}{(\Delta x)^2}\,\sin^2\left[\frac{(k_m\,\Delta x)}{2}\right] \geq 0$$

Since $a\,(\Delta t)/(\Delta x)^2$ is always positive, this condition always holds.

(*ii*)
$$1 - \frac{4a\,\Delta t}{(\Delta x)^2}\,\sin^2\left[\frac{(k_m\,\Delta x)}{2}\right] \geq -1$$

Thus

$$\frac{4a\,\Delta t}{(\Delta x)^2}\,\sin^2\left[\frac{(k_m\,\Delta x)}{2}\right] - 1 \leq 1$$

For the above condition to hold

$$\frac{a\,(\Delta t)}{(\Delta x)^2} \leq \frac{1}{2} \tag{4.34}$$

Equation 4.34 gives the stability requirement for which the solution of the difference Equation 4.16 will be stable. It can be said that for a given Δx, the allowed value of Δt must be small enough to satisfy Equation 4.34. For $a\,(\Delta t)/(\Delta x)^2 \leq (1/2)$ the error will not grow for subsequent marching steps in t, and the numerical solution will proceed in a stable manner. On the contrary, if $a\,(\Delta t)/(\Delta x)^2 > (1/2)$, then the error will progressively become larger and the calculation will be useless.

The above mentioned analysis using Fourier series is called *von Neumann* stability analysis.

4.4.2 First Order Wave Equation

Before we proceed further, let us look at the system of first order equations which are frequently encountered in a class of fluid flow problems. Consider the second-order wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{4.35}$$

Here, c is the wave speed and u is the wave amplitude. This can be written as a system of two first order equations. If $v = \partial u/\partial t$ and $w = c\,(\partial u/\partial x)$, then we may write $\partial v/\partial t = c\,(\partial w/\partial x)$ and $\partial w/\partial t = c\,(\partial v/\partial x)$.

Rather the system of equation may be written as

$$\frac{\partial u}{\partial t} + [A]\,\frac{\partial u}{\partial x} = 0$$

which is a first-order wave equation.

It is implicit that $u = \left\{ \begin{array}{c} v \\ w \end{array} \right\}$ and $A = \left[\begin{array}{cc} 0 & -c \\ -c & 0 \end{array} \right]$. The eigenvalues λ of the $[A]$ matrix are found by

$$\det\,[A - \lambda\, I] = 0, \quad \text{or} \quad \lambda^2 - c^2 = 0.$$

Roots of this characteristic equation are $\lambda_1 = +c \quad \lambda_2 = -c$ representing two travelling waves with speeds given by

$$\left(\frac{dx}{dt}\right)_1 = c \quad \text{and} \quad \left(\frac{dx}{dt}\right)_2 = -c$$

The system of equations in this example is hyperbolic and it has also been seen that the eigenvalues of the A matrix represent the characteristic differential representation of the wave equation. Euler's equations may be treated as a system of first-order wave equations. For Euler's equations, in two dimensions, we can write a system of first order as

$$\frac{\partial E}{\partial t} + [A]\,\frac{\partial E}{\partial x} + [B]\,\frac{\partial E}{\partial y} = -\frac{1}{\rho} S \tag{4.36}$$

where

$$E = \left\{ \begin{array}{c} u \\ v \end{array} \right\}, \quad A = \left[\begin{array}{cc} u & 0 \\ 0 & u \end{array} \right],$$

$$B = \left[\begin{array}{cc} v & 0 \\ 0 & v \end{array} \right] \quad \text{and} \quad S = \left\{ \begin{array}{c} \partial p/\partial x \\ \partial p/\partial y \end{array} \right\}$$

4.4.3 Stability of Hyperbolic and Elliptic Equations

Let us examine the stability characteristics of the first order wave equation given by

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = 0 \tag{4.37}$$

Here we shall represent the spatial derivative by the central difference form

$$\frac{\partial u}{\partial x} = \frac{u_{i+1}^{n} - u_{i-1}^{n}}{2\,\Delta x} \tag{4.38}$$

We shall replace the time derivative with a first order difference, where $u(t)$ is represented by an average value between grid points $(i+1)$ and $(i-1)$, i.e.

$$u\,(t) = \frac{1}{2}\,\left(u_{i+1}^{n} + u_{i-1}^{n}\right)$$

Then

$$\frac{\partial u}{\partial t} = \frac{u_{i}^{n+1} - \frac{1}{2}\,(u_{i+1}^{n} + u_{i-1}^{n})}{\Delta t} \tag{4.39}$$

Substituting Equations 4.38 and 4.39 into 4.37, we have

$$u_{i}^{n+1} = \frac{u_{i+1}^{n} + u_{i-1}^{n}}{2} - c\,\frac{\Delta t}{\Delta x}\,\frac{\left(u_{i+1}^{n} - u_{i-1}^{n}\right)}{2} \tag{4.40}$$

The time derivative approximated by Equation 4.39 is called the *Lax method of discretization*, after the well known mathematician Peter Lax who first proposed it. If we once again assume an error of the form

$$\epsilon_m\,(x,t) = e^{at}\,e^{ik_m\,x}$$

as done previously, and substitute this form into Equation 4.40, following the same arguments as applied to the analysis of Equation 4.16, the amplification factor becomes

$$e^{a\,\Delta t} = \cos\,(k_m\,\Delta x) - i\,C\,\sin\,(k_m\,\Delta x)$$

where $C = c\,(\Delta t/\Delta x)$. The stability requirement is $|\,e^{a\Delta t}\,| \leq 1$. After algebraic simplification, the stability condition culminates in

$$C = c\,\frac{\Delta t}{\Delta x} \leq 1 \tag{4.41}$$

In Equation 4.41, C is the Courant number. This equation restricts $\Delta t \leq \Delta x/c$ for the solution of Equation 4.40 to be stable. The condition posed by Equation 4.41 is called the *Courant-Friedrichs-Lewy* condition, generally referred as the CFL condition.

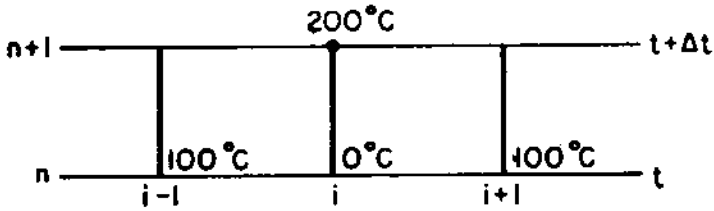


Figure 4.3: Physical Violations Resulting from $r = 1$

Physical Example of Unstable Calculation

Let us take the heat conduction equation once again,

$$\frac{\partial u}{\partial t} = a\,\frac{\partial^2 u}{\partial x^2}$$

Applying FTCS discretization scheme depicts simple explicit representation as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = a\,\frac{(u_{i+1}^n - 2u_i^n + u_{i-1}^n)}{(\Delta x)^2}$$

or

$$u_i^{n+1} = r\,(u_{i+1}^n + u_{i-1}^n) + (1 - 2r)\,u_i^n, \quad \text{where} \quad r = a\,\frac{\Delta t}{(\Delta x)^2}$$

This is stable only if $r \leq 1/2$.

Let us consider a case when $r > 1/2$. For $r = 1$, which is greater than the stability restriction, we get $u_i^{n+1} = 1\,(100 + 100) + (1 - 2) \cdot 0 = 200^o$C, which is impossible. The values of u are shown in Figure 4.3.

Next, an example demonstrating the application of the von Neumann method to *multidimensional problems* is taken up. Consider the vorticity transport equation:

$$\frac{\partial \omega}{\partial t} + u\,\frac{\partial \omega}{\partial x} + v\,\frac{\partial \omega}{\partial y} = \nu\,\left[\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right] \tag{4.42}$$

We shall extend the von Neumann stability analysis for this equation, assuming u and v as constant coefficients (within the framework of linear stability analysis). Using FTCS scheme

$$\begin{aligned}\frac{\omega_{i,j}^{n+1} - \omega_{i,j}^n}{\Delta t} = & - u\,\frac{\omega_{i+1,j}^n - \omega_{i-1,j}^n}{2\,\Delta x} - v\,\frac{\omega_{i,j+1}^n - \omega_{i,j-1}^n}{2\,\Delta y} \\ & + \nu\,\left[\frac{\omega_{i+1,j}^n - 2\,\omega_{i,j}^n + \omega_{i-1,j}^n}{(\Delta x)^2}\right] \\ & + \nu\,\left[\frac{\omega_{i,j+1}^n - 2\,\omega_{i,j}^n + \omega_{i,j-1}^n}{(\Delta y)^2}\right]\end{aligned} \tag{4.43}$$

Let us consider $N = D + \epsilon$ with a two dimensional error distribution

$$\epsilon = e^{at} \sum_m e^{(ik_m x + ik_m y)} \tag{4.44}$$

where N is the numerical solution obtained from computer, D the exact numerical solution of the FDE and ϵ error. Substituting 4.44 into 4.43, and using the trigonometric identities, we obtain

$$G = \frac{\epsilon_{i,j}^{n+1}}{\epsilon_{i,j}^{n}} = \frac{e^{a\,(t+\Delta t)}\, e^{i\,k_m\,(x+y)}}{e^{at}\, e^{ik_m\,(x+y)}} = e^{a\Delta t}$$

where, from Equation 4.43

$$\begin{aligned} G &= 1 - 2\,(d_x + d_y) + 2d_x \cos(k_m\,\Delta x) + 2d_y \cos(k_m\,\Delta y) \\ &\quad -i\,[C_x \sin(k_m\,\Delta x) + C_y \sin(k_m\,\Delta y)] \end{aligned}$$

$$d_x = \frac{\nu\,\Delta t}{(\Delta x)^2}, \quad d_y = \frac{\nu\,\Delta t}{(\Delta y)^2}, \quad C_x = \frac{u\,\Delta t}{\Delta x}, \quad C_y = \frac{v\,\Delta t}{\Delta y}$$

The obvious stability condition $\mid G \mid \leq 1$, leads to

$$d_x + d_y \leq \frac{1}{2}, \quad C_x + C_y \leq 1 \tag{4.45}$$

When

$$d_x = d_y = d \quad (\text{for } \Delta x = \Delta y),\ d \leq \frac{1}{4}$$

we get

$$\frac{\nu\,(\Delta t)}{(\Delta x)^2} \leq \frac{1}{4}$$

This is twice as restrictive as the one dimensional diffusive limitation (compare with Equation 4.34). Again, for the special case ($u = v$ and $\Delta x = \Delta y$)

$$C_x = C_y = C, \quad \text{hence} \quad C \leq \frac{1}{2}$$

which is also twice as restrictive as one dimensional convective limitation (compare with Equation 4.41).

Next, let us look at the stability requirements for the second-order wave equation given by

$$\frac{\partial^2 u}{\partial t^2} = c^2\,\frac{\partial^2 u}{\partial x^2}$$

We replace both the spatial and time derivatives with the central difference scheme (which is second order accurate)

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{(\Delta t)^2} = c^2 \left[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}\right] \tag{4.46}$$

Again assume

$$N = D + \epsilon \tag{4.47}$$

and

$$\epsilon_i^n = e^{at}\, e^{ik_m x} \tag{4.48}$$

Substituting 4.47 and 4.48 in 4.46 and dividing both sides by $e^{at}\, e^{ik_m \Delta x}$, we get

$$e^{a\,\Delta t} - 2 + e^{-a\,\Delta t} = C^2 \left[e^{ik_m \Delta x} + e^{-ik_m \Delta x} - 2\right] \tag{4.49}$$

where

$$C, \text{ the Courant number} = \frac{c\,(\Delta t)}{\Delta x}$$

Using trigonometric identities in Equation 4.49, we get

$$e^{a\Delta t} + e^{-a\Delta t} = 2 - 4\,C^2 \sin^2 \frac{k_m\,\Delta x}{2} \tag{4.50}$$

and, the amplification factor

$$G = \left|\frac{\epsilon_i^{n+1}}{\epsilon_i^n}\right| = |\, e^{a\,\Delta t}\,| \tag{4.51}$$

Using Equation 4.50 we arrive at

$$e^{2a\,\Delta t} - 2\left[1 - 2\,C^2 \sin^2 \frac{k_m\,\Delta x}{2}\right] e^{a\Delta t} + 1 = 0 \tag{4.52}$$

which is a quadratic equation for $e^{a\Delta t}$. This equation, quite obviously, has two roots, and the product of the roots is equal to +1. Thus it follows that the magnitude of one of the roots (value of $e^{a\Delta t}$) must exceed 1 unless both the roots are equal to unity.

However, $e^{a\Delta t}$ is the magnification factor. If its value exceeds 1, the error will grow exponentially, in turn leading to an unstable situation. These possibilities mean that Equation 4.52 should possess complex roots in order to have both the values of $e^{a\Delta t}$ equal to unity. This implies that the discriminant of Equation 4.52 should be negative.

$$\left(1 - 2C^2 \sin^2 \frac{k_m\,\Delta x}{2}\right)^2 - 1 < 0$$

or

$$C^2 < \frac{1}{\sin^2 k_m\,\Delta x/2} \tag{4.53}$$

which is always true if $C < 1$. Hence CFL condition ($C < 1$), must again be satisfied for the stability of second-order hyperbolic equations.

In the light of above discussion, we can say that a finite difference procedure will be unstable if for that procedure, the solution becomes unbounded, i.e.,

the error grows exponentially as the calculation progresses in the marching direction. In order to have a stable calculation, we pose different conditions based on stability analysis. Here we have discussed the von Neumann stability analysis which is indeed applicable to linear differential equations.

Situations may arise where the amplification factor is always less than unity. These procedures are referred to as unconditionally stable. In a similar way for some procedures we may get an amplification factor which is always greater than unity. Such methods are unconditionally unstable.

Over and above, it should be realized that such stability analyses are not really adequate for practical complex problems. Difficulties arise when the velocity components and grid spacings change with position and time. In actual fluid flow problems, the stability restrictions are applied locally. The mesh is scanned for the most restrictive value of the stability limitations, and the resulting minimum Δt is used throughout the mesh. For a differential equation with variable coefficients, the von Neumann condition is only necessary but not sufficient. Generally, a stability calculation procedure of universal applicability is not available. For nonlinear problems we need numerical experimentation in order to identify stable solutions. The stability analysis applied to the linearized equations will describe the initial parameter space of practical utility.

4.5 Fundamentals of Fluid Flow Modelling

We have discussed finite difference methods with respect to the solution of linear problems such as heat conduction. The problems of fluid mechanics are more complex in character. The governing partial differential equations form a nonlinear system which must be solved for the unknown pressures, densities, temperatures and velocities.

Before entering into the domain of actual flow modelling, we shall discuss some subtle points of fluid flow equations with the help of a model equation. The model equation should have convective, diffusive and time dependent terms. Burgers (1948) introduced a simple nonlinear equation which meets the aforesaid requirements and is given as:

$$\frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} = \nu\frac{\partial^2 \zeta}{\partial x^2} \tag{4.54}$$

Equation 4.54 is called *Burgers equation.* Here, u is the velocity, ν is the coefficient of diffusivity and ζ is any property which can be transported and diffused. If the viscous term (diffusive term) on the right hand side is neglected, the remaining equation may be viewed as a simple analog of Euler's equation

$$\frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} = 0 \tag{4.55}$$

Now we shall see the behaviour of Burgers equation for different kinds of discretization methods. In particular, we shall study their influence on conservative and transportive property, and artificial viscosity.

4.5.1 Conservative Property

A finite difference equation possesses conservative property if it preserves integral conservation relations of the continuum. Let us consider the vorticity transport equation

$$\frac{\partial \omega}{\partial t} = -(\nabla \cdot \mathbf{V})\,\omega + \nu\,\nabla^2\,\omega \tag{4.56}$$

where ∇ is nabla or differential operator, $\mathbf{V}$ the fluid velocity and ω the vorticity. If we integrate this over some fixed space region $\Re$, we get

$$\int_{\Re} \frac{\partial \omega}{\partial t}\, d\Re = -\int_{\Re} (\nabla \cdot \mathbf{V}\omega)\, d\Re + \int_{\Re} \nu\,\nabla^2\,\omega\, d\Re \tag{4.57}$$

The first term of Equation 4.57 can be written as

$$\int_{\Re} \frac{\partial \omega}{\partial t}\, d\Re = \frac{\partial}{\partial t} \int_{\Re} \omega\, d\Re$$

The second term of Equation 4.57 may be expressed as

$$-\int_{\Re} \nabla \cdot \mathbf{V}\omega\, d\Re = -\int_{Ao} (\mathbf{V}\omega) \cdot n dA$$

Here Ao is the boundary of $\Re$, n is unit normal vector and dA is the differential element of Ao. The remaining term of Equation 4.57 may be written as

$$\nu \int_{\Re} \nabla^2\,\omega\, d\Re = \nu \int_{Ao} (\nabla\,\omega) \cdot n dA$$

As such,

$$\nu \int_{Ao} (\nabla\,\omega) \cdot n dA = \nu \int_{\Re} \nabla \cdot (\nabla\omega)\, d\Re = \nu \int_{\Re} \nabla^2\,\omega\, d\Re$$

Combining these equations, we can write

$$\frac{\partial}{\partial t} \int_{\Re} \omega\, d\Re = -\int_{Ao} (\mathbf{V}\omega) \cdot n dA + \nu \int_{Ao} (\nabla\,\omega) \cdot n dA \tag{4.58}$$

which implies that the time rate of accumulation of ω in $\Re$ is equal to net advective flux rate of ω across Ao into $\Re$ plus net diffusive flux rate of ω across Ao into $\Re$.

The concept of conservative property is to maintain this integral relation in finite difference representation.

For clarity, again let us consider inviscid Burgers Equation 4.55. This time we let $\zeta = \omega =$vorticity, which means

$$\frac{\partial \omega}{\partial t} = -\frac{\partial (u\,\omega)}{\partial x} \tag{4.59}$$

The finite difference analog is given by FTCS method as

$$\frac{\omega_i^{n+1} - \omega_i^n}{\Delta t} = -\frac{u_{i+1}^n \, \omega_{i+1}^n - u_{i-1}^n \, \omega_{i-1}^n}{2\Delta x} \tag{4.60}$$

Let us consider a region $\Re$ running from $i = I_1$ to $i = I_2$ (Figure 4.4). We evaluate the integral $\frac{1}{\Delta t} \sum_{i=I_1}^{I_2} \omega_i \, \Delta x$ as

$$\begin{aligned} \frac{1}{\Delta t} \left[\sum_{i=I_1}^{i=I_2} \omega_i^{n+1} \, \Delta x - \sum_{i=I_1}^{I_2} \omega_i^n \, \Delta x \right] &= \sum_{i=I_1}^{I_2} -\frac{(u_{i+1}^n \, \omega_{i+1}^n) - (u_{i-1}^n \, \omega_{i-1}^n)}{2 \, \Delta x} \, \Delta x \\ &= \frac{1}{2} \sum_{i=I_1}^{i=I_2} \left[(u \, \omega)_{i-1}^n - (u \, \omega)_{i+1}^n \right] \end{aligned} \tag{4.61}$$

Summation of the right hand sides finally gives

$$\begin{aligned} & \frac{1}{\Delta t} \left[\sum_{i=I_1}^{I_2} \omega_i^{n+1} \, \Delta x - \sum_{i=I_1}^{I_2} \omega_i^n \, \Delta x \right] \\ = \; & \frac{1}{2} \left[(u \, \omega)_{I_1-1}^n + (u \, \omega)_{I_1}^n \right] - \frac{1}{2} \left[(u \, \omega)_{I_2}^n + (u \, \omega)_{I_2+1}^n \right] \\ = \; & (u \, \omega)_{I_1 - \frac{1}{2}} - (u \, \omega)_{I_2 + \frac{1}{2}} \end{aligned} \tag{4.62}$$

Equation 4.62 states that the rate of accumulation of ω_i in $\Re$ is identically equal to the net advective flux rate across the boundary of $\Re$ running from $i = I_1$ to $i = I_2$. This is analogous to inviscid part of the integral Equation 4.58. Thus the finite difference analog Equation 4.60 has preserved the conservative property. As such, conservative property depends on the form of the continuum equation used. Let us take non-conservative form of inviscid Burgers Equation 4.55 as

$$\frac{\partial \omega}{\partial t} = -u \frac{\partial \omega}{\partial x} \tag{4.63}$$

Using the same FTCS differencing technique as before, we can write

$$\frac{\omega_i^{n+1} - \omega_i^n}{\Delta t} = -u_i^n \left[\frac{\omega_{i+1}^n - \omega_{i-1}^n}{2 \, \Delta x} \right] \tag{4.64}$$

Now, the integration over $\Re$ running from $i = I_1$ to $i = I_2$, yields

$$\begin{aligned} \frac{1}{\Delta t} \left[\sum_{i=I_1}^{I_2} \omega_i^{n+1} \Delta x - \sum_{i=I_1}^{I_2} \omega_i^n \Delta x \right] &= \sum_{i=I_1}^{I_2} -u_i^n \frac{(\omega_{i+1}^n - \omega_{i-1}^n)}{2 \, \Delta x} \\ &= \frac{1}{2} \sum_{i=I_1}^{I_2} \left[u_i^n \omega_{i-1}^n - u_i^n \omega_{i+1}^n \right] \end{aligned} \tag{4.65}$$

While performing the summation of the right hand side of Equation 4.65, it can

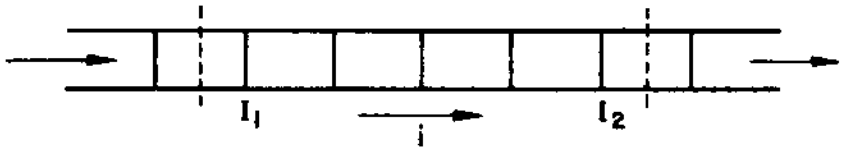


Figure 4.4: Domain running from $i = I_1$ to $i = I_2$

be observed that the terms corresponding to inner cell fluxes do not cancel out. Consequently an expression in terms of fluxes at the inlet and outlet section, as it was found earlier, could not be obtained. Hence, the finite difference analog Equation 4.64 has failed to preserve the integral Gauss-divergence property, i.e. the conservative property of the continuum.

The quality of preserving the conservative property is of special importance with regards to the methods involving finite-volume approach (a special form of finite difference equations). The use of conservative form depicts that the advective flux rate of ω out of a control volume at the interface $i = I_2 + 1/2$ is exactly equal to flux rate of ω into the next control volume and so on. The meaning of calling Equation 4.59 as *conservative form* is now clearly understood. However, the conservative form of advective part is of prime importance for modelling fluid flow and is often referred to as weak conservative form. For the incompressible flow, in cartesian coordinate system this form is:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\,\nabla^2 u \\
\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial vw}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\,\nabla^2 v \\
\frac{\partial w}{\partial t} + \frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial w^2}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\,\nabla^2 w
\end{aligned} \tag{4.66}$$

If all the terms in the flow equations are recast in the form of first order derivatives of x, y, z and t, the equations are said to be in *strong conservation form*. We shall write the strong conservation form of Navier-Stokes equation in cartesian coordinate system:

$$\begin{aligned}
\frac{\partial u}{\partial t} &+ \frac{\partial}{\partial x}\left(u^2 + \frac{p}{\rho} - \nu\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(uv - \nu\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(uw - \nu\frac{\partial u}{\partial z}\right) = 0 \\
\frac{\partial v}{\partial t} &+ \frac{\partial}{\partial x}\left(uv - \nu\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(v^2 + \frac{p}{\rho} - \nu\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial z}\left(vw - \nu\frac{\partial v}{\partial z}\right) = 0 \\
\frac{\partial w}{\partial t} &+ \frac{\partial}{\partial x}\left(uw - \nu\frac{\partial w}{\partial x}\right) + \frac{\partial}{\partial y}\left(vw - \nu\frac{\partial w}{\partial y}\right) \\
&+ \frac{\partial}{\partial z}\left(w^2 + \frac{p}{\rho} - \nu\frac{\partial w}{\partial z}\right) = 0
\end{aligned} \tag{4.67}$$

4.5.2 The Upwind Scheme

Once again, we shall start with the inviscid Burgers Equation 4.55. Regarding discretization, we can think about the following formulations

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} + u\,\frac{\zeta_{i+1}^n - \zeta_i^n}{\Delta x} = 0 \tag{4.68}$$

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} + u\,\frac{\zeta_{i+1}^n - \zeta_{i-1}^n}{2\,\Delta x} = 0 \tag{4.69}$$

If von Neumann stability analysis is applied to these schemes, we find that both methods are unconditionally unstable.

A well known remedy for the difficulties encountered in such formulations is the upwind scheme which is described by Gentry, Martin and Daly (1966) and Runchal and Wolfshtein (1969). Equation 4.68 can be made stable by substituting the forward space difference by a backward space difference, provided that the carrier velocity u is positive. If u is negative, a forward difference must be used to assure stability. For full Burgers Equation 4.54, the formulation of the diffusion term remains unchanged and only the convective term (in conservative form) is calculated in the following way (Figure 4.5):

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} = -\frac{u\,\zeta_i^n - u\,\zeta_{i-1}^n}{\Delta x} + \quad \text{viscous term}, \quad \text{for} \quad u > 0 \tag{4.70}$$

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} = -\frac{u\,\zeta_{i+1}^n - u\,\zeta_i^n}{\Delta x} + \quad \text{viscous term}, \quad \text{for} \quad u < 0 \tag{4.71}$$

It is also well known that upwind method of discretization is very much necessary in convection (advection) dominated flows in order to obtain numerically stable results. The upwind bias retains the transportive property of the flow equation. Let us take a closer look at the transportive property and related upwind bias.

4.5.3 Transportive Property

A finite difference formulation of a flow equation possesses the transportive property if the effect of a perturbation is convected (advected) only in the direction of the velocity. Consider the model Burgers equation in conservative form

$$\frac{\partial \zeta}{\partial t} = -\frac{\partial u\,\zeta}{\partial x}$$

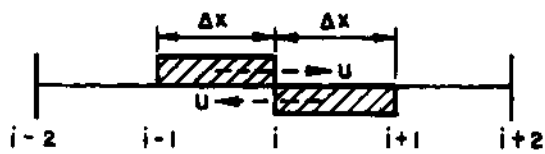


Figure 4.5: The Upwind Scheme

Let us examine a method which is central in space. Using FTCS we get

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} = -\frac{u\,\zeta_{i+1}^n - u\,\zeta_{i-1}^n}{2\,\Delta x} \tag{4.72}$$

Consider a perturbation $\epsilon_m = \delta$ in ζ. A physical perturbation in ζ will spread in all directions due to diffusion. We are taking an inviscid model equation and we want the perturbation to be carried along only in the direction of the velocity. So, for $u > 0$, $\epsilon_m = \delta$ (perturbation at mth space location), all other $\epsilon = 0$. Therefore, at a point $(m+1)$ downstream of the perturbation

$$\frac{\zeta_{m+1}^{n+1} - \zeta_{m+1}^n}{\Delta t} = -\frac{0 - u\,\delta}{2\,\Delta x} = +\frac{u\,\delta}{2\,\delta x}$$

which is acceptable. However, at the point of perturbation $(i = m)$,

$$\frac{\zeta_m^{n+1} - \zeta_m^n}{\Delta t} = -\frac{(0-0)}{2\,\Delta x} = 0$$

which is not very reasonable. But at the upstream station $(i = m-1)$ we observe

$$\frac{\zeta_{m-1}^{n+1} - \zeta_{m-1}^n}{\Delta t} = -\frac{(u\,\delta - 0)}{2\,\Delta x} = -\frac{u\,\delta}{2\,\delta x}$$

which indicates that the transportive property is violated.

On the contrary, let us see what happens when an upwind scheme is used. We know that for $u > 0$

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} = -\frac{u\,\zeta_i^n - u\,\zeta_{i-1}^n}{\Delta x} \tag{4.73}$$

Then for $\epsilon_m = \delta$ at the downstream location $(m+1)$

$$\frac{\zeta_{m+1}^{n+1} - \zeta_{m+1}^n}{\Delta t} = -\frac{0 - u\,\delta}{\Delta x} = \frac{u\,\delta}{\delta x}$$

which follows the rationale for the transportive property.

At point m of the disturbance

$$\frac{\zeta_m^{n+1} - \zeta_m^n}{\Delta t} = -\frac{u\,\delta - 0}{\Delta x} = -\frac{u\,\delta}{\delta x}$$

which means that the perturbation is being transported out of the affected region. Finally, at $(m-1)$th station, we observe that

$$\frac{\zeta_{m-1}^{n+1} - \zeta_{m-1}^n}{\Delta t} = -\frac{(0-0)}{\Delta x} = 0$$

This signifies that no perturbation effect is carried upstream. In other words, the upwind method maintains unidirectional flow of information. In conclusion, it can be said that while space centered differences are more accurate than upwind differences, as indicated by the Taylor series expansion, the whole system is not more accurate if the criteria for accuracy includes the transportive property as well.

4.5.4 Upwind Differencing and Artificial Viscosity

Consider the model Burgers Equation 4.54 and focus attention on the inertia terms

$$\frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} = \nu \frac{\partial^2 \zeta}{\partial x^2}$$

As seen, the simple upwind scheme gives

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} = -\frac{u\,\zeta_i^n - u\,\zeta_{i-1}^n}{\Delta x} + \ldots \quad \text{for} \quad u > 0$$

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} = -\frac{u\,\zeta_{i+1}^n - u\,\zeta_i^n}{\Delta x} + \ldots \quad \text{for} \quad u < 0$$

From Taylor series expansion, we can write

$$\zeta_i^{n+1} = \zeta_i^n + \Delta t \left.\frac{\partial \zeta}{\partial t}\right|_i^n + \frac{(\Delta t)^2}{2} \left.\frac{\partial^2 \zeta}{\partial t^2}\right|_i^n + \ldots \tag{4.74}$$

$$\zeta_{i\pm1}^n = \zeta_i^n \pm \Delta x \left.\frac{\partial \zeta}{\partial x}\right|_i^n + \frac{(\Delta x)^2}{2} \left.\frac{\partial^2 \zeta}{\partial x^2}\right|_i^n \pm \ldots \tag{4.75}$$

Substituting Equations 4.74 and 4.75 into 4.70 gives (dropping the superscript n and subscript i)

$$\begin{aligned} & \frac{1}{\Delta t}\left[\Delta t \frac{\partial \zeta}{\partial t} + \frac{1}{2}(\Delta t)^2 \frac{\partial^2 \zeta}{\partial t^2} + O(\Delta t)^3\right] \\ = & -\frac{u}{\Delta x}\left[\Delta x \frac{\partial \zeta}{\partial x} - \frac{(\Delta x)^2}{2}\frac{\partial^2 \zeta}{\partial x^2} + O(\Delta x)^3\right] \\ & +[\text{Diffusive term}] \end{aligned}$$

or

$$\frac{\partial \zeta}{\partial t} = -u \frac{\partial \zeta}{\partial x} + \frac{1}{2}\left[u\,\Delta x\left(1 - \frac{u\,\Delta t}{\Delta x}\right)\right]\frac{\partial^2 \zeta}{\partial x^2} + \nu \frac{\partial^2 \zeta}{\partial x^2} + O(\Delta x)^2$$

which may be rewritten as

$$\frac{\partial \zeta}{\partial t} = -u \frac{\partial \zeta}{\partial x} + \nu \frac{\partial^2 \zeta}{\partial x^2} + \nu_e \frac{\partial^2 \zeta}{\partial x^2} + \quad \text{higher order terms} \tag{4.76}$$

where

$$\nu_e = \frac{1}{2}\left[u\,\Delta x\,(1 - C)\right], \quad C \text{ (Courant number)} = \frac{u\,\Delta t}{\Delta x}$$

In deriving Equation 4.76, $\partial^2 \zeta / \partial t^2$ was taken as $u^2\,\partial^2 \zeta / \partial x^2$. However, the non-physical coefficient ν_e leads to a diffusion like term which is dependent on the discretization procedure. The coefficient ν_e is known as numerical or *artificial viscosity*. Let us look at the expression

$$\nu_e = \frac{1}{2}\left[u\,\Delta x\,(1 - C)\right] \quad \text{for} \quad u > 0 \tag{4.77}$$

somewhat more critically. On one hand we have considered that $u > 0$ and on the other CFL condition demands that $C < 1$ (so that the algorithm can work). As a consequence, ν_e is always a positive non-zero quantity. If, instead of analysing the transient equation, we put $\partial\zeta/\partial t = 0$ in Equation 4.70 and expand it in Taylor series, we obtain

$$\nu_e = \frac{1}{2}\, u\, \Delta x \tag{4.78}$$

Let us now consider a two dimensional convection-diffusion equation with viscous diffusion in both direction (Equation 4.42 but with $\omega = \zeta$). For u_i, $v_i > 0$, upwind differencing gives

$$\begin{aligned}\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^{n}}{\Delta t} = & - \frac{u\,\zeta_{i,j}^{n} - u\,\zeta_{i-1,j}^{n}}{\Delta x} - \frac{v\,\zeta_{i,j}^{n} - v\,\zeta_{i,j-1}^{n}}{\Delta y} \\ & + \nu \left[\frac{\zeta_{i+1,j}^{n} - 2\zeta_{i,j}^{n} + \zeta_{i-1,j}^{n}}{(\Delta x^2)} \right. \\ & + \left. \frac{\zeta_{i,j+1}^{n} - 2\,\zeta_{i,j}^{n} + \zeta_{i,j-1}^{n}}{(\Delta y)^2} \right]\end{aligned} \tag{4.79}$$

The Taylor series procedure, as performed on Equation 4.76 will lead to

$$\frac{\partial\zeta}{\partial t} = -u\,\frac{\partial\zeta}{\partial x} - v\,\frac{\partial\zeta}{\partial y} + (\nu + \nu_{ex})\,\frac{\partial^2\zeta}{\partial x^2} + (\nu + \nu_{ey})\,\frac{\partial^2\zeta}{\partial y^2} \tag{4.80}$$

where

$$\nu_{ex} = \frac{1}{2}\, u\, \Delta x\, (1 - C_x),\ \nu_{ey} = \frac{1}{2}\, v\, \Delta y\, (1 - C_y); \quad \text{with} \quad C_x = \frac{u\Delta t}{\Delta x},\ C_y = \frac{v\Delta t}{\Delta y}$$

For the special case of $u \approx v$ and $\Delta x = \Delta y$, CFL condition is $C_x = C_y \leq (1/2)$. This indicates that for a stable calculation, artificial viscosity will necessarily be present. However, for a steady state analysis, we get

$$\nu_{ex} = \frac{1}{2}\, u\, \Delta x; \quad \nu_{ey} = \frac{1}{2}\, v\, \Delta y \tag{4.81}$$

We have observed that some amount of upwind effect is indeed necessary to maintain transportive property of flow equations while the computations based on upwind differencing often suffer from false diffusion (inaccuracy!). One of the plausible improvements is the usage of higher order upwind method of differencing. Next subsection discusses this aspect of improving accuracy.

4.5.5 Second Upwind Differencing or Hybrid Scheme

According to the second upwind differencing, if u is the velocity in x direction and ζ is any property which can be convected and diffused, then

$$\frac{\partial\,(u\zeta)}{\partial x}\,|_{i,j} = \frac{u_R\,\zeta_R - u_L\,\zeta_L}{\Delta x} \tag{4.82}$$

One point to be carefully observed from Equation 4.82 is that the second upwind should be written in conservative form. However, the definition of u_R and u_L are (Figure 4.6):

$$u_R = \frac{u_{i,j} + u_{i+1,j}}{2}; \quad u_L = \frac{u_{i-1,j} + u_{i,j}}{2} \tag{4.83}$$

Now,

$$\zeta_R = \zeta_{i,j} \quad \text{for} \quad u_R > 0; \zeta_R = \zeta_{i+1,j} \quad \text{for} \quad u_R < 0 \tag{4.84}$$

and

$$\zeta_L = \zeta_{i-1,j} \quad \text{for} \quad u_L > 0; \zeta_L = \zeta_{i,j} \quad \text{for} \quad u_L < 0 \tag{4.85}$$

Hence, we get

$$\begin{aligned} \frac{\partial (u\,\zeta)}{\partial x} &= \frac{1}{\Delta x}\left[\left(\frac{u_{i,j} + u_{i+1,j}}{2}\right)\zeta_{i,j} - \left(\frac{u_{i-1,j} + u_{i,j}}{2}\right)\zeta_{i-1,j}\right] \\ &\quad \text{for} \quad u_R > 0,\, u_L > 0 \end{aligned} \tag{4.86}$$

Let us discretize the second term of the convection part of unsteady x-direction momentum equation. We have chosen this in order to cite a meaningful example of second upwind differencing. Using Equation 4.86 we can write

$$\begin{aligned} \frac{\partial u^2}{\partial x}\Big|_{i,j} &= \frac{1}{\Delta x}\left[\left(\frac{u_{i,j} + u_{i+1,j}}{2}\right)u_{i,j} - \left(\frac{u_{i-1,j} + u_{i,j}}{2}\right)u_{i-1,j}\right] \\ &= \frac{1}{\Delta x}\left[\left(\frac{u_{i,j} + u_{i+1,j}}{2}\right)\left(\frac{u_{i,j} + u_{i+1,j} + u_{i,j} - u_{i+1,j}}{2}\right)\right. \\ &\quad \left. - \left(\frac{u_{i-1,j} + u_{i,j}}{2}\right)\left(\frac{u_{i-1,j} + u_{i,j} + u_{i-1,j} - u_{i,j}}{2}\right)\right] \\ &= \frac{1}{4\Delta x}\left[(u_{i,j} + u_{i+1,j})\left[(u_{i,j} + u_{i+1,j}) + (u_{i,j} - u_{i+1,j})\right]\right. \\ &\quad \left. - (u_{i-1,j} + u_{i,j})\left[(u_{i-1,j} + u_{i,j}) + (u_{i-1,j} - u_{i,j})\right]\right] \\ &= \frac{1}{4\Delta x}\left[(u_{i,j} + u_{i+1,j})^2 + (u_{i,j}^2 - u_{i+1,j}^2)\right. \\ &\quad \left. - (u_{i-1,j} + u_{i,j})^2 - (u_{i-1,j}^2 - u_{i,j}^2)\right] \end{aligned} \tag{4.87}$$

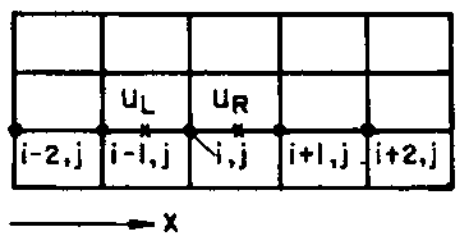


Figure 4.6: Definition of u_L and u_R

Here we introduce a factor η which can express Equation 4.87 as a weighted average of central and upwind differencing. Invoking the weighted average concept in Equation 4.87, we obtain

$$\begin{aligned}\frac{\partial u^2}{\partial x} &= \frac{1}{4\Delta x}\Big[(u_{i,j}+u_{i+1,j})^2+\eta\mid(u_{i,j}+u_{i+1,j})\mid(u_{i,j}-u_{i+1,j})\\ &\quad -(u_{i-1,j}+u_{i,j})^2-\eta\mid(u_{i-1,j}+u_{i,j})\mid(u_{i-1,j}-u_{i,j})\Big] \qquad (4.88)\end{aligned}$$

where $0 < \eta < 1$. For $\eta = 0$, Equation 4.88 becomes centered in space and for $\eta = 1$ it becomes full upwind. So η brings about the upwind bias in the difference quotient. If η is small, Equation 4.88 tends towards being centered in space. This upwind method was first introduced by Gentry, Martin and Daly (1966). Some more stimulating discussions on the need of upwinding and its minimization has been presented by Roache (1972) who has also pointed out that the second-upwind-formulation possesses both the conservative and transportive properties and produces reasonable accuracy provided the upwind factor (formally called donar cell factor) is not large. In principle, the weighted average differencing scheme can as well be called as *hybrid* scheme (see Raithby and Torrance, 1974) and can always be increased by a suitable adjustment of the value of the upwind factor η.

4.5.6 More Suggestions for Improvement

Several researchers have tried to resolve the difficulty associated with the discretization of the first order terms which needs some amount of artificial viscosity for stability. Substantial progress has been made on the development of higher order schemes which are suitable over a large range of velocities. However, none of these prescriptions are universal. Depending on the nature of flow and geometry one can always go for the best suited algorithm. Now we shall discuss one such algorithm which has been proposed by Khosla and Rubin (1974).

Consider the Burgers Equation 4.54 once again. The derivatives in this equation are discretized in the following way.
For $u > 0$,

$$\begin{aligned}\frac{\partial\zeta}{\partial t} &= \frac{\zeta_i^{n+1}-\zeta_i^n}{\Delta t} \quad \text{(Forward time)}\\ \frac{\partial\zeta}{\partial x} &= \frac{\zeta_i^{n+1}-\zeta_{i-1}^{n+1}}{\Delta x}+\frac{\zeta_{i+1}^n-2\zeta_i^n+\zeta_{i-1}^n}{2\Delta x}\end{aligned}$$

This is modified central difference in space, which for a converged solution ($\zeta_i^{n+1} = \zeta_i^n$) reduces to space centered scheme

$$\frac{\partial^2\zeta}{\partial x^2} = \frac{\zeta_{i+1}^{n+1}-2\zeta_i^{n+1}+\zeta_{i-1}^{n+1}}{(\Delta x)^2}$$

This is central difference in space. Substituting the above quotients in Equation 4.54, one finds

$$-A_i\,\zeta_{i+1}^{n+1} + B_i\,\zeta_i^{n+1} - C_i\,\zeta_{i-1}^{n+1} = D_i \tag{4.89}$$

where

$$A_i = \frac{\nu\,\Delta t}{(\Delta x)^2},\; C_i = \frac{u\,\Delta t}{\Delta x} + \frac{\nu(\Delta t)}{(\Delta x)^2},\; B_i = 1 + \frac{u\,\Delta t}{\Delta x} + 2\,\frac{\nu(\Delta t)}{(\Delta x)^2}$$

and

$$D_i = \left(1 + \frac{u\Delta t}{\Delta x}\right)\zeta_i^n - \frac{u\,\Delta t}{2\Delta x}\left(\zeta_{i+1}^n + \zeta_{i-1}^n\right) \tag{4.90}$$

For $u_i > 0$, A_i, B_i and $C_i > 0$ and $B_i > A_i + C_i$. The system of equations produced from Equation 4.89 is always diagonally dominant and capable of providing a stable solution. As the solution progresses (i.e., $u_i^n \to u_i^{n+1}$), the convective term approaches second order accuracy. This idea of implementing higher order upwind is known as *deferred correction procedure.*

Another widely suggested improvement is known as third order upwind differencing (Kawamura *et al.*, 1986). The following example illustrates the essence of this discretization scheme.

$$\begin{aligned}\left(u\,\frac{\partial u}{\partial x}\right)_{i,j} &= u_{i,j}\left[\frac{-u_{i+2,j} + 8\,(u_{i+1,j} - u_{i-1,j}) + u_{i-2,j}}{12(\Delta x)}\right] \\ &+ \;|\,u_{i,j}\,|\;\left[\frac{u_{i+2,j} - 4u_{i+1,j} + 6u_{i,j} - 4u_{i-1,j} + u_{i-2,j}}{4(\Delta x)}\right]\end{aligned} \tag{4.91}$$

Higher order upwinding is an emerging area of research in Computational Fluid Dynamics. However, no unique suggestion has been evolved as an optimal method applicable to wide variety of problems. Interested readers are referred to Vanka (1987), Fletcher (1988) and Rai and Moin (1991) for more stimulating information on related topics.

4.6 Some Non-Trivial Problems with Discretized Equations

The discussion in this section is based upon some ideas indicated by Hirt (1968) which are applied to model Burgers equation. The FTCS differencing scheme will produce the finite difference equation as

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} + u\,\frac{\zeta_{i+1}^n - \zeta_{i-1}^n}{2\,\Delta x} = \nu\left[\frac{\zeta_{i+1}^n - 2\zeta_i^n + \zeta_{i-1}^n}{(\Delta x)^2}\right] \tag{4.92}$$

From this form, the modified partial differential equation becomes

$$\frac{\partial \zeta}{\partial t} + u\,\zeta_x = \left(\nu - \frac{u^2\,\Delta t}{2}\right)\zeta_{xx} \tag{4.93}$$

We define

$$r = \frac{\nu\,(\Delta t)}{(\Delta x)^2}; \quad C = \frac{u\,\Delta t}{\Delta x} = \text{Courant number}$$

It is interesting to note that the values $r = 1/2$ and $C = 1$ (which are extremum conditions of von Neumann stability analysis) unfortunately eliminate the viscous diffusion completely in Equation 4.93 and produce a solution from Equation 4.92 directly as $\zeta_i^{n+1} = \zeta_i^n$. This is unacceptable. From Equation 4.93, it is clear that in order to obtain a solution for convection diffusion equation, we should have

$$\nu - \frac{u^2\,\Delta t}{2} > 0$$

For a meaningful physical result in the case of inviscid flow we require

$$\nu - \frac{u^2\,\Delta t}{2} = 0$$

Combining these two criteria, for a meaningful solution

$$\nu - \frac{u^2\,\Delta t}{2} \geq 0$$

or

$$\nu \left[1 - \frac{1}{2}\,\frac{u\,\Delta t}{\Delta x} \cdot \frac{u\,\Delta x}{\nu}\right] \geq 0$$

Here we define the mesh-Reynolds-number or cell-Peclet-number as

$$Re_{\Delta x} = \frac{u\,\Delta x}{\nu} = Pe_{\Delta x}$$

in terms of which

$$\nu \left[1 - \frac{1}{2} C\, Re_{\Delta x}\right] \geq 0$$

or

$$Re_{\Delta x} \leq \frac{2}{C} \tag{4.94}$$

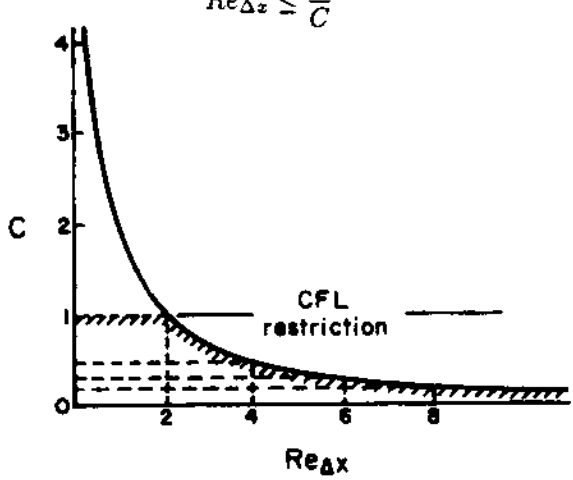


Figure 4.7: Limiting Line ($Re_{\Delta x} \leq \frac{2}{C}$)

The plot of C vs $Re_{\Delta x}$ is shown in Figure 4.7 in order to describe the significance of Equation 4.94. From the CFL condition, we know that the stability requirement is $C \leq 1$. Under such a restriction, below $Re_{\Delta x} = 2$, the calculation is always stable. The interesting information is that it is possible to cross the cell-Reynolds-number of 2 if C is made less than unity.

References

Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, New York, USA, 1984.

Burgers, J.M., A Mathematical Model Illustrating the Theory of Turbulence, Adv. Appl. Mech., Vol 1, pp 171-199, 1948.

DuFort, E.C. and Frankel, S.P., Stability Conditions in the Numerical Treatment of Parabolic Differential Equations, Mathematical Tables and Other Aids to Computation, Vol 7, pp 135-152, 1953.

Fletcher, C.A.J., *Computational Techniques for Fluid Dynamics*, Vol 1 (Fundamentals and General Techniques), Springer Verlag, 1988.

Gentry, R.A., Martin, R.E., and Daly, B.J., An Eulerian Differencing Method for Unsteady Compressible Flow Problems, J. Comput. Phys., Vol 1, pp 87-118, 1966.

Hirt, C.W., Heuristic Stability Theory of Finite Difference Equations, J. Comp. Phys., Vol 2, pp 339-355, 1968.

Kawamura, T., Takami, H., and Kuwahara, K., Computation of High Reynolds Number Flow around a Circular Cylinder with Surface Roughness, Fluid Dynamics Research, Vol 1, pp 145-162, 1986.

Khosla, P.K. and Rubin, S.G., A Diagonally Dominant Second Order Accurate Implicit Scheme, Computers and Fluids, Vol 2, pp 207-209, 1974.

Lax, P.D. and Wendroff, B., Systems of Conservation Laws, Comm. Pure Appl. Math., Vol 13, pp 217-237, 1960.

Rai, M.M. and Moin, P., Direct Simulations of Turbulent Flow Using Finite Difference Schemes, J. Comput. Phys., Vol 96, pp 15-53, 1991.

Raithby, G.D. and Torrance, K.E., Upstream - weighted Differencing Schemes and Their Applications to Elliptic Problems Involving Fluid Flow, Computers and Fluids, Vol 2, pp 191-206, 1974.

Roache, P.J. *Computational Fluid Dynamics*, Hermosa, Albuquerque, New Mexico, 1972 (revised printing 1985).

Runchal, A.K. and Wolfshtein, M., Numerical Integration Procedure for the Steady State Navier-Stokes Equations, J. Mech. Eng. Sci., Vol 11, pp 445-453, 1969.

Thomas, L.H., Elliptic Problems in Linear Difference Equations over a Network, Watson Sci. Compt. Lab. Rept., Columbia University, New York, 1949.

Vanka, S.P., Second Order Differencing in Recirculating Flow, AIAA Journal, Vol 22, pp 1435-1441, 1987.

Problems

1. Consider the nonlinear equation

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

where μ is a constant, u the x component of velocity, and y is the direction normal to x.

   (a) Is this equation in conservative form? If not, suggest a conservative form of the equation.

   (b) Develop a finite difference equivalent for this equation which will march in the x direction.

2. Establish the truncation error of the following finite difference approximation to $\partial T/\partial y$ at the point (i, j) for a uniform mesh

$$\frac{\partial T}{\partial y} = \frac{-3T_{i,j} + 4T_{i,j+1} - T_{i,j+2}}{2(\Delta y)}.$$

What is the order of the truncation error? If you want to apply a second-order-accurate boundary condition for $\partial T/\partial y = 0$ at the boundary (refer to Figure 4.8), can you make use of the above mentioned expression? If yes, what should be the expression for $T_{i,j}$ at the boundary?

3. The Lax-Wendroff finite difference scheme (Lax and Wendroff, 1960) can be derived from a Taylor series expansion in the following manner:

$$u_i^{n+1} = u_i^n + \Delta t u_t + 1/2\,(\Delta t)^2 u_{tt} + O[(\Delta t)^3]$$

Using the wave equations

$$u_t = -cu_x$$
$$u_{tt} = c^2 u_{xx}$$

the Taylor series expansion may be written as

$$u_i^{n+1} = u_i^n - c(\Delta t)u_x + \frac{1}{2}c^2(\Delta t)^2 u_{xx} + O[(\Delta t)^3]$$

If u_x and u_{xx} are replaced by central difference quotients, we get

$$u_i^{n+1} = u_i^n - \frac{c(\Delta t)}{2(\Delta x)}\,(u_{i+1}^n - u_{i-1}^n) + \frac{c^2(\Delta t)^2}{2(\Delta x)^2}\,(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

Prove that the CFL condition is the stability requirement for the above discretization scheme.

4. A three level explicit discretization of

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

can be written as

$$\begin{aligned}\frac{0.5T_i^{n-1} - 2T_i^n + 1.5T_i^{n+1}}{\Delta t} &= \alpha \left[\frac{(1+d)\left(T_{i-1} - 2T_i + T_{i+1}\right)^n}{(\Delta x)^2} \right. \\ &\quad \left. - \frac{d\left(T_{i-1} - 2T_i + T_{i+1}\right)^{n-1}}{(\Delta x)^2} \right]\end{aligned}$$

Expand each term as a Taylor series to determine the truncation error of the complete equation for arbitrary values of d. Suggest the general technique where for a functional relationship between d and $r\left[= \alpha\Delta t/(\Delta x)^2\right]$ the scheme will be fourth-order accurate in Δx.

5. Consider the equation

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right]$$

where T is the dependent variable which is convected and diffused. The independent variables, x and y, are in space while t is the time (evolution) coordinate. The coefficient u, v and α can be treated as constant. Employing forward difference for the first-order derivatives and central-second difference for the second-order derivatives, obtain the finite difference equation. What is the physical significance of the difference between the above equation and the equation actually being solved? Suggest a method to overcome this difference.

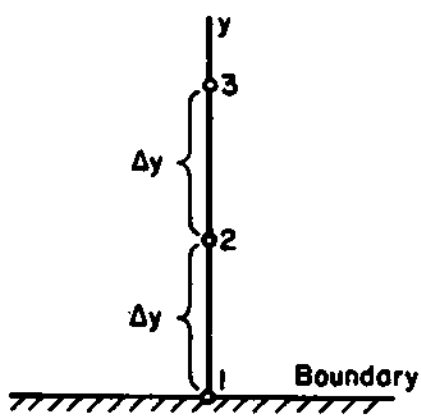


Figure 4.8: Grid Points at a Boundary

5
Finite Difference Applications in Heat Conduction and Convection

Finite difference method is a simple yet powerful technique for solving a variety of heat transfer and flow problems in regular geometries. In this chapter, the modelling of heat conduction and convection problems by the finite difference technique is discussed. Several illustrative examples are presented for steady and transient heat transfer in rectangular, cylindrical and spherical geometries, along with brief discussions on the accuracy and stability of the respective numerical schemes.

The central theme in finite difference simulation is the conversion of a physical problem involving the continuous variation of a field variable $\phi(x, y, z, t)$, into an approximate numerical formulation containing discrete values of ϕ at spatial points (x_i, y_i, z_i) and time levels t^n. The spatial points are called *nodes* of the finite difference grid. The governing differential equations of the physical problem are approximated at the nodes, to generate algebraic equations for the nodal values of ϕ.

Application of the finite difference method to any physical problem consists of the following three steps:

1. Division of spatial domain into an orthogonal computational grid;
2. Discretization of the governing equations and boundary conditions in space and time to derive the approximately equivalent algebraic equations for each node; and
3. Solution of the discretized (algebraic) equations by a suitable matrix inversion or iterative technique.

The discretization procedure involves the replacement of *derivatives* in the governing differential equations and boundary conditions by the corresponding *difference* expressions. Conventionally, an orthogonal grid is employed for the

purpose of discretization, since the derivatives with respect to a particular coordinate are approximated by uni-directional difference expressions in that direction. Application of the finite difference method to irregular geometries is somewhat complicated, involving either domain transformation or cumbersome interpolation at nodes adjacent to the boundary. In recent years, however, the extension of the finite difference method to non-orthogonal grids has also been carried out by some researchers, with the help of generalized interpolation procedures. Here, we shall confine ourselves to the applications in regular geometries.

5.1 Heat Conduction

The transient energy balance equation for heat conduction through an isotropic solid is given by (Holman, 1997)

$$\rho \frac{\partial(CT)}{\partial t} = \nabla \cdot (k\nabla T) + Q \tag{5.1}$$

where k is the thermal conductivity, C is the specific heat, ρ is the density, for the solid material and Q is the rate of heat generation per unit volume. For obtaining the temperature field $T(x, y, z, t)$ within the solid, Equation 5.1 has to be solved subject to prescribed heat transfer conditions at the boundaries of the domain and a given initial temperature field at time $t = 0$.

For numerical simulation by finite differences, the energy balance Equation 5.1 can be directly taken up as the starting point for discretization, or it may be integrated over chosen control volumes around each node of the grid before the derivatives are approximated. The latter method (known as the *control volume* approach) is quite useful when material properties vary in space. Apart from the energy balance equation, the boundary conditions are also to be suitably discretized. The spatial discretization of the governing equations is performed over the finite difference grid, while timewise discretization is carried out by marching in time, in steps of Δt.

Following sections describe the procedures for discretizing the governing equations and the matrix methods for solving the discretized equations. Various examples of modelling steady heat conduction in rectangular, cylindrical and spherical geometries and the methods for solving problems of transient heat conduction and convective heat transfer are discussed.

5.1.1 Steady Heat Conduction in a Rectangular Geometry

In rectangular geometry, an orthogonal finite difference grid is formed by an array of x = constant, y = constant and z = constant lines. The nodes are placed at the intersection of the x, y and z grid lines. Normally, the spacings $\Delta x, \Delta y$ and Δz between the respective grid lines are taken to be constant throughout

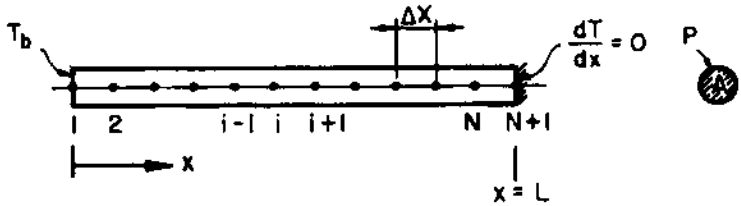


Figure 5.1: Heat Conduction in a Fin

the mesh. However, when there is a steep variation in the temperature gradient, non-uniform step sizes are used in order to economically obtain accurate solutions. Specifically, more grid points are placed at high gradient locations and less points are used for the low gradient regions. The coefficient matrix for the system of discretized equations has tridiagonal, pentadiagonal and septadiagonal structures for one, two and three dimensional problems respectively.

The following examples illustrate the application of the finite difference method to steady heat conduction in a Cartesian geometry.

Example 5.1. Heat dissipation through a straight fin

Consider steady heat loss through a long straight fin (Figure 5.1). The temperatures of the fin base and the surrounding fluid are T_b and T_f respectively. The one-dimensional heat conduction equation for the fin is given by (Ozisik, 1987):

$$\frac{d^2T}{dx^2} - \frac{hP}{kA}(T - T_f) = 0 \tag{5.2}$$

where h is the heat transfer coefficient between fluid and fin surface, and P and A are respectively the perimeter and the cross-sectional area of the fin geometry. Assuming that the heat loss from the end face ($x = L$) of the fin is negligible, the boundary conditions for the problem can be written as

$$T = T_b \text{ at } x = 0 \tag{5.3}$$

$$\frac{dT}{dx} = 0 \text{ at } x = L \tag{5.4}$$

For the sake of simplicity, let us consider a finite difference grid of N equal steps (i.e. $(N+1)$ grid points) laid along the length of the fin (Carnahan *et al.*, 1969; Ozisik, 1994). The step size Δx for this grid is equal to L/N. The discretized form of the energy balance equation can now be obtained by replacing the second derivative of temperature in Equation 5.2 by its equivalent difference approximation. Using the central difference form of the second derivative, the nodal equation for any node i in the grid is found as (Chapter 4)

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} - \frac{hP}{kA}(T_i - T_f) = 0 \tag{5.5}$$

The discretization error for the above equation is $O(\triangle x^2)$. The nodal equations for the interior nodes, which are generated by substituting $i = 2, 3, ..., N$ in Equation 5.5, are given by

$$\begin{aligned} T_1 - (2+A)T_2 + T_3 &= -AT_f, \quad i = 2; \\ T_2 - (2+A)T_3 + T_4 &= -AT_f, \quad i = 3; \\ &\cdots \\ T_{N-1} - (2+A)T_N + T_{N+1} &= -AT_f, \quad i = N; \end{aligned} \tag{5.6}$$

where $A = (hP/kA)\triangle x^2$. For the boundary nodes, nodal equations are derived from the boundary conditions. At the fin base (for $i = 1$)

$$T_1 = T_b \quad \text{(known base temperature)} \tag{5.7}$$

On the end surface, the zero heat flux condition is discretized by introducing a hypothetical node $(N + 2)$ beyond the boundary. We shall refer to this additional node as the *image node*. The central difference expression for the derivative dT/dx at node $(N + 1)$ is given by

$$\frac{dT}{dx} = \frac{T_{N+2} - T_N}{2\triangle x} = 0$$

or

$$T_{N+2} = T_N \tag{5.8}$$

Applying the energy balance principle for the end surface, the corresponding discretized equation can be derived as

$$T_N - (2+A)T_{N+1} + T_{N+2} = -AT_f \quad \text{for } i = N+1$$

After substituting for the temperature of the node $(N + 2)$ from Equation 5.8, the above equation reduces to the form

$$2T_N - (2+A)T_{N+1} = -AT_f \tag{5.9}$$

Now, Equation 5.9 can be taken as the nodal equation for the boundary node $(N+1)$. Considering the Taylor series expansions of the differential Equation 5.2 and the boundary condition (5.4), it is easily shown that Equation 5.9 is second order accurate (i.e. a discretization error of $O(\triangle x^2)$). Here, by combining the discrete forms of the energy balance equation and the derivative boundary condition through the use of the image node $(N + 2)$, a nodal equation with second order accuracy has been derived for the boundary . On the other hand, if the backward difference expression is used for the derivative at $x = L$, there

is no need to consider the extra node outside the boundary. The equation for the node $(N+1)$ then becomes

$$\frac{dT}{dx} = \frac{T_{N+1} - T_N}{\Delta x} = 0$$

or

$$T_{N+1} - T_N = 0 \tag{5.10}$$

Although the above nodal equation is derived in a much simpler fashion, it is not a suitable choice since it is only first order accurate (namely, the error is $O(\Delta x)$. Therefore, the form given in Equation 5.9 is preferred over that of Equation 5.10.

It is also possible to obtain nodal equations with higher order accuracy on the boundary, by fitting a polynomial form for the variation of temperature, as discussed in Chapter 4 (Tables 4.1-4.2). For instance, fitting a quadratic formula between the values of T_{N+1}, T_N and T_{N-1}, the temperature variation near the boundary $x = L$ can be expressed as

$$\begin{aligned} T(x) &= a + b\,(x - L) + c\,(x - L)^2 \\ &= T_{N+1} + \frac{3T_{N+1} - 4T_N + T_{N-1}}{2\Delta x}\,(x - L) \\ &+ \frac{T_{N+1} - 2T_N + T_{N-1}}{2\Delta x^2}\,(x - L)^2 \end{aligned}$$

Using the above polynomial expression, the boundary condition at $x = L$ Equation 5.4 can be discretized as

$$\left.\frac{dT}{dx}\right|_{x=L} = \frac{3T_{N+1} - 4T_N + T_{N-1}}{2\Delta x} = 0$$

Therefore, the nodal equation for $x = L$ is given by

$$3T_{N+1} - 4T_N + T_{N-1} = 0 \tag{5.11}$$

From Taylor series expansions of the temperature profile around the node $(N+1)$, it can be proved that the discretization error for the above equation is $O(\Delta x^2)$. The polynomial-fitting method can be employed for obtaining nodal equations with even third or fourth order accuracy on boundary, by using suitable number of nodes in the polynomial approximation. The only minor drawback is that the nodal equations obtained by this technique often disturb the tri-diagonal structure (in one dimensional problems) for the coefficient matrix of discretized equations.

The final system of algebraic equations for the fin problem can be explicitly derived as follows. Consider a grid of $N = 4$, with image point method employed for the boundary condition at $x = L$. Then

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -(2+A) & 1 & 0 & 0 \\ 0 & 1 & -(2+A) & 1 & 0 \\ 0 & 0 & 1 & -(2+A) & 1 \\ 0 & 0 & 0 & 2 & -(2+A) \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \begin{bmatrix} T_b \\ -AT_f \\ -AT_f \\ -AT_f \\ -AT_f \end{bmatrix}$$

It is evident that the coefficient matrix in the above equation has a tridiagonal structure. The highly efficient Thomas algorithm can be employed for the solution of this matrix equation (Chapter 2).

Example 5.2. Two dimensional heat conduction in rectangular geometry

Figure 5.2a shows a long rectangular bar which experiences two dimensional steady heat conduction in the x-y plane. In the absence of heat sources, the heat conduction equation for this geometry is given by

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{5.12}$$

The following boundary conditions are applied on the side surfaces of the bar:

$$\begin{aligned} T = T_0 \text{ at } x &= 0 \quad \text{(prescribed temperature)} \\ \frac{\partial T}{\partial y} = 0 \text{ at } y &= 0 \quad \text{(insulated boundary)} \\ -k\,\frac{\partial T}{\partial x} = q \text{ at } x &= L \quad \text{(applied heat flux)} \\ -k\,\frac{\partial T}{\partial y} = h\,(T - T_f) \text{ at } y &= H \quad \text{(convective heat transfer)} \end{aligned} \tag{5.13}$$

Let us employ an equi-spaced rectangular grid as shown in Figure 5.2b. The step sizes for this grid are given by

$$\Delta x = \frac{L}{N}\ ,\ \Delta y = \frac{H}{M}$$

where N and M are the number of divisions in the x and y directions respectively. For identification of each node in the grid, it is necessary to assign numbers to the nodes. The numbering could be done in x direction from left to right along each y = constant grid line, or in y direction from bottom to top along each x = constant line. However, it is often convenient to use separate indices (i and j) for the two directions. For a $(N+1) \times (M+1)$ grid, these

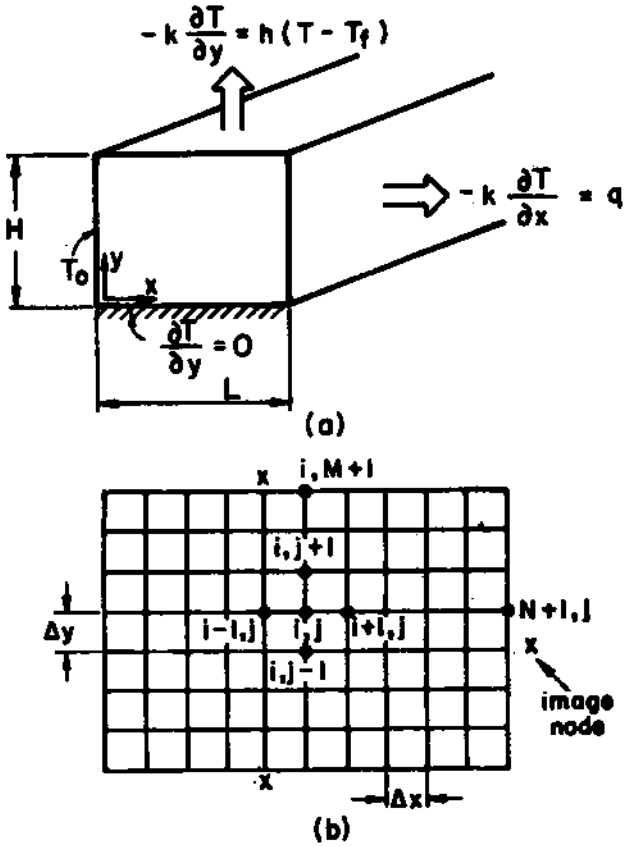


Figure 5.2: Two-dimensional Heat Conduction in a Bar, a. Physical Problem, b. Grid

directional indices vary from 1 to $(N+1)$ and 1 to $(M+1)$ as shown in Figure 5.2b.

Let us now consider the discretized form of the heat balance Equation 5.12 at a node (i, j). Using central difference approximations for the second derivatives, the nodal equation of (i, j) is given by

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

Defining the grid aspect ratio $\beta = (\Delta x/\Delta y)$, the above equation can be simplified as

$$T_{i-1,j} + T_{i+1,j} + \beta^2\, T_{i,j-1} + \beta^2\, T_{i,j+1} - 2(1+\beta^2)T_{i,j} = 0 \qquad (5.14)$$

An important point to be noted here with regard to the discretization procedure is that one dimensional difference expressions have been used in the x- and y-directions, for approximating the second derivatives in the heat conduction equation. Such a simplification is possible only on an orthogonal grid. A detailed discussion of the complexities associated with finite difference applications on a non-orthogonal grid has been presented in Chapter 13.

The nodal equations for all the interior nodes ($i = 2, 3..., N; j = 2, 3..., M$) can now be obtained using Equation 5.14. For deriving the nodal equations of the boundary nodes, the discretized forms of the boundary conditions have to be considered. The prescribed temperature condition at $x = 0$ becomes

$$T_{1,j} = T_0 \quad \text{for} \quad 1 \leq j \leq (M+1) \tag{5.15}$$

For the other three boundary conditions which involve the temperature derivative, one may consider image nodes outside the respective boundaries for deriving second order accurate nodal equations. On the lower side (for $j = 1$), the insulated boundary condition can be discretized as

$$\frac{dT}{dy} = \frac{T_{i,2} - T_{i,0}}{2\Delta y} = 0$$

or

$$T_{i,2} = T_{i,0}$$

Here, $(i, 0)$ is an image node located at a distance of Δy below the lower boundary for each i. Applying the heat balance equation at the lower boundary, we obtain

$$T_{i-1,1} + T_{i+1,1} + \beta^2 T_{i,0} + \beta^2 T_{i,2} - 2(1+\beta^2)T_{i,1} = 0$$

Combining the above two equations, the final form of the nodal equation for the lower side is given by

$$T_{i-1,1} + T_{i+1,1} + 2\beta^2 T_{i,2} - (1+\beta^2)\, T_{i,1} = 0 \quad \text{for } 2 \leq i \leq N \tag{5.16}$$

For the right boundary, the applied heat flux condition is implemented as follows. Considering an image point $(N+2, j)$ outside the boundary, the discretized form of the boundary condition becomes

$$-k\frac{dT}{dx} = -k\frac{T_{N+2,j} - T_{N,j}}{2\Delta x} = q$$

or

$$T_{N+2,j} = -\frac{2q\Delta x}{k} + T_{N,j}$$

Combining the above equation with the heat balance equation for the boundary node $(N+1, j)$, the nodal equations for the right boundary are given by

$$\begin{aligned} 2T_{N,j} + \beta^2 T_{N+1,j-1} + \beta^2 T_{N+1,j+1} &- 2(1+\beta^2)T_{N+1,j} \\ &= \frac{2q\Delta x}{k} \text{ for } 2 \le j \le M \end{aligned} \quad (5.17)$$

In a similar manner, the nodal equations for the top boundary can be obtained by discretizing the convective condition using image points and substituting for the image point temperature in the heat balance equation from the boundary condition. The resulting nodal equations are

$$\begin{aligned} T_{i-1,M+1} + T_{i+1,M+1} &+ 2\beta^2 T_{i,M} - 2(1+\beta^2 + \beta^2 \frac{h\Delta y}{k} T_{i,M+1} \\ &= -(2\beta^2 \frac{h\Delta y}{k} T_f \text{ for } 2 \le i \le N \end{aligned} \quad (5.18)$$

Now let us consider the corner nodes of the rectangular domain. The nodal equations of the corner points lying on the prescribed temperature boundary $(x = 0)$ have already been included in Equation 5.15. For $i = 1$ and $j = 1$ or $M + 1$, these equations are of the form

$$\begin{aligned} T_{1,1} &= T_0 \\ T_{1,M+1} &= T_0 \end{aligned} \quad (5.19)$$

For the corner node $(N+1, 1)$, two image points $(N+1, 0)$ and $(N+2, 1)$ are considered, Figure 5.3. The boundary conditions on the lower and the right surfaces are discretized using these image nodes. After combining the discretized boundary conditions with the energy balance equation (5.14), the following nodal equation is obtained:

$$T_{N,1} + \beta^2 T_{N+1,2} - (1+\beta^2)\, T_{N+1,1} = \frac{q\ \Delta x}{k} \quad (5.20)$$

Similarly, by incorporating the boundary conditions of applied heat flux and convective heat exchange with the help of two image points (Figure 5.4), the equation for the node $(N+1, M+1)$ is obtained as

$$\begin{aligned} T_{N,M+1} + \beta^2 T_{N+1,M} &- (1+\beta^2+\beta^2 \frac{h\Delta y}{k})T_{N+1,M+1} \\ &= -\beta^2 \frac{h\Delta y}{k} T_f + \frac{q\Delta x}{k} \end{aligned} \quad (5.21)$$

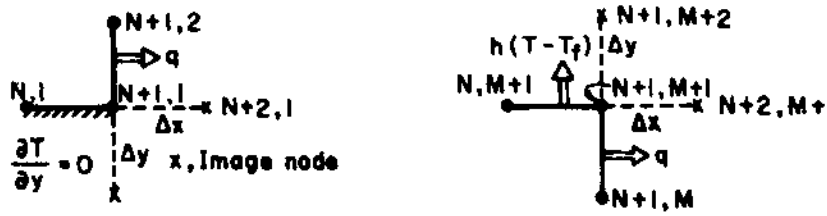


Figure 5.3: Use of Image Nodes for Corner Points

Equations 5.14 to 5.19 are second order accurate for determining the temperatures of the interior and the boundary nodes.

Until now, we have employed the double index notation (i, j) to identify each node. For setting up the matrix form of equations, it is necessary to number the nodes sequentially in the x- or y- direction. Let us number the nodes in x-direction from left to right. The number 'n' of a node in this scheme is related to the directional indices (i, j) through the relationship

$$n = (j-1)(M+1) + i \tag{5.22}$$

With the above numbering scheme, the matrix equation for $N = M = 3$ is obtained as

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & a & 1 & 0 & 0 & b' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & a & 1 & 0 & 0 & b' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & a' & 0 & 0 & 0 & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & b & 0 & 0 & 1 & a & 1 & 0 & 0 & b & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & b & 0 & 0 & 1 & a & 1 & 0 & 0 & b & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & b & 0 & 0 & 2 & a & 0 & 0 & 0 & b & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 1 & a & 1 & 0 & 0 & b & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 1 & a & 1 & 0 & 0 & b & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 2 & a & 0 & 0 & 0 & b \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b' & 0 & 0 & 1 & d & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b' & 0 & 0 & 1 & d & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 1 & d'
\end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \\ T_{10} \\ T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \\ T_{15} \\ T_{16} \end{bmatrix}$$

$$= [T_0 \; 0 \; 0 \; c \; T_0 \; 0 \; 0 \; 2c \; T_0 \; 0 \; 0 \; 2c \; T_0 \; 2e \; 2e \; e + c]^T$$

where $a = -2(1+\beta^2), b = \beta^2, a' = a/2, b' = 2b, c = (q\ \Delta x/k), d = -2(1 + \beta^2 + \beta^2 h\ \Delta y/k)$, $d' = d/2$ and $e = -(\beta^2 h\ \Delta y/k)$. The above matrix equation

can be solved with the help of any standard procedure such as the Gaussian elimination, for obtaining the nodal temperatures. For large number of nodes in the domain, the matrix solution procedure may not be efficient unless some special technique, which takes advantage of the high sparsity of the coefficient matrix is used. Indeed, it is evident that for the two-dimensional example considered here, the coefficient matrix has a penta-diagonal structure. Efficient matrix algorithms are available for the storage and solution of such systems (Anderson *et al.*, 1984).

A simple iterative technique which finds wide use in the solution of algebraic equation systems of the above type is the line-by-line method. In this method, the nodal equations of grid points lying on a grid line (x=constant or y=constant) are solved together. For instance, consider a y=constant grid line. On this line, the index i varies from 1 to $(N+1)$ while j has a fixed value. Let us rewrite Equation 5.14 in the form

$$T_{i-1,j} + T_{i+1,j} - 2(1+\beta^2)T_{i,j} = -\beta^2 T_{i,j-1} - \beta^2 T_{i,j+1} \tag{5.23}$$

If the above equation is applied for $i = 2$ to N (for a fixed j), we obtain the set of nodal equations for all the interior nodes of the grid line under consideration. The nodal equations for the boundary points on this line can be written as

$$T_{1,j} = T_0 \tag{5.24}$$

and

$$2T_{N,j} - 2(1+\beta^2)T_{N+1,j} = -\beta^2 T_{N+1,j-1} - \beta^2 T_{N+1,j+1} + \frac{2q\triangle x}{k} \tag{5.25}$$

The set of Equations 5.23 - 5.25 can be written in the matrix form:

$$[A]\ \{T\} = [B] \tag{5.26}$$

corresponding to the nodal index values of $i = 1$ to $N+1$, at a constant j. It can be verified easily that the coefficient matrix $[A]$ has a tridiagonal structure. Also, $\{T\}$ consists of the nodal temperatures on the selected grid line, while $[B]$ is a function of the nodal temperatures of the neighbouring grid lines. Providing an initial guess for the nodal temperatures of the entire domain (for evaluating $[B]$), Equation 5.26 can be solved by the Thomas algorithm for each value of j ranging from 1 to $(M+1)$. After one such cycle, the updated nodal temperatures are used as the initial guess for the next cycle of line-by-line solution. This iterative procedure is continued until the nodal temperatures converge within a prescribed tolerance.

In order to achieve a rapid convergence, the line-by-line technique is applied for the x-grid lines in one cycle and the y-grid lines during the next cycle. While applying this procedure on the x=constant grid lines, the entries for the nodes $(i, j-1), (i, j)$ and $(i, j+1)$ are kept on the left hand side and those of nodes $(i-1, j)$ and $(i+1, j)$ are transferred to the right hand side. The resulting tridiagonal matrix system is solved in the same manner as described earlier.

In some special applications, the point-by-point iterative scheme may be employed for solving the system of discretized equations. In this procedure, only the entry corresponding to the node i, j (whose temperature is currently being solved) is kept on the left hand side and all the terms containing the neighbouring temperatures in the nodal equations are transferred to the right hand side. Each node is visited sequentially and its temperature is updated using the available guess values of the neighbouring node temperatures during any iteration cycle. Obviously, this method has a slow rate of convergence and is therefore preferred only when the iterative solutions of the discretized equations tend to become unstable, probably due to the presence of some strong non-linearities in the problem. Both line-by-line and point-by-point techniques can be over-relaxed (in linear problems) or under-relaxed (in non-linear problems).

The double index (i, j) notation is quite useful for the line-by-line or point-by-point iterative solution schemes. For complete matrix inversion, the sequential numbering of nodes is essential. In some numerical formulations, a general notation of E, W, N, S (implying east, west, north and south) is employed during the discretization stage and it is converted to the multi-indexing or the sequential numbering schemes at the time of solving the algebraic equations.

Example 5.3. One-dimensional conduction with variable thermal conductivity and a heat source

It was mentioned earlier that the nodal equations can be obtained by integration of the energy balance equation over a control volume also. This method is more advantageous for problems with varying material properties. For instance, consider the following one dimensional heat conduction equation

$$\frac{d}{dx}\left[k\,\frac{dT}{dx}\right] = 0 \tag{5.27}$$

where k, the thermal conductivity is a function of x. Let us integrate the above equation over the computational cell shown in Figure 5.4 a, around a node P. We shall denote the node on the east side of P as E and the node on the west side as W. The boundary faces of the control volume in the east and west sides are denoted as e and w respectively. Now, integration of Equation 5.27 leads to:

$$\left[k\,\frac{dT}{dx}\right]_e - \left[k\,\frac{dT}{dx}\right]_w = 0 \tag{5.28}$$

where the subscripts e and w indicate that the quantities inside the brackets are evaluated at the respective control volume faces. It is desirable that all quantities in the above equation be expressed in terms of their magnitudes at the nodes E, W and P. This is because the integration over the control volume is only an intermediate step and the final discretized equations concern only with nodal quantities. The temperature gradients at the control volume boundaries are written in central difference form as

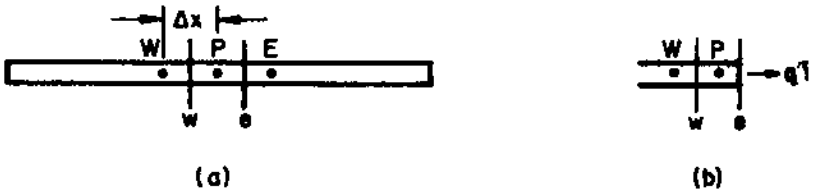


Figure 5.4: Choice of Control Volume

$$\left[k\,\frac{dT}{dx}\right]_e = \frac{T_E - T_P}{\Delta x}$$
$$\left[k\,\frac{dT}{dx}\right]_w = \frac{T_P - T_W}{\Delta x} \tag{5.29}$$

The thermal conductivity values k_e and k_w can be calculated as

$$k_e = \frac{(k_E + k_P)}{2}$$
$$k_w = \frac{(k_P + k_W)}{2} \tag{5.30}$$

if k varies smoothly between the nodes E, P and W. On the other hand, if $k(x)$ varies sharply with x, the conductivity value at each control volume face has to be calculated as the harmonic average between the nodal values, instead of the arithmetic average (Patankar, 1980). The significance of the harmonic average is understood as follows. Consider heat conduction through two slabs which are in perfect thermal contact with each other, as shown in Figure 5.5. The equivalent resistance to heat conduction is given by

$$R_{eq} = R_1 + R_2 = \frac{\delta}{k_1 A} + \frac{\delta}{k_2 A} = \frac{\delta}{k_{eq} A} \tag{5.31}$$

where

$$\frac{1}{k_{eq}} = \frac{1}{k_1} + \frac{1}{k_2} \tag{5.32}$$

Thus, taking k_{eq} as the harmonic average properly accounts for the equivalent resistance when discrete changes in k occur between two neighbouring nodes. If k_1 and k_2 are not very different from each other, then $k_{eq} = k_1 k_2/(k_1 + k_2) = (k_{av} - \varepsilon)(k_{av} + \varepsilon)/(2k_{av}) \approx k_{av}/2$, where k_{av} is the arithmetic average $(k_1 + k_2)/2$ and ε is equal to $(k_2 - k_1)/2$. From the above arguments, it is evident that the conductivity at the cell face can be calculated as the arithmetic average between the nodal values, as long as the ratio $\Delta k/k_{av} \ll 1$, where Δk is the difference in the nodal thermal conductivity values.

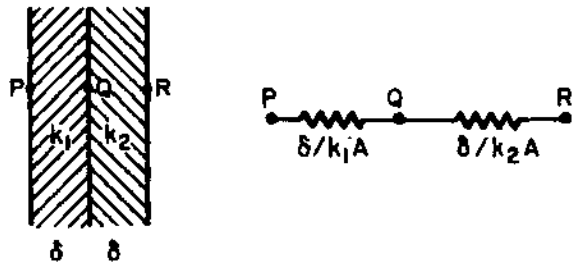


Figure 5.5: Evaluation of Equivalent Thermal Conductivity

A major advantage of the control volume approach lies in its easy application to problems of coupled heat transfer through different materials. In the direct differencing approach, such problems are handled only by solving for each region separately and coupling the solutions at the interface through continuity conditions for heat flux and temperature. In the integral approach, there is no need for such separate treatment of each region; by selecting the equivalent thermal properties for the control volume around the interface appropriately, it is possible to provide a unified solution methodology for the whole domain of the problem. Another advantage of this method is that the prescribed heat flux condition can be implemented directly, without discretization of the temperature gradient at the boundary. By choosing the control volume face (e or w) to coincide with the boundary, Equation 5.28 is simplified through the substitution of the heat flux value at the boundary (Figure 5.4b). The control volume approach can also easily account for variations in the spatial step size, as described later in Example 5.4.

When thermal conductivity varies as a function of temperature, the heat conduction equation becomes non-linear. However, using the Kirchoff transformation, it is possible to reduce the non-linear equation to a linear form. For example, consider the one dimensional equation

$$\frac{d}{dx}\left(k(T)\,\frac{dT}{dx}\right) = 0 \tag{5.33}$$

Defining the variable

$$g(x) = \int_{T_0}^{T} \frac{k(T)}{k(T_0)}\,dT \tag{5.34}$$

where T_0 is any suitable reference temperature, it can be shown that

$$\frac{dg}{dx} = k(T)\,\frac{dT}{dx}$$

using Leibnitz rule. With the substitution of the above expression, the heat conduction Equation 5.33 simplifies to the form

$$\frac{d^2 g}{dx^2} = 0 \tag{5.35}$$

Let us consider the boundary conditions of prescribed boundary temperature ($T = T_0$ at $x = 0$) and applied heat flux ($-kdT/dx = q$ at $x = L$), for the one dimensional heat conduction Equation 5.33. These boundary conditions transform as

$$g = 0 \text{ at } x = 0 \tag{5.36}$$

and

$$-\frac{dg}{dx} = q \text{ at } x = L \tag{5.37}$$

Note that we can use the prescribed boundary temperature T_0 itself as the reference temperature for the definition of g in Equation 5.34. Obtaining the solution for the transformed Equations 5.35 and (5.36) and (5.37) is quite straightforward. In order to get the temperature profile, however, Equation 5.34 has to be solved at every node. It may be necessary to employ an iterative procedure for this purpose, if $k(T)$ does not have a simple form.

Let us now consider the problem of one-dimensional steady heat conduction with a heat source, $Q(x)$. The governing differential equation for this problem is given by

$$\frac{d}{dx}\left[k\,\frac{dT}{dx}\right] + Q(x) = 0 \tag{5.38}$$

Integrating the above equation over a control volume around the node P, we obtain

$$\left[k\,\frac{dT}{dx}\right]_e - \left[k\,\frac{dT}{dx}\right]_w + \int_{x_e}^{x_w} Q(x)\,dx = 0 \tag{5.39}$$

For a uniform step size Δx, Equation 5.39 can be rewritten as

$$k_e\,\frac{T_E - T_P}{\Delta x} - k_w\,\frac{T_P - T_W}{\Delta x} + Q(x_P)\Delta x = 0 \tag{5.40}$$

Note that the rate of heat generation $Q(x)$ has been treated like a constant (equal to the average value of $Q(x)$ over the cell), in the evaluation of the heat source integral. This is in view of the fact that the discretization error for the conduction flux terms in Equation 5.40 is $O(\Delta x^2)$ and therefore, it is sufficient to approximate the heat source term with an error of $O(\Delta x^2)$. The nodal equation for P can now be written as

$$k_e T_E + k_w T_W - (k_e + k_w)T_P + Q(x_p)\,\Delta x^2 = 0 \tag{5.41}$$

For increasing the accuracy of representation of the heat source term, a linear variation within the control volume can be used for $Q(x)$ during integration. This leads to

$$\int_{x_e}^{x_w} Q(x)\, dx = \frac{(Q_e + Q_w)}{2}\, \Delta x + O(\Delta x^3) \simeq \frac{(Q_E + 2Q_P + Q_W)}{4}\, \Delta x \quad (5.42)$$

If the rate of heat generation is a function of temperature, then obtained as nodal equations with an accuracy of $O(\Delta x^2)$ can be

$$k_e T_E + k_w T_W - (k_e + k_w) T_P + Q(T_P)\, \Delta x^2 = 0 \quad (5.43)$$

When $Q(T)$ is non-linear, the solution procedure for the resulting system of algebraic equations has to be iterative in nature. Here, we may evaluate the conduction terms at the current iteration level and the heat source term at the previous iteration level. Thus, the nodal Equation 5.43 can be rewritten as

$$k_e T_E^{n+1} + k_w T_W^{n+1} - (k_e + k_w) T_P^{n+1} + Q(T_P^n)\, \Delta x^2 = 0 \quad (5.44)$$

where the superscripts $(n+1)$ and n denote current and previous iteration level solutions respectively. For each node, the rate of heat generation $Q(T_P^n)$ can be treated as a constant and transferred to the right hand side of the equation. On the other hand, for faster convergence of the iterative procedure, the source term can also be linearized with respect to T_P^{n+1} in one of the many possible ways. For instance, consider the form $Q(T) = AT^2$, for the rate of heat generation. This function can be linearized as

$$Q(T_P) \simeq (AT_P^n)\, T_P^{n+1} \quad (5.45)$$

Alternatively, one can use the Taylor expansion (up to the first term) and obtain

$$Q(T_P) \simeq Q(T_P^n) + (2AT_P^n)(T_P^{n+1} - T_P^n) = 2\, A\, T_P^n\, T_P^{n+1} - A\, (T_P^n)^2 \quad (5.46)$$

Even an *ad hoc* expression such as

$$Q(T_P) \simeq 2\, A (T_P^n)^2 - (A\, T_P^n)\, T_P^{n+1} \quad (5.47)$$

could be considered as a possibility. Since $T_P^n \simeq T_P^{n+1}$ during the later stages of convergence, all the three expressions given above reduce to the form $Q = AT^2$, as $n \to \infty$. However, all of these expressions may not increase the rate of convergence and some might even lead to numerical instability. It can be shown that if $Q(T)$ is linearized as $Q(T_P) = \alpha\, T_P^{n+1} + \beta$, the coefficient α should be negative for the iterative scheme to be stable.

With regard to the solution of Equation 5.44, the substitution of Equation 5.45 or 5.46 for $Q(T)$, could make the iterative procedure unstable for $A > 0$, while the use of Equation 5.47 will provide a stable scheme with a slow rate of convergence. If $A < 0$, the forms given in Equations 5.45 and 5.46 lead to

stable solutions, with the rate of convergence for Equation 5.46 being higher. The form of Equation 5.47 is clearly not preferable since it has a positive value of α. In general, when the Taylor series expansion has a positive coefficient for the linear term, it is better to evaluate $Q(T)$ as $Q(T_P^n)$ itself, corresponding to the case of $\alpha = 0$.

With appropriate discretization of the heat source term, nodal equations for the interior and boundary nodes can be obtained and solved as a system of linear equations.

Example 5.4. Solution of two dimensional heat conduction equation on a non-uniform mesh

The governing differential equation for two-dimensional steady heat conduction with varying thermal conductivity $k = k(x, y)$, is given by

$$\frac{\partial}{\partial x}\left\{k\frac{\partial T}{\partial x}\right\} + \frac{\partial}{\partial y}\left\{k\frac{\partial T}{\partial y}\right\} = 0 \tag{5.48}$$

Let us consider a mesh with step sizes varying in both the x and y directions, as shown in Figure 5.6a. Integration of the energy balance equation over a control volume around the point P (Figure 5.6b) yields

$$\left\{\left[k\frac{\partial T}{\partial x}\right]_e - \left[k\frac{\partial T}{\partial x}\right]_w\right\} \Delta y + \left\{\left[k\frac{\partial T}{\partial y}\right]_n - \left[k\frac{\partial T}{\partial y}\right]_s\right\} \Delta x = 0$$

Now, substituting central difference expressions for each of the heat fluxes at the control volume faces (e, w, n, s) leads to

$$\frac{k_e}{\Delta x_e}T_E + \frac{k_w}{\Delta x_w}T_W + \frac{\beta k_n}{\Delta y_n}T_N + \frac{\beta k_s}{\Delta y_s}T_S - \left[\frac{k_e}{\Delta x_e} + \frac{k_w}{\Delta x_w} + \frac{\beta k_n}{\Delta y_n} + \frac{\beta k_s}{\Delta y_s}\right] T_p = 0 \tag{5.49}$$

where $\beta = \Delta x/\Delta y, \Delta x = (\Delta x_e + \Delta x_w)/2, \Delta y = (\Delta y_n + \Delta y_s)/2$ and k_e, k_w, k_n, and k_s are the equivalent thermal conductivities between the respective neighbouring nodes and the central node P. As stated earlier, they are to be computed as harmonic averages of the neighbouring values. For a smooth variation of k in the neighbourhood of P, the equivalent conductivities can be taken as the arithmetic averages between the nodes, namely

$$k_e = \frac{k_E + k_P}{2} \quad ; \quad k_w = \frac{k_W + k_P}{2}$$

$$k_n = \frac{k_N + k_P}{2} \quad ; \quad k_s = \frac{k_S + k_P}{2} \tag{5.50}$$

In the derivation of the nodal equations by the control volume approach so far, the subscripts of E, N, S and W have been employed for representing the

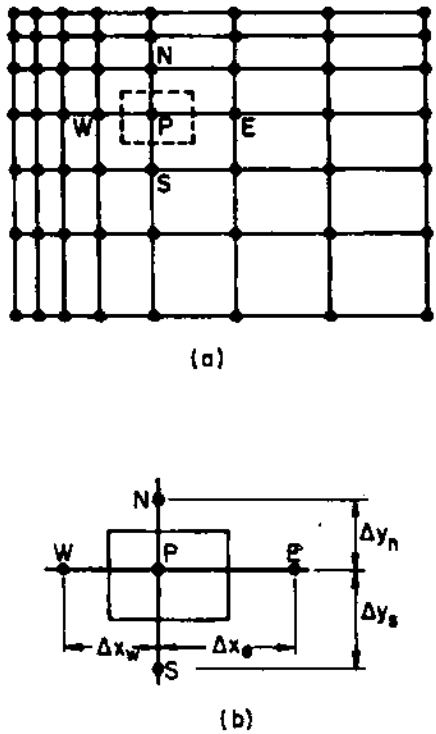


Figure 5.6: Control Volume Formulation for a Non-uniform Grid

neighbouring nodes around the point P. During the solution of the discretized equations, however, one may use the actual node numbers in place of the directional subscripts for the direct matrix algorithms or the double index (i, j) notation for the iterative line-by-line method. For the point-by-point method, Equation 5.49 is re-written as

$$T_P^{n+1} = \frac{\sum_i \alpha_i T_i^*}{\sum_i \alpha_i}$$

where the index i varies over E, W, N ans S. The coefficients α_i are immediately derivable from Equation 5.49. Index n represents the current iteration level. If $*$ on the right hand side is taken to be n, the point-by-pont method classifies as *Jakobi* iteration. If it is interpreted as n or $n+1$, depending on the availability of the latest value of T_i, the method is identical to the *Gauss-Seidel* technique (Chapter 2).

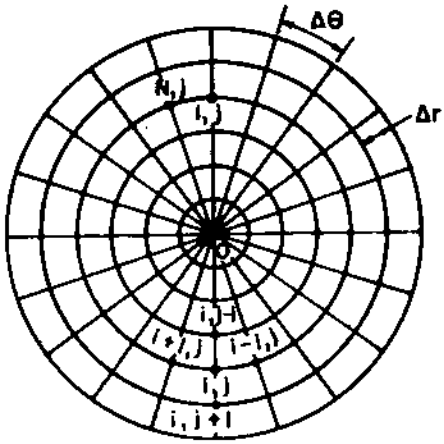


Figure 5.7: Finite Difference Mesh in Cylindrical Geometry

5.1.2 Steady Heat Conduction in Cylindrical and Spherical Geometries

Finite difference simulation in cylindrical and spherical geometries is similar to that in rectangular geometry, except for a few minor differences. The finite difference grid may be made up of curved grid lines in some coordinate directions and it is necessary to account for the change in the length/area of the control volume faces for these directions. Another important factor for consideration is that certain lines or points of singularity are inherent in these systems, where the grid becomes irregular. Special methods are necessary to discretize the governing equations and boundary conditions at such locations. For instance, the axis ($r = 0$) is a line of singularity in the cylindrical coordinate system where all the radial grid lines converge at a single node O (Figure 5.7). Due to this, the number of nodes in the angular direction on each r = constant grid line is equal, except at $r = 0$. Moreover, the centre does not possess uniquely identifiable neighbours on either side along orthogonal grid lines, unlike rest of the nodes of the mesh. Because of such singularities, special care is required while discretizing the equations at $r = 0$. For the spherical geometry, the axes $\theta = 0$ and π as well as the origin $r = 0$, exhibit singularities in the mesh structure.

Another peculiar feature of the curvilinear geometries is that no boundary conditions are required in the angular direction if the corresponding angular coordinate undergoes a complete revolution, i.e. $\theta = 0$ to $\theta = 2\pi$. Since θ equal to 0 or 2π implies the same location, continuity conditions for temperature and heat flux are imposed across the mesh boundaries in the angular direction. Lastly, something can be said about the structure of the coefficient matrix for the discretized equations. Due to the mesh irregularities at the singular points

Figure 5.8: Grid for Radial Conduction in a Cylinder

and lines and the continuity conditions in the angular direction, the matrices do not possess tridiagonal, pentadiagonal and septa- diagonal structure for one, two and three dimensional problems respectively, as discussed in the case of rectangular coordinates.

The following examples illustrate the implementation of the finite difference technique for heat conduction in curvilinear geometries.

Example 5.5 Heat conduction in a cylindrical geometry

Here, one dimensional heat conduction in radial direction and the two dimensional conduction problem in r-θ coordinates are described.

Consider steady radial conduction in a long cylinder, with a heat source $Q(r)$. The energy balance equation for this situation is given by

$$k\left\{\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right\} + Q(r) = 0 \tag{5.51}$$

Let the surface of the cylinder be maintained at a temperature of T_s . This is the only imposed boundary condition for the problem. The other condition is to be derived from the prevailing symmetry at the centre. For an axisymmetric problem in cylindrical geometry, it can be shown that $\partial T/\partial r = 0$ at $r = 0$, which serves as the second boundary condition.

Consider a radial grid with constant step size Δr, as shown in Figure 5.8. At a node i, the discretized form of the heat conduction Equation 5.51 becomes

$$k\left\{\frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta r^2} + \frac{T_{i+1} - T_{i-1}}{r_i\, 2\Delta r}\right\} + Q(r_i) = 0 \text{ for } i = 2 \ldots N \tag{5.52}$$

For the surface node $(N+1)$, the prescribed temperature condition gives

$$T_{N+1} = T_s \tag{5.53}$$

For deriving a second order accurate nodal equation at the centre ($i = 1$), the discretized form of the differential equation needs to be combined with the boundary condition at $r = 0$. However, an examination of the governing Equation 5.51 indicates that at $r = 0$, the second term cannot be handled directly since the factor $1/r$ approaches ∞. Therefore, a special procedure has

to be introduced to handle this term. A simple way of removing the difficulty is to apply the L'Hospital rule. Thus

$$\mathrm{Lt}_{r\to 0}\ \frac{1}{r}\ \frac{\partial T}{\partial r} = \frac{\mathrm{Lt}_{r\to 0}\ \frac{\partial}{\partial r}\left\{\frac{\partial T}{\partial r}\right\}}{\mathrm{Lt}_{r\to 0}\ \frac{\partial}{\partial r}\{r\}} = \frac{\partial^2 T}{\partial r^2} \tag{5.54}$$

The modified differential equation for $r = 0$ is then given by

$$k\left\{2\frac{\partial^2 T}{\partial r^2}\right\} + Q(r) = 0 \tag{5.55}$$

Combining the discretized form of the above equation with that of the boundary condition at $r = 0$, the second order accurate nodal equation for the centre ($i = 1$) is obtained by the image point approach as

$$4k(T_2 - T_1) + Q(0)\Delta r^2 = 0 \tag{5.56}$$

The system of Equations 5.52, 5.53 and 5.56 can be arranged in a tridiagonal form and solved by the Thomas algorithm. The main feature to be observed in the present 1D example is the application of the L'Hospital rule for resolving the difficulty at $r = 0$. However, this trick works only when axisymmetry prevails at $r = 0$. For a general situation, the nodal equation is obtained by considering a local Cartesian mesh as described in the following example.

Consider heat conduction in the r-θ plane of a long cylinder. The steady conduction equation in the absence of heat source is given by

$$\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{1}{r^2}\frac{\partial^2 T}{\partial \theta^2} = 0 \tag{5.57}$$

The finite difference mesh for the problem is shown in Figure 5.7. Let M and N be the number of steps in the r and θ directions respectively. Using the double index notation, the discretized form of the governing equation at node (i, j) is given by

$$\frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta r^2} + \frac{T_{i,j+1} + T_{i,j-1}}{r_j 2\Delta r} + \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{r_j^2 \Delta\theta^2} = 0 \quad \text{for } 2 \le i \le N, 2 \le j \le M \tag{5.58}$$

Let the surface temperature be specified as

$$T_{i,M+1} = T_s(i) \tag{5.59}$$

for each angle θ_i . The continuity conditions at $\theta = 0$ and 2π are incorporated by substituting (Figure 5.9a)

$$T_{i+1,j} = T_{1,j} \quad \text{for } i = N \tag{5.60}$$

$$T_{i-1,j} = T_{N,j} \quad \text{for } i = 1 \tag{5.61}$$

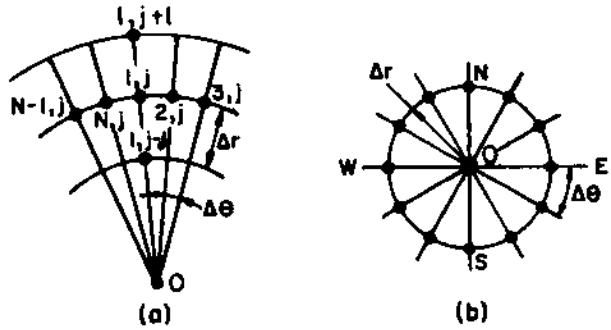


Figure 5.9: Implementing Boundary Conditions in Cylindrical Geometry

in the Equation 5.58, since nodes $(1, j)$ and (N, j) are neighbours for each j. At the centre $r = 0$, the gradient $\partial T/\partial r$ depends on the angle θ, and it is not possible to simplify the governing equation by the use of L'Hospital rule. Therefore, we shall consider a local Cartesian mesh as shown in Figure 5.9b. On this mesh, the heat conduction Equation 5.9a in Cartesian coordinates, is discretized as

$$\frac{T_E + T_W + 2T_0}{\triangle r^2} + \frac{T_N + T_S - 2T_0}{\triangle r^2} = 0 \tag{5.62}$$

where E, N, W and S are the neighbouring nodes as shown in Figure 5.9b and $\triangle x = \triangle y = \triangle r$. It can be verified easily that the penta-diagonality of the matrix structure is disturbed by the nodal equations for $\theta = 0$ and 2π (Equations 5.60 and 5.61) and the nodal equation for $r = 0$ (Equation 5.62).

Example 5.6. Axisymmetric steady heat conduction in spherical coordinates

The heat conduction equation in spherical (r, θ) coordinates is given by (Ozisik, 1980)

$$\frac{\partial^2 T}{\partial r^2} + \frac{2}{r}\frac{\partial T}{\partial r} + \frac{1}{r^2}\left\{\frac{\partial^2 T}{\partial \theta^2} + \cot\theta\,\frac{\partial T}{\partial \theta}\right\} = 0 \tag{5.63}$$

Considering the computational mesh inside a sphere (Figure 5.10) over a half-plane, the nodal equation for an interior point (i, j) is derived as

$$\frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\triangle r^2} + \frac{2}{r_j}\frac{T_{i,j+1} + T_{i,j-1}}{2\triangle r} + \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{r_j^2 \triangle\theta^2} + \cot\theta_i\,\frac{T_{i+1,j} + T_{i-1,j}}{r_j^2 2\triangle\theta} = 0$$

$$\text{for } i = 2, \ldots N \text{ and } j = 2, \ldots M \tag{5.64}$$

Prescribing the temperature of the sphere as a function of the angle θ, the equations for the surface nodes are given by

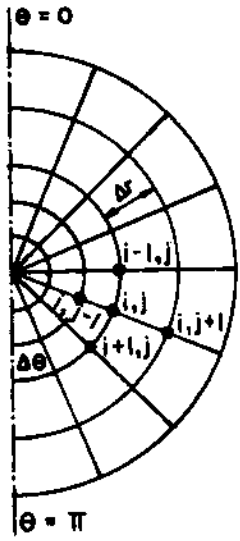


Figure 5.10: Mesh for Axisymmetric Conduction Inside a Sphere

$$T_{i,M+1} = T_s(i) \tag{5.65}$$

On the axis $\theta = 0$ or π, the derivative $\partial T/\partial\theta \rightarrow 0$ due to axisymmetry while $\cot\theta \rightarrow \infty$. Therefore, by resolving the last term of Equation 5.63 through L'Hospital rule, it can be shown that

$$\mathrm{Lt}_{\theta \rightarrow 0},\ \cot\theta \frac{\partial T}{\partial\theta} = \frac{\partial^2 T}{\partial\theta^2} \text{ and } \mathrm{Lt}_{\theta \rightarrow \pi},\ \cot\theta \frac{\partial T}{\partial\theta} = \frac{\partial^2 T}{\partial\theta^2} \tag{5.66}$$

These expressions can be substituted in Equation 5.63 for nodes on the axis ($\theta = 0$ or π) and the corresponding nodal equations can be derived by discretizing the modified forms of Equation 5.63.

For the centre $r = 0$, a local Cartesian mesh has to be used. The heat conduction equation for this node is

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

Referring to Figure 5.11, the above equation can be discretized as

$$\frac{T_E + T_E - 2T_0}{\Delta r^2} + \frac{T_E + T_E - 2T_0}{\Delta r^2} + \frac{T_N + T_S - 2T_0}{\Delta r^2} = 0$$

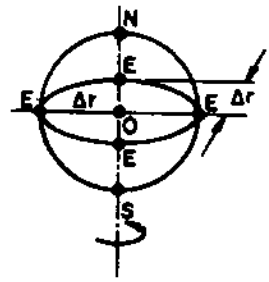


Figure 5.11: Local Cartesian Mesh at the Centre of a Sphere

or

$$4T_E + T_N + T_S - 6T_0 = 0 \tag{5.67}$$

Now, Equation 5.67 is the required nodal equation for the centre ($r = 0$). Similar to the case of cylindrical geometry, it can be shown that the penta-diagonality of the matrix structure is disturbed by the nodal equation of the centre. If the line-by-line iterative method is adopted (Equation 5.23), the terms which introduce irregularities in the coefficient matrix structure can also be transferred to the right hand side and evaluated based on the guess temperature profile.

5.1.3 Transient Heat Conduction

In the previous discussions, procedures of obtaining the nodal equations for steady conduction problems by direct differencing and the control volume approach were described. For transient situations, the spatial discretization can be performed using these methods, in exactly the same manner as described earlier. For discretizing with respect to time, one of the three basic schemes outlined in Chapter 4, i.e. the explicit, implicit and the semi-implicit (or Crank-Nicolson) procedures, can be used. For illustrating the three time discretization schemes, let us consider the following one-dimensional transient conduction equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{5.68}$$

where $\alpha = k/\rho C$. At a node i, the right hand side term can be discretized in space as usual, by substituting the central difference expression for the second derivative as

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2}$$

This gives the time derivative of temperature at the node i as

$$\frac{\partial T}{\partial t} = f(T_{i+1}, T_i, T_{i-1}) \tag{5.69}$$

The question that immediately springs to mind is, at what time level should the function f be evaluated? One may use the initial temperatures which are available at the beginning of a time step or one may express f in terms of the unknown temperatures which are to be determined during that particular time step. It is also possible to evaluate f from a combination of both the known and unknown temperatures. Thus , denoting the temperatures at the current time level by the superscript $(n+1)$ and those at the previous time level by (n), we can write different expressions for the time derivative as

$$\frac{\partial T}{\partial t} = f(T_{i+1}^{n+1}, T_i^{n+1}, T_{i-1}^{n+1}) \tag{5.70}$$

$$\frac{\partial T}{\partial t} = f(T_{i+1}^{n}, T_i^{n}, T_{i-1}^{n}) \tag{5.71}$$

$$\frac{\partial T}{\partial t} = 0.5\ \{f(T_{i+1}^{n}, T_i^{n}, T_{i-1}^{n}) + f(T_{i+1}^{n+1}, T_i^{n+1}, T_{i-1}^{n+1})\ \} \tag{5.72}$$

Here, Equations 5.70, 5.71 and 5.72 correspond to the implicit, explicit and the Crank-Nicolson schemes respectively. For a given problem, the choice of a particular scheme depends on the computational accuracy required, the affordable computational time and storage memory. The explicit method is the cheapest (on a per time step basis), but requires that a stability criterion be satisfied. The implicit method is unconditionally stable, but requires more calculations per time step. The Crank-Nicolson technique is also stable, requires more calculations per time step than implicit, but is second order accurate in time. The comparative features of the three methods outlined in Chapter 4 should be reviewed before making a suitable choice.

Example 5.7. Transient heat conduction in a slab

Consider one dimensional transient conduction of heat in a large rectangular slab, due to the sudden imposition of different surface temperatures at time $t = 0$. Such a situation can arise in practice, when a slab at uniform initial temperature T_i is immersed suddenly into a bath of liquid at T_0 (Figure 5.12). The governing equation for this problem is identical to Equation 5.68, while the boundary conditions are given by

$$\begin{aligned} T &= T_0 \text{ at } x = 0 \\ T &= T_0 \text{ at } x = L \end{aligned} \tag{5.73}$$

The initial condition for the problem is

$$T = T_i \text{ for } 0 < x < L, \text{ at } t = 0 \tag{5.74}$$

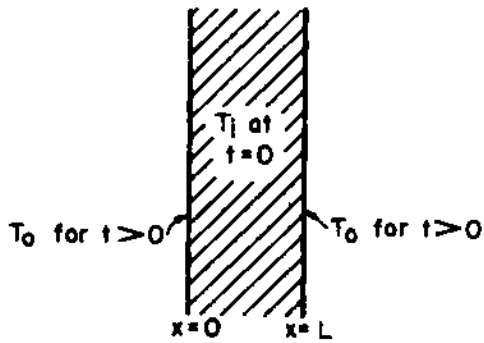


Figure 5.12: Transient Conduction Within a Rectangular Slab

Discretized equations for this problem can be derived by the explicit method as:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \alpha \frac{T_{j+1}^n + T_{j-1}^n - 2T_j^n}{\Delta x^2}$$

for any interior node $j = 2,3...M$, where M is the total number of spatial steps ($M = L/\Delta x$). The above equation can be rearranged as

$$T_j^{n+1} = (1 - 2A)T_j^n + AT_{j+1}^n + AT_{j-1}^n \text{ for } 2 \leq j \leq M \tag{5.75}$$

where $A = (\alpha\ \Delta t/\Delta x^2)$. The boundary conditions are incorporated as

$$\begin{aligned} T_1^{n+1} &= T_0 \\ T_{M+1}^{n+1} &= T_0 \end{aligned} \tag{5.76}$$

For the first time step (with $n = 0$), all the temperatures on the right hand side of Equation 5.75 are set equal to the initial temperature T_i. The $(n + 1)$th level temperatures are evaluated through the solution of nodal Equations 5.75 and 5.76. In the explicit method, for each node j, the updating of the temperature with respect to time can be performed independently from its neighbours. Therefore, no matrix inversion (or iterative line-by-line solution) is required and the computational effort required for each time step is very less. However, from stability considerations, the step sizes for space and time are restricted by the relation

$$A = \frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{5.77}$$

Let us consider the implicit solution of the same problem mentioned above. Here, the discretized equation for each node j is given by

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \alpha \frac{T_{j+1}^{n+1} + T_{j-1}^{n+1} - 2T_j^{n+1}}{\Delta x^2}$$

Rearranging the above equation, we obtain

$$(1 + 2A)T_j^{n+1} - AT_{j-1}^{n+1} - AT_{j+1}^{n+1} = T_j^n \tag{5.78}$$

for each interior node. For the boundary nodes, Equation 5.76 is still applicable. It is evident that matrix inversion is needed for obtaining the solution by the implicit method, since the temperature of node (j) at the time level $(n + 1)$ is coupled with those of the neighbouring nodes at the same time level. For the one dimensional situation considered here, the coefficient matrix multiplying the unknown temperatures on the left-hand side, has tridiagonal structure. This system, as before can be solved by the Thomas algorithm. Although the computation per time step is greater for the implicit method, the total computational time could still be less compared to the other schemes, since there are no restrictions on the size of the time step Δt for stability. The discretization error in approximating the term $\partial T/\partial t$ is $O(\Delta t)$, for the implicit and explicit schemes.

For the Crank-Nicolson scheme, the equations for the interior nodes are given by

$$(1 + 2A)T_j^{n+1} - AT_{j-1}^{n+1} - AT_{j+1}^{n+1} = (1 - 2A)T_j^n + AT_{j+1}^n + AT_{j-1}^n \quad \text{for } 2 \leq j \leq M \tag{5.79}$$

where $A = (\alpha\, \Delta t/2\Delta x^2)$. Obviously, the Crank-Nicolson technique also requires matrix inversion. This is the most expensive per time step, among the three methods. However, the time-discretization error for the Crank-Nicolson scheme is $O(\Delta t^2)$.

For multidimensional problems, the Alternating Direction Implicit (ADI) method is widely used (Anderson, *et al.*, 1984). This is quite similar to the line-by-line technique discussed for steady problems, with the difference that the iterative cycles of sweeping in different directions in the steady method are replaced by partial time step advancements corresponding to each direction. Here, we shall illustrate the ADI scheme for the two dimensional transient conduction problem in a rectangular geometry. Consider the equation

$$\frac{\partial T}{\partial t} = \alpha \left\{ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right\} \tag{5.80}$$

The discretization of this equation over a time step Δt between the nth and $(n + 1)$th time levels can be written in terms of two half-steps as

$$\left.\frac{\partial T}{\partial t}\right|^{n+1/2} = \alpha\left\{\frac{\partial^2 T^{n+1/2}}{\partial x^2} + \frac{\partial^2 T^{n}}{\partial y^2}\right\} \tag{5.81}$$

$$\left.\frac{\partial T}{\partial t}\right|^{n+1} = \alpha\left\{\frac{\partial^2 T^{n+1/2}}{\partial x^2} + \frac{\partial^2 T^{n+1}}{\partial y^2}\right\} \tag{5.82}$$

In the first half time step, the x-direction derivative term is evaluated at the current time level $(n + 1/2)$, while the y-derivative is not advanced in time. For the second half time step, the y-derivative is updated and the x-derivative term is left as it is. Substituting the corresponding difference expressions for all the terms, the discretized forms of Equations (5.81 and 5.82) are given by

$$\frac{T_{i,j}^{n+1/2} - T_{i,j}^{n}}{(\Delta t/2)} = \alpha \left\{\frac{T_{i+1,j}^{n+1/2} + T_{i-1,j}^{n+1/2} - 2T_{i,j}^{n+1/2}}{\Delta x^2} + \frac{T_{i,j+1}^{n} + T_{i,j-1}^{n} - 2T_{i,j}^{n}}{\Delta y^2}\right\} \tag{5.83}$$

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n+1/2}}{(\Delta t/2)} = \alpha \left\{\frac{T_{i+1,j}^{n+1/2} + T_{i-1,j}^{n+1/2} - 2T_{i,j}^{n+1/2}}{\Delta x^2} + \frac{T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1} - 2T_{i,j}^{n+1}}{\Delta y^2}\right\} \tag{5.84}$$

Rearranging these equations, we obtain

$$\begin{aligned}(1 + 2A)T_{i,j}^{n+1/2} - AT_{i+1,j}^{n+1/2} - AT_{i-1,j}^{n+1/2} &= (1 - 2B)T_{i,j}^{n} \\ &+ BT_{i,j+1}^{n} + BT_{i,j-1}^{n}\end{aligned} \tag{5.85}$$

$$\begin{aligned}(1 + 2B)T_{i,j}^{n+1} - BT_{i,j+1}^{n+1} - BT_{i,j-1}^{n+1} &= (1 - 2A)T_{i,j}^{n+1/2} \\ &+ AT_{i+1,j}^{n+1/2} + AT_{i-1,j}^{n+1/2}\end{aligned} \tag{5.86}$$

where $A = (\alpha\ \Delta t/2\Delta x^2)$ and $B = (\alpha\ \Delta t/2\Delta y^2)$. Equation 5.85 can be applied (along with the discretized x-direction boundary conditions) to solve the nodal equations for each j = constant grid line, by the tridiagonal matrix solution procedure. Similarly, Equation 5.86 can be used for updating the temperature field during the second half time step, by the application of tridiagonal matrix procedure along each i = constant grid line.

For three dimensional problems, the time step Δt is split into three $(\Delta t/3)$ steps and over each of the partial time steps, terms corresponding to a particular direction are treated implicitly, with others evaluated explicitly. By the

application of line-by-line method in each direction, the temperature field is updated repetitively using the highly efficient Thomas algorithm. Thus, ADI provides an inexpensive way of calculating transient temperature fields in multidimensional problems.

5.2 Finite Difference Applications in Convective Heat Transfer

Convection of heat by a fluid flowing over a heated surface is commonly encountered in several heat transfer applications. For a complete solution of the convective heat transfer problem, most often it is required to solve the flow and energy balance equations together. However, here we shall only discuss the handling of the energy balance equation by the finite difference method. The flow solution algorithms are discussed in detail in the following chapters.

The main feature which needs to be incorporated for convective problems (in addition to those already discussed for the conduction mode of heat transfer) is that of providing the appropriate *upwind bias* to the convective derivatives. The need for upwinding in convective problems can be seen from the following arguments. Consider the steady one dimensional convection-diffusion equation

$$u \frac{dT}{dx} = \alpha \frac{d^2T}{dx^2} \tag{5.87}$$

with constant flow velocity u and diffusion coefficient α. If central difference expressions are substituted for the derivatives in Equation 5.87, the following discretized equation is obtained.

$$T_{i+1} \left(1 - \frac{\mathrm{Pe}_c}{2}\right) + T_{i-1} \left(1 + \frac{\mathrm{Pe}_c}{2}\right) = 2T_i \tag{5.88}$$

where Pe_c , the cell Peclet number, is defined as $\mathrm{Pe}_c = (u\ \Delta x/\alpha)$.

For $\mathrm{Pe}_c > 2$, the coefficient of T_{i+1} on the left hand side of Equation 5.88 is negative, while the coefficient for the T_i term on the right hand side is positive. As a consequence, an increase in the value of T_{i+1} will result in a decrease in the value of T_i . But, from physical considerations it is obvious that increasing the temperature of a particular node should lead to only an increase in the temperature of the neighbouring node, whether heat is transmitted by conduction or convection. It can be verified that application of Equation 5.88 to all the nodes in the domain introduces non-physical oscillations in the predicted temperature solution for $\mathrm{Pe}_c > 2$, due to the occurrence of coefficients with opposite sign in Equation 5.88. Similarly, if u is negative, it can be shown that numerical instabilities arise for $\mathrm{Pe}_c < -2$. Thus, the numerical scheme based on central differences is unstable unless the cell Peclet number satisfies the condition $|\mathrm{Pe}_c| < 2$.

When the flow velocity is very high, the central difference based discretization may place a severe constraint on the size of the spatial step Δx. The alter-

native for removing the cell size restriction is the upwind difference approach, in which the temperature derivative of the convective term is discretized giving more bias to the upstream neighbours of the node under consideration. Considering one upstream node, the first order upwind formula for the temperature gradient (with a discretization error of order $O(\Delta x)$) can be obtained as

$$\frac{dT}{dx} = \frac{T_i - T_{i-1}}{\Delta x} \quad \text{for } u > 0$$
$$\frac{dT}{dx} = \frac{T_{i+1} - T_i}{\Delta x} \quad \text{for } u < 0 \tag{5.89}$$

With first order upwinding, the discretized form of Equation 5.87 can be written in the general form (Roache, 1998):

$$C_p T_{i+1} + C_0 T_i + C_m T_{i-1} = 0 \tag{5.90}$$

$$\begin{aligned} \text{where} \quad C_p &= 1 \text{ for } u > 0 \text{ and } C_p = 1 + |\text{Pe}| \text{ for } u < 0 \\ C_0 &= -(2 + |\text{Pe}_c|) \text{ for } u > 0 \text{ and } u < 0 \\ \text{and} \quad C_m &= 1 + \text{Pe}_c \text{ for } u > 0 \text{ and } C_m = 1 \text{ for } u < 0. \end{aligned} \tag{5.91}$$

Equation 5.90 can be used to generate the discretized equations for all the interior nodes $i = 2 \ldots N$. For the boundary nodes 1 and $(N+1)$, the boundary conditions have to be incorporated, as discussed earlier for heat conduction. The above system of equations are stable for any step size, although the discretization error may be quite high for large Δx.

For greater accuracy, higher order upwind schemes could be employed for handling the convective term. For instance, second order accuracy can be achieved by taking two upstream nodes with respect to the node under consideration, during the discretization process. If $(i-1)$ and $(i-2)$ are the two immediate upstream neighbours of node (i), the temperature derivative can be approximated in terms of these nodal temperatures as

$$\frac{dT}{dx} = \frac{3T_i - 4T_{i-1} + T_{i-2}}{2\Delta x} \text{ for } u > 0 \tag{5.92}$$

One possible way of constructing nodal equations with second order upwind bias is to substitute the Equation 5.92 into 5.87, with central difference employed for the conduction term. Upwind schemes can also be constructed by involving some downstream and upstream neighbours of the node i (say, $i-m, i-m+1, ..., i-1, i+1, i+2, ...i+n$, where $m > n$). But these higher order schemes are normally used only in conjunction with explicit transient simulations.

In the context of control volume approach, the QUICK scheme of Leonard (1979) is fairly popular (also see Chapter 7, for a variety of improved upwind

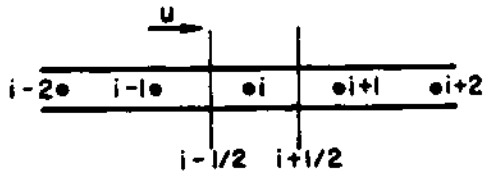


Figure 5.13: Application of QUICK Scheme

schemes). Consider the following convective-diffusion equation in conservative form

$$\frac{\partial(uT)}{\partial x} = \frac{\partial}{\partial x}\left\{\alpha\,\frac{\partial T}{\partial x}\right\} \tag{5.93}$$

where u and α may be functions of x. Integrating the above equation with respect to x over a control volume (Figure 5.13) around the node i, we obtain

$$(uT)_{i+1/2} - (uT)_{i-1/2} - \left[\alpha\,\frac{\partial T}{\partial x}\right]_{i+1/2} + \left[\alpha\,\frac{\partial T}{\partial x}\right]_{i-1/2} = 0 \tag{5.94}$$

Now, the diffusion terms can be approximated by central differences as usual, resulting in the expressions

$$\left[\alpha\,\frac{\partial T}{\partial x}\right]_{i+1/2} = \alpha_{i+1/2}\,\frac{T_{i+1}-T_i}{\Delta x};\quad \left[\alpha\,\frac{\partial T}{\partial x}\right]_{i-1/2} = \alpha_{i-1/2}\,\frac{T_i-T_{i-1}}{\Delta x}; \tag{5.95}$$

where $\alpha_{i+1/2}$ and $\alpha_{i-1/2}$ are the equivalent diffusivities at the cell faces.

Convective terms can be differenced with second order upwind bias as

$$\begin{aligned}(\phi_{i+1/2}) &= 0.5\,(\phi_i+\phi_{i+1}) - \beta(\phi_{i-1} - 2\phi_i + \phi_{i+1})\\ \text{and, } (\phi_{i-1/2}) &= 0.5\,(\phi_i+\phi_{i-1}) - \beta(\phi_{i-2} - 2\phi_{i-1} + \phi_i)\end{aligned} \tag{5.96}$$

where $\phi = (uT)$ and β is a constant which is suitably selected. The QUICK scheme has $\beta = 1/8$, while the value $\beta = 0$ corresponds to pure central differencing. The second order accurate expressions given above, can be derived through Taylor series expansions around the locations $i-1/2$ and $i+1/2$. Substituting for the convective and diffusive fluxes from Equations 5.95 and 5.96 into the discretized energy balance Equation 5.94, nodal equations with second order upwinding bias can be obtained for all the interior nodes.

Apart from upwinding, there are a few other aspects which require close attention in convective heat transfer problems. For example, the prescription of a proper downstream boundary condition is a non-trivial matter. Moreover, in some problems, the diffusion of heat in the flow direction can be neglected

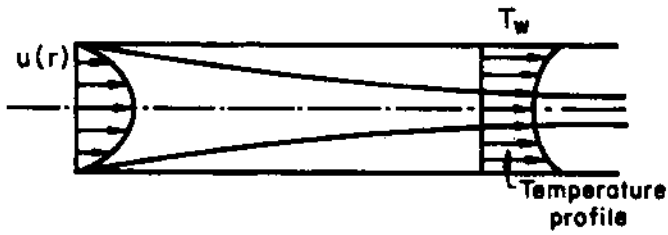


Figure 5.14: Thermally Developing Flow Inside a Circular Pipe

in comparison with the convection of heat in the same direction. In such cases, the energy balance equation becomes a parabolic partial differential equation, which can be solved by marching in the flow direction. In fact, advancement in the flow direction can be treated analogous to advancement in time, and the explicit, implicit and Crank-Nicolson schemes can be applied. The following examples illustrate these special aspects of convective heat transfer problems.

Example 5.8. Thermally developing flow inside a circular pipe

Consider the steady, hydrodynamically fully-developed, thermally developing laminar flow inside a circular pipe (Figure 5.14). These conditions may be approximately met during the steady flow of a high Prandtl number fluid such as oil inside a long pipe. The dimensionless energy balance equation for this situation is of the form

$$\text{Pe}\ u(r)\ \frac{\partial T}{\partial z} = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\ \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \tag{5.97}$$

where Pe is the Peclet number based on the radius of the pipe and (r, z) are respectively the radial and the axial coordinates. The dimensionless flow velocity has a parabolic profile of the form

$$u(r) = 1.5\ (1 - r^2) \tag{5.98}$$

Let us assume the following boundary conditions for the problem. At the inlet of the pipe ($z = 0$), the temperature profile $T_0(r)$ is known. A uniform temperature T_w is maintained at the pipe wall ($r = 1$). An important parameter of the problem which needs to be predicted by the numerical simulation is the thermal development length, i.e. the distance from the inlet where the fluid temperature becomes nearly equal to the wall temperature. Mathematically, an infinite distance would be required for the temperature of the fluid to approach T_w, but a practical limit can be defined in terms of a prescribed deviation (say, 0.1%) from the wall temperature. Specifically, one can look for convergence in the bulk mean temperature $T_b(z)$, defined as

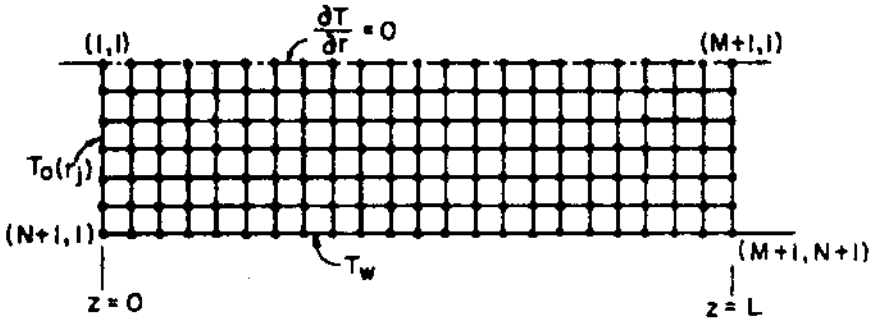


Figure 5.15: Computational Domain for Developing Flow in a Pipe

$$T_b(z) = \frac{\int_0^R 2\pi r u(r) T(r,z) dr}{\int_0^R 2\pi r u(r) dr}$$

For computational purposes, an outer boundary can be selected at $z = L$, where $L \gg 1$. The appropriate value of L needs to be determined by numerical experimentation.

Owing to axisymmetry, the half- plane defined by $0 \leq r \leq 1; 0 \leq z \leq L$, can be taken as the computational domain (Figure 5.15). Using the first order upwind scheme for the convective term

$$u(r)\frac{\partial T}{\partial z}$$

and central differences for the conduction terms, the nodal equation at a grid point (i,j) is given by

$$\begin{aligned} \text{Pe}\, u(r_j)\, \frac{T_{i,j} - T_{i-1,j}}{\triangle z} &- \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\triangle r^2} - \frac{1}{r_j}\frac{T_{i,j+1} - T_{i,j-1}}{2\triangle r} \\ &- \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\triangle z^2} = 0 \end{aligned} \tag{5.99}$$

for $2 \leq j \leq N$ and $2 \leq i \leq M$, where N and M are the total number of spatial steps in the radial and axial directions respectively. The nodal equations for the inlet boundary are given by

$$T_{i,j} = T_0(r_j) \tag{5.100}$$

For the pipe wall, we obtain

$$T_{i,N+1} = T_w \tag{5.101}$$

On the axis, axisymmetry leads to

$$\frac{\partial T}{\partial r} = 0$$

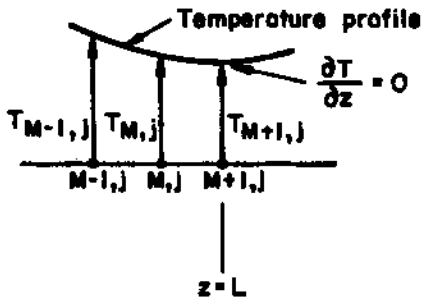


Figure 5.16: Effect of Vanishing First Derivative Condition at the Exit Boundary

Resolving the term $1/r\ \partial T/\partial r$ in the governing equation by L'Hospital rule, the above boundary condition can be incorporated with the help of the image point method, to yield the following nodal equation on the axis

$$\text{Pe}\ u(0)\ \frac{T_{i,1} - T_{i-1,1}}{\triangle z} - \frac{4T_{i,2} - 4T_{i,1}}{\triangle r^2} - \frac{T_{i+1,1} + T_{i-1,1} - 2T_{i-1,1}}{\triangle z^2} = 0 \quad (5.102)$$

Now, we need to impose some suitable boundary condition on the downstream boundary. One possibility is to assume

$$T_{M+1,j} = T_w \quad (5.103)$$

since it is anticipated that the fluid temperature should asymptotically reach the limiting value of T_w, for large length L. While this condition will yield reasonable results for very large L, its performance at moderate values of L is poor. For minimizing the computational effort, a condition which can represent the asymptotic behaviour well even at moderate length, is preferable. Another possibility is to assume the derivative in the z-direction as equal to zero at the downstream boundary, since the fully developed profile is invariant in the axial direction by definition. Therefore, one may prescribe the condition

$$\frac{\partial T}{\partial z} = 0 \text{ at } z = L \quad (5.104)$$

However, making the first derivative zero could result in a mirror reflection-type variation (Figure 5.16) with a local extremum at $z = L$, for moderate values of L. Obviously, such a profile is meaningless in the context of the thermally developing solution.

Let us examine the consequences of prescribing zero second derivative in the axial direction, at $z = L$. This condition, also known as the *continuity* condition, ensures a smooth variation of the temperature profile at the outflow boundary. Also, it precludes the reflection-type variation and results in a monotonically

decaying solution (Figure 5.16). Therefore, the second derivative condition is the best choice among the three discussed above and it can be incorporated as

$$\frac{\partial^2 T}{\partial z^2} = 0 \text{ at } z = L \tag{5.105}$$

In discretized form, this leads to the nodal equation

$$\frac{T_{M-1,j} - 2T_{M,j} + T_{M+1,j}}{\Delta z^2} = 0 \tag{5.106}$$

An alternative approach to the outflow boundary condition based on the proposal of Orlanski (1976) is discussed in Chapter 7.

Equations 5.99, 5.100-5.102 and 5.106 can now be solved, to obtain the nodal temperaures within the computational domain. A check on the correctness of the computed solution is to verify that $T_{M+1,j} \to T_w$ at the exit boundary, for various values of the channel length L.

In the above example, the temperature derivative in the convective term was approximated using first order upwind scheme. For greater accuracy, higher order upwinding can be implemented, for example, using the QUICK technique. One small difficulty in employing the higher order upwind expressions is that they cannot be directly used for the set points adjacent to the inlet section, since these nodes do not have the adequate number of upstream neighbours. However, this difficulty can be overcome easily, by using a very small value of Δz for the first few axial steps, along with first order upwinding. Due to small Δz, the discretization error can be kept low even with first order upwinding.

In the above example, the flow velocity was assumed to be known and the finite difference technique was applied only to the heat transfer equation. However, in most of the convective heat transfer problems, it is also required to solve for the velocity profiles, before obtaining the temperature solution. For general multi-dimensional flow problems, finite difference based flow solvers such as MAC and SIMPLE are used to obtain the velocity and pressure fields (Chapter 7). But for the special case of boundary-layer flows, even the techniques discussed so far in this chapter can be directly applied. The coupled solution of flow and heat transfer for the boundary-layer over a flat plate is discussed in Example 5.9.

Example 5.9. Heat transfer in the boundary-layer over a flat plate

Consider the steady, two-dimensional incompressible laminar flow over a flat plate. Far stream velocity of the flow is uniform and parallel to the plate. The flow and energy balance equations for this simplified situation are given by (Schlichting, 2000):

$$u\,\frac{\partial u}{\partial x} + v\,\frac{\partial u}{\partial y} = \nu\,\left[\frac{\partial^2 u}{\partial y^2}\right] \tag{5.107}$$

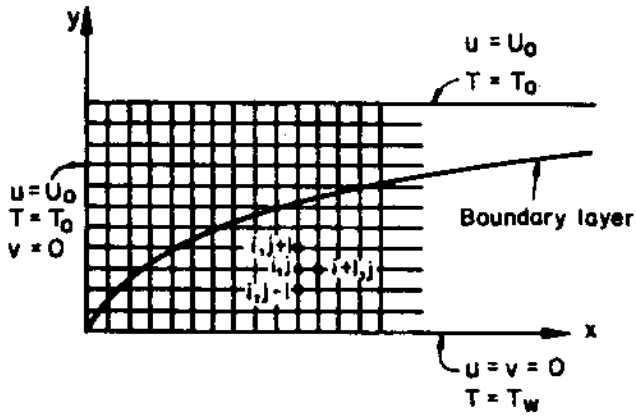


Figure 5.17: Boundary Layer Flow Over a Flat Plate

$$u\,\frac{\partial T}{\partial x} + v\,\frac{\partial T}{\partial y} = \alpha\,\left[\frac{\partial^2 T}{\partial y^2}\right] \tag{5.108}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.109}$$

where ν and α are respectively the momentum and thermal diffusivities. Let U_0 and T_0 be the velocity and temperature of the approaching flow and T_w the wall temperature maintained at a constant value over the whole length of the plate (Figure 5.17). The boundary conditions for the problem are

$$\begin{aligned} u = 0, v = 0 \text{ and } T = T_w \text{ at } y &= 0 \quad \text{(plate surface)} \\ u = U_0, v = 0 \text{ and } T = T_0 \text{ at } x &= 0 \quad \text{(approach flow conditions)} \\ u \to U_0 \text{ and } T \to T_0 \text{ as } y &\to \infty \quad \text{(far stream conditions)} \end{aligned} \tag{5.110}$$

Evidently, the far stream conditions of uniform velocity and fluid temperature have to be applied at some finite distance H from the plate, during the numerical implementation. Since it is well known that for laminar flows the momentum boundary layer thickness $\delta = O(L/\mathrm{Re}^{-1/2})$ and the thermal boundary layer thickness $\delta_t = O(L/\mathrm{Pe}^{-1/2})$, the distance H can be selected as

$$H = \max\left\{C(L/\mathrm{Re}^{-1/2}),\ C(L/\mathrm{Pe}^{-1/2})\right\} \tag{5.111}$$

where the Reynolds number Re$= U_0 L/\nu$ and the Peclet number Pe = Re$\times$Pr = Re (ν/α). In the above expressions, L is the length of the plate and C a constant of $O(1)$. In most of the boundary layer flow problems, it is sufficient to assume a value of $C \approx 10$ ($C = 7$ or 8 would often suffice).

Equations 5.107-5.109 are a parabolic system of partial differential equations, which admit a marching solution procedure for the x direction. In fact, the explicit, implicit and semi-implicit schemes can be applied for this purpose, as discussed earlier in the context of transient conduction problems. Let us consider the prediction of the velocity and temperature profiles for the location x_{i+1}, by marching over the spatial step Δx between x_i and x_{i+1}. Applying an explicit scheme, the nodal equations for a point (i, j) are given by

$$u_{i,j}\,\frac{u_{i+1,j}-u_{i,j}}{\Delta x}+v_{i,j}\,\frac{u_{i,j+1}-u_{i,j-1}}{2\Delta y}-\nu\frac{u_{i,j+1}+u_{i,j-1}-2u_{i,j}}{\Delta y^2}=0 \quad (5.112)$$

$$u_{i,j}\,\frac{T_{i+1,j}-T_{i,j}}{\Delta x}+v_{i,j}\,\frac{T_{i,j+1}-T_{i,j-1}}{2\Delta y}-\alpha\frac{T_{i,j+1}+T_{i,j-1}-2T_{i,j}}{\Delta y^2}=0 \quad (5.113)$$

$$\frac{u_{i+1,j}-u_{i,j}}{\Delta x}+\frac{v_{i+1,j+1}-v_{i+1,j-1}}{2\Delta y}=0 \quad (5.114)$$

Note that the y-component of velocity v in Equation 5.109 has been differenced implicitly, leading to Equation 5.114. Since $\partial v/\partial x$ does not appear in any of the governing equations, the v-component can not be advanced explicitly in x-direction. Therefore, the nodal values of v are solved from the continuity equation, by marching in the y-direction starting from the wall $y = 0$. Equations 5.112 and 5.113 can be rearranged in the form

$$\begin{aligned} u_{i+1,j} = u_{i,j} &- \frac{v_{i,j}\,\Delta x}{u_{i,j}\,2\Delta y}\,(u_{i,j+1}-u_{i,j-1}) \\ &+\frac{\nu\,\Delta x}{u_{i,j}\,\Delta y^2}\,(u_{i,j+1}+u_{i,j-1}-2u_{i,j}) \end{aligned} \quad (5.115)$$

$$\begin{aligned} T_{i+1,j} = T_{i,j} &- \frac{v_{i,j}\,\Delta x}{u_{i,j}\,2\Delta y}\,(T_{i,j+1}-T_{i,j-1}) \\ &+\frac{\alpha\,\Delta x}{u_{i,j}\,\Delta y^2}\,(T_{i,j+1}+T_{i,j-1}-2T_{i,j}) \end{aligned} \quad (5.116)$$

Now, the nodal values of u and T at the section x_{i+1} can be evaluated using Equations 5.115 and 5.116, for $2 \le j \le M$. The boundary conditions for the problem are implemented in a straightforward manner as follows:

$$\begin{aligned} u_{1,j} &= U_0,\ v_{1,j}=0 \text{ and } T_{1,j}=T_0, \text{ for } 2\le j\le M; \\ u_{i,1} &= 0,\ v_{i,1}=0 \text{ and } T_{i,1}=T_w, \text{ for } 1\le i\le N+1; \\ u_{i,M+1} &= U_0, \text{ and } T_{i,M+1}=T_0, \text{ for } 1\le i\le N+1. \end{aligned} \quad (5.117)$$

Instead of the explicit method discussed above, it is also possible to employ the implicit scheme for discretizing the Equations 5.107 and 5.108. However, the resulting algebraic equations will be non-linear, if the velocity coefficients u and v of the convective terms are also evaluated at the section x_{i+1}. These non-linear equations require an iterative solution method, for example the Newton-Raphson scheme (Chapter 2). On the other hand, the velocity coefficients can be evaluated at x_i and weakly implicit solutions can be obtained.

5.3 Closure

In this chapter, various applications of the finite difference method to conduction and convection heat transfer problems have been discussed. It is evident from the examples presented that the finite difference formulation is very simple to apply. A wide variety of heat transfer and flow problems in regular geometries can be solved by this technique.

References

Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer,* Hemisphere Publishing Corporation, New York, 1984.

Carnahan B., Luther H.A. and Wilkes J.O., *Applied Numerical Methods,* John Wiley Sons, New York, 1969.

Conte, S. D. and de Boor, C., *Elementary Numerical Analysis: an Algorithmic Approach,* Mc Graw Hill, NeW York, 1981.

Fadeeva, V. N., *Computational Methods of Linear Algebra,* Dover Publications, New York, 1959.

Fletcher C. A. J., *Computational Techniques for Fluid Dynamics,* Vol I and II, Sringer Verlog, Berlin, 1988.

Holman, J.P., *Heat Transfer,* 8th edition, McGraw-Hill, New York, 1997.

Jaluria, Y. and Torrance, K. E., *Computational Heat Transfer,* Hemisphere Publishing Corporation, New York, 1986.

Kays, W.M. and Crawford, M.E., *Convective Heat and Mass Transfer,* McGraw-Hill, New York, 1980 (third edition: 1993).

Leonard, B. P., A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, Comput. Methods Appl. Mech. Eng., Vol. 19, pp. 59- 98, 1979.

Orlanski, I., A Simple Boundary Condition for Unbounded Flows, J. Computational Phys., Vol. 21, pp 251-269, 1976.

Ozisik, M.N., *Heat Conduction,* John Wiley, New York, 1980; second edition, 1993.

Ozisik, M.N., *Finite Difference Methods in Heat Transfer,* CRC Press, Boca Raton, USA, 1994.

Patankar, S. V., *Numerical Heat Transfer and Fluid Flow,* Hemisphere Publishing Co., New York, 1980.

Roache, P.J., *Fundamentals of Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico, 1998.

Schlichting, H., *Boundary-layer Theory*, McGraw-Hill, New York, 1979 (8th edition: 2000).

Smith, G. D., *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Clarendon Press, Oxford, 1978.

Problems

1. A practical approach towards estimating the order of accuracy of a discretization procedure is the following. For a simple geometry, for example the one in Figure 5.2, it is possible to develop the analytical solution, the numerical solution and hence the RMS error ϵ defined as

$$\epsilon^2 = \frac{1}{L^2}\int_0^L \int_0^L (T_{\text{numerical}} - T_{\text{analytical}})^2 dxdy$$

The error should decrease with the grid spacing (or the number of nodes) linearly for a first order scheme, and quadratically for a second order scheme. This trend is most conveniently seen on a log-log graph. Using this approach, confirm the orders of discretization errors quoted in this chapter.

2. A long two dimensional parallel plate channel experiences steady heat transfer by conduction, with boundary conditions as shown in Figure 5.18. Set up the nodal equations for this domain, after taking full advantage of the symmetry present in the problem.

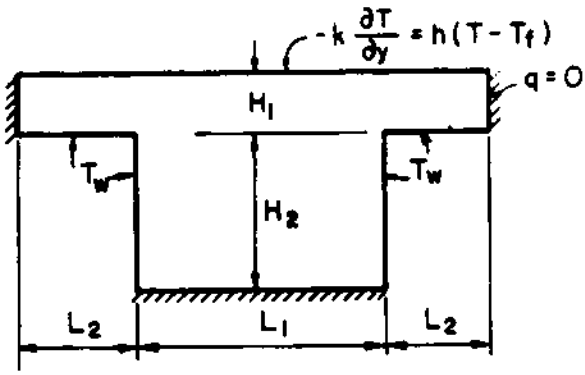


Figure 5.18: Heat Conduction in a Two Dimensional Channel

3. A long cylinder with an initial temperature of T_i is immersed into a hot oil bath at temperature T_b. Assuming that the cylinder surface immediately reaches the bath temperature, formulate the nodal equations for temperature within the cylinder with second order discretization error.

4. A sphere, initially at a temperature of T_i, loses heat by convection to a fluid at temperature T_f. Assuming that the heat transfer coefficient h at any point is

of the form

$$h = -ax^{0.5} + b$$

where x is the distance from the forward stagnation point (along the spherical surface), formulate the nodal equations for temperature variation inside the sphere by finite differences.

5. A fin has a conical shape, with circular cross section. The thermal conductivity of the fin material varies as a linear function of the temperature. Apply the control volume method of setting up the nodal equations, assuming steady heat conduction within the fin. Formulate the more general problem of an optimum fin that maximizes heat transfer subjected to a constraint on its weight. The formulation can be in terms of a variational prinicple, from which the appropriate differential equation can be extracted. Solve the differential equation (with boundary conditions) by the control volume method. Assume suitable dimensional parameters for developing the computer program. The fin material may be taken as aluminium, and the surrounding fluid as air.

6. For hydrodynamically developed, thermally developing pipe flow of Example 5.9, develop the finite difference expressions by using the QUICK scheme. Assume that the flow is steady, laminar and incompressible. Develop a computer program to solve for temperature distribution. Select the geometric dimensions of the tube appropriately. Plot the variation of the bulk mean temperature of the fluid with distance for Peclet numbers upto 1000. Repeat the problem for constant wall temperature and constant heat flux boundary conditions. The Nusselt number values for thermally fully developed flow have been derived by Kays and Crawford (1993).

6
Solution of Viscous Incompressible Flows by the Stream Function-Vorticity Formulation

The stream function-vorticity formulation is a powerful approach for solving the incompressible Navier-Stokes equations in two-dimensional and axisymmetric geometries. It has been used in several applications such as the modeling of turbulent flows, analysis of laminar to turbulent flow transition and studies on free and mixed convection. One of the major attractive features of this method is that it does not involve the solution of the pressure field. For incompressible flows, pressure is a difficult variable to handle due to the fact that there is no direct equation for its prediction. Most of the flow solvers based on the velocity-pressure formulation employ the continuity equation as a constraint to derive the correct pressure field through an elaborate, iterative solution procedure (Chapter 7). In the stream function-vorticity method, pressure is completely eliminated by cross-differentiation of the momentum equations. Another advantage of this formulation is that the governing equations for stream function and vorticity constitute a second order elliptic system whose properties are well understood. Therefore, it is easy to construct numerical schemes for the solution of stream function and vorticity, which are highly robust and stable.

However, the stream function-vorticity method has its share of drawbacks as well, the main drawback being that it cannot be applied to three dimensional flow problems. Secondly, the distribution of pressure, which is eliminated during the solution stage, is required subsequently at least on the solid boundary, for calculating the drag forces. Recovering pressure from the computed stream

function and vorticity solutions involves additional calculations. One other minor disadvantage of the technique is that it requires the estimation of vorticity at the boundaries, for computing the flow solution in the whole domain. Since there is no direct boundary condition for the surface vorticity, it has to be computed numerically from the available stream function field and the accuracy of overall solution procedure critically depends on how this boundary condition is handled. Despite the above-mentioned drawbacks, it is still widely used on account of its robustness during numerical simulation.

Examples of flow prediction, in Cartesian, cylindrical and spherical coordinates using the stream function-vorticity formulation are described in the following sections.

6.1 Two Dimensional Incompressible Viscous Flow in a Lid-driven Cavity

For two dimensional steady incompressible flow in a rectangular geometry, the governing equations for stream function and vorticity are the following (Chapter 1; also see Currie, 1993):

(i) Definition of vorticity:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \tag{6.1}$$

where ψ is the stream function and ω is the vorticity.

(ii) Convective-diffusive transport equation for vorticity

$$u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu \left\{ \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right\} \tag{6.2}$$

where the velocity components u and v are given by

$$u = \frac{\partial \psi}{\partial y} \text{ and } v = -\frac{\partial \psi}{\partial x} \tag{6.3}$$

The boundary conditions for the lid-driven cavity problem (Figure 6.1) are:

$$\begin{aligned} u &= 0, v = 0 \text{ at } x = 0 \; ; \; u = 0, v = 0 \text{ at } y = 0 \; ; \\ u &= 0, v = 0 \text{ at } x = L \; ; \quad u = U_o, v = 0 \text{ at } y = H \end{aligned} \tag{6.4}$$

where U_0 is the velocity of the lid. All the four boundary conditions given in Equation 6.4 arise due to no-slip at the respective solid boundaries.

The velocity boundary conditions can be converted in terms of stream function and vorticity as follows:

On the walls $x = 0$ and $x = L$, $u = \partial\psi/\partial y = 0$. Therefore, ψ has a constant value over these surfaces. Similarly, one can show that ψ is constant on $y = 0$ and $y = H$, since $v = 0$ along the bottom and top walls. From

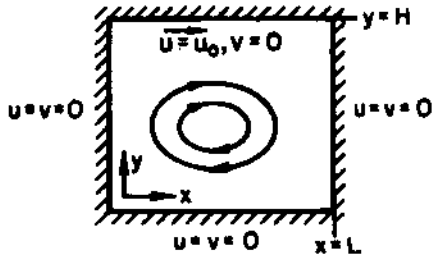


Figure 6.1: Flow Inside a Lid-driven Cavity

the geometry of the problem, it is evident that the same stream line passes through all the four surfaces, $x = 0; x = L; y = 0$ and $y = H$. Otherwise, there will be a jump in the stream function value at the corners and this violates the condition $\psi =$ constant on each of the boundaries. The magnitude of the stream function on the boundary streamline can be conveniently set as zero. In fact, any constant value other than zero will equally serve the purpose. Thus, the boundary conditions for stream function are:

$$\begin{aligned} \psi &= 0 \text{ on } x = 0, 0 \leq y \leq H \; ; \psi = 0, \text{ on } y = 0, 0 \leq x \leq L \; ; \\ \psi &= 0 \text{ on } x = L, 0 \leq y \leq H \; ; \psi = 0, \text{ on } y = H, 0 \leq x \leq L \end{aligned} \tag{6.5}$$

While deriving the above equations, we have used only one condition, i.e. normal velocity $= 0$ for each of the solid surfaces. Now, the boundary conditions for vorticity at the walls have to be derived from the tangential velocity conditions which have not been used so far. The procedure of obtaining the surface vorticity at the walls is described below.

Equation 6.1 can be simplified on the cavity walls $x = 0$ and $x = L$ as:

$$\omega = -\frac{\partial^2 \psi}{\partial x^2} \tag{6.6}$$

for ψ does not vary with y on these surfaces. Similarly, on the boundaries $y = 0$ and $y = H$, it is easily verified that

$$\omega = -\frac{\partial^2 \psi}{\partial y^2} \tag{6.7}$$

The condition of zero tangential velocity on the side walls leads to

$$v = \frac{\partial \psi}{\partial x} = 0 \text{ on } x = 0 \text{ and } x = L \tag{6.8}$$

Thus, wall vorticity on these sides must be evaluated using Equation 6.6, subject to the constraint of Equation 6.8. For the bottom and the top surfaces, the tangential velocity conditions are given by

$$u = \frac{\partial \psi}{\partial y} = 0 \text{ on } y = 0 \tag{6.9}$$

$$\text{and } u = \frac{\partial \psi}{\partial y} = U_0 \text{ on } y = H \tag{6.10}$$

The wall-vorticity on these boundaries can be obtained from Equation 6.7, with one of the above conditions used as a constraint.

We now briefly describe the implementation of the ideas discussed so far in a computational approach. Consider a numerical grid as shown in Figures 6.2(a-b) and apply the finite difference discretization to replace all derivatives in Equations 6.1 and 6.2 with the corresponding central difference expressions. This leads to

$$\frac{\psi_{i+1,j} + \psi_{i-1,j} - 2\psi_{i,j}}{\Delta x^2} + \frac{\psi_{i,j+1} + \psi_{i,j-1} - 2\psi_{i,j}}{\Delta y^2} = -\omega_{i,j} \tag{6.11}$$

and

$$\begin{aligned} & u_{i,j}\left[\frac{\omega_{i+1,j} - \omega_{i-1,j}}{2\Delta x}\right] + v_{i,j}\left[\frac{\omega_{i,j+1} - \omega_{i,j-1}}{2\Delta y}\right] \\ & = \nu\left\{\frac{\omega_{i+1,j} + \omega_{i-1,j} - 2\omega_{i,j}}{\Delta x^2} + \frac{\omega_{i,j+1} + \omega_{i,j-1} - 2\omega_{i,j}}{\Delta y^2}\right\} \end{aligned} \tag{6.12}$$

$$\text{where} \quad u_{i,j} = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta y} \text{ and } v_{i,j} = -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x}$$

In the above expressions $\Delta x, \Delta y$ are spatial step sizes and i, j are nodal indices in the x and y directions respectively. The step sizes are assumed to be constant in each of these directions.

The discretized equations (6.11) and (6.12) can be rearranged as:

$$\psi_{i+1,j} + \psi_{i-1,j} - 2(1+\beta^2)\psi_{i,j} + \beta^2(\psi_{i,j+1} + \psi_{i,j-1}) = -\Delta x^2 \omega_{i,j} \tag{6.13}$$

and

$$\begin{aligned} & \omega_{i+1,j} + \omega_{i-1,j} - 2(1+\beta^2)\,\omega_{i,j} + \beta^2(\omega_{i,j+1} + \omega_{i,j-1}) \\ & -\frac{u_{i,j}\Delta x}{2\nu}\,(\omega_{i+1,j} - \omega_{i-1,j}) - \frac{v_{i,j}\Delta y}{2\nu}\,\beta^2\,(\omega_{i,j+1} - \omega_{i,j-1}) = 0 \end{aligned} \tag{6.14}$$

where $\beta = \Delta x/\Delta y$ and i, j are the indices of interior nodes such that $2 \le i \le imax - 1$ and $2 \le j \le jmax - 1$. Each of the above two equations represents a pentadiagonal matrix system. In general, these can be solved by any technique such as matrix inversion, line-by-line method, or point-by-point iteration. Since

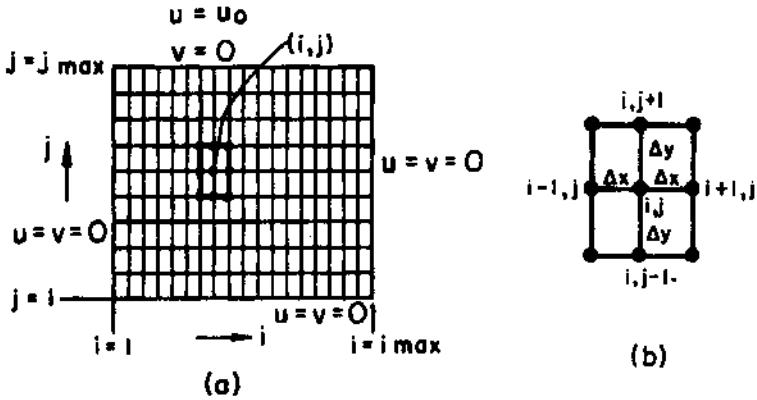


Figure 6.2: a. Numerical Grid for Lid-driven Cavity Flow, b. Cell Around i, j

the equation system is non-linear and involves iterative solution in any case, the line-by- line method is preferred over pentadiagonal matrix inversion. However, for external flows, the point-by-point method is normally recommended, atleast for the ψ-equation (6.13), since the magnitude of ψ approaches ∞ in the far-stream. Simultaneous handling of large values at far stream and small values near the object surface, can introduce numerical instabilities. Another important point to be kept in mind is that for the above numerical scheme to be stable, the grid Reynolds numbers appearing in Equation 6.14 are restricted by the conditions

$$\left|\frac{u_{i,j}\Delta x}{\nu}\right| \leq 2 \text{ and } \left|\frac{v_{i,j}\Delta y}{\nu}\right| \leq 2 \tag{6.15}$$

for each node (Chapters 4-5). Thus, when central difference expressions are used for all terms, the step lengths Δx and Δy are limited by the above criteria. The upwind scheme can be employed for eliminating the step size restrictions for stability and this aspect is discussed in Section 6.2.

Let us now turn our attention to the implementation of the boundary conditions. The conditions for stream function (Equation 6.5) are readily written as:

$$\begin{gathered}\psi_{1,j} = 0 \text{ for } 1 \leq j \leq j\text{max}; \ \psi_{i,1} = 0 \text{ for } 1 \leq i \leq i\text{max}; \\ \psi_{i\text{max},j} = 0 \text{ for } 1 \leq j \leq j\text{max}; \psi_{i,j\text{max}} = 0 \text{ for } 1 \leq i \leq i\text{max}.\end{gathered} \tag{6.16}$$

For evaluating wall-vorticity, let us first consider the surface $x = 0$ (left side wall). On this boundary, Equation 6.6 can be expressed in the following finite difference form

$$\omega_{1,j} = -\left\{\frac{\psi_{2,j} + \psi_{0,j} - 2\psi_{1,j}}{\Delta x^2}\right\} \tag{6.17}$$

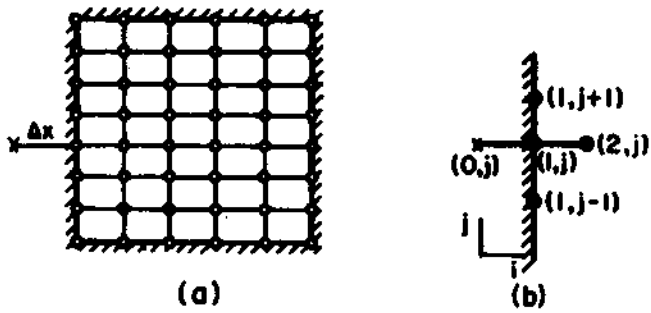


Figure 6.3: Use of Image Node in Surface Vorticity Estimation, • Actual node, × Image Node

where $(0, j)$ is an image point on the left side of $x = 0$ surface (Figure 6.3). Now, discretizing Equation 6.8 at the point $(1, j)$ leads to

$$\left.\frac{\partial \psi}{\partial x}\right|_{1,j} = \frac{\psi_{2,j} - \psi_{0,j}}{2\Delta x} = 0$$

Hence

$$\psi_{0,j} = \psi_{2,j} \tag{6.18}$$

Combining Equations 6.17 and 6.18, the wall-vorticity is obtained as

$$\omega_{1,j} = -\left\{\frac{2\psi_{2,j} - 2\psi_{1,j}}{\Delta x^2}\right\} = -\frac{2\psi_{2,j}}{\Delta x^2} \text{ for } 2 \leq j \leq j\text{max} - 1 \tag{6.19}$$

In a similar fashion, it can be shown that for the lower surface

$$\omega_{i,1} = -\frac{2\psi_{i,2}}{\Delta y^2}, \quad \text{for } \ 2 \leq i \leq i\text{max} - 1 \tag{6.20}$$

For the right side surface

$$\omega_{i\text{max},j} = -\frac{2\psi_{i\text{max}-1,j}}{\Delta x^2}, \text{ for } 2 \leq j \leq j\text{max} - 1 \tag{6.21}$$

In each of the above cases, the image point has been taken outside the cavity and its contribution to the wall-vorticity is evaluated using the corresponding tangential velocity condition. For the top wall, the above procedure is applied as follows:

$$\begin{aligned} \omega_{i,j\text{max}} &= -\frac{\partial^2 \psi}{\partial y^2} = -\frac{\psi_{i,j\text{max}-1} + \psi_{i,j\text{max}+1} - 2\psi_{i,j\text{max}}}{\Delta y^2} \\ &= -\frac{(\psi_{i,j\text{max}-1} + \psi_{i,j\text{max}+1})}{\Delta y^2} \end{aligned} \tag{6.22}$$

The tangential velocity condition for the top wall is discretized as

$$u = \frac{\partial \psi}{\partial y} = \frac{\psi_{i,j\max+1} - \psi_{i,j\max-1}}{2\Delta y} = U_0 \tag{6.23}$$

Eliminating the contribution of the image point $(i, j\max+1)$ between Equations 6.22 and 6.23, we obtain

$$\omega_{i,j\max} = -\frac{(2\psi_{i,j\max-1} + 2U_0\Delta y)}{\Delta y^2}, \text{ for } 2 \leq i \leq \text{imax} - 1 \tag{6.24}$$

The vorticity values for the corner nodes of the cavity need not be evaluated as they do not appear in the pentadiagonal matrix system arising from Equation 6.14.

Now, the discretized Equations 6.13, 6.14, 6.16, 6.19, 6.20, 6.21 and 6.24 have to be solved together for obtaining the stream function and vorticity values in the whole domain. This nonlinear algebraic system can be solved only by iterative schemes, assuming initial guess profiles for vorticity and stream function. The initial profiles may be chosen in any convenient manner. In the absence of any better guess, even zero values could be assumed, as long as the trivial solution is not a particular solution for the problem. With reference to the lid-driven cavity flow considered here, the initial profiles could be constructed by simple linear interpolation between the boundary values as:

$$\psi_{i,j} = 0 \text{ and } \omega_{i,j} = -\frac{2U_0}{\Delta y}\frac{j}{j\max} \text{ for all } (i,j) \tag{6.25}$$

Starting with the guessed profiles, the stream function and vorticity solutions are improved with the help of the nodal equations presented above. During any particular iteration, the velocity components appearing in the nonlinear convective terms of Equation 6.14 are evaluated using the stream function values of the previous iteration. The stream function solution, in turn, is updated using the latest vorticity values in Equation 6.13. After every stream function calculation, the vorticity at the walls is improved through Equations 6.19, 6.20, 6.21 and 6.24, while at interior nodes is calculated from Equation 6.14. For strongly nonlinear problems (for example, high Reynolds number flows), under relaxed point-by-point iterative schemes may be employed in the form:

$$\begin{aligned} \psi_{i,j}^{k+1} &= \psi_{i,j}^{k} + W_s(\psi_{i,j}^{*} - \psi_{i,j}^{k}) \\ \omega_{i,j}^{k+1} &= \omega_{i,j}^{k} + W_v(\omega_{i,j}^{*} - \omega_{i,j}^{k}) \end{aligned} \tag{6.26}$$

where the superscripts (k) and $(k+1)$ denote variables corresponding to the respective iterations and $(*)$ denotes the values predicted by the discretized equations after the kth iteration. If relaxation is not applied, $(*)$ quantities are set equal to the $(k+1)$th iteration values. In the above equation, W_s and W_v are respectively the relaxation parameters for the stream function and the vorticity

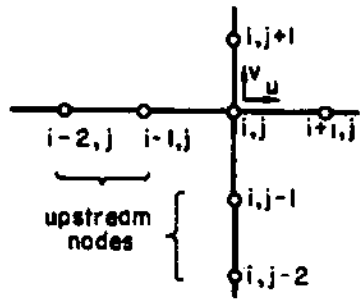


Figure 6.4: Nodes appearing in the Upwind Scheme

solutions. For under relaxation, these are chosen in the range $0 < W_s < 1$ and $0 < W_v < 1$. Normally, it may be necessary to under relax atleast the surface vorticity values. Over-relaxation with $W_s > 1$ or $W_v > 1$, may be applied for the interior nodes if the rate of convergence at the later stages of the calculation is slow.

6.2 Incorporation of Upwind Scheme

If central difference expressions are employed for the convective terms, the maximum spatial step sizes are limited by Equation 6.15. For high Reynolds number flows, the step sizes Δx and Δy may turn out to be very small and the numerical computation can become very expensive. In such cases, for removing the cell size restrictions, it is necessary to employ upwind schemes which are numerically stable even at large step sizes.

Upwind schemes provide a stronger bias to the upstream side of a node than the downstream side, while evaluating the convective contributions. In the simplest scheme of upwinding, the derivatives in convective terms are evaluated using only the upstream neighbour of a node (Figure 6.4).

Considering one upstream node in each direction, the convective terms of vorticity transport equation are evaluated as:

$$
\begin{aligned}
u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} &= u_{i,j}\left[\frac{\omega_{i,j} - \omega_{i-1,j}}{\Delta x}\right] + v_{i,j}\left[\frac{\omega_{i,j} - \omega_{i,j-1}}{\Delta y}\right], \\
&\quad \text{if } u_{i,j} > 0,\ v_{i,j} > 0; \\
&= u_{i,j}\left[\frac{\omega_{i+1,j} - \omega_{i,j}}{\Delta x}\right] + v_{i,j}\left[\frac{\omega_{i,j} - \omega_{i,j-1}}{\Delta y}\right], \\
&\quad \text{if } u_{i,j} < 0,\ v_{i,j} > 0; \\
&= u_{i,j}\left[\frac{\omega_{i,j} - \omega_{i-1,j}}{\Delta x}\right] + v_{i,j}\left[\frac{\omega_{i,j+1} - \omega_{i,j}}{\Delta y}\right], \\
&\quad \text{if } u_{i,j} > 0,\ v_{i,j} < 0;
\end{aligned}
$$

$$= \quad u_{i,j}\left[\frac{\omega_{i+1,j}-\omega_{i,j}}{\Delta x}\right] + v_{i,j}\left[\frac{\omega_{i,j+1}-\omega_{i,j}}{\Delta y}\right],$$
$$\text{if } u_{i,j} < 0,\ v_{i,j} < 0.$$

The convective terms can be discretized with upwind bias as shown above or they could be expressed in central difference form (Equation 6.14) if the stability criteria are not violated. A general equation for the discretized vorticity transport equation which includes the upwind and central difference schemes can be developed as follows:

$$\begin{aligned}
&\omega_{i+1,j} + \omega_{i-1,j} - 2(1+\beta^2)\omega_{i,j} + \beta^2(\omega_{i,j+1} + \omega_{i,j-1}) \\
&\quad \frac{-u_{i,j}\Delta x}{\nu}(\text{cxp}\times\omega_{i+1,j} + \text{cxo}\times\omega_{i,j} + \text{cxm}\times\omega_{i-1,j}) \\
&\quad \frac{-v_{i,j}\Delta y}{\nu}(\text{cyp}\times\omega_{i,j+1} + \text{cyo}\times\omega_{i,j} + \text{cym}\times\omega_{i,j-1}) = 0
\end{aligned} \tag{6.27}$$

where

$$\begin{aligned}
&\text{for } \left|\frac{u_{i,j}\Delta x}{\nu}\right| \le 2,\ \text{cxp} = 1/2,\ \text{cxo} = 0 \text{ and cxm} = -1/2 \\
&\qquad\qquad \text{(central difference)} \\
&\text{for } \left|\frac{u_{i,j}\Delta x}{\nu}\right| > 2,\ \text{cxp} = 0,\ \text{cxo} = 1 \text{ and cxm} = -1, \text{ if } u > 0 \\
&\qquad \text{cxp} = 1,\ \text{cxo} = -1 \text{ and cxm} = 0, \text{ if } u < 0 \\
&\text{for } \left|\frac{v_{i,j}\Delta y}{\nu}\right| \le 2,\ \text{cyp} = 1/2,\ \text{cyo} = 0 \text{ and cym} = -1/2 \\
&\qquad\qquad \text{(central difference)} \\
&\text{for } \left|\frac{v_{i,j}\Delta y}{\nu}\right| > 2,\ \text{cyp} = 0,\ \text{cyo} = 1 \text{ and cym} = -1, \text{ if } v > 0 \\
&\qquad \text{cyp} = 1,\ \text{cyo} = -1 \text{ and cym} = 0, \text{ if } v < 0
\end{aligned}$$

The discretization error in the above equations is first order in Δx and Δy with upwinding and second order if central difference expression is employed for the convective terms. It is possible to obtain higher order accuracy even with upwinding, if the derivatives $\partial\omega/\partial x$ and $\partial\omega/\partial y$ in the convective terms are evaluated using more than one upstream node with respect to the node under consideration (Figure 6.4), as discussed in Chapter 5.

6.3 Estimation of Discretization error

Proper handling of the surface vorticity condition is crucial for the overall accuracy of the stream function-vorticity formulation. Here, we examine the discretization error involved in different treatments of the wall-vorticity condition. For instance, in the lid- driven cavity flow problem, the discretization errors for Equations 6.19, 6.20, 6.21 and 6.24 are $O(\Delta x)$ and $O(\Delta y)$, for the vertical and the horizontal surfaces respectively. Let us prove this estimate of error for

the right side wall ($x = 0$). Considering the Taylor series expansion of stream function near the $x = 0$ boundary, we obtain:

$$\begin{aligned}
\psi_{2,j} &= \psi_{1,j} + \left.\frac{\partial \psi}{\partial x}\right|_{1,j} \Delta x + \left.\frac{\partial^2 \psi}{\partial x^2}\right|_{1,j} \frac{\Delta x^2}{2!} \\
&+ \left.\frac{\partial^3 \psi}{\partial x^3}\right|_{1,j} \frac{\Delta x^3}{3!} + \left.\frac{\partial^4 \psi}{\partial x^4}\right|_{1,j} \frac{\Delta x^4}{4!} + O(\Delta x^5) \\
&= \left.\frac{\partial^2 \psi}{\partial x^2}\right|_{1,j} \frac{\Delta x^2}{2!} + \left.\frac{\partial^3 \psi}{\partial x^3}\right|_{1,j} \frac{\Delta x^3}{3!} \\
&+ \left.\frac{\partial^4 \psi}{\partial x^4}\right|_{1,j} \frac{\Delta x^4}{4!} + O(\Delta x^5)
\end{aligned} \tag{6.28}$$

after simplifying with the help of boundary conditions (6.5) and (6.8). Substituting for the second derivative of stream function from Equation 6.6, Equation 6.28 reduces to the form

$$\omega_{1,j} = \frac{-2\psi_{2,j}}{\Delta x^2} + \left.\frac{\partial^3 \psi}{\partial x^3}\right|_{1,j} \frac{\Delta x}{3} + \left.\frac{\partial^4 \psi}{\partial x^4}\right|_{1,j} \frac{\Delta x^2}{12} + O(\Delta x^3) \tag{6.29}$$

Comparing Equations 6.19 and 6.29, it is clear that the discretization error in the surface vorticity expression (Equation 6.19) is $O(\Delta x)$ only. It can be similarly shown for all the other surfaces that the discretization error in the corresponding vorticity boundary condition is $O(\Delta x)$ or $O(\Delta y)$. Thus, even though the discretization for the interior nodes may have higher accuracy, the overall prediction could suffer from the less accurate treatment of wall-vorticity.

For deriving wall-vorticity expressions of higher order accuracy, the following procedure can be employed. Let us again consider the right side wall ($x = 0$) for the lid-driven cavity flow problem. Differentiating Equation 6.1 with respect to x at the boundary $x = 0$, we obtain

$$\left.\frac{\partial \omega}{\partial x}\right|_{x=0} = -\left[\frac{\partial^3 \psi}{\partial x^3} + \frac{\partial^3 \psi}{\partial x \partial y^2}\right]\Bigg|_{x=0} = -\left.\frac{\partial^3 \psi}{\partial x^3}\right|_{x=0} \tag{6.30}$$

The above expression has been simplified using the condition that $\psi =$ constant for all y on the boundary $x = 0$. Now, substituting Equation 6.30 into 6.29, the wall vorticity expression can be rewritten as:

$$\begin{aligned}
\omega_{1,j} &= \frac{-2\psi_{2,j}}{\Delta x^2} - \left.\frac{\partial \omega}{\partial x}\right|_{1,j} \frac{\Delta x}{3} + \left.\frac{\partial^4 \psi}{\partial x^4}\right|_{1,j} \frac{\Delta x^2}{12} + O(\Delta x^3) \\
&= \frac{-2\psi_{2,j}}{\Delta x^2} - \left\{\frac{(\omega_{2,j} - \omega_{1,j})}{\Delta x} + O(\Delta x)\right\} \frac{\Delta x}{3} \\
&+ \left.\frac{\partial^4 \psi}{\partial x^4}\right|_{1,j} \frac{\Delta x^2}{12} + O(\Delta x^3)
\end{aligned} \tag{6.31}$$

Rearranging Equation 6.31, the following second order expression is obtained for the surface vorticity:

$$\omega_{1,j} = -\frac{3\psi_{2,j}}{\Delta x^2} - \frac{1}{2}\omega_{2,j} + O(\Delta x^2) \tag{6.32}$$

In a similar manner, the wall vorticity estimates for the other surfaces can also be obtained with a discretization error of $O(\Delta x)$ or $O(\Delta y)$.

6.4 Application to Cylindrical Coordinates

Consider steady, incompressible flow over an infinitely long cylinder of radius R_0. The stream function and vorticity equations for this flow geometry are given in terms of the cylindrical polar coordinates (r, θ) as:

$$\frac{\partial^2\psi}{\partial r^2} + \frac{1}{r}\frac{\partial\psi}{\partial r} + \frac{1}{r^2}\frac{\partial^2\psi}{\partial\theta^2} = -\omega \tag{6.33}$$

$$v_r\frac{\partial\omega}{\partial r} + \frac{v_\theta}{r}\frac{\partial\omega}{\partial\theta} = \nu\left\{\frac{\partial^2\omega}{\partial r^2} + \frac{1}{r}\frac{\partial\omega}{\partial r} + \frac{1}{r^2}\frac{\partial^2\omega}{\partial\theta^2}\right\} \tag{6.34}$$

where the radial and tangential velocity components are

$$v_r = \frac{1}{r}\frac{\partial\psi}{\partial\theta} \text{ and } v_\theta = -\frac{\partial\psi}{\partial r} \tag{6.35}$$

The boundary conditions for stream function and vorticity are now derived. The stream function must be a constant on the axis ($\theta = 0$ and $\theta = \pi$) since $v_\theta = 0$ by axisymmetry. The solid surface $r = R_0$ is also a constant stream line since the normal velocity $v_r = 0$ on the cylinder. For this flow geometry, it is evident that the axis as well as the cylinder surface are parts of the same stream line. The value of the stream function can be set equal to zero on these surfaces (Figure 6.5). Thus

$$\begin{aligned} \psi = 0 \text{ on } \theta = 0, \quad r \geq R_0 \\ \psi = 0 \text{ on } \theta = \pi, \quad r \geq R_0 \\ \psi = 0 \text{ on } r = R_0, \quad 0 \leq \theta \leq \pi \end{aligned} \tag{6.36}$$

In the far stream, velocity becomes uniform and equal to U_0 as $r \to \infty$. In terms of stream function, this condition is expressed as:

$$\psi \to U_0 r \sin\theta \text{ as } r \to \infty \tag{6.37}$$

Usually, a large cylindrical surface ($r = R_\infty$) is considered for implementing the far stream condition (Figure 6.5). Therefore, the outer boundary condition (Equation 6.37) is approximated as:

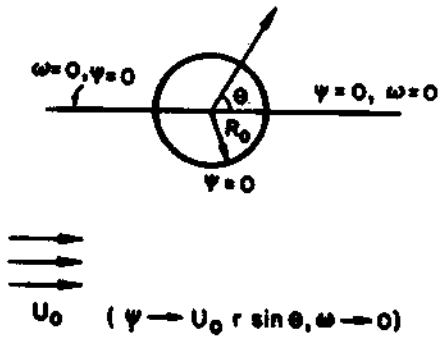


Figure 6.5: Flow Over a Circular Cylinder

$$\psi = U_0 R_\infty \sin\theta \text{ at } r = R_\infty \tag{6.38}$$

For vorticity, the far stream condition is given by

$$\omega = 0 \text{ at } r = R_\infty$$

since the velocity is uniform and velocity gradients are zero. On the axis ($\theta = 0$ and π), $\partial v_r/\partial\theta = 0$ due to axisymmetry and $\psi = 0$ for all r. Thus, from Equations 6.33 and 6.35, it can be seen that

$$\begin{aligned} \omega &= 0 \text{ on } \theta = 0, \quad r \geq R_0 \\ \text{and} \qquad \omega &= 0 \text{ on } \theta = \pi, \quad r \geq R_0 \end{aligned} \tag{6.39}$$

On the solid surface $r = R_0$, however, vorticity is not zero. Here, the wall vorticity has to be derived in much the same way as in the cavity problem. Starting with Equation 6.33 and incorporating the no-slip conditions of $v_r = 0$ and $v_\theta = 0$ (or alternatively, $\psi = 0$ and $\partial\psi/\partial r = 0$ for all θ), the expression for the wall vorticity reduces to

$$\frac{\partial^2\psi}{\partial r^2} = -\omega \tag{6.40}$$

Now, the wall vorticity can be calculated using the image point method or from expressions similar to Equation 6.32. Considering a curvilinear grid as shown in Figure 6.6, the derivatives of stream function or vorticity can be replaced by their discrete (central difference) forms:

$$\frac{\partial\omega}{\partial\theta} = \frac{\omega_{i+1,j} - \omega_{i-1,j}}{2\triangle\theta}; \frac{\partial\omega}{\partial r} = \frac{\omega_{i,j+1} - \omega_{i,j-1}}{2\triangle r}$$

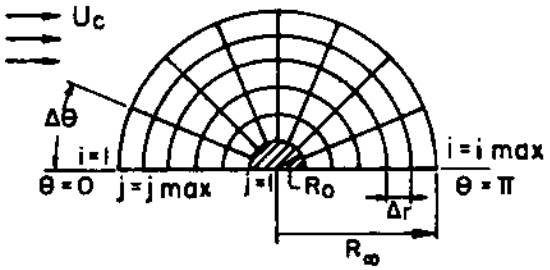


Figure 6.6: Finite Difference Grid for Flow Over a Cylinder

$$\frac{\partial^2\omega}{\partial\theta^2} = \frac{\omega_{i+1,j} + \omega_{i-1,j} - 2\omega_{i,j}}{\triangle\theta^2}; \quad \frac{\partial^2\omega}{\partial r^2} = \frac{\omega_{i,j+1} + \omega_{i,j-1} - 2\omega_{i,j}}{\triangle r^2} \tag{6.41}$$

Similarly, expressions for stream function derivatives can also be obtained. Substituting these expressions into the governing equations (6.33) and (6.34), the nodal equations of all the interior nodes ($i = 2, ...imax - 1, j = 2, ...jmax - 1$) are derived. Finally, the set of discretized equations and boundary conditions are solved by iterative methods. For high Reynolds number flows, upwinding can also be implemented based on the magnitudes of v_r and v_θ, in a similar manner as described in Section 6.2.

The important fact to be kept in mind while simulating flows in curvilinear geometries is that due to curvature, certain terms in the governing equations or boundary conditions may take a 0/0 form. In such cases, one can either resolve the 0/0 form using the L'Hospital rule (if possible) or employ a local cartesian mesh at the point of singularity, as described in Chapter 4. The use of L'Hospital rule to resolve 0/0 forms on the axis for the flow over a sphere is illustrated in the following section.

6.5 Flow over a Sphere

Consider steady uniform axisymmetric flow over a solid sphere (Figure 6.7). The dimensionless governing equations for the stream function-vorticity formulation in spherical geometry are given by (Hamielec *et al.*, 1967):

$$E^2\psi = \omega r \ \sin\theta \tag{6.42}$$

and

$$\left[\frac{\partial\psi}{\partial r}\frac{\partial}{\partial\theta}\left\{\frac{\omega}{r\sin\theta}\right\} - \frac{\partial\psi}{\partial\theta}\frac{\partial}{\partial r}\left\{\frac{\omega}{r\sin\theta}\right\}\right] = \frac{2}{\mathrm{Re}}E^2(\omega r\sin\theta) \tag{6.43}$$

where Re is the Reynolds number based on the sphere diameter and

$$E^2 = \frac{\partial^2}{\partial r^2} + \frac{\sin\theta}{r^2}\frac{\partial}{\partial\theta}\left\{\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right\}$$

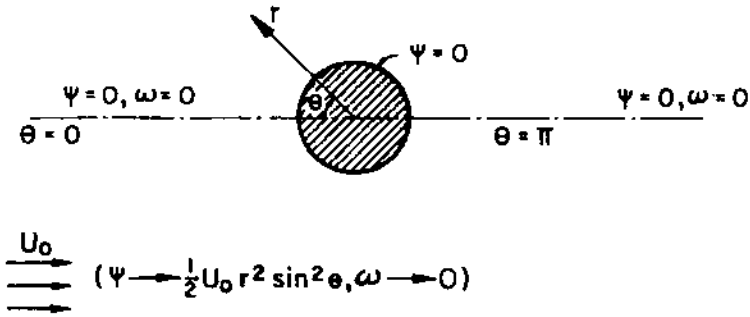


Figure 6.7: Flow Over a Solid Sphere

The velocity components v_r and v_θ are related to the stream function through the expressions

$$v_\theta = \frac{1}{r\sin\theta}\frac{\partial\psi}{\partial r} \tag{6.44}$$

and

$$v_r = -\frac{1}{r^2\sin\theta}\frac{\partial\psi}{\partial\theta} \tag{6.45}$$

The boundary conditions of $v_r = 0$ and $v_\theta = 0$ on the surface of the sphere lead to

$$\psi = 0 \text{ and, } \omega = \frac{1}{\sin\theta}E^2\psi \text{ for } r = 1, \quad 0 < \theta < \pi \tag{6.46}$$

On the axis ($\theta = 0$ and π), based on the conditions of axisymmetry, it can be shown that

$$\psi = 0 \text{ and } \omega = 0 \tag{6.47}$$

The far stream condition of uniform velocity can be written as

$$\psi \to \frac{1}{2}r^2\sin^2\theta \text{ and, } \omega \to 0, \text{ as, } r \to \infty. \tag{6.48}$$

These far stream conditions have to be applied on a finite spherical envelope of radius R_∞, similar to the case of flow over a cylinder.

Equations 6.42-6.48 can be discretized with suitable upwind bias, using procedures described earlier. A little bit of caution is to be exercised, however, while discretizing the term

$$\frac{\partial}{\partial\theta}\frac{\omega}{r\sin\theta}$$

in Equation 6.43 for nodes adjacent to the axis. The expression inside the bracket has a $\frac{0}{0}$ form for $\theta = 0$ and π, which has to be resolved by L'Hospital rule. It is easily verified by the application of L'Hospital rule that

$$\left.\frac{\omega}{r\sin\theta}\right|_{\theta=0} = \left.\frac{1}{r}\frac{\partial\omega}{\partial\theta}\right|_{\theta=0}; \quad \left.\frac{\omega}{r\sin\theta}\right|_{\theta=\pi} = -\left.\frac{1}{r}\frac{\partial\omega}{\partial\theta}\right|_{\theta=\pi} \tag{6.49}$$

Note that the derivative of vorticity with respect to angle θ is not zero on the axis, although vorticity itself is zero due to axisymmetry. The above derivative terms can be discretized to the desired order of accuracy using one-sided difference expressions.

The stream function-vorticity approach can be easily extended to two dimensional or axisymmetric flows with heat and mass transfer. Two phase flow problems are also handled without much difficulty, since the velocity and stress boundary conditions at the interface can be converted in terms of stream function and vorticity. For complex flow problems, this approach offers an advantage that streamlines can be readily plotted, giving valuable insight about the flow structure. For instance, the streamline pattern for flow over a condensing drop is shown in Figure 6.8, for two different levels of condensation (Sundararajan and Ayyaswamy, 1984). It is evident from the streamline plots that the structure of the separated flow region (steady wake) is significantly altered by a change in the condensation level.

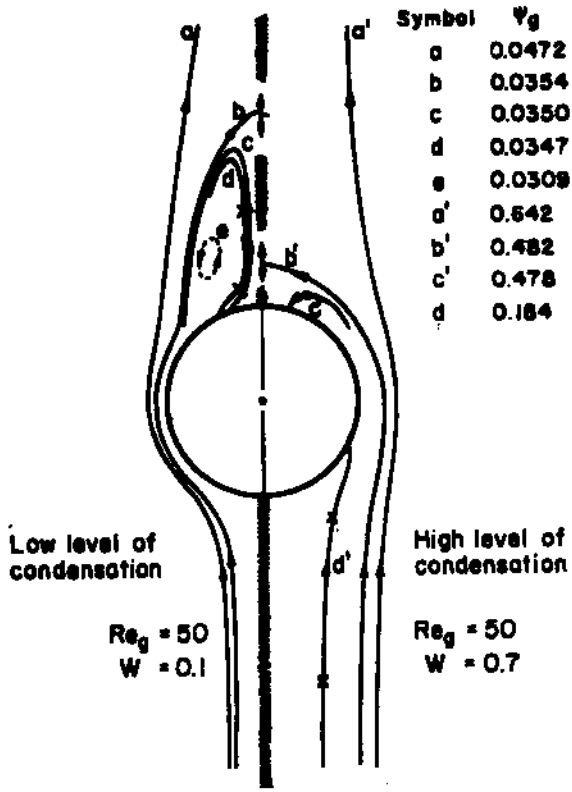


Figure 6.8: Streamline Pattern for a Drop Experiencing Condensation

After computing the stream function and vorticity values, the velocity field can be deduced easily, using the relationships between the stream function and the velocity components. For obtaining the drag force or the pressure field, however, some additional computations have to be performed. The procedure of calculating the pressure profile and the drag on an object from the computed ψ-ω solutions is described below.

6.6 Derivation of Surface Pressure and Drag

The Navier-Stokes equations can be written in terms of vorticity by replacing some of the velocity derivatives. For example, the momentum equations in the radial and angular directions for cross flow over a circular cylinder are given by

$$r - \text{momentum: } \frac{\partial}{\partial r}(p + \frac{1}{2}\rho q^2) = -v_r\omega + \frac{\mu}{r}\frac{\partial \omega}{\partial \theta} \tag{6.50}$$

$$\theta - \text{momentum: } \frac{1}{r}\frac{\partial}{\partial \theta}(p + \frac{1}{2}\rho q^2) = -v_\theta\omega - \mu\frac{\partial \omega}{\partial r} \tag{6.51}$$

where $q^2 = v_r^2 + v_\theta^2$. Integrating Equation 6.50 from the outer boundary $r = R_\infty$ up to the surface of the cylinder $r = R_0$ along the axis $\theta = 0$ leads to

$$(p_\infty + \frac{1}{2}\rho U_0^2) - p_s = \int_{R_0}^{R_\infty} (\frac{\mu}{r}\frac{\partial \omega}{\partial \theta})\Big|_{\theta=0} dr \tag{6.52}$$

after incorporating the condition of $\omega = 0$ on the axis. In Equation 6.52, p_s and p_∞ are the stagnation and far stream pressures respectively. The variation of pressure in the θ direction along the surface of the cylinder is obtained by integrating the θ-momentum equation as:

$$\frac{1}{R_0}\left\{(p + \frac{1}{2}\rho q^2) - p_s\right\} = \int_0^\theta (v_\theta\omega - \mu\frac{\partial \omega}{\partial r})\Big|_{r=R_0} d\theta$$

After simplification (using boundary conditions of $v_r = v_\theta = 0$ on $r = R_0$), the following expression is obtained for the surface pressure:

$$p = p_s - R_0\int_0^\theta \mu\frac{\partial \omega}{\partial r}\Big|_{r=R_0} d\theta \tag{6.53}$$

Discretizing the derivatives of vorticity in Equations 6.52 and 6.53 suitably, numerical quadratures such as Simpson's rule can be applied to evaluate the surface pressure. For good accuracy of the pressure profile, the derivatives in the above equations should be calculated using at least 3 or 4 points in the relevant direction.

In some applications, the pressure field in the entire domain may be required. For such cases, a Poisson equation can be deduced for pressure by applying the divergence operator to the momentum equations. For instance, let us perform the operation

$$(\frac{\partial}{\partial r} + \frac{1}{r})(\text{Equation 6.50}) \; + \frac{1}{r}\frac{\partial}{\partial \theta} \;\; (\text{Equation 6.51})$$

for the flow over a cylinder. This results in the equation

$$\nabla^2(p + \frac{1}{2}\,\rho q^2) = (\frac{\partial}{\partial r} + \frac{1}{r})(-v_r\omega) + \frac{1}{r}\frac{\partial}{\partial \theta}(v_\theta \omega) \tag{6.54}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}$$

Thus, a second order elliptic equation is obtained for pressure. It can be discretized and solved by the finite difference procedures discussed in Chapters 4 and 5. Note that the pressure Poisson equation requires complete specification of pressure (or its derivatives) over the physical domain. In the far field, pressure may be taken to be prescribed. On the solid surface, the normal derivative of pressure from Equation 6.50 can be applied as a boundary condition.

6.7 Closure

The stream function-vorticity method is a robust procedure for solving incompressible flows in two dimensional and axisymmetric geometries. The surface pressure and drag forces also can be evaluated with some additional processing. Three dimensional flows however, cannot be handled by this method. For such general applications, it is preferable to use the velocity-pressure formulation (Chapter 7).

References

Batchelor, G.K., *An Introduction to Fluid Dynamics*, Cambridge University Press, UK, 1993.

Chhabra, R.P., *Bubbles, Drops and Particles in Non-Newtonian Fluids*, CRC Press, Boca Raton, FL, USA, 1993.

Currie, I.G., *Fundamental Mechanics of Fluids*, McGraw-Hill, New York, 1993.

Gresho, P.M. and Sani, R.L., Introducing Four Benchmark Solutions, Int. J. Numerical Methods in Fluids, Vol. 11, pp 951-952, 1990.

Hamielec,A.E., Hoffman,T.W. and Ross,L.L., *Numerical Solution of the Navier Stokes Equation for Flow Past Sphere, part I: Viscous Flow around Spheres with and without Radial Mass Flux*, AIChE, V13, No. 2, pp. 212 - 219, 1967.

Kays, W.M. and Crawford, M.E., *Convective Heat and Mass Transfer*, McGraw-Hill, New York, 1980 (third edition: 1993).

Sparrow, E. M., Lin, S. H. and Lundgren, T. S., Flow Development in the Hydrodynamic Entrance Region of Tubes and Ducts, Physics of Fluids, Vol 7, pp. 338-347, 1964.

Sundararajan,T. and Ayyaswamy,P.S., *Hydrodynamics and Heat Transfer Associated with Condensation on a Moving Drop: Solutions for Intermediate Reynolds Numbers*, J. Fluid Mech., Vol. 149, pp. 33- 58, 1984.

Wilcox, D.C., *Turbulence Modeling for CFD*, DCW Industries, California, second edition, 1998.

Problems

1. Set up the stream function-vorticity formulation for steady boundary layer flow over a flat plate. Assume that the far stream velocity is uniform and the flow is laminar and incompressible.

2. Rework Problem 1, when there is uniform suction at the wall. (Hint: ψ is not constant on the wall).

3. Repeat Problem 1 for a turbulent boundary-layer using a suitable eddy viscosity model (Chapter 1; also see Wilcox, 1998).

4. Develop expressions for upwinding (Section 6.2) purely in terms of stream function ψ at the node (i, j) as well as its neighbours.

5. Generalize the formulation given in the present Chapter for unsteady flow and heat transfer. Thus solve for velocity and temperature distributions adjacent to a heated vertical flat plate in the free, mixed and forced convection regimes. The flow may be treated as laminar. Consider a Reynolds number range of upto 10^4 and Grashof number of less than 10^6, both based on the plate length.

6. Determine the nature of developing flow field in a parallel plate channel (Re < 500). Also solve for the pressure drop in the channel. Compare the friction factor variation with distance and Reynolds number against the data of Sparrow *et al.* (1964). Repeat the problem for a channel with a sudden expansion and compare the results with those of Gresho and Sani (1990) in the Reynolds number range of 100-1000 and an expansion ratio of 1:2.

7. Develop the stream function-vorticity formulation (namely, equations and boundary conditions) for steady axisymmetric flow in a circular tube in r-z coordinates.

8. Determine the nature of developing flow field in a circular tube (Re < 500). Also solve for the pressure drop in the tube. Compare the friction factor variation with distance and Reynolds number against the data of Kays and Crawford (1993).

9. Show that the creeping flow approximation (Re < 1) decouples the stream function and vorticity equations to yield

$$\nabla^4 \psi = 0$$

$$\nabla^2 \omega = 0$$

Express these equations in spherical coordinates with azimuthal symmetry, as in Section 6.5. Derive the necessary boundary conditions for flow past a rigid sphere placed in an infinite fluid medium. Derive a dimensionless expression for drag acting on the sphere as a function of Reynolds number and compare with the analytical result (Batchelor, 1993). Study the effect of the radius of the outer envelop on the numerical solution.

10. Repeat the problem of creeping flow past a sphere to a periodic array of spheres. Extend the formulation to flow of a power-law fluid in a periodic array of spheres (Chhabra, 1993).

11. The gas inside a rising bubble experiences circulatory flow due to viscous shear stresses at the bubble surface. Derive the finite difference nodal equations in terms of ψ and w for this flow, assuming that the circulation velocity at the bubble surface is known. Extend the solution to include flow outside the bubble by applying matching conditions of pressure and shear stress on the bubble boundary. The fluid outside the bubble may be taken as a liquid.

7
Solution of Navier-Stokes Equations for Incompressible Flows Using MAC and SIMPLE Algorithms

7.1 Introduction

In Cartesian coordinates, the governing equations for incompressible three-dimensional laminar flow are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{7.1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right] \tag{7.2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (v^2)}{\partial y} + \frac{\partial (vw)}{\partial z} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right] \tag{7.3}$$

$$\frac{\partial w}{\partial t} + \frac{\partial (uw)}{\partial x} + \frac{\partial (vw)}{\partial y} + \frac{\partial (w^2)}{\partial z} = -\frac{\partial p}{\partial z} + \frac{1}{\mathrm{Re}}\left[\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right] \tag{7.4}$$

In this chapter no assumption is made about the relative magnitude of the velocity components; consequently the full Navier-Stokes equations are solved.

Methods described in this section will be based, essentially on the finite difference and finite volume discretizations and on the solution of a Poisson equation to determine the pressure distribution. It may be mentioned that these methods use primitive variables u, v, w and p as function of x, y, z, t and Reynolds number Re. This approach is to be contrasted with the stream function-vorticity formulation of Chapter 6. Admittedly, the present route is algebraically more complicated, but is clearly preferable in the simulation of three-dimensional flow fields. In addition, it yields directly the pressure distribution, unlike the ψ-ω calculation, where a separate pressure equation has to be solved.

In the primitive variables approach, it is implicitly assumed that the x-component momentum equation determines the x-component velocity, and so on, while the mass balance equation solves for pressure. The fact that pressure does not appear in the mass balance equation at all is a source of concern, and is addressed by the algorithms developed in the present chapter. The second issue that requires careful treatment is the inherent nonlinearity of the Navier-Stokes equations.

7.2 Staggered Grid

As discussed earlier, the major difficulty encountered during solution of incompressible flow is the non-availability of any obvious equation for pressure. This difficulty can be resolved in the stream function-vorticity approach. This approach loses its advantage when three-dimensional flow is computed because of the fact that a single scalar stream function does not exist in three dimensional space. A three-dimensional problem demands a primitive-variable approach. Effort has been made so that two as well as three-dimensional problems could be computed following a primitive variable approach without encountering non-physical wiggles in pressure distribution. As a remedy it has been suggested to employ a different grid for each-dependent variables. Such a staggered grid for the dependent variables in a flow field was first used by Harlow and Welch (1965), in their very well known MAC (Marker and Cell) method. Since then, it has been used by many researchers. Specifically, SIMPLE (Semi Implicit Method for Pressure Linked Equations) procedure of Patankar and Spalding (1972) has become popular. Figure 7.1 shows a two dimensional staggered grid where dependent variables ($u_{i,j}, v_{i,j}$ and $p_{i,j}$) with the same indices are staggered to one another. Extension to three dimensions is straight-forward. The computational domain is divided into number of cells which are shown as "main control volume" in Figure 7.1. The location of the velocity components are at the centre of the cell faces to which they are normal. If a uniform grid is used, the locations are exactly midway between the grid points. In such an arrangement, pressure difference between two adjacent cells is the driving force for the velocity component located between the interface of these cells. The finite difference approximation is now physically meaningful and the computed pressure field will have to be reasonably smooth to correspond to a correct velocity field.

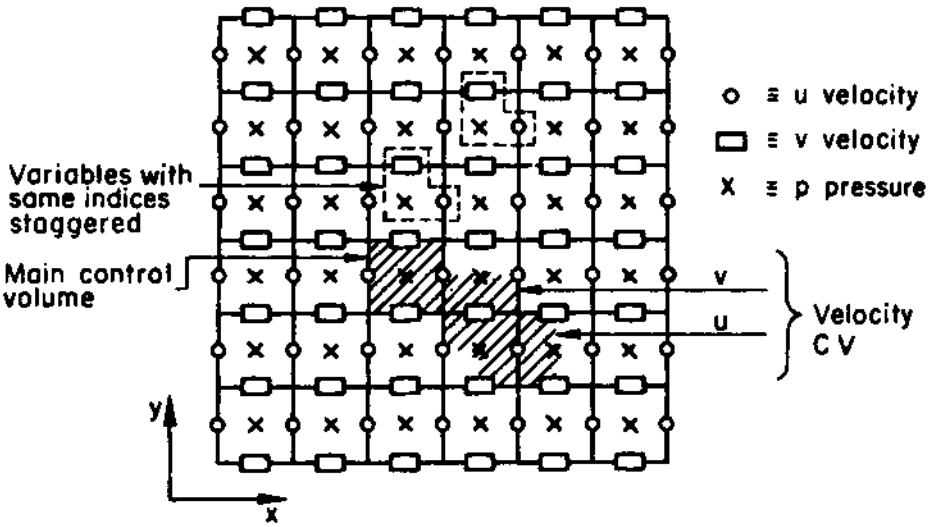


Figure 7.1: Staggered Grid

In this connection, it worth mentioning that a non-staggered grid will admit an oscillatory pressue field over-and-above the correct pressure variation appropriate for the problem of interest. Called the *checker-board pattern*, it has been discussed at length by Patankar (1980) and Ferziger and Peric (2002).

An important advantage of staggered grids is that transport rates across the faces of the control volumes can be computed without interpolation of velocity components.

The detailed outline of two different solution procedures for the full Navier-Stokes equations with primitive variables on a staggered grid will be discussed in subsequent sections. A similar development for the non-staggered (also called *collocated*) grid is presented in Chapter 12. Both chapters describe algorithms that broadly fall in the category of *projection* techniques (Peyret and Taylor, 1983).

7.3 Solution of the Unsteady Navier-Stokes Equations

7.3.1 Introduction to the MAC Method

The MAC method of Harlow and Welch is one of the earliest and most useful methods for solving full Navier-Stokes equations. This method necessarily deals with a Poisson equation for the pressure and momentum equations for the computation of velocity. It was basically developed to solve problems with free surfaces, but can be applied to any incompressible fluid flow problem. A

modified version of the original MAC method due of Hirt and Cook (1972) has been used by many researchers to solve a variety of fluid flow problems.

This text discusses the modified MAC method and highlights the salient features of the solution algorithm so that the readers will be able to write a computer program with some confidence. The important ideas on which the MAC algorithm is based are:

1. Unsteady Navier-Stokes equations for incompressible flow in weak conservative form (see Chapter 4) and the continuity equation are the governing equations.

2. Description of the problem is elliptic in space and parabolic in time. Solution will be marched in the time direction. At each time step, a converged solution in space domain is obtained but this converged solution at any time step may not be the solution of the physical problem.

3. If the problem is steady, in its physical sense, then after a finite number of steps in time direction, two consecutive time-steps will show identical solutions. However, in a machine-computation this is not possible hence a very small value, say, STAT is predefined. Typically, STAT may be chosen between 10^{-3} and 10^{-5}. If the maximum discrepancy of any of the velocity components for two consecutive time steps at any location over the entire space does not exceed STAT then it can be said that the steady solution has been evolved.

4. If the physical problem is basically unsteady in nature, the aforesaid maximum discrepancy of any dependent variable for two consecutive time steps will never be less than STAT. However, for such a situation, a velocity component can be stored over a long duration of time and plot of the velocity component against time depicts the character of flow. Such a flow may be labelled as simply"unsteady".

5. With the help of momentum equations, we compute explicitly a provisional value of the velocity components for the next time step.

   Consider the weak conservative form of nondimensional momentum equation in the x direction:

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}} \nabla^2 u$$

   It is assumed that at $t = n$th level, we have a converged solution. Then for the next time step

$$\frac{\bar{u}^{n+1}_{i,j,k} - u^n_{i,j,k}}{\Delta t} = [\mathrm{SDISCU}]^n_{i,j,k}$$

   or

$$\bar{u}^{n+1}_{i,j,k} = u^n_{i,j,k} + \Delta t\, [\mathrm{SDISCU}]^n_{i,j,k} \tag{7.5}$$

$[\text{SDISCU}]^n_{i,j,k}$ consists of spatial discretization of the convective and diffusive terms of the x-momentum equations, and the pressure gradient. Similarly, the provisional values for $\tilde{v}^{n+1}_{i,j,k}$ and $\tilde{w}^{n+1}_{i,j,k}$ can be explicitly computed. These explicitly advanced velocity components may not constitute a realistic flow field. A divergence free velocity field has to exist in order to describe a plausible incompressible flow situation. Now, with these provisional $\tilde{u}^{n+1}_{i,j,k}$, $\tilde{v}^{n+1}_{i,j,k}$ and $\tilde{w}^{n+1}_{i,j,k}$ values, the continuity equation is evaluated in each cell. If the calculation of $\nabla \cdot \mathbf{u}$ produces a nonzero value, there must be some amount of mass accumulation or annihilation in each cell which is not physically possible, leading to an incorrect pressure field. Therefore, pressure at any cell is directly linked with the value of $(\nabla \cdot \mathbf{u})$ at that cell. Now, on one hand the pressure has to be corrected with the help of nonzero divergence value and on the other, the velocity components have to be adjusted. The correction procedure continues through an iterative cycle till the divergence free velocity field is ensured. Details of this procedure will be discussed in a subsequent section.

6. Boundary conditions are to be applied after each explicit evaluation for the time step is accomplished. Since the governing equations are elliptic in space, boundary conditions on all confining surfaces are required. Moreover, the boundary conditions are also to be applied after every pressure-velocity iteration. The five principal kinds of boundary conditions to be considered are rigid no-slip walls, free-slip walls, in flow and outflow boundaries, and periodic (repeating) boundaries.

7.3.2 MAC Formulation

The region in which computations are to be performed is divided into a set of small cells having edge lengths δx, δy and δz (Figure 7.2). With respect to this set of computational cells, velocity components are located at the centre of the cell faces to which they are normal and pressure and temperature are defined at the centre of the cells. Cells are labelled with an index (i, j, k) which denotes the cell number as counted from the origin in the x, y and z directions respectively. Also $p_{i,j,k}$ is the pressure at the centre of the cell (i, j, k), while $u_{i,j,k}$ is the x direction velocity at the centre of the face between cells (i, j, k) and $(i+1, j, k)$, and so on (Figure 7.3). Because of staggered grid arrangements, the velocities are not defined at the nodal points, but whenever required, they are to be found by interpolation. For example, with uniform grids, we can write $u_{i-1/2,j,k} = \frac{1}{2}\left[u_{i-1,j,k} + u_{i,j,k}\right]$. Where a product or square of such quantity appears, it is to be averaged first and then the product to be formed.

Convective terms are discretized using a weighted average of second upwind and space centered scheme (Hirt *et al.*, 1975). Diffusive terms are discretized by a central difference scheme. Let us consider the discretized terms of the x-momentum equation (Figure 7.3):

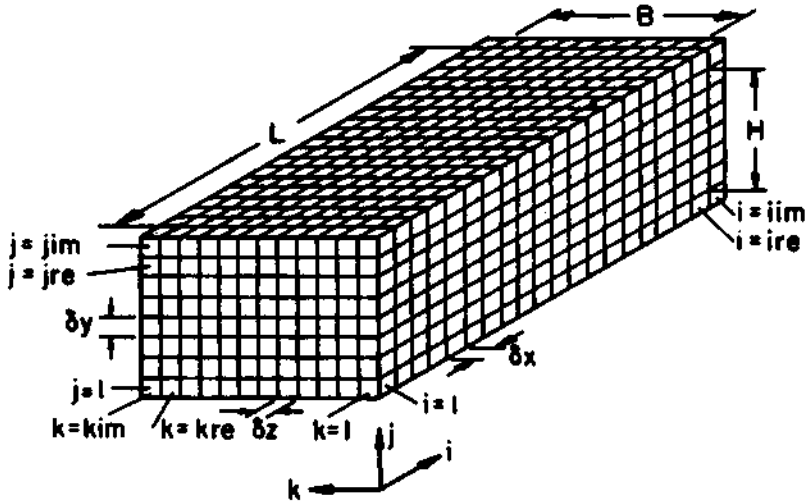


Figure 7.2: Discretization of a Three-Dimensional Domain

$$
\begin{aligned}
\frac{\partial (u^2)}{\partial x} &= \frac{1}{4\delta x}\,[(u_{i,j,k} + u_{i+1,j,k})\,(u_{i,j,k} + u_{i+1,j,k}) \\
&\quad + \alpha \mid (u_{i,j,k} + u_{i+1,j,k}) \mid (u_{i,j,k} - u_{i+1,j,k}) \\
&\quad - (u_{i-1,j,k} + u_{i,j,k})\,(u_{i-1,j,k} + u_{i,j,k}) \\
&\quad - \alpha \mid (u_{i-1,j,k} + u_{i,j,k}) \mid (u_{i-1,j,k} - u_{i,j,k})] \\
&\equiv DUUDX
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial (uv)}{\partial y} &= \frac{1}{4\delta y}\,[(v_{i,j,k} + v_{i+1,j,k})\,(u_{i,j,k} + u_{i,j+1,k}) \\
&\quad + \alpha \mid (v_{i,j,k} + v_{i+1,j,k}) \mid (u_{i,j,k} - u_{i,j+1,k}) \\
&\quad - (v_{i,j-1,k} + v_{i+1,j-1,k})\,(u_{i,j-1,k} + u_{i,j,k}) \\
&\quad - \alpha \mid (v_{i,j-1,k} + v_{i+1,j-1,k}) \mid (u_{i,j-1,k} - u_{i,j,k})] \\
&\equiv DUVDY
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial (uw)}{\partial z} &= \frac{1}{4\delta z}\,[(w_{i,j,k} + w_{i+1,j,k})\,(u_{i,j,k} + u_{i,j,k+1}) \\
&\quad + \alpha \mid (w_{i,j,k} + w_{i+1,j,k}) \mid (u_{i,j,k} - u_{i,j,k+1}) \\
&\quad - (w_{i,j,k-1} + w_{i+1,j,k-1})\,(u_{i,j,k-1} + u_{i,j,k}) \\
&\quad - \alpha \mid (w_{i,j,k-1} + w_{i+1,j,k-1}) \mid (u_{i,j,k-1} - u_{i,j,k})] \\
&\equiv DUWDZ
\end{aligned}
$$

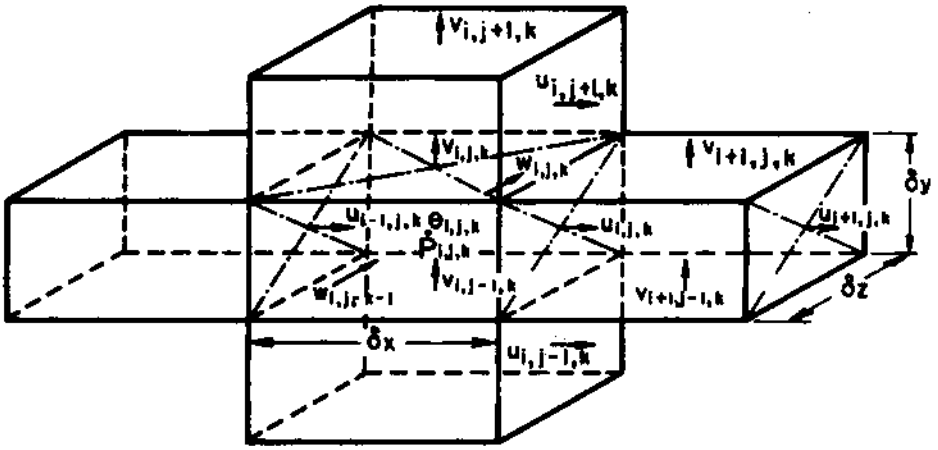


Figure 7.3: Three-dimensional staggered grid showing the locations of the discretized variables

$$\frac{\partial p}{\partial x} = \frac{p_{i+1,j,k} - p_{i,j,k}}{\delta x} \equiv DPDX$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{(\delta x)^2} \equiv D2UDX2$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{(\delta y)^2} \equiv D2UDY2$$

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{(\delta z)^2} \equiv D2UDZ2$$

with

$$\begin{array}{ll} \alpha \longrightarrow 1 & \text{Scheme} \longrightarrow \text{Second upwind} \\ \alpha \longrightarrow 0 & \text{Scheme} \longrightarrow \text{Space centered} \end{array}$$

Factor α is chosen in such a way that the differencing scheme retains "something" of second order accuracy and the required upwinding is done for the sake of stability. A typical value of α is between 0.2 and 0.3. As mentioned earlier, the quantity $\tilde{u}^{n+1}_{i,j,k}$ is now evaluated explicitly from the discretized form of Equation 7.2 as

$$\tilde{u}^{n+1}_{i,j,k} = u^n_{i,j,k} + \delta t\,[\text{SDISCU}]^n_{i,j,k}$$

where

$$\begin{aligned} [\text{SDISCU}]^n_{i,j,k} = \; & [(-DUUDX - DUVDY - DUWDZ) \\ & -DPDX + (1/\text{Re})\,(D2UDX2 \\ & +D2UDY2 + D2UDZ2)] \end{aligned}$$

Similarly, we evaluate

$$\tilde{v}^{n+1}_{i,j,k} = v^n_{i,j,k} + \delta t\,[\text{SDISCV}]^n_{i,j,k} \tag{7.6}$$

$$\tilde{w}^{n+1}_{i,j,k} = w^n_{i,j,k} + \delta t\,[\text{SDISCW}]^n_{i,j,k} \tag{7.7}$$

As discussed earlier, the explicitly advanced tilde velocities may not necessarily lead to a flow field with zero mass divergence in each cell. This implies that, at this stage the pressure distribution is not correct. Pressure in each cell will be corrected in such a way that there is no *net* mass flow in or out of the cell. In the original MAC method, the corrected pressures were obtained from the solution of a Poisson equation for pressure. A related technique developed by Chorin (1967) involved a simultaneous iteration on pressures and velocity components. Viecelli (1971) showed that the two methods as applied to MAC are equivalent. We shall make use of the iterative correction procedure of Chorin (1967) in order to obtain a divergence-free velocity field. The mathematical methodology of this iterative pressure-velocity correction procedure will be discussed herein.

The relationship between the explicitly advanced velocity component and velocity at the previous time step may be written as

$$\tilde{u}^{n+1}_{i,j,k} = u^n_{i,j,k} + \delta t\,\frac{\left[p^n_{i,j,k} - p^n_{i+1,j,k}\right]}{\delta x} + \delta t\,[\text{CONDIFU}]^n_{i,j,k} \tag{7.8}$$

where

$$[\text{CONDIFU}]^n_{i,j,k} = [\text{SDISCU}]^n_{i,j,k} + [DPDX]^n_{i,j,k}$$

On the other hand, the corrected velocity-component (unknown) will be related to the correct pressure (also unknown) in the following way:

$$u^{n+1}_{i,j,k} = u^n_{i,j,k} + \delta t\left[\frac{p^{n+1}_{i,j,k} - p^{n+1}_{i+1,j,k}}{\delta x}\right] + \delta t\,[\text{CONDIFU}]^n_{i,j,k} \tag{7.9}$$

From Equations 7.8 and 7.9

$$u^{n+1}_{i,j,k} - \tilde{u}^{n+1}_{i,j,k} = \frac{\delta t\,\left[p'_{i,j,k} - p'_{i+1,j,k}\right]}{\delta x}$$

where the pressure correction may be defined as

$$p'_{i,j,k} = p^{n+1}_{i,j,k} - p^n_{i,j,k}$$

Neither the pressure corrections nor $u^{n+1}_{i,j,k}$ are known explicitly at this stage. Hence, one cannot be calculated with the help of the other. Calculations are done in an iterative cycle and we write

$$u^{n+1}_{i,j,k} \longrightarrow \tilde{u}^{n+1}_{i,j,k} + \frac{\delta t\,\left[p'_{i,j,k} - p'_{i+1,j,k}\right]}{\delta x}$$

In a similar way, we can formulate the following array:

Corrected		Estimated	Correction	
$u_{i,j,k}^{n+1}$	$\longrightarrow$	$\tilde{u}_{i,j,k}^{n+1}$	$+\dfrac{\delta t\left[p'_{i,j,k}-p'_{i+1,j,k}\right]}{\delta x}$	(7.10)
$u_{i-1,j,k}^{n+1}$	$\longrightarrow$	$\tilde{u}_{i-1,j,k}^{n+1}$	$-\dfrac{\delta t\left[p'_{i,j,k}-p'_{i-1,j,k}\right]}{\delta x}$	(7.11)
$v_{i,j,k}^{n+1}$	$\longrightarrow$	$\tilde{v}_{i,j,k}^{n+1}$	$+\dfrac{\delta t\left[p'_{i,j,k}-p'_{i,j+1,k}\right]}{\delta y}$	(7.12)
$v_{i,j-1,k}^{n+1}$	$\longrightarrow$	$\tilde{v}_{i,j-1,k}^{n+1}$	$-\dfrac{\delta t\left[p'_{i,j,k}-p'_{i,j-1,k}\right]}{\delta y}$	(7.13)
$w_{i,j,k}^{n+1}$	$\longrightarrow$	$\tilde{w}_{i,j,k}^{n+1}$	$+\dfrac{\delta t\left[p'_{i,j,k}-p'_{i,j,k+1}\right]}{\delta z}$	(7.14)
$w_{i,j,k-1}^{n+1}$	$\longrightarrow$	$\tilde{w}_{i,j,k-1}^{n+1}$	$-\dfrac{\delta t\left[p'_{i,j,k}-p'_{i,j,k-1}\right]}{\delta z}$	(7.15)

The correction is done through continuity equation. Plugging-in the above relationship into the continuity Equation 7.1 yields

$$\begin{aligned}
&\left[\frac{u_{i,j,k}^{n+1}-u_{i-1,j,k}^{n+1}}{\delta x}+\frac{v_{i,j,k}^{n+1}-v_{i,j-1,k}^{n+1}}{\delta y}+\frac{w_{i,j,k}^{n+1}-w_{i,j,k-1}^{n+1}}{\delta z}\right]\\
=&\left[\frac{\tilde{u}_{i,j,k}^{n+1}-\tilde{u}_{i-1,j,k}^{n+1}}{\delta x}+\frac{\tilde{v}_{i,j,k}^{n+1}-\tilde{v}_{i,j-1,k}^{n+1}}{\delta y}+\frac{\tilde{w}_{i,j,k}^{n+1}-\tilde{w}_{i,j,k-1}^{n+1}}{\delta z}\right]\\
-&\ \delta t\left[\frac{p'_{i+1,j,k}-2p'_{i,j,k}+p'_{i-1,j,k}}{(\delta x)^2}+\frac{p'_{i,j+1,k}-2p'_{i,j,k}+p'_{i,j-1,k}}{(\delta y)^2}\right.\\
+&\ \left.\frac{p'_{i,j,k+1}-2p'_{i,j,k}+p'_{i,j,k-1}}{(\delta z)^2}\right]
\end{aligned}$$

or

$$\begin{aligned}
&\left[\frac{u_{i,j,k}^{n+1}-u_{i-1,j,k}^{n+1}}{\delta x}+\frac{v_{i,j,k}^{n+1}-v_{i,j-1,k}^{n+1}}{\delta y}+\frac{w_{i,j,k}^{n+1}-w_{i,j,k-1}^{n+1}}{\delta z}\right]\\
=&\left[\frac{\tilde{u}_{i,j,k}^{n+1}-\tilde{u}_{i-1,j,k}^{n+1}}{\delta x}+\frac{\tilde{v}_{i,j,k}^{n+1}-\tilde{v}_{i,j-1,k}^{n+1}}{\delta y}+\frac{\tilde{w}_{i,j,k}^{n+1}-\tilde{w}_{i,j,k-1}^{n+1}}{\delta z}\right]\\
&+\frac{2\delta t\,(p'_{i,j,k})}{\delta x^2}+\frac{2\delta t\,(p'_{i,j,k})}{\delta y^2}+\frac{2\delta t\,(p'_{i,j,k})}{\delta z^2}
\end{aligned}$$

In deriving the above expression, it is assumed that the pressure corrections in the neighbouring cells are zero. The first term between square brackets is the

divergence of the velocity field in a finite difference form. It is given a symbol $(\text{Div})_{i,j,k}$. Returning to the algebraic calculations, we can write

$$0 = (\text{Div})_{i,j,k} + p'_{i,j,k}\left[2\delta t\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2}\right)\right]$$

$$\text{or} \quad p'_{i,j,k} = \frac{-(\text{Div})_{i,j,k}}{\left[2\delta t\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2}\right)\right]} \tag{7.16}$$

In order to accelerate the calculation, the pressure correction equation is modified as

$$p'_{i,j,k} = \frac{-\omega_o\,(\text{Div})_{i,j,k}}{\left[2\delta t\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2}\right)\right]} \tag{7.17}$$

where ω_o is the overrelaxation factor. A value of $\omega_o = 1.7$ is commonly used (Saha *et al.*, 1999). The value of ω_o that gives the most rapid convergence, should be determined by numerical experimentation.

After calculating $p'_{i,j,k}$, the pressure in the cell (i, j, k) is adjusted as

$$p^{n+1}_{i,j,k} \longrightarrow p^{n}_{i,j,k} + p'_{i,j,k} \tag{7.18}$$

Now the pressure and velocity components for each cell are corrected through an iterative procedure in such a way that for the final pressure field, the velocity divergence in each cell vanishes. The process is continued till a divergence-free velocity field is reached with a prescribed upper bound; here a value of 0.0001 is recommended.

Finally we discuss another important observation. If the velocity boundary conditions are correct and a divergence-free converged velocity field has been obtained, eventually correct pressure will be determined in all the cells including the cells at the boundary. Thus, this method avoids the application of pressure boundary conditions. This typical feature of modified MAC method has been discussed in detail by Peyret and Taylor (1983). However, it was also shown by Brandt, Dendy and Ruppel (1980) that the aforesaid pressure-velocity iteration procedure of correcting pressure is equivalent to the solution of the full Poisson equation for pressure.

7.3.3 Boundary Conditions

So far we have not discussed the boundary conditions. However, they are imposed by setting appropriate velocities in the fictitious cells surrounding the physical domain (Figure 7.4).

Consider, for example, the bottom boundary of the computing (physical) mesh. If this boundary is to be a rigid no-slip wall, the normal velocity on the wall must be zero and the tangential velocity components should also be zero. Here we consider a stationary wall. With reference to Figure 7.4, we have

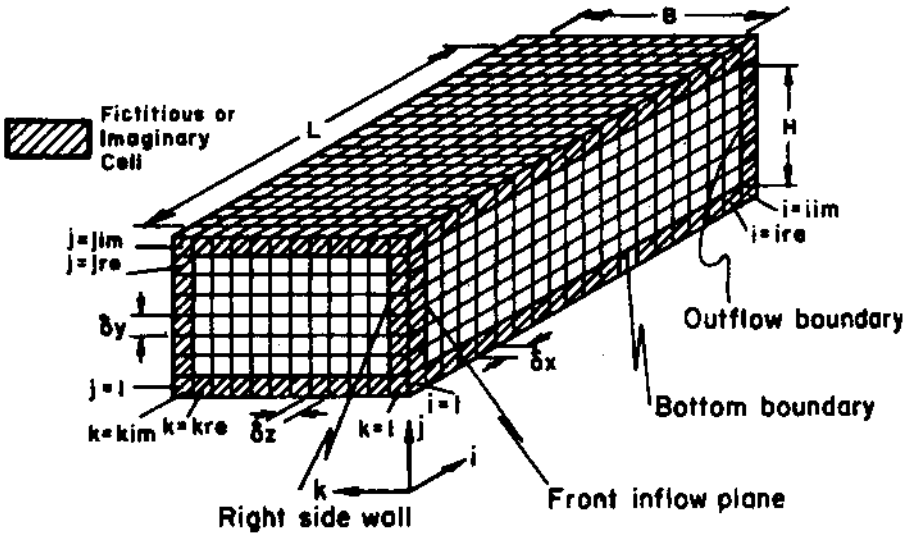


Figure 7.4: Boundary conditions and fictitious boundary cells

$$\left.\begin{aligned} v_{i,1,k} &= 0 \\ u_{i,1,k} &= -u_{i,2,k} \\ w_{i,1,k} &= -w_{i,2,k} \end{aligned}\right\} \quad \begin{array}{c}\text{for } i = 2 \text{ to ire} \\ \text{and } k = 2 \text{ to kre}\end{array}$$

If the right side wall is a free-slip (vanishing shear) boundary, the normal velocity must be zero and the tangential velocities should have no normal gradient.

$$\left.\begin{aligned} w_{i,j,1} &= 0 \\ u_{i,j,1} &= u_{i,j,2} \\ v_{i,j,1} &= v_{i,j,2} \end{aligned}\right\} \quad \begin{array}{c}\text{for } i = 2 \text{ to ire} \\ \text{and } j = 2 \text{ to jre}\end{array}$$

If the front plane is to be provided with inflow boundary condition, it should be specified properly. Any desired functional relationship may be recommended. Generally, normal velocity components are set to zero and a uniform or parabolic axial velocity may be deployed. Hence, with reference to Figure 7.4, we can write

$$\left.\begin{aligned} v_{1,j,k} &= -v_{2,j,k} \\ w_{1,j,k} &= -w_{2,j,k} \\ u_{1,j,k} &= 1.0, \quad \text{or} \\ u_{1,j,k} &= 1.5\left[1 - ((j_m - j)/j_m)^2\right] \end{aligned}\right\} \quad \begin{array}{c}\text{for } j = 2 \text{ to jre} \\ \text{and } k = 2 \text{ to kre}\end{array}$$

where j_m is the horizontal midplane.

Continuative or outflow boundaries always pose a problem for low-speed calculations, because whatever prescription is chosen it can affect the entire

flow field upstream. What is needed is a prescription that permits fluid to flow out of the mesh with a minimum of upstream influence. Commonly used conditions for such a boundary is $\nabla \mathbf{u} \cdot \mathbf{n} = \mathbf{0}$, where $\mathbf{n}$ is the unit normal vector and $\mathbf{u}$ is the velocity vector. In the MAC family of algorithms, the derivatives of the dependent variables in flow direction are set to zero in order to ensure smooth transition through the outflow boundary (Orlanski, 1976). Symbolically, we write

$$\frac{\partial \phi}{\partial t} + u_{\text{av}} \frac{\partial \phi}{\partial x} = 0$$

Here, ϕ represents any of the velocity components (u, v or w) and u_{av} is the average streamwise velocity at the outflow section. The outflow boundary condition is implemented as follows:

$$\text{For} \quad i = ire, \quad j = 2 \ldots jre, \quad k = 2 \ldots kre$$

$$u_{i,j,k}^{n+1} = u_{i,j,k}^{n} - \frac{u_{\text{av}} \Delta t}{\Delta x} \left(u_{i,j,k}^{n} - u_{i-1,j,k}^{n} \right)$$

$$v_{i,j,k}^{n+1} = v_{i,j,k}^{n} - \frac{u_{\text{av}} \Delta t}{\Delta x} \left(v_{i,j,k}^{n} - v_{i-1,j,k}^{n} \right)$$

$$w_{i,j,k}^{n+1} = w_{i,j,k}^{n} - \frac{u_{\text{av}} \Delta t}{\Delta x} \left(w_{i,j,k}^{n} - w_{i-1,j,k}^{n} \right)$$

7.3.4 Numerical Stability Considerations

For accuracy, the mesh size must be chosen small enough to resolve the expected spatial variations in all dependent variables.

Once a mesh has been chosen, the choice of the time increment is governed by two restrictions, namely, the Courant-Friedrichs-Lewy (CFL) condition and the restriction on the basis of grid-Fourier numbers.

According to the CFL condition, material cannot move through more than one cell in one time step, because the difference equations assume fluxes only between the adjacent cells. Therefore the time increment must satisffy the inequality

$$\delta t < \min \left\{ \frac{\delta x}{|\, u \,|}, \frac{\delta y}{|\, v \,|}, \frac{\delta z}{|\, w \,|} \right\} \tag{7.19}$$

where the miminum is with respect to every cell in the mesh. Typically, δt is chosen equal to one-fourth to one-third of the minimum cell transit time. When the viscous diffusion terms are more important, the condition necessary to ensure stability is dictated by the restriction on the grid Fourier numbers, which results in

$$\nu \, \delta t < \frac{1}{2} \cdot \frac{(\delta x^2 \delta y^2 \delta z^2)}{(\delta x^2 + \delta y^2 + \delta z^2)} \tag{7.20}$$

in dimensional form. After nondimensionalization, this leads to

$$\delta t < \frac{1}{2} \cdot \frac{(\delta x^2 \delta y^2 \delta z^2)}{(\delta x^2 + \delta y^2 + \delta z^2)} \cdot \text{Re} \tag{7.21}$$

The final δt for each time increment is the minimum of the δt's obtained from Equations 7.19 and 7.21.

The last quantity needed to ensure numerical stability is the upwind parameter α. In general, α should be slightly larger than the maximum value of $\mid u\delta t/\delta x \mid$ or $\mid v\delta t/\delta y \mid$ occurring in the mesh, that is

$$\max\left\{\mid \frac{u\delta t}{\delta x} \mid, \mid \frac{v\delta t}{\delta y} \mid, \mid \frac{w\delta t}{\delta z} \mid\right\} \leq \alpha < 1 \tag{7.22}$$

As a ready prescription, a value between 0.2 and 0.4 can be used for α. If α is too large, an unnecessary amount of numerical diffusion (artificial viscosity) may be introduced.

7.3.5 Higher Order Upwind Differencing

More accurate solutions are obtained if the convective terms are discretized by higher order schemes. Davis and Moore (1982) use the MAC method with a multidimensional third-order upwinding scheme. Needless to mention that their marching algorithm for the momentum equation is explicit and the stability restriction concerning the CFL condition $[u\delta t/\delta x \leq 1 \quad \text{and} \quad v\delta t/\delta y \leq 1]$ is satisfied.

The multidimensional third-order upwinding is, in principle similar to one-dimensional quadratic upstream interpolation scheme introduced by Leonard (1979). Consider Figure 7.5. Let ϕ be any property which can be convected and diffused. The convective term $\partial(u\phi)/\partial x$ may be represented as

$$\frac{\partial(u\phi)}{\partial x} = \frac{(u\phi)_{i+\frac{1}{2},j} - (u\phi)_{i-\frac{1}{2},j}}{\delta x} \tag{7.23}$$

where the variables $\phi_{i+\frac{1}{2},j}$ and $\phi_{i-\frac{1}{2},j}$ are defined as

$$\phi_{i+\frac{1}{2},j} = 0.5\,(\phi_{i,j} + \phi_{i+1,j}) - \frac{\xi}{3}\,(\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}) \quad \text{for} \quad u_{i,j} > 0 \tag{7.24}$$

and

$$\phi_{i-\frac{1}{2},j} = 0.5\,(\phi_{i,j} + \phi_{i-1,j}) - \frac{\xi}{3}\,(\phi_{i-2,j} - 2\phi_{i-1,j} + \phi_{i,j}) \quad \text{for} \quad u_{i,j} > 0 \tag{7.25}$$

The parameter ξ can be chosen to increase accuracy or to alter the diffusion-like characteristics. It may be pointed out that $\xi = 3/8$ corresponds to the QUICK scheme of Leonard (1979).

Let us consider two-dimensional momentum equation in weak conservative form which is given by

$$\frac{\partial u}{\partial t} + \frac{\partial(u^2)}{\partial x} + \frac{\partial(uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] \tag{7.26}$$

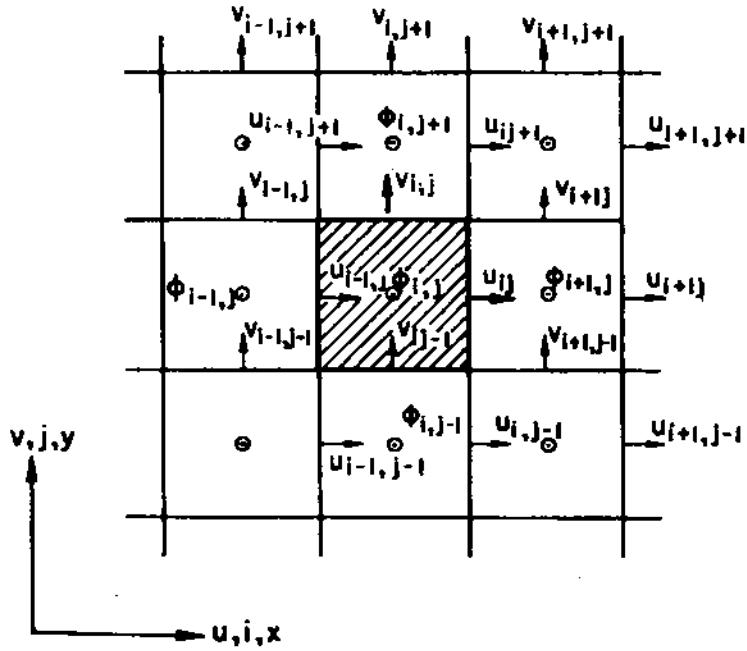


Figure 7.5: Dependent Variables (u, v and ϕ) on a rectangular mesh

In non-conservative form this may be written as

$$\frac{\partial u}{\partial t} + u\,\frac{\partial u}{\partial x} + v\,\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] \tag{7.27}$$

Here we introduce a term transport-velocity. The transport velocities for the second and third terms on the left hand side are u and v respectively. While dealing with the equations in conservative form, we shall keep this in mind. For example, during discretization of the term $\partial(uv)/\partial y$ of Equation 7.26, we should remember that v is the transport-velocity associated with this term. It is customary to define the transport-velocity at the nodal point where the equation is being defined. In case of the term $\partial(uv)/\partial y$ we have to refer to Figure 7.6 and write down the product term uv as

$$(uv)_{i,j} = 0.25\,u_{i,j}\,(v_{i,j} + v_{i+1,j} + v_{i,j-1} + v_{i+1,j-1}) \tag{7.28}$$

Finally the discretization of the term $\partial(uv)/\partial y$ for the x-momentum equation will be accomplished in the following way:

$$\begin{aligned}\frac{\partial uv}{\partial y} &= 0.25\frac{1}{8\delta y}\\ &\quad [3u_{i,j+1}\,(v_{i,j+1} + v_{i,j} + v_{i+1,j+1} + v_{i+1,j})\end{aligned}$$

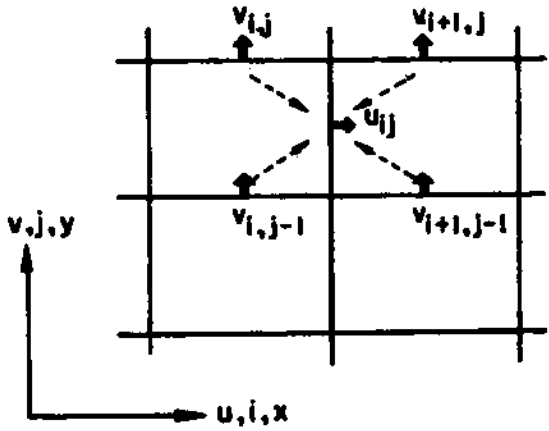


Figure 7.6: Definition of the transport velocity at a point where the momentum equation is being discretized

$$
\begin{aligned}
&+3u_{i,j}\left(v_{i,j}+v_{i,j-1}+v_{i+1,j}+v_{i+1,j-1}\right)\\
&-7u_{i,j-1}\left(v_{i,j-1}+v_{i,j-2}+v_{i+1,j-1}+v_{i+1,j-2}\right)\\
&+u_{i,j-2}\left(v_{i,j-2}+v_{i,j-3}+v_{i+1,j-2}+v_{i+1,j-3}\right)] \text{ for } V>0
\end{aligned}
\tag{7.29}
$$

$$
\begin{aligned}
\frac{\partial uv}{\partial y} &= 0.25\frac{1}{8\delta y}\\
&-[u_{i,j+2}\left(v_{i,j+2}+v_{i,j+1}+v_{i+1,j+2}+v_{i+1,j+1}\right)\\
&+7u_{i,j+1}\left(v_{i,j+1}+v_{i,j}+v_{i+1,j+1}+v_{i+1,j}\right)\\
&-3u_{i,j}\left(v_{i,j}+v_{i,j-1}+v_{i+1,j}+v_{i+1,j-1}\right)\\
&-3u_{i,j-1}\left(v_{i,j-1}+v_{i,j-2}+v_{i+1,j-1}+v_{i+1,j-2}\right)] \text{ for } V<0
\end{aligned}
\tag{7.30}
$$

where $V = v_{i,j} + v_{i+1,j} + v_{i,j-1} + v_{i+1,j+1}$

7.3.6 Sample Results

For unsteady laminar flow past a rectangular obstacle in a channel, Mukhopadhyay, Biswas and Sundararajan (1992) use the MAC algorithm to explicitly march in time. Their results corroborated with the experimental observation of Okajima (1982). A typical example of numerical flow visualization depicting the development of von-Karman vortex street is illustrated in Figure 7.7. The cross-stream velocity vectors behind a delta-wing placed inside a channel are shown in Figure 7.8. These results were obtained by Biswas and Chattopadhyay (1992) who used MAC to solve for a three-dimensional flow field in a channel

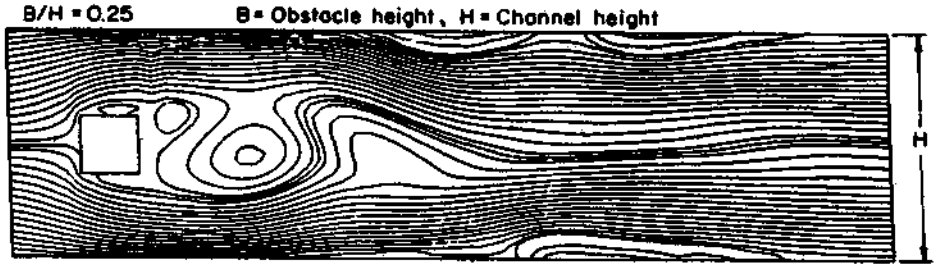


Figure 7.7: Streamlines around a Bluff-body in a Duct: $Re_B = 162$.

containing delta-wing as a vortex generator. The MAC algorithm has been extensively used by many researchers to solve flows in complex geometry. Braza, Chassaing and Ha-Minh (1986) investigated the dynamic characteristics of the pressure and velocity fields of the unsteady wake behind a circular cylinder using MAC algorithm. Recently, Robichaux, Tafti and Vanka (1992) deployed MAC algorithm for Large Eddy Simulation (LES) of turbulent channel flows. Of course, they performed the time integration of the discretized equations by using a fractional step method (Kim and Moin, 1985). Another investigation by Kim and Benson (1992) suggests that the MAC method is significantly accurate and at the same time the computational effort is reasonable.

7.4 Solution of Energy Equation

The energy equation for incompressible flows, neglecting mechanical work and gas radiation, may be written as

$$\rho c_p \frac{DT}{Dt^*} = k\nabla^2 T + \mu\phi^* \tag{7.31}$$

where ϕ^* is the viscous dissipation given as

$$\begin{aligned}\phi^* &= 2\left[\left(\frac{\partial u^*}{\partial x^*}\right)^2 + \left(\frac{\partial v^*}{\partial y^*}\right)^2 \left(\frac{\partial w^*}{\partial z^*}\right)^2\right] + \left\{\frac{\partial u^*}{\partial x^*} + \frac{\partial u^*}{\partial y^*}\right\}^2 \\ &+ \left\{\frac{\partial w^*}{\partial y^*} + \frac{\partial v^*}{\partial z^*}\right\}^2 + \left\{\frac{\partial w^*}{\partial x^*} + \frac{\partial u^*}{\partial z^*}\right\}^2\end{aligned}$$

Equation 7.31 may be nondimensionalized in the following manner:

$$\begin{aligned} u &= \frac{u^*}{U_\infty}, v = \frac{v^*}{U_\infty}, w = \frac{w^*}{U_\infty}, \quad \theta = \frac{T - T_\infty}{T_w - T_\infty} \\ x &= \frac{x^*}{L}, v = \frac{y^*}{L}, z = \frac{z^*}{L} \quad \text{and} \quad t = \frac{t^*}{L/U_\infty} \end{aligned}$$

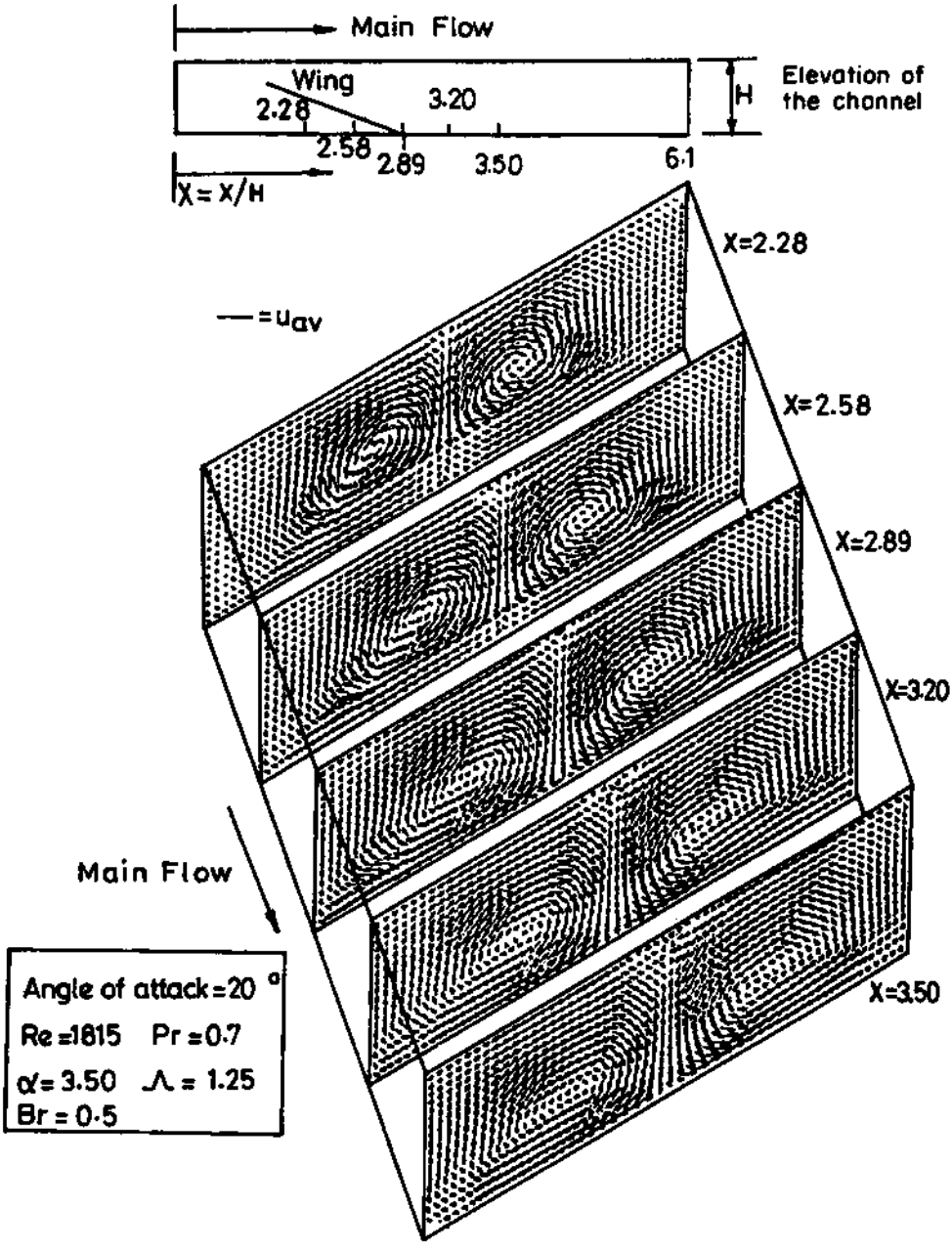


Figure 7.8: Generation and deformation of longitudinal vortices in a channel

Substituting the above variables in Equation 7.31 we obtain

$$\frac{\rho c_p U_\infty (T_w - T_\infty)}{L}\left[\frac{\partial\theta}{\partial t} + u\frac{\partial\theta}{\partial x} + v\frac{\partial\theta}{\partial y} + w\frac{\partial\theta}{\partial z}\right]$$
$$= \frac{(T_w - T_\infty)\,k}{L^2}\left[\frac{\partial^2\theta}{\partial x^2} + \frac{\partial^2\theta}{\partial y^2} + \frac{\partial^2\theta}{\partial z^2}\right] + \frac{\mu U_\infty^2}{L^2}\phi \tag{7.32}$$

where ϕ is nondimensional form of ϕ^*. The normalized energy equation can be written as

$$\frac{\partial\theta}{\partial t} + u\frac{\partial\theta}{\partial x} + v\frac{\partial\theta}{\partial y} + w\frac{\partial\theta}{\partial z} = \frac{1}{\mathrm{Pe}}\left[\frac{\partial^2\theta}{\partial x^2} + \frac{\partial^2\theta}{\partial y^2} + \frac{\partial^2\theta}{\partial z^2}\right] + \frac{\mathrm{Ec}}{\mathrm{Re}}\phi \tag{7.33}$$

where Pe, the Peclet number is given as

$$\frac{1}{\mathrm{Pe}} = \frac{k(T_w - T_\infty)}{L^2}\cdot\frac{L}{\rho c_p U_\infty (T_w - T_\infty)}$$
$$\text{or}\quad \frac{1}{\mathrm{Pe}} = \frac{k}{L\cdot\rho\cdot c_p U_\infty} = \frac{k}{\mu c_p}\cdot\frac{\mu}{\rho L U_\infty} = \frac{1}{\mathrm{Pr}}\cdot\frac{1}{\mathrm{Re}}$$

Further, Ec the Eckert number is

$$\frac{\mathrm{Ec}}{\mathrm{Re}} = \frac{\mu U_\infty^2}{L^2}\cdot\frac{L}{\rho c_p U_\infty (T_w - T_\infty)} = \frac{U_\infty^2}{c_p (T_w - T_\infty)}\cdot\frac{1}{\rho U_\infty L/\mu}$$

7.4.1 Retention of Dissipation

The dissipation term ϕ is frequently neglected while solving the energy equation for incompressible flows. As the Mach number $\mathrm{M} \longrightarrow 0$, $\mathrm{Ec} \longrightarrow 0$. However, even at a low Mach number, ϕ can be important if $(T_w - T_\infty)$ is very small. Let us look at these aspects. Since

$$\mathrm{Ec} = \frac{U_\infty^2}{c_p (T_w - T_\infty)},\ \frac{1}{\mathrm{Ec}} = \frac{c_p T_\infty}{U_\infty^2}\left[\frac{T_w}{T_\infty} - 1\right]$$

and

$$\frac{c_p T_\infty}{U_\infty^2} = \frac{c_p \gamma R T_\infty}{\gamma R U_\infty^2}$$

where R is the gas constant $= c_p - c_v$, and $\gamma = c_p/c_v$.
The local speed of sound is $C = \sqrt{\gamma R T_\infty}$, and Mach number $\mathrm{M}_\infty = U_\infty/C$.
We can now write

$$\frac{c_p T_\infty}{U_\infty^2} = \frac{c_p}{\gamma (c_p - c_v)}\left[\frac{1}{\mathrm{M}_\infty^2}\right] = \frac{c_p}{\gamma c_p (1 - \frac{1}{\gamma})}\left[\frac{1}{\mathrm{M}_\infty^2}\right] = \frac{1}{(\gamma - 1)}\frac{1}{\mathrm{M}_\infty^2}$$

Hence

$$\frac{1}{\mathrm{Ec}} = \frac{1}{(\gamma - 1)\,\mathrm{M}_\infty^2}\left(\frac{T_w}{T_\infty} - 1\right)$$

or

$$\mathrm{Ec} = \frac{(\gamma - 1)\,\mathrm{M}_\infty^2}{((T_w/T_\infty) - 1)}$$

In general, for incompressible flows $\mathrm{M}_\infty < 0.3$ and $\gamma \geq 1$. Hence Ec is small. But for very small temperature difference, i.e. if T_w/T_∞ is slightly larger than 1. Ec might assume a large value and importance of including dissipation arises.

However, for computing incompressible convective flows, the viscous dissipation is neglected in this chapter and we start with the steady state energy equation.

7.5 Solution Procedure

The steady state energy equation, neglecting the dissipation term, may be written in the following conservative form as

$$\frac{\partial(u\theta)}{\partial x} + \frac{\partial(v\theta)}{\partial y} + \frac{\partial(w\theta)}{\partial z} = \frac{1}{\mathrm{Pe}}\left[\frac{\partial^2\theta}{\partial x^2} + \frac{\partial^2\theta}{\partial y^2} + \frac{\partial^2\theta}{\partial z^2}\right] \tag{7.34}$$

Equation 7.34 may be rewritten as

$$\nabla^2\theta = \mathrm{Pe}\,[\mathrm{CONVT}]_{i,j,k}^m \tag{7.35}$$

where $[\mathrm{CONVT}]_{i,j,k}$ is the discretized convective terms on the left hand side of Equation 7.34 and m stands for iterative counter. To start with, we can assume any guess value of θ throughout the flow field. Since u, v, w are known from the solution of momentum equation, hence Equation 7.34 is now a linear equation. However, from the guess values of θ and known correct values of u, v, and w the left hand side of Equation 7.34 is evaluated. A weighted average scheme or QUICK scheme may be adapted for discretization of the convective terms. After discretizing and evaluating right hand side of Equation 7.35 we obtain a Poisson equation for temperature with a source term on the right hand side. Now, we shall follow a SOR technique for solving Equation 7.35. Consider a discretized equation as

$$\begin{aligned}\frac{\theta_{i+1,j,k} - 2\theta_{i,j,k} + \theta_{i-1,j,k}}{(\delta x)^2} &+ \frac{\theta_{i,j+1,k} - 2\theta_{i,j,k} + \theta_{i,j-1,k}}{(\delta y)^2} \\ &+ \frac{\theta_{i,j,k+1} - 2\theta_{i,j,k} + \theta_{i,j,k-1}}{(\delta x)^2} = S^{*m}\end{aligned}$$

where

$$S^{*m} \equiv \mathrm{Pe}\,[\mathrm{CONVT}]_{i,j,k}^m$$

or

$$A^{*m} - \theta_{i,j,k}\left[\frac{2}{(\delta x)^2} + \frac{2}{(\delta y)^2} + \frac{2}{(\delta z)^2}\right] = S^{*m}$$

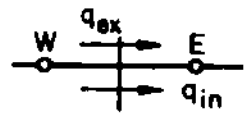


Figure 7.9: Consistency of Flux

or

$$\theta_{i,j,k} = \frac{A^{*m} - S^{*m}}{\left[\frac{2}{(\delta x)^2} + \frac{2}{(\delta y)^2} + \frac{2}{(\delta z)^2}\right]} \tag{7.36}$$

where

$$A^{*m} = \frac{\theta^m_{i+1,j,k} + \theta^m_{i-1,j,k}}{(\delta x)^2} + \frac{\theta^m_{i,j+1,k} + \theta^m_{i,j-1,k}}{(\delta y)^2} + \frac{\theta^m_{i,j,k+1} + \theta^m_{i,j,k-1}}{(\delta z)^2}$$

$\theta_{i,j,k}$ in Equation 7.36 may be assumed to be the most recent value and it may be written as $\theta^{m'}_{i,j,k}$. In order to accelerate the speed of computation we introduce an overrelaxation factor ω. Thus

$$\theta^{m+1}_{i,j,k} = \theta^m_{i,j,k} + \omega\left[\theta^{m'}_{i,j,k} - \theta^m_{i,j,k}\right] \tag{7.37}$$

where $\theta^m_{i,j,k}$ is the previous value, $\theta^{m'}_{i,j,k}$ the most recent value and $\theta^{m+1}_{i,j,k}$ the calculated better guess. The procedure will continue till the required convergence is achieved. This is equivalent to Gauss-Seidel procedure for solving system of linear equations.

7.6 SIMPLE Formulation

7.6.1 Introduction

This family of algorithms is based on a finite volume discretization of the governing equations on a staggered grid. The method was introduced by Patankar and Spalding (1972) and Patankar (1980).

Before presenting the SIMPLE flow-solver, we have to be acquainted with some terminologies and ideas. These ideas will be discussed in this section. We have to remember these concepts while formulating the problems to be solved by the SIMPLE algorithm.

7.6.2 The Four Basic Rules

Rule 1: Consistency of flux across the control volume faces:

When a face is common to two adjacent control volumes, the flux across it must be represented by the same expression for the two control volumes (Figure 7.9).

$$q_{ex} = q_{in} \tag{7.38}$$

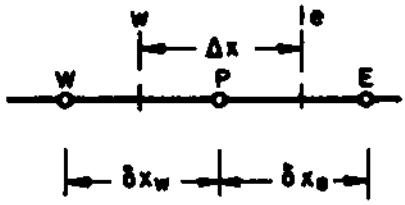


Figure 7.10: One-dimensional Control Volume

Rule 2: Positive Coefficients:

Value of a dependent variable at a grid point (P) may be expressed in terms of its neighbours (E and W). Hence with reference to Figure 7.10, the resulting expression will be $a_P\phi_P = a_E\phi_E + a_W\phi_W$ where ϕ is the dependent variable and a's are the coefficients. The east and the west neighbours of point P on a rectangular grid (Figure 7.10) are indicated by E and W respectively. However, the arrangement will be such that the coefficients a_E, a_W and a_P linked with the dependent variable ϕ will be considered positive

$$a_P,\ a_E,\ a_W > 0$$

Rule 3: Negative-slope linearization of the source term:

The strategy of finite volume method is to develop an algebraic relationship among the dependent variables (ϕ's) at neighbouring grid points by integrating the governing equation with respect to independent variables over the local neighbourhood of point P (refer to Figure 7.10). While doing this the source terms (which are not linked to a_E and a_W) may be linearized as:

$$S = S_c + S_P\,\phi_P$$

where ϕ_P is the dependent variable, and the coefficient S_P must always be less than or equal to zero.

Rule 4: Sum of Neighbouring Coefficients:

It should be noted that, in presence of source terms,

$$a_P = \sum a_{nb} - S_P\,\Delta v$$

where, $\underline{nb}$ stands for $\underline{\text{neighbours}}$ and Δv is the elemental control volume when

$$S_P = 0, \quad a_P = \sum a_{nb}.$$

7.6.3 Discretization Technique on a One-dimensional Control Volume

In order to appreciate the discretization equation for steady conduction in any dimension, let us review the governing differential equation for one-dimensional heat conduction problem. Steady one dimensional equation for conduction is

$$\frac{d}{dx}\left(k\,\frac{\partial T}{\partial x}\right)+S=0$$

Integrating over a control volume (refer to Figure 7.10), we get

$$\left(k\,\frac{dT}{dx}\right)_e-\left(k\,\frac{dT}{dx}\right)_w+\int_w^e\left(S_c+S_p\,T_p\right)dx=0$$

A linear profile assumption for temperature yields

$$k_e(T_E-T_P)/\delta x_e-k_w(T_P-T_W)/\delta x_w+(S_c+S_PT_P)\,\Delta x=0$$

or

$$a_P\,T_P=a_E\,T_E+a_W\,T_W+b,$$

where

$$a_E=k_e/\delta x_e,\quad a_W=k_w/\delta x_w$$

$$a_P=a_E+a_W-S_P\Delta x\quad\text{and}\quad b=S_c\Delta x$$

In general, it follows that

$$a_P\,T_P=\sum a_{nb}\,T_{nb}+b \tag{7.39}$$

7.6.4 Discretization of One-dimensional Convection-diffusion Equation

Unsteady convection-diffusion equation, in one-dimension may be expressed as:

$$\frac{\partial}{\partial t}(\rho\phi)+\frac{\partial}{\partial x_j}(\rho u_j\phi)=\frac{\partial}{\partial x_j}\left(\Gamma\,\frac{\partial\phi}{\partial x_j}\right)+S \tag{7.40}$$

where ϕ is any quantity which can be convected by the mean flow and diffused across a gradient $\nabla\phi$. Assuming that the flow field is known, we want to calculate the distribution of ϕ. Let us consider a steady case

$$\frac{d}{dx}(\rho u\phi)=\frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right) \tag{7.41}$$

We integrate over the control volume (refer to Figure 7.10) to get,

$$(\rho u\phi)_e-(\rho u\phi)_w=\left(\Gamma\,\frac{d\phi}{dx}\right)_e-\left(\Gamma\,\frac{d\phi}{dx}\right)_w$$

or

$$(\rho u)_e \frac{1}{2}(\phi_E + \phi_P) - (\rho u)_w \frac{1}{2}(\phi_P + \phi_W) = \frac{\Gamma_e(\phi_E - \phi_P)}{\delta x_e} - \frac{\Gamma_w(\phi_P - \phi_W)}{\delta x_w}$$

or

$$\phi_P \left[\frac{(\rho u)_e}{2} - \frac{(\rho u)_w}{2} + \frac{\Gamma_e}{\delta x_e} + \frac{\Gamma_w}{\delta x_w}\right] = \phi_E \left[\frac{\Gamma_e}{\delta x_e} - \frac{(\rho u)_e}{2}\right] + \phi_W \left[\frac{(\rho u)_w}{2} + \frac{\Gamma_w}{\delta x_w}\right]$$

In the above equation, following parameters are very important

- ρu indicates the strength of convection ($\equiv F$)
- $\Gamma/\delta x$ is the <u>diffusion conductance</u> ($\equiv D$)
- D always remains <u>positive</u>
- (ρu) or F can take positive or negative values depending on the direction of flow.

Finally the form of the discretization equation can be written as

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \tag{7.42}$$

where

$$a_E = D_e - \frac{F_e}{2} \tag{7.43}$$

$$a_W = D_w + \frac{F_w}{2} \tag{7.44}$$

We also observe

$$\begin{aligned} a_P &= \frac{F_e}{2} - \frac{F_w}{2} + D_e + D_w \\ &= D_e + \frac{F_e}{2} + D_w - \frac{F_w}{2} \\ &= D_e - \frac{D_e}{2} + D_w + \frac{F_w}{2} + F_e - F_w \\ &= a_E + a_W + (F_e - F_w) \end{aligned}$$

Since by continuity $F_e = F_w$, Rule 4 is satisfied and we get

$$a_P = a_E + a_W \tag{7.45}$$

7.6.5 A More General Formulation

In two dimensions, the x momentum equation is given by

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = -\frac{\partial p}{\partial x} \tag{7.46}$$

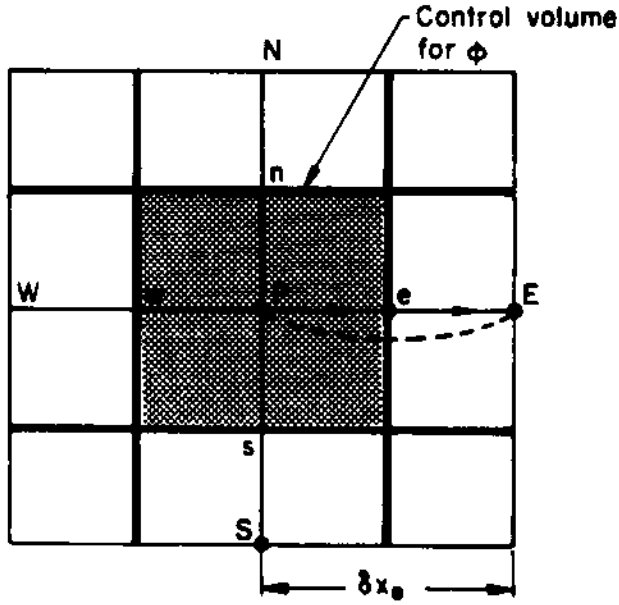


Figure 7.11: Control Volume in Two Dimensions

where with reference to Figure 7.11

$$J_x = \rho u \phi - \Gamma \frac{\partial \phi}{\partial x}, \quad \text{and} \quad J_y = \rho v \phi - \Gamma \frac{\partial \phi}{\partial y}$$

J_e is the combined convective diffusive flux crossing an interface. It may also be represented as J_e^* as:

$$J_e^* = \frac{J_e \, \delta x_e}{\Gamma} = \frac{(\rho u)_e \, \phi \, \delta x_e}{\Gamma} - \frac{\partial \phi}{\partial (x/\delta x_e)}$$

or

$$J_e^* = \mathrm{P_e} \phi - \frac{d\phi}{d(x/\delta x_e)}$$

$\mathrm{P_e}$ being the cell Peclet number. For simplicity we shall call it Peclet number. It can also be proposed

$$J_e^* = \mathrm{P_e} \left[\alpha \phi_P + (1-\alpha)\phi_E\right] - \beta \left(\phi_P - \phi_E\right)$$

α and β are dimensionless multipliers that depend on $\mathrm{P_e}$. Rewriting this equation as

$$J_e^* = B \, \phi_P - A \phi_E \tag{7.47}$$

we see that

$$B = \alpha \mathrm{P_e} - \beta \quad \text{and} \quad A = -\left[\beta + \mathrm{P_e}\,(1-\alpha)\right]$$

The coefficient A is associated with the point which is at the downstream of the interface grid point and B is connected with the point which is at the upstream of the interface-grid-point.

If ϕ_P and ϕ_E are equal then the diffusion flux must be zero and J_e^* reduces to the convective flux only, i.e.

$$J_e^* = \mathrm{P_e}\,\phi_P = \mathrm{P_e}\,\phi_E \tag{7.48}$$

Combination of Equations 7.47 and 7.48 produces

$$B = A + \mathrm{P_e}$$

Now, Equation 7.47 can be rewritten as

$$J_e^* = \mathrm{P_e}\,\phi_P + A\,(\phi_P - \phi_E) \tag{7.49}$$

and $\mathrm{P_e}$ is essentially $(\rho u)_e\,\delta x_e/\Gamma_e$ or

$$\mathrm{P_e} = \frac{(\rho u)_e[\Delta y]}{(\Gamma_e/\delta x_e)[\Delta y]}$$

$$\text{So} \quad \mathrm{P_e} = \frac{F_e}{D_e} = \frac{\text{Convection transport at } e}{\text{Diffusive transport at } e}$$

and we have already accepted J_e^* as $J_e\,\delta x_e/\Gamma_e = J_e/D_e$.

We can write Equation 7.49 once again as

$$J^* = \frac{J_e}{D_e} = \frac{F_e}{D_e}\,\phi_P + A\,(\phi_P - \phi_E)$$

or

$$J_e - F_e\phi_P = D_e\,A\,(\phi_P - \phi_E)$$

If both the directions of (ρu) are considered, then for $F_e < 0$, i.e. for $(\rho u)_e < 0$), J_e can be written as

$$J_e - F_e\phi_E = AD_e\,(\phi_P - \phi_E)$$

and for $F_e > 0$ ($(\rho u)_e > 0$,)

$$J_e - F_e\phi_P = AD_e\,(\phi_P - \phi_E)$$

So, we can write a more general expression as

$$J_e - F_e\phi_P = a_E\,(\phi_P - \phi_E) \tag{7.50}$$

where

$$a_E = AD_e + \max[-F_e, 0]$$

The symbol max[,] signifies greater of the two quantities enclosed within the brackets.

Hence if, $\quad F_e > 0, \quad a_E = AD_e \quad$ and $\quad J_e - F_e\phi_P = AD_e\,(\phi_P - \phi_E)$.

$$\text{If} \quad F_e < 0, \quad a_E = AD_e - F_e \quad \text{and}$$

$$J_e - F_e\phi_P = (AD_e - F_e)(\phi_P - \phi_E) = AD_e\phi_P - F_e\phi_p - AD_e\phi_E + F_e\phi_E$$

$$\text{or} \quad J_e - F_e\phi_E = AD_e(\phi_P - \phi_E).$$

Similarly, we can write

$$a_W = AD_W + \max[F_w, 0]$$

To summarize, we can write

$$a_E = D_e A(|\,\mathrm{P_e}\,|) + \max[-F_e, 0] \tag{7.51}$$

$$a_W = D_w A(|\,\mathrm{P_e}\,|) + \max[F_w, 0] \tag{7.52}$$

$$\text{and} \quad a_P = (a_E + a_W) + (F_e - F_w). \tag{7.53}$$

Value of the function $A(|\,\mathrm{P_e}\,|)$ depends upon the differencing scheme chosen. Expressions for $A(|\,\mathrm{P_e}\,|)$ for different schemes provided by Patankar (1980) are:

Scheme	Formula for $A(\|\,\mathrm{P_e}\,\|)$
Central Difference	$1 - 0.5\,\|\,\mathrm{P_e}\,\|$
Upwind	1
Hybrid	$\max[0,\ 1 - 0.5\,\|\,\mathrm{P_e}\,\|]$
Power Law	$\max[0, (1 - 0.5\,\|\,\mathrm{P_e}\,\|)^5]$
Exponential	$\|\,\mathrm{P_e}\,\|/\{\exp(\|\,\mathrm{P_e}\,\|) - 1\}$

The differencing schemes and the forms of functions A are discussed in the next subsection. However, for completeness, we can write the following for two dimensional problems

$$a_N = D_n A(|\,\mathrm{P_e}\,|) + \max[-F_n, 0] \tag{7.54}$$

$$\text{and} \quad a_S = D_s A(|\,\mathrm{P_e}\,|) + \max[F_s, 0] \tag{7.55}$$

where the coefficients for the north and the south neighbours (Figure 7.11) are indicated by a_N and a_S respectively.

7.6.6 Discussion on Various Differencing Schemes

(a) Central-difference approximation

The choice of any discretization scheme depends on the relative importance of convection and diffusion. When convection is not very large, central differencing can be used to yield results of high accuracy. For central differencing, we can write the discretized equation (Figure 7.12) as:

$$(\rho u)_e(\phi_E + \phi_P)/2 - (\rho u)_w(\phi_P + \phi_W)/2 = \frac{\Gamma_e(\phi_E - \phi_P)}{\delta x_e} - \frac{\Gamma_w(\phi_P - \phi_W)}{\delta x_w}$$

or

$$a_P\phi_P = a_E\phi_E + a_W\phi_W$$

where

$$a_E = D_e - F_e/2; \quad a_W = D_w + F_w/2.$$

The distance between P and E is δx_e and the distance between W and P is δx_w. We also know that

$$a_P = a_E + a_W$$

and that the scheme is second order accurate. However, in some circumstances we face difficulties with such discretization scheme.

Consider Figure 7.12.

$$\text{Let} \quad F = \rho_u = 4, \quad D = \Gamma/\delta x = 1, \quad \phi_E = 200 \quad \text{and} \quad \phi_W = 100.$$

Obviously ϕ_P should be bounded by the values of 100 and 200. Use of the scheme given above yields

$$a_E = 1 - 2 = -1, \quad a_W = 3, \quad a_P = 2 \quad \text{and}$$

$2\phi_P = -\phi_E + 3\phi_W = 300 - 200 = 100$ i.e., $\phi_p = 50$!! This is unacceptable.

Finally, for central difference scheme, we have the following observations:

- Unrealistic result because some coefficients become negative. This is violation of Rule 2.
- Scheme is satisfactory for only low values of F/D or low cell Peclet numbers.

(b) Upwind scheme

In order to remove the difficulties encountered with the central-difference approximation, the upwind-scheme is suggested. Upwinding means that the value of ϕ at an interface is taken as the value of ϕ at the grid point on the upwind side of that face. The idea is that the scheme should account for the special influence of upstream points. Hence, with reference to Figure 7.13,

$$\phi_e = \phi_P \quad \text{if} \quad F_e > 0; \quad \phi_e = \phi_E \quad \text{if} \quad F_e < 0.$$

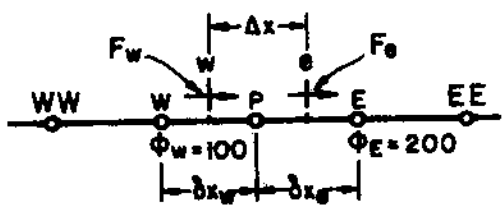


Figure 7.12: Grid-point nomenclature for the one-dimensional problem

Once again we can write the discretized equation as:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W$$

where, $a_E = D_e + \max[-F_e, 0]$, $a_W = D_W + \max[+F_w, 0]$ and $a_P = a_E + a_W$ and $\max[A, B]$ stands for greater of A and B.

The upwind scheme works well from a stability viewpoint, but it is criticized owing to loss of accuracy and false diffusion; this aspect has already been discussed in Chapter 4.

(c) The exact solution and the exponential scheme

The governing equation for steady convection-diffusion in one dimension is

$$\frac{d}{dx}(\rho u \phi) = \frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right)$$

The boundary conditions are (Figure 7.14),

$$x = 0 \quad \phi = \phi_P$$

$$x = \delta x_e \quad \phi = \phi_E$$

This problem can be analytically solved as follows. Writing

$$\frac{d}{dx}\left(\rho u \phi - \Gamma \frac{d\phi}{dx}\right) = 0 \quad \text{and integrating, we get}$$

$$\rho u \phi - \Gamma \frac{d\phi}{dx} = C_1$$

$$\text{or} \quad \frac{d\phi}{\rho u \phi - C_1} = \frac{dx}{\Gamma}$$

$$\text{or} \quad \frac{1}{\rho u} \ln(\rho u \phi - C_1) = \frac{x}{\Gamma} + C_2$$

and finally

$$\phi = \frac{1}{\rho u}\left[e^{\rho u (x/\Gamma + C_2)} + C_1\right]$$

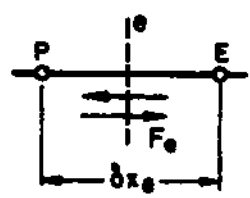


Figure 7.13: Upwind Scheme

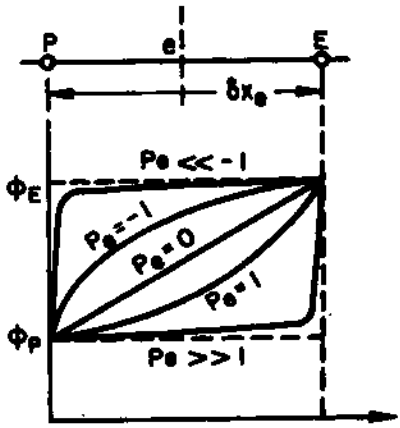


Figure 7.14: Exact solution for the one dimensional convection-diffusion problem

At $x = 0$, $\phi = \phi_P$, and we can write

$$\phi_P = \frac{1}{\rho u}\left(e^{\rho u C_2} + C_1\right) \tag{7.56}$$

At $x = \delta x_e$, $\phi = \phi_E$ and this means

$$\phi_E = \frac{1}{\rho u}\left[e^{\rho u(\delta x_e/\Gamma + C_2)} + C_1\right] \tag{7.57}$$

From Equations 7.56 and 7.57 we get:

$$\begin{aligned} C_1 &= \frac{\rho u}{\left[e^{(\rho u\,\delta x_e)/\Gamma} - 1\right]}\left[e^{(\rho u\,\delta x_e)/\Gamma}\phi_P - \phi_E\right] \\ \text{and} \qquad C_2 &= \frac{1}{\rho u}\left[\ln\frac{\rho u(\phi_E - \phi_P)}{e^{(\rho u \delta x_e)/\Gamma} - 1}\right] \end{aligned} \tag{7.58}$$

Hence we can write

$$\phi = \frac{1}{\rho u}\left[\frac{\rho u(\phi_E - \phi_P)}{\left(-1 + e^{(\rho u \delta x_e)/\Gamma}\right)}e^{(\rho u x)/\Gamma} + \frac{\rho u\left\{\left(e^{(\rho u \delta x_e)/\Gamma}\right)\phi_P - \phi_E\right\}}{\left(e^{(\rho u \delta x_e)/\Gamma} - 1\right)}\right]$$

Since we know the cell Peclet number $\mathrm{P_e} = (\rho u)\,\delta x_e/\Gamma$, we can write

$$\phi = \frac{(\phi_E - \phi_P)\,e^{\rho u x/\Gamma} + \phi_P\,e^{\rho u \delta x_e/\Gamma} - \phi_P + \phi_P - \phi_E}{e^{\rho u \delta x_e/\Gamma} - 1}$$

or

$$\phi = \phi_P + \frac{(\phi_E - \phi_P)\,\left(e^{\rho u x/\Gamma} - 1\right)}{e^{\rho u \delta x_e/\Gamma} - 1}$$

$$\frac{\phi - \phi_P}{\phi_E - \phi_P} = \frac{e^{\rho u x/\Gamma} - 1}{e^{\rho u \delta x_e/\Gamma} - 1}$$

or

$$\frac{\phi - \phi_P}{\phi_E - \phi_P} = \frac{e^{\mathrm{P}_e x/\delta x_e} - 1}{e^{\mathrm{P}_e} - 1} \tag{7.59}$$

This is the exact solution based on which the equations may be discretized between any two points, P and E. The nature of exact solution is understood from Figure 7.14.

Let us use the exact solution (Equation 7.59) to compute ϕ_e and ϕ_w in the difference equation

$$(\rho u)_e\, \phi_e - (\rho u)_w\, \phi_w = \Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} - \Gamma_w \frac{\phi_P - \phi_W}{\delta x_w}$$

or

$$a_P \phi_P = a_E\, \phi_E + a_W\, \phi_W$$

The coefficients a's (invoking Equation 7.59) are

$$a_E = F_e/\{\exp\,(\mathrm{P_e}) - 1\}$$

$$a_W = F_w \exp\,(P_w)/\{\exp\,(P_w - 1)\}$$

$$a_P = a_E + a_W \quad \text{and} \quad \mathrm{P_e} = F_e/D_e$$

The discretization done following the exact solution is known as the *exponential scheme*. Despite its accuracy and highly desirable behaviour, it is not very widely used because exponentials are expensive to compute.

(d) The power-law scheme

Earlier section already pointed out that for one-dimensional convection-diffusion problem, the exponential scheme provide exact results. While it is time-consuming, an efficient scheme that closely approximates the exponential is the *power-law* scheme (Patankar, 1980) is discussed as follows:

The discretized equation in one-dimension may be once again written as

$$a_p\, \phi_P = a_E\, \phi_E + a_W\, \phi_W$$

The coefficients due to power-law scheme are

$$a_E = D_e \max[0, (1 - 0.1 \mid \mathrm{P_e} \mid)^5] + \max[0, -F_e]$$

$$a_W = D_w \max[0, (1 - 0.1 \mid P_w \mid)^5] + \max[0, F_w]$$

$$a_P = a_E + a_W$$

From the plot of a_E/D_e as a function of $\mathrm{P_e}$, we can appreciate the accuracy of power-law scheme (Figure 7.15).

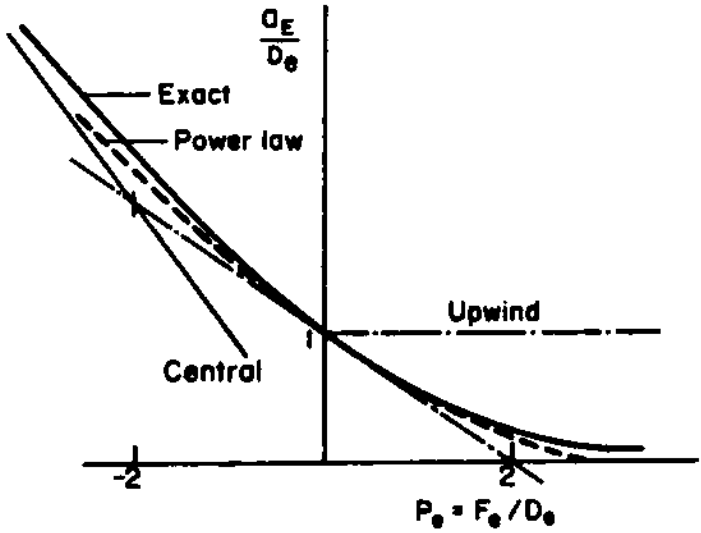


Figure 7.15: Variation of the Coefficient a_E with Peclet number

7.7 Formulation of the Flow Problem

7.7.1 Introduction

We now discuss the SIMPLE algorithm applied to the incompressible flow equations. The momentum equations are given by

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) + S \tag{7.60}$$

where S refers to body force terms. Consider the flow to be laminar and Newtonian. Following are the salient features of the flow field calculation.

- Momentum equations have the same form as the general ϕ equation as discussed earlier. Thus we can use the discretization methods described earlier to solve the momentum equations.
- Non-linearity is resolved through iteration.
- The pressure gradient term in the momentum equation causes difficulty. There is no obvious equation for pressure.
- Pressure is indirectly specified by the continuity equation. A correct pressure field when substituted into the momentum equation will yield a velocity field that satisfies the continuity equation.

The two-dimensional form of the flux equation can be written as

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = S \tag{7.61}$$

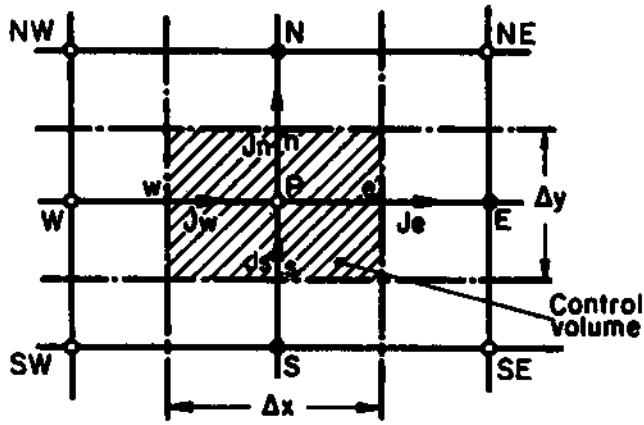


Figure 7.16: Control volume for two dimensional flow field

where J_x and J_y are the total (convection plus diffusion) fluxes defined by

$$J_x = \rho u \phi - \Gamma \frac{\partial \phi}{\partial x}$$

and

$$J_y = \rho v \phi - \Gamma \frac{\partial \phi}{\partial y}.$$

u and v are velocity components in x and y directions. Integration of the flux equation over the control volume (Figure 7.16) will produce

$$\frac{(\rho_P \, \phi_P - \rho_P^o \, \phi_P^o) \, \Delta x \Delta y}{\Delta t} + J_e - J_w + J_n - J_s = (S_c + S_P \phi_P) \, \Delta x \, \Delta y \quad (7.62)$$

where the source term has been linearized in the usual manner. The unsteady terms ρ_P and ϕ_P are assumed to be uniform over the whole control volume. These are current values (at the end of time step). The initial values (i.e., the values at the beginning of the time step) are denoted by ρ_P^o and ϕ_P^o. The fluxes are total fluxes over the control volume faces. For example $J_e = \int J_x dy$, and so on.

Now consider the continuity equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) = 0 \quad (7.63)$$

This may be integrated over the control volume and the resulting equation becomes

$$\frac{(\rho_P - \rho_P^o) \, \Delta x \Delta y}{\Delta t} + F_e - F_w + F_n - F_s = 0 \quad (7.64)$$

where F_e, F_w, F_n and F_s are the mass flow rates through the faces of control volume. If (ρu) at point e is considered as mass flux over the interface, then we

can write

$$F_e = (\rho u)_e \, \Delta y, \quad F_w = (\rho u)_w \, \Delta y, \quad F_n = (\rho v)_n \, \Delta x, \quad \text{and} \quad F_s = (\rho v)_s \, \Delta x.$$

Multiply Equation 7.64 by ϕ_P and subtract from Equation 7.62 to obtain

$$\begin{aligned} \frac{(\phi_P - \phi_P^o)\, \rho_P^o \Delta x \Delta y}{\Delta t} &+ (J_e - F_e\, \phi_P) - (J_w - F_w\, \phi_P) \\ &+ (J_n - F_n\, \phi_P) - (J_s - F_s\, \phi_P) \\ &= (S_c + S_p\, \phi_p)\, \Delta x\, \Delta y \end{aligned} \tag{7.65}$$

At this point, we recall that Equation 7.50 provided a method for expressing terms such as $(J_e - F_e\, \phi_P)$. Similarly $(J_w - F_w\, \phi_P)$ and other terms may be expressed. With this approach we write

$$J_e - F_e \phi_P = a_E\, (\phi_P - \phi_E) \tag{7.66}$$

$$J_w - F_w \phi_P = a_W\, (\phi_W - \phi_P) \tag{7.67}$$

$$J_n - F_n \phi_P = a_N\, (\phi_P - \phi_N) \tag{7.68}$$

$$J_s - F_s \phi_P = a_S\, (\phi_S - \phi_P) \tag{7.69}$$

Hence, the discretization Equation 7.65 can be written as

$$a_P\, \phi_P = a_E\, \phi_E + a_W\, \phi_W + a_N\, \phi_N + a_S\, \phi_S + b \tag{7.70}$$

where

$$a_E = D_e\, A\, (|\, \mathrm{P_e}\, |) + \max[-F_e, 0] \tag{7.71}$$

$$a_W = D_w\, A\, (|\, P_w\, |) + \max[F_w, 0] \tag{7.72}$$

$$a_N = D_n\, A\, (|\, P_n\, |) + \max[-F_n, 0] \tag{7.73}$$

$$a_S = D_s\, A\, (|\, P_s\, |) + \max[F_s, 0] \tag{7.74}$$

$$a_P = a_E + a_W + a_N + a_S + a_P^o - S_P\, \Delta x\, \Delta y \tag{7.75}$$

$$b = S_c\, \Delta x\, \Delta y + a_P^o\, \phi_P^o \tag{7.76}$$

$$a_P^o = \frac{\rho_P^o \Delta_x\, \Delta_y}{\Delta t} \tag{7.77}$$

In the above expressions, ϕ_P^o and ρ_P^o refer to the known values of ϕ_P and ρ_P at time t.

All other values ($\phi_P, \phi_E, \phi_N, \phi_S$ and so on) are unknown values at time $(t + \Delta t)$. The flow rates have already been defined as

$$F_e = (\rho u)_e\, \Delta y$$

$$F_w = (\rho u)_w\, \Delta y$$

$$F_n = (\rho v)_n\, \Delta x$$

$$F_s = (\rho v)_s \, \Delta x$$

and the corresponding conductances are

$$D_e = \frac{\Gamma_e}{\delta x_e} \Delta y \tag{7.78}$$

$$D_w = \frac{\Gamma_w}{\delta x_w} \Delta y \tag{7.79}$$

$$D_n = \frac{\Gamma_n}{\delta y_n} \Delta x \tag{7.80}$$

$$D_s = \frac{\Gamma_s}{\delta y_s} \Delta x \tag{7.81}$$

The cell Peclet numbers are $P_e = F_e/D_e$, $P_w = F_w/D_w$, $P_n = F_n/D_n$ and $P_s = F_s/D_s$ respectively. The function $A(|P|)$ depends on discretization schemes. If power-law scheme is used, then

$$A(|\ P\ |) = \max\ [0, (1 - 0.1\ |\ P\ |)^5]$$

The physical significance of the various coefficients of Equation 7.70 can now be appreciated. The neighbour coefficients a_E, a_W, a_N and a_S represent the convection and diffusion influence at four faces of the control volume in terms of flow rate F and conductance D. The term $a_P^o \, \phi_P^o$ is the known ϕ content of the control volume (at time t) divided by time-step.

7.7.2 Momentum Equation

It can be said that, if the pressure field is given, the solution of the momentum equations can be obtained by employing the formulation described above for the general variable ϕ. In momentum equation ϕ stands for relevant velocity component, and Γ and S are attributed to their appropriate meaning of viscosity and source term. The use of the staggered grid makes the discretized momentum equation somewhat different from the discretization equations for ϕ's that are calculated for the main grid points.

A staggered control volume for x momentum equation is shown in Figure 7.17. Its faces lie between the point e and the corresponding locations for u neighbours. The control volume is, however, staggered in relation to normal control volume around the main grid point P. This layout emphasises one of the main advantages of the staggered grid. The difference $P_P - P_E$ can be used to calculate the pressure force acting on the control volume for the velocity u. The calculation of diffusion coefficient and mass flow rate at the faces of u control volume would require a calculation similar to Equation 7.70. The resulting discretization equation can be written as

$$a_e \, u_e = \sum a_{nb} \, u_{nb} + b + (P_P - P_E) \, A_e \tag{7.82}$$

The four u_{nb} neighbours are shown outside the control volume. The neighbour coefficients a_{nb} account for the combined convection-diffusion influence at the control-volume faces. The term b is defined in the same manner as in Equation 7.70, but the pressure gradient is not included in the source term quantities S_c and S_p and is separately written as $(P_P - P_E)\,A_e$ which makes it convenient to determine a pressure field for our purpose. A_e is the area on which the differential pressure acts.

7.7.3 Discretized Continuity and Momentum Equations

Using a staggered grid we can write the following discretized equations in two dimensions

- Continuity (refer to Main Control Volume in Figure 7.18)

$$\frac{(\rho_P - \rho_P^o)\,\Delta x\,\Delta y}{\Delta t} + \{(\rho u)_e - (\rho u)_w\}\Delta y + \{(\rho v)_n - (\rho v)_s\}\,\Delta x = 0 \quad (7.83)$$

- X-momentum (refer to U-Control Volume in Figure 7.18)

$$a_e u_e = \sum a_{nb}\,u_{nb} + b + A_e(P_P - P_E) \quad (7.84)$$

- Y-momentum (refer to V-Control Volume in Figure 7.18)

$$a_n\,v_n = \sum a_{nb}\,v_{nb} + b + A_n\,(P_P - P_N) \quad (7.85)$$

The momentum equation can be solved only when the pressure field is given or is somehow estimated. Unless the correct pressure field is employed, the resulting velocity field will not satisfy the continuity equation. Such an imperfect velocity field based on a guessed pressure field P^* will be denoted by u^*, v^*. This starred velocity field will result from the solution of the following discretization equations

$$a_e\,u_e^* = \sum a_{nb}\,u_{nb}^* + b + (P_P^* - P_E^*)\,A_e$$

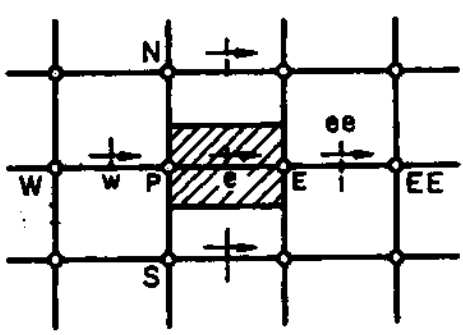


Figure 7.17: Control volume for velocity component u

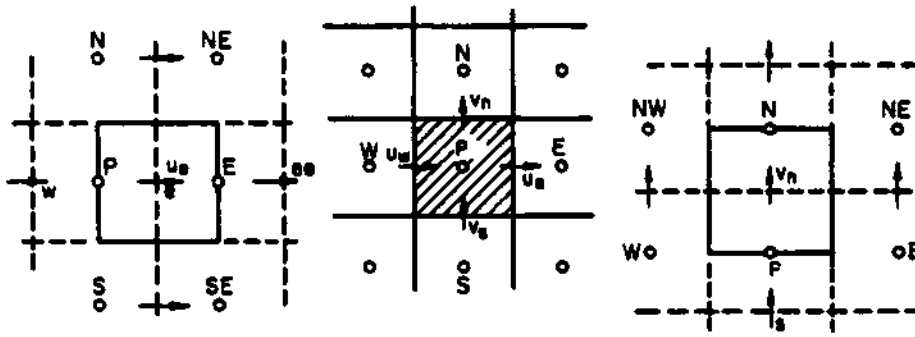


Figure 7.18: (a) U-Control Volume, (b) Main Control Volume and (c) V-Control Volume

$$a_n v_n^* = \sum a_{nb} v_{nb}^* + b + (P_P^* - P_N^*) A_n$$

We want to find out a way of improving the guessed pressure P^* such that the resulting starred velocity field will progressively get closer to satisfying the continuity equation. Let us propose that the correct pressure P is obtained from

$$P = P^* + P'$$

where P' will be called the pressure correction. Next, we would be interested in knowing how the velocities will respond to the changed pressure.

7.7.4 The Pressure Correction Equation

For a correct pressure P, we can write

$$a_e u_e = \sum a_{nb} u_{nb} + b + A_e (P_P - P_E)$$

For a guessed pressure p^*, we get

$$a_e u_e^* = \sum a_{nb} u_{nb}^* + b + A_e (P_P^* - P_E^*) \tag{7.86}$$

Subtracting Equation 7.86 from 7.84 and denoting $P - P^* = P'$, we get

$$a_e (u_e - u_e^*) = \sum a_{nb} (u_{nb} - u_{nb}^*) + A_e (P_P' - P_E') \tag{7.87}$$

We neglect first term on right hand side of Equation 7.87 to get velocity correction equation

$$u_e = u_e^* + d_e (P_P' - P_E') \quad \text{where,} \quad d_e = \frac{A_e}{a_e} \tag{7.88}$$

Similarly we can write

$$v_n = v_n^* + d_n (P_P' - P_N') \tag{7.89}$$

The next task is to substitute Equations 7.88 and 7.89 into discretized continuity equation developed from Equation 7.83. Equation 7.83 may be rewritten as

$$\frac{(\rho_P - \rho_P^o)}{\Delta t}\,\Delta x\,\Delta y + \rho_e\,u_e\,\Delta y - \rho_w\,u_w\,\Delta y + \rho_n v_n\,\Delta x - \rho_s v_s \Delta x = 0 \quad (7.90)$$

The resulting equation from Equations 7.88, 7.89 and 7.90 yields the pressure correction for P' equation as:

$$a_P P'_P = a_E P'_E + a_W P'_W + a_N P'_N + a_S P'_S + b \quad (7.91)$$

where

$$\begin{aligned} a_E &= \rho_e\,d_e\,\Delta y && (7.92)\\ a_W &= \rho_w\,d_w\,\Delta y && (7.93)\\ a_N &= \rho_n\,d_n\,\Delta x && (7.94)\\ a_S &= \rho_s\,d_s\,\Delta x \quad \text{and} && (7.95)\\ a_P &= a_E + a_w + a_N + a_S && (7.96) \end{aligned}$$

and

$$b = -\left[\frac{\rho_P - \rho_P^o}{\Delta t}\,\Delta x\,\Delta y + \rho_e\,u_e^*\,\Delta y - \rho_w\,u_w^*\,\Delta y + \rho_n v_n^* \Delta x - \rho_s v_s^* \Delta x\right] \quad (7.97)$$

Note that b in pressure correction equation is equivalent to the continuity equation for the starred velocity field. If u^* and v^* assume the correct values, b will be zero. Also when P' obtained by solving the above equation is substituted into the velocity correction equations, we get $u_e = u_e^* + d_e\,(P'_p - P'_E)$, $v_n = v_n^* + d_n\,(P'_p - P'_N)$, and so on. The resulting velocity field exactly satisfies the continuity equation.

Before proceeding, further, we would like to quickly mention the evaluation technique of the momentum and pressure correction equation.

7.7.5 Two-dimensional System of Equations and Line-by-line TDMA

Whether the discretized momentum equations (7.84 and 7.85) or the pressure correction equation 7.91, the final outcome is a system of algebraic equation given by $a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b$. The current values of the dependent variables (ϕ's) are to be evaluated from the known values of the coefficient (a's). The evaluation algorithm will be same for the momentum equations and the pressure correction equation. On a rectangular grid, the dependent variables at one point (i, j) may be expressed in terms of its neighbours (Figure 7.19) as

$$a_{i,j}u_{i,j} = b_{i,j}\,u_{i+1,j} + c_{i,j}\,u_{i-1,j} + d_{i,j}u_{i,j+1} + e_{i,j}\,u_{i,j-1} + f_{i,j}$$

where, $b_{i,j}$ is equivalent to $a_E, c_{i,j}$ to $a_W, d_{i,j}$ to a_N, $e_{i,j}$ to a_S and $f_{i,j}$ to b. The evaluation of u can be accomplished in the following ways:

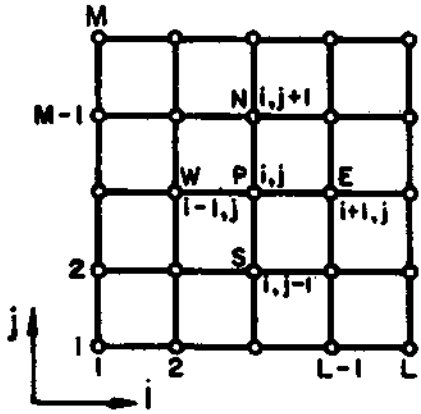


Figure 7.19: Dependent variable at (i, j) expressed in terms of neighbouring points

(a) Vertical Sweep Upwards may be written in a pseudo FORTRAN code as

$$\text{DO} \quad 10 \quad J = 2, M-1$$

$$\text{DO} \quad 10 \quad I = 2, L-1$$

$$10 \quad a_{ij}\, u_{i,j} = b_{ij}\, u_{i+1,j} + c_{ij}\, u_{i-1,j} + (d_{ij}\, \bar{u}_{i,j+1} + e_{ij} \bar{u}_{i,j-1} + f_{ij})$$

Here $\bar{u}$ is the currently available value in storage and all the coefficients including $f_{i,j}$ are known. For each j, we shall get a system of equations if we substitute $i = 2 \ldots L-1$. In other words, for each j, a tridiagonal matrix is available which can be solved for all i row of points at that j. Once one complete row is evaluated for any particular j, the next j will be taken up, and so on.

(b) Horizontal sweep forward may be written in pseudo FORTRAN code as

$$\text{DO} \quad 20 \quad I = 2, L-1$$

$$\text{DO} \quad 20 \quad J = 2, M-1$$

$$20 \quad a_{ij}\, u_{i,j} = d_{ij}\, u_{i,j+1} + e_{ij}\, u_{i,j-1} + (b_{ij}\, \bar{u}_{i+1,j} + c_{ij} \bar{u}_{i-1,j} + f_{ij})$$

Again $\bar{u}$ is the currently available value in storage from previous calculations. For each i, we get a system of equation if we substitute $j = 2 \ldots M-1$. A tridiagonal matrix is available for each i. Once one complete column of points are evaluated for any particular i, the next i will be taken up and so on.

The vertical sweep upward and downward are repeated. Similarly the horizontal sweep forward rearward are also repeated until convergence is achieved.

For solving the linear system, the tridiagonal matrix algorithm (TDMA) due to Thomas (1949) is utilized. The above mentioned evaluation procedure is known as line-by-line TDMA.

7.8 SIMPLE Algorithm

We now consolidate the development given in Sections 7.6 and 7.7. The resulting scheme is called as SIMPLE. The salient operations of this method in order of their execution, as suggested by Patankar (1980), are:

1. Guess P^*
2. Solve momentum equation to get u^*, v^*
3. Calculate the source term b for P' equation. If $b = 0$ STOP
4. Solve P' equation
5. Update velocities using $u_e = u_e^* + d_e\,(P_p' - P_E')$, $\quad v_n = v_n^* + \cdots$
6. Update pressure using $P = P^* + \alpha_u P'$ with α_u set equal to about 0.8
7. Solve for other variables
8. Return to step 2 and repeat until convergence

The SIMPLE algorithm and its variants have dominated the entire field of numerical simulation of incompressible flows, over the past two decades. Some modifications to the application of SIMPLE have been suggested by Van Doormal and Raithby (1984). Garg and Maji (1987) have applied this modified method of Van Doormal and Raithby (called SIMPLEC) for the solution of viscous flows through periodically converging-diverging pipes. They do not have, whatsoever, any limit to the ratio of wall amplitude to pitch. Figure 7.20 shows streaklines of the periodic mixed-convection flow of a radial jet on an impingement plate. These results have been obtained by Potthast *et al.* (1994). Karki (1986) has extended SIMPLE algorithm for solving viscous flow problems in complex geometries. These problems include both incompressible and compressible flows with shocks.

In recent years, much effort has been put into the development of numerical procedures for the calculation of steady incompressible three-dimensional flows in complex geometries. SIMPLE based finite volume methods employing cartesian velocity components and non-staggered variable arrangement have been successfully used to obtain steady and unsteady solutions of 3-D Navier-Stokes equations on curvilinear boundary fitted grids. Vanka *et al.* (1980) in their three-dimensional solution, use the curvilinear coordinates and store the cartesian velocity components at the corners of the control volumes and pressure at the centre. The interest in using collocated grid arrangements for boundary fitted curvilinear coordinates has grown since Peric (1985) published an

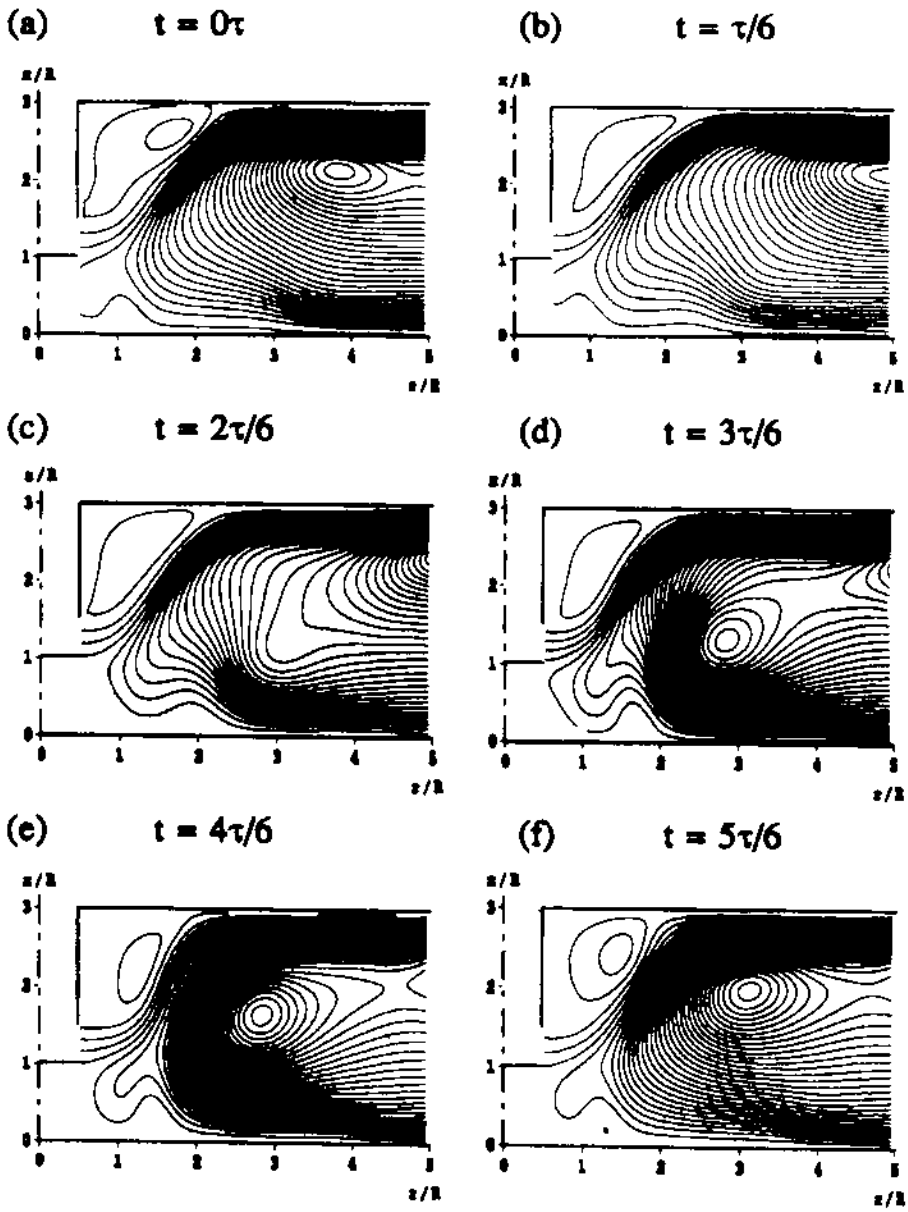


Figure 7.20: Streaklines of the periodic mixed-convection flow of a radial jet impinging on a plate; Re = 200, Gr = 10000, τ is the time period.

application of SIMPLE algorithm based on this approach. Some of the contributions of Kost *et al.* (1991) and Majumdar *et al.* (1992) may be mentioned in connection with the development of SIMPLE based finite volume methods for solving incompressible flows in arbitrary geometry. A SIMPLE-like finite volume transient algorithm has been developed by Mukhopadhyay *et al.* (1993) for viscous flows in irregular geometry. The algorithm uses some of the ideas of finite element methods such as element-wise interpolation and transformation of nonorthogonal element geometry into a square computational element while solving the integral conservation equations. Eswaran and Prakash (1998) have proposed a finite volume based algorithm for solving the incompressible fluid flow equations in a complex geometry. The algorithm has been successfully applied to solve flow and heat transfer problems in a channel with built-in oval tube and vortex generators (Prabhakar *et al.*, 2002).

References

Biswas, G. and Chattopadhyay, H., Heat Transfer in a Channel with Built-in Wing-type Vortex Generators, Int. J. Heat Mass Transfer, Vol 35, pp 803-814, 1992.

Brandt, A., Dendy, J. E. and Ruppel, H., the Multigrid Method for Semi-Implicity Hydrodynamic Codes, Journal of Comput. Phys., Vol 34, pp 348-370, 29-1980.

Braza, M., Chassaing, P. and Ha Minh, H., Numerical Study and Physical Analysis of the Pressure and Velocity Fields in the Near Wake of a Circular Cylinder, J. Fluid Mech., Vol 165, pp 79-130, 1986.

Chorin, A. J., A Numerical Method for Solving Incompressible Viscous Flow Problems, Journal of Comput. Phys., Vol 2, pp 12-26, 1967.

Davis, R. W., and Moore, E. F., A Numerical Study of Vortex Shedding from Rectangles, J. Fluid Mech., Vol 116, pp 475-506, 1982.

Garg, V. K. and Maji, P. K., Flow through a Converging-Diverging Tube with Constant Wall Enthalpy, Numerical Heat Transfer, Vol 12, pp 285-305, 1987.

Harlow, F. H. and Welch, J. E., Numerical Calculation of Time-dependent Viscous Incompressible Flow of Fluid with Free Srufaces, The Phys. of Fluids, Vol 8, pp 2182-2188, 1965.

Hirt, C. W. and Cook, J. L., Calculating Three Dimensional Flows around Structures and Over Rough Terrain, Journal of Comput. Phys., Vol 10, pp 324-340, 1972.

Hirt, C. W. Nichols, B. D. and Romero, N. C., SOLA - A Numerical Solution Algorithm for Transient Fluid Flows, LA - 5852, Los Alamos Scientific Laboratory Report, 1975.

Eswaran, V. and Prakash, S., A Finite Volume Method for Navier-Stokes Equations, Proceedings of the Third Asian CFD Conference, Bangalore, Vol. 1, pp 127-133, 1998.

Karki, K. C., A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries, Ph.D. thesis, University of Minnesota, 1986.

Kim, J. and Moin, P., Application of a Fractional Step Method to Incompressible Navier-Stokes Equations, Journal of Comput. Phys., Vol 59, pp 308-323, 1985.

Kim, S. W. and Benson, T. J., Comparison of the SMAC, PISO and Iterative Time-Advancing Schemes for Unsteady Flows, Computers Fluids, Vol 21, pp 435-454, 1992.

Kost, A., Bai, L., Mitra, N. K. and Fiebig, M., Calculation Procedure for Unsteady Incompressible 3-D Flows in Arbitrary Shaped Domains, Proc of the Ninth GAMM-Conference on Numerical Methods in Fluid Mechanics, pp 269-278, 1991.

Leonard, B. P., A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, Comp. Methods Appl. Mech. Engr., Vol 19, pp 59-98, 1979.

Majumdar, S. Rodi, W. and Zhu, J., Three-Dimensional Finite-Volume Method for Incompressible Flows with Complex Boundaries, Journal of Fluids Engineering (ASME), Vol 114, pp 496-503, 1992.

Mukhopadhyay, A., Biswas, G. and Sundararajan, T., Numerical Investigation of Confined Wakes behind a Square Cylinder in a Channel, Int. J. Numer. Methods Fluids, Vol 14, pp 1473-1484, 1992.

Mukhopadhyay, A., Sundararajan, T. and Biswas, G., An Explicit Transient Algorithm for Predicting Incompressible Viscous Flows in Arbitrary Geometry, Int. J. Numer. Methods Fluids, Vol 17, pp 975-993, 1993.

Okajima, A., Strouhal Numbers of Rectangular Cylinders, J. Fluid. Mech., Vol 123, pp 379-398, 1982.

Orlanski, I., A Simple Boundary Condition for Unbounded Flows, J. Computational Phys., Vol. 21, pp 251-269, 1976.

Patankar, S. V., A Calculation Procedure for Two-dimensional Elliptic Situations, Numerical Heat Transfer, Vol 4, pp 409-425, 1981.

Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Co., New York, 1980.

Patankar, S. V. and Spalding, D. B., A Calculation Procedure for Heat Mass and Momentum Transfer in Three Dimensional Parabolic Flows, Int. J. Heat and Mass Transfer, Vol 15, pp 1787-1805, 1972.

Peric, M., A Finite Volume Method for the Prediction of Three Dimensional Fluid Flow in Complex Ducts, Ph.D. thesis, University of London, 1985. Also, Ferziger, J.H. and Peric, M., *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, (third edition) 2002.

Peyret, R. and Taylor, T. D., *Computational Methods for Fluid Flow*, Springer Verlag, New York, 1983.

Potthast, F., Laschefski, H., Mitra, N. K. and Biswas, G., Numerical Investigation of Flow Structure and Mixed Convection Heat Transfer of Impinging Radial and Axial Jets, Numerical Heat Transfer - Part A, Vol 26, pp 123-140, 1994.

Prabhakar, V., Biswas, G., and Eswaran, V., Numerical Prediction of Heat Transfer in a Channel with a Built-in Oval Tube and two differently shaped Vortex Generators, Numerical Heat Transfer Part A , Vol. 41, pp 307-329, 2002.

Robichaux, J., Tafti, D.K., and Vanka, S.P., Large-eddy Simulation of Turbulence on CM-2, Numerical Heat Transfer, Part B, Vol. 21, pp 367-388, 1992.

Saha, A.K., Biswas, G., and Muralidhar, K., Numerical Study of the Turbulent Unsteady Wake behind a Partially Enclosed Square Cylinder using RANS, Computer Methods in Applied Mechanics and Engineering, Vol 178, pp 323-341, 1999.

Thomas, L. H., Elliptic Problems in Linear Difference Equations over a Network, Watson Sci. Comput. Lab. Rept., Columbia University, New York, 1949.

Van Doormal, J. P. and Raithby, G. D., Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows, Numerical Heat Transfer, Vol 7, pp 147-163, 1984.

Vanka, S. P., Chen, B.C.-J. and Sha, W. T., A Semi-Implicit Calculation Procedure for Flow Described in Body-Fitted Coordinate Systems, Numerical Heat Transfer, Vol 3, pp 1-19, 1980.

Viecelli, J. A., A Computing Method for Incompressible Flows Bounded by Moving Walls, Journal of Comput. Phys., Vol 8, pp 119-143, 1971.

Problems

1. Develop the MAC and SIMPLE algorithms describes in this chapter for a steady one dimensional flow field $u(y)$, driven by a constant (but as yet undetermined) pressure gradient. For definiteness, flow in a parallel plate channel can be considered.
2. For a two-dimensional flow, show that the MAC type iterative correction of pressure and velocity field through the implicit continuity equation is equivalent to the solution of Poisson equation for pressure. How does the procedure avoid the need of applying pressure-boundary-conditions directly?
3. Compare the solutions obtained by the stream function-vorticity method with those presented in this chapter for the following problems:
   (a) Developing flow in a parallel plates channel;
   (b) Flow past a circular cylinder (Re $<$ 200);
   (c) Flow past a periodic array of cylinders.
4. Apply MAC method to solve the shear driven cavity flow problem shown in Figure 7.21. Take a grid size of 51×51 and solve the flow equations for Re $= 400$.

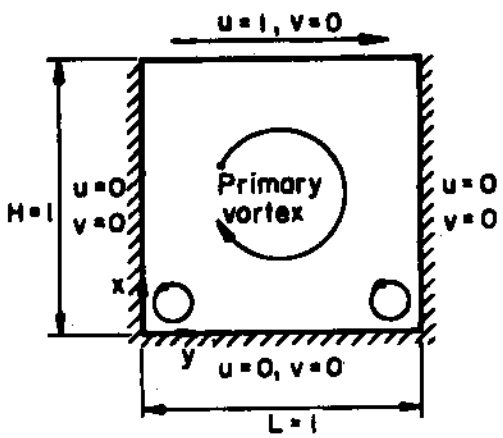


Figure 7.21: Shear driven Cavity

5. The Navier-Stokes equations in three-dimensional spherical-polar coordinate system in θ direction may be written as

$$\rho\left(\frac{\partial V_\theta}{\partial t}+V_r\frac{\partial V_\theta}{\partial r}+\frac{V_\theta}{r}\frac{\partial V_\theta}{\partial \theta}+\frac{V_\phi}{r\sin\theta}\cdot\frac{\partial V_\theta}{\partial \phi}+\frac{V_rV_\theta}{r}-\frac{V_\phi^2\cot\theta}{r}\right)$$

$$= -\frac{\partial p}{r\partial\theta}+\mu\left[\frac{\partial^2 V_\theta}{\partial r^2}+\frac{2}{r}\frac{\partial V_\theta}{\partial r}+\frac{1}{r^2}\frac{\partial V_\theta}{\partial \theta^2}+\frac{\cot\theta}{r^2}\frac{\partial V_\theta}{\partial\theta}\right.$$

$$\left.+\frac{1}{r^2\sin^2\theta}\cdot\frac{\partial^2 V_\theta}{\partial\phi^2}+\frac{2}{r^2}\frac{\partial V_r}{\partial\theta}-\frac{V_\theta}{r^2\sin^2\theta}-\frac{2\cot\theta\csc\theta}{r^2}\frac{\partial V_\theta}{\partial\phi}\right]$$

The continuity equation is given by

$$\frac{\partial(\rho V_r)}{\partial r}+\frac{2}{r}(\rho V_r)+\frac{1}{r}\frac{\partial}{\partial\theta}(\rho V_\theta)+\frac{\cot\theta}{r}\rho V_\theta+\frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}(\rho V_\phi)=0.$$

Discretize the weak conservative form of the θ-momentum equation on a grid shown in Figure 7.22 using a weighted-average scheme.

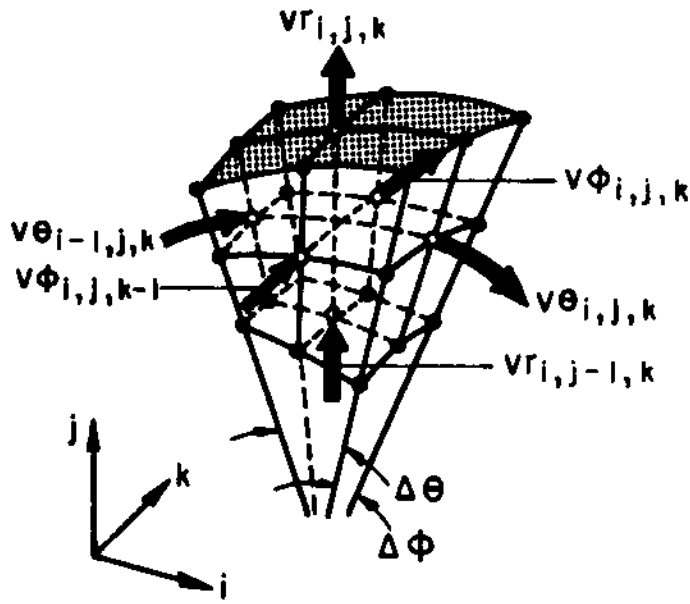


Figure 7.22: Location of velocity components for a staggered grid arrangement in a spherical polar coordinate system.

6. Obtain the exact solution of the equation given by

$$\frac{d}{dx}\left(\rho uT-\Gamma\frac{\partial T}{\partial x}\right)=0$$

where, ρu is the convective flux at point e (refer to Figure 7.11), and Γ is the diffusion coefficient. Both ρu and Γ are constant over the control volume. The boundary conditions are: at $x=0$, $T=T_P$ and at $x=\delta x_e$, $T=T_E$.

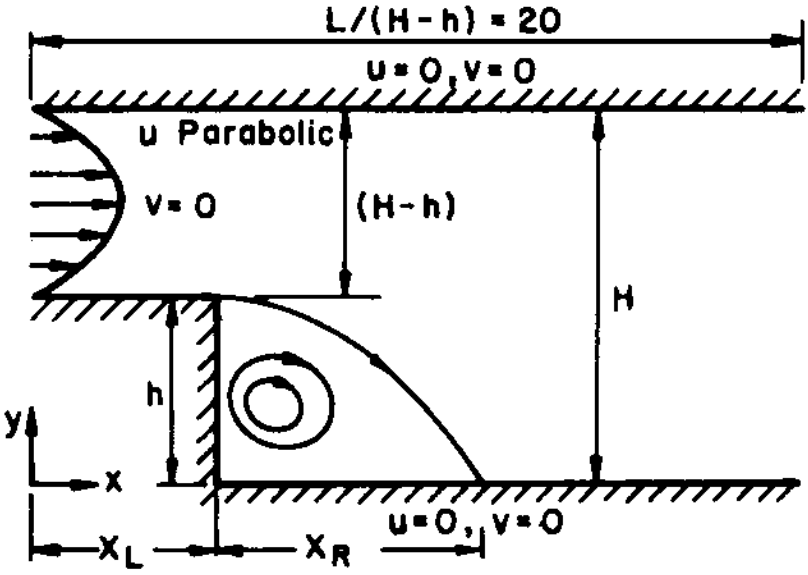


Figure 7.23: Flow over a Backward Facing Step

7. Solve two-dimensional Navier-Stokes equation for the flow over a backward-facing-step problem illustrated in Figure 7.23. Use SIMPLE algorithm and a 81×41 grid in the region $0 \leq X \leq X_L$. In the region, $X_L \leq X \leq L$ you can take a grid of your choice. Find out the value of the reattachment length X_R for Reynolds numbers of 200, 300 and 400. The distance $X_L = (H - h)$ and $\mathrm{Re} = U_{\max}\,(H - h)/\nu$.

8. Repeat the channel flow problems given above in the turbulent flow regime using an appropriate algebraic eddy viscosity model (Chapter 1).

9. Consider the channel flow and backward-facing-step configurations once again. Combine the flow and thermal energy equations and determine the local and global heat transfer rates when one of the solid surfaces is heated, the incoming fluid is cold and all other solid surfaces are thermally insulated.

8
Finite Element Method

8.1 Introduction

The chapter introduces finite element method (FEM) as applied to the solution of flow and heat transfer problems. FEM uses concepts such as interpolation and weighted integration which are not explicitly visible in the finite difference method. Development of a computer code that employs FEM is more time consuming compared to an equivalent code that uses finite differences. The additional effort is usually justified since for a class of linear problems it can be rigorously shown that the Galerkin FEM solution is the *best* in comparison to all other solutions on a prescribed grid. Here the comparison is in terms of the error which is the difference between the approximate numerical solution and the analytical solution of the governing differential equation. This result does not carry over to flow problems. However other advantages accrued through the use of FEM such as easy treatment of complex geometry and gradient boundary conditions make it competetive with the finite difference method.

From a mathematical point-of-view, the finite element method is an extension of various numerical techniques that are employed with variational principles and the method of least squares. It is a computationally efficient means of implementing these techniques. What is new is the choice of trial functions; in FEM these are piecewise polynomials. FEM was originally developed for the calculation of stresses in complex structures made of discrete interconnected elements. In these problems it has the following physical interpretation. A structure is an assembly of individual structural components *(finite elements)* that can deform within the elastic limit. They are joined at a finite number of points *(nodes)*. In each element the displacement is assumed to be a simple function of position, usually a polynomial. The solution corresponds to the structural configuration that minimizes the total potential energy arising from external loads and internal strain (Zienkiewicz and Taylor, 1991). This physical interpretation has led to widespread use of FEM in continuum mechanics.

8.2 Weighted Residual and Variational Formulations

For simplicity the initial discussion is restricted to linear differential equations. Consider the differential equation

$$Lu = f(\mathbf{x}) \tag{8.1}$$

where the solution $u(\mathbf{x})$ must be determined. Here L is a linear differential operator and is taken for definiteness as

$$L = -\left(\frac{\partial}{\partial x_i}(A(\mathbf{x})\frac{\partial u}{\partial x_j}\right) + q(\mathbf{x})u$$

and f, q and matrix A are prescribed functions of the spatial coordinate $\mathbf{x}$. For

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$q = 0$ and $f = 0$, Equation 8.1 reduces to the Laplace equation $u_{xx} + u_{yy} = 0$ in Cartesian coordinates. In two dimensions Equation 8.1 can be shown to be elliptic. Let Ω be the extent of the physical domain and $\partial\Omega$ the boundary of Ω. We assume a Dirichlet boundary condition u prescribed on $\partial\Omega$ for Equation 8.1.

In the definition of the operator L, i and j take the values of $1..n_d$, where n_d is the dimensionality of the physical space and $x_i = (x, y, z)$ depending on whether $i = 1$, 2 or 3. Repeated indices indicate summation; if A_{ij} are elements of matrix A, the expanded form of Equation 8.1a in two dimensions is

$$L = -\frac{\partial}{\partial x}\,(A_{11}\frac{\partial u}{\partial x}) - \frac{\partial}{\partial x}\,(A_{12}\frac{\partial u}{\partial y}) - \frac{\partial}{\partial y}\,(A_{21}\frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}\,(A_{22}\frac{\partial u}{\partial y}) + qu$$

Engineering problems are commonly formulated in terms of differential equations. However it might be more convinient to set up a variational principle for a certain class of physical problems. In two dimensions this principle is of the form

$$I(u) = \int_{\Omega} F\,(\mathbf{x}, u, u_x, u_y)\,d\mathbf{x} = \text{extremum} \tag{8.2}$$

where $u = u(\mathbf{x})$ is the dependent variable that must be determined and $\mathbf{x} = (x, y)$. The precise form of F is known from the physical principle governing the problem. Numerical methods can be directly applied to the variational principle to generate approximate discrete values of u. Alternatively a differential form can be derived for u using the Euler-Lagrange equation

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial x}(\frac{\partial F}{\partial u_x}) - \frac{\partial}{\partial y}(\frac{\partial F}{\partial u_y}) = 0$$

In practice obtaining an expression for F from the differential equation $Lu = f$ is quite difficult. However for the linear operator defined by Equation 8.1a, F can be explicitly shown to be

$$F = u(Lu - 2f)$$

It can be easily verified that the Euler-Lagrange equation for this functional is equivalent to the differential equation $Lu = f$.

The difficulty of deriving a variational principle from a differential equation can be circumvented by constructing a minimization problem using the least squares method. For an equation $Lu = f$ this problem is

$$J(u) = \int_\Omega (Lu - f)^2 d\mathbf{x} = \text{minimum} \tag{8.3}$$

Numerical methods that apply to the classical variational problem can be extended to the least squares formulation as well.

A third approach of problem-solving that is also of considerable importance goes by the name *weighted-residual formulation*. Here the residue $Lu - f$ is made zero in a global average sense with respect to an appropriately chosen test function $v(\mathbf{x})$. Hence $u(\mathbf{x})$ is the solution of the integral equation

$$\int_\Omega (Lu - f)v(\mathbf{x})d\mathbf{x} = 0 \tag{8.4}$$

For the specific form of operator L and boundary conditions being considered in this section we require v to be twice differentiable and zero on the boundaries.

The function which satisfies Equation 8.1 at every point in the physical region is called the **classical** solution of the problem. It is obvious that the classical solution if available will identically satisfy the variational principle (Equation 8.2), the least squares problem (Equation 8.3) and the weighted residual Equation 8.4. These alternative approaches become prominent when it is cumbersome to derive the analytical solution of the differential equation. The variational and the weighted residual methods have an inherent advantage owing to their formulation. In many cases the integrals appearing in them can be evaluated once and the order of the highest derivative can be reduced from n to $n-1$. For the operator L of Equation 8.1, the following can be derived.

$$\begin{aligned}
\text{Variational principle}: \quad & \int_\Omega \left[A_{ij} \frac{\partial u}{\partial x_i} \frac{\partial u}{\partial x_j} + (qu - 2f)u \right] d\mathbf{x} &= \text{minimum} \\
\text{Weighted Residual}: \quad & \int_\Omega \left[A_{ij} \frac{\partial u}{\partial x_j} \frac{\partial v}{\partial x_i} + (qu - f)v(\mathbf{x}) \right] d\mathbf{x} &= 0, \\
& i, j = 1.. \quad n_d
\end{aligned} \tag{8.5}$$

A matching equation for the least squares problem does not exist. The advantage gained by integrating once are

1. u and v can be required to be differentiable only once (but not twice differentiable). This enables one to search for a solution from a larger class of functions.

2. The coefficients A_{ij} need to be finite valued and can be piecewise continuous.

These advantages are of practical value. In a heat conduction problem A_{ij} would be interpreted as thermal conductivity. Equations 8.5 would then permit modelling composite regions with sharp interfaces across which the material properties change abruptly. Solutions obtained after one (weighted) integration of the governing equation are called **weak** solutions.

8.3 Rayleigh-Ritz Method

It is a numerical procedure for solving problems that involve extremizing functionals. Let

$$u^h(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \phi_k(\mathbf{x})$$

be an approximation of the classical solution $u(\mathbf{x})$. Here α_k's are constant coefficients and ϕ_k's called *basis functions* are prescribed functions that satisfy boundary conditions and are parameterized by integer k. For example ϕ_k could represent polynomials of order k. Determination of u^h is now reduced to finding the coefficients α_k. The variational formulation in terms of u^h is

$$I(u^h) = \int_\Omega u^h(Lu^h - 2f)d\mathbf{x} = \text{minimum}$$

and so $\partial I/\partial \alpha_k = 0$ for each k. Carrying out the differentiation with respect to α_k results in the algebraic system of equations

$$\int_\Omega \left(A_{ij}\frac{\partial u^h}{\partial x_i}\frac{\partial \phi_k}{\partial x_j} \; + \; q(\mathbf{x})u^h\phi_k \; - \; f(\mathbf{x})\phi_k \right) d\mathbf{x} = 0, \quad k = 1..N \tag{8.6}$$

Repeated indices once again represent summation. When the least squares formulation is used, the computational steps are

$$I(u^h) = \int_\Omega (Lu^h - f)^2 d\mathbf{x} = \text{minimum}$$

Requiring $\partial I/\partial \alpha_k = 0$ for each k, we get

$$\int_\Omega \left(L\phi_k \sum_{l=1}^{N} \alpha_l L\phi_l - fL\phi_k\right) d\mathbf{x} = 0, \quad k = 1..N \tag{8.7}$$

The differences between Equations 8.6 and 8.7 follow from the discussion given earlier. In a clssical variational formulation of a second order differential equation the basis functions should be once differentiable. In a least squares method the basis functions must be twice differentiable since they are operated upon by L. As a rule, least squares technique employs high order polynomials as the basis functions. However this method can be used with any operator L and is hence widely applicable. Variational principles can be derived in a systematic manner for linear differential equations only when certain conditions are satisfied. For non-linear problems they may not exixt at all.

The notation $(.\ ,\ .)$ is commonly used to denote the inner product, for example, $(u, v) = \int_\Omega u(\mathbf{x})v(\mathbf{x})d\mathbf{x}$. The results after applying the Rayleigh-Ritz method can now be written in compact form as:

$$\begin{aligned} \text{Variational Principle} \quad &: \quad a(u^h, \phi_k) = (f, \phi_k), \quad k = 1..N \\ \text{Least squares} \quad &: \quad \sum_{k=1}^{N} \alpha_k (L\phi_k, L\phi_l) = (f, L\phi_l), \quad l = 1..N \end{aligned}$$

Here $a(u, v)$ is the bilinear form of operator L defined as $\int_\Omega (A_{ij}\partial u/\partial x_i \partial v/\partial x_j + quv)d\mathbf{x}$. These equations form a system of algebraic equations from which α_k can be determined.

Consider the solution of the governing differential equation using the weighted residual formulation (Equation 8.4) with $v(\mathbf{x})$ chosen as ϕ_k and u as u^h. Integrating $\int \phi_k Lud\mathbf{x}$ by parts we get

$$a(u^h, \phi_k) = (f, \phi_k), \quad k = 1..N$$

This is identical to the result obtained using the Rayleigh-Ritz method for the classical variational principle. When the test function $v(\mathbf{x})$ is chosen as identical to the basis functions of the unknown dependent variable, the weighted residual method is called *Galerkin* (also, Bubnov-Galerkin) method. Other possibilities exist. If $v(\mathbf{x})$ is a delta function $\delta(\mathbf{x} - \mathbf{x}_k)$ it is called a *collocation* method. The reduced equation after integration for the collocation method is

$$Lu^h = f, \text{at each } x_k, \quad k = 1..N \tag{8.8}$$

Here the governing equation is solved exactly at a finite number of points.

The Galerkin method applied to the weighted residual formulation is equivalent to the Rayleigh-Ritz method applied to the classical variational principle. However it has practical utility encompassing problems that cannot be handled using the Rayleigh-Ritz method. These include non-linear problems, complex geometry and boundary conditions, as well as problems for which a variational principle is itself not available. The finite element method is based on the Galerkin formulation is the thrust of the present chapter and is covered in later

sections. For completeness the above methods are illustrated here through the following numerical example.

Consider the solution of the equation, $-(u''+u) = x$ subject to homogeneous boundary conditions $u(0) = u(1) = 0$. Comparing with the standard form of the operator L one can identify

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$q = -1$ and $f(\mathbf{x}) = x$. The dependent variable is $u = u(x)$, a function of one independent variable. The operator L itself is d^2/dx^2. The analytical solution of this problem is $u(x) = (\sin x/\sin 1) - x$. The application of the four methods namely Rayleigh-Ritz, least squares, Galerkin and collocation is described below. For comparison the finite difference solution is also given. In each approach except finite differences the dependent variable is expanded in the form $u(x) \approx u^h(x) = \sum_{k=1}^{N} \alpha_k \phi_k(x)$. For simplicity N is chosen as 2 and the ϕ_k's are: $\phi_1 = x(1-x)$ and $\phi_2 = x^2(1-x)$. The basis functions automatically satisfy the boundary conditions and the goal of each approximate techique is to compute the values of α_k which weakly satisfy the governing differential equation.

Rayleigh-Ritz Method

The applicable variational principle for the model differential equation given above is

$$I(u) = \int_{x=0}^{1} (u'' + u + 2f)u dx = \text{extremum}.$$

Integrating the first term by parts simplifies to

$$I(u) = \int_{x=0}^{1} (-(u')^2 + u^2 + 2fu)dx = \text{extremum}$$

Substituting $u = \sum_k \alpha_k \phi_k$ in the expression for I and differentiating with respect to α_k results in the following expression valid for each k:

$$\int_0^1 \left(-\frac{\partial u}{\partial x}\frac{\partial u'}{\partial \alpha_k} + u\frac{\partial u}{\partial \alpha_k} + f\frac{\partial u}{\partial \alpha_k}\right) dx = 0, \quad k = 1..N$$

Using the results $\partial u/\partial \alpha_k = \phi_k, \partial u'/\partial \alpha_k = \partial \phi_k/\partial x$ and $\partial u/\partial x = \sum_{l=1}^{N} \alpha_l \partial \phi_l/\partial x$ this equation reduces to

$$\int_0^1 \left[-\frac{\partial \phi_k}{\partial x}\sum_{l=1}^{N} \alpha_l \frac{\partial \phi_l}{\partial x} + \phi_k \sum_{l=1}^{N} \alpha_l \phi_l + \phi_k f\right] dx = 0, \quad k = 1..N$$

Performing the necessary integration one obtains the following system of algebraic equations:

$$\begin{aligned} k = 1 \quad &: \quad [-1/3 + 1/30]\alpha_1 + [-1/6 + 1/60]\alpha_2 = -1/12 \\ k = 2 \quad &: \quad [-1/6 + 1/60]\alpha_2 + [-2/15 + 1/105]\alpha_2 = -1/20 \end{aligned}$$

Hence $\alpha_1 = 71/369$ and $\alpha_2 = 7/41$.

Method of Least Squares

Start with the statement

$$\int_0^1 (Lu - f)^2 dx = \text{minimum}$$

As mentioned earlier this is equivalent to the system of equations

$$\int_0^1 (\sum_{l=1}^{N} \alpha_l L\phi_l L\phi_k - fL\phi_k)dx = 0, \quad k = 1..N$$

For $N = 2$ the system of equations in expanded form is

$$\begin{aligned} k = 1: \quad & \int_0^1 (\alpha_1 (L\phi_1)^2 + \alpha_2 L\phi_1 L\phi_2 - fL\phi_1)dx = 0 \\ k = 2: \quad & \int_0^1 (\alpha_1 L\phi_1 L\phi_2 + \alpha_2 (L\phi_2)^2 - fL\phi_2)dx = 0 \end{aligned}$$

Using the results $L\phi_1 = 2 - x(1-x)$ and $L\phi_2 = (6x - 2) - x^2(1-x)$, these equations reduce to

$$\begin{aligned} 101/30\ \alpha_1 + 131/60\ \alpha_2 &= 11/12 \\ 131/60\ \alpha_1 + 131/35\ \alpha_2 &= 19/20 \end{aligned}$$

Hence $\alpha_1 = 0.17319$ and $\alpha_2 = 0.15279$.

Galerkin Method

In the weighted residual formulation we choose the test function as $\phi_k(x)$ and arrive at the system of equations

$$\int_0^1 \phi_k (Lu - f)\ dx = 0, \quad k = 1..N$$

Since the ϕ_k's are chosen in such a way that they satisfy the same boundary conditions as u the integral above can be simplified using integration by parts. This yields

$$\int_0^1 (-\frac{\partial \phi_k}{\partial x}\frac{\partial u}{\partial x} + \phi_k u + \phi_k f)dx = 0$$

The integral expression is identical to the equation obtained using Rayleigh-Ritz method. Hence $\alpha_1 = 71/369$ and $\alpha_2 = 7/41$.

Collocation Method

In this method the governing equation is exactly satisfied at a pre-determined set of points called collocation points. For an expansion $u^h = \sum_{k=1}^{N} \alpha_k \phi_k$, N such points are required. In the present example $N = 2$ and these points are chosen as $x = 1/3$ and $x = 2/3$. Hence the governing equations are

$$\begin{aligned} x &= 1/3, \qquad & \alpha_1 L\phi_1 + \alpha_2 L\phi_2 &= f(1/3) \\ x &= 2/3, \qquad & \alpha_1 L\phi_1 + \alpha_2 L\phi_2 &= f(2/3) \end{aligned}$$

The algebraic equations obtained after evaluating terms such as $L\phi_1$ are

$$\begin{aligned} 16/9 \;\; \alpha_1 - 2/27\, \alpha_2 &= 1/3 \\ 16/9 \;\; \alpha_1 + 50/27\, \alpha_2 &= 2/3 \end{aligned}$$

The α_k's can now be calculated as $\alpha_1 = 25/128$ and $\alpha_2 = 3/16$.

Finite Differences

A finite difference procedure leading to a system of two algebraic equations will involve four nodes (1...4) located at $x = 0, \frac{1}{3}, \frac{2}{3}$ and 1 resulting in a grid spacing $\Delta x = 1/3$. Using central differences the system of equations can be readily shown to be:

$$\begin{aligned} &\text{Node 1} \quad && u_1 = 0 \\ &\text{Node 2} \quad && (1 - 2/\Delta x^2)\; u_2 + 1/\Delta x^2\; u_3 = -1/3 \\ &\text{Node 3} \quad && 1/\Delta x^2\; u_2 + (1 - 2/\Delta x^2)\; u_3 = -2/3 \\ &\text{Node 4} \quad && u_4 = 0 \end{aligned}$$

Hence $u_2 = 0.05608$ and $u_3 = 0.0689$. Values of u at any other point is obtained using linear interpolation between the pair of surrounding nodes.

A comparison of the five solutions with the analytical solution is given in Table 8.1.

Table 8.1 Comparison of Numerical Solutions for a Linear Problem.

x	Analytical	Rayleigh-Ritz	Galerkin	Least Squares	Collocation	Finite Differences
0	0	0	0	0	0	0
1/4	0.044	.044	.044	.0396	.0454	.042
1/2	.070	.069	.069	.0624	.0722	.0625
3/4	.060	.060	.060	.0539	.0629	.0517
1	0	0	0	0	0	0

Table 8.1 shows that the best approximate solution is provided by the Rayleigh-Ritz and the Galerkin methods. In fact the following result can be rigorously proved. *The Galerkin method applied to a linear self-adjoint second-order differential operator and the Rayleigh-Ritz method applied to the classical variational principle of this differential operator will give the best approximation to the analytical solution compared to all other methods.* The comparison is to be made on a prescribed grid that produces equal number of unknowns for all the competing methods. Equation 8.1 involves a linear operator L which is also self-adjoint, i.e., the inner product $(v, Lu) = (Lu, v)$ in Ω for any pair of functions u and v that are twice differentiable and vanish on $\partial\Omega$. Most differential equations that one encounters in flow and heat transfer are neither linear nor self-adjoint and the result given above is not of real practical use. However the Galerkin finite element method described in this chapter allows a systematic procedure for controlling errors and treating complex geometry, gradient boundary conditions, iterative schemes and matrix inversion. There is growing evidence that FEM permits massive parallelization of the calculations and is an important numerical tool for the future.

Before illustrating the application of FEM to model problems an overview of the method as a whole is presented here. FEM employs the weighted residual formulation with piecewise interpolant polynomials as the basis functions. These polynomial functions are defined on a subdomain having a finite support and are zero outside. This subdomain is called a finite element. The entire domain is made up of contiguous finite elements. The polynomial function over an element is parametrized by values of the dependent variable at discrete points. These points are called nodes. The degree of the polynomial must be equal to the number of nodes per element. The test function used in weighted integration is identical to the polynomial basis function and integration is performed over the subdomain. Algebraic equations generated at the element level are assembled to form a matrix equation for the region as a whole. The solution for the dependent variable is obtained pointwise by inverting this system of equations. The sequence of finite element calculations for any problem involves: *(i)* Element generation and node numbering, *(ii)* Weighted integration at the element level, *(iii)* Matrix assembly, *(iv)* Application of boundary conditions, *(v)* Matrix preconditioning and inversion, *(vi)* Iteration for non-linearity, and

(vii) Convergence checks. These steps are illustrated through examples in the succeeding sections.

8.4 Interpolation

8.4.1 One Dimensional Regions

Basis functions used in finite element method arise from polynomials used for interpolation. Consider interpolation for a function $f(x)$ in terms of its values $f_1, f_2 \cdots f_N$ evaluated at nodes $1..N$. Irrespective of the exact form of $f(x)$ one can expand

$$f(x) = \sum_{i=1}^{N} a_i x^{i-1}$$

and determine the coefficients a_i by requiring $f = f_i$ at $x = x_i$. Hence a_i's are determined from the matrix equation

$$\begin{bmatrix} 1 & x_1 & \dots & x_1^{N-1} \\ 1 & \vdots & \dots & x_2^{N-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_N & \dots & x_N^{N-1} \end{bmatrix} \begin{Bmatrix} a_1 \\ \vdots \\ \vdots \\ a_N \end{Bmatrix} = \begin{Bmatrix} f_1 \\ \vdots \\ \vdots \\ f_N \end{Bmatrix}$$

With a_i's known, the function $f(x)$ can be evaluated at any value of x between x_1 and x_N using the expansion for $f(x)$ given above. Although a considerable amount of theoretical results are available for estimating interpolation errors, only the following are of relevance to the present chapter. If $f(x)$ is an mth order polynomial, values of $N \geq m + 1$ will produce zero interpolation error. If $f(x)$ is a continuous function with derivatives of all orders well-defined, then the interpolation error monotonically decreases as N increases. Since the interpolation coefficients a_i are determined by inverting a matrix it follows that the nodes $x_1, \dots x_N$ must be distinct. In practice the nodes may or may not be equally spaced. The extent of non-uniformity of grid spacing permitted in a calculation is closely linked with the round-off errors arising from machine computation. It is also important that all nodes x_i should not fall at the roots of the function $f(x)$. This guarantees that the right hand side of the matrix equation is not a zero vector.

The procedure described above is called Lagrangian interpolation. For $f(x) = \sin \pi x, 0 < x < 1$, values of a_i as a function of N and the rms-error defined as

$$E = \left[\int_0^1 (f(x) - \sum_{i=1}^{N} a_i x^{i-1})^2 dx \right]^{1/2}$$

are summarized in Table 8.2.

Table 8.2 Lagrangian Interpolation Error

N	x_i	a_i	E
2	0, 1	0, 0	0.707
2	1/3, 2/3	0.866, 0.0	0.384
3	0, 1/2, 1	0, 3.99, −3.99	0.034
4	0, 1/3, 2/3, 1	0, 3.897, −3.897, 0	0.027

The expansion for $f(x)$ in terms of powers of x can be recast in the form of a linear superposition of elementary polynomials N_i, i.e.

$$f(x) = \sum_{i=1}^{N} c_i N_i(x)$$

Here c_i's are constant coefficients and $N_i(x)$, called the Lagrange polynomial of order $N-1$, is defined as

$$\begin{aligned} N_i(x) &= \frac{(x-x_1)(x-x_2).....(x-x_{i-1})(x-x_{i+1})...(x-x_N)}{(x_i-x_1)(x_i-x_2).....(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_N)} \\ &= \Pi^N_{\substack{j=1 \\ j\neq i}} \frac{(x-x_j)}{(x_i-x_j)}, \; i = 1..N \end{aligned}$$

Here the points x_i are not necessarily equally spaced. The polynomials N_i have a unique property of being unity when $x = x_i$ and zero when $x = x_j (j \neq i)$, $j = 1....N$. The matrix equation for c_i can now be shown to be, $Ic_i = f_i$, where I is an identity matrix. Hence $c_i = f_i$ and the expansion for $f(x)$ becomes

$$f(x) = \sum_{i=1}^{N} f(x_i) N_i = \sum_{i=1}^{N} f_i N_i$$

The polynomials N_i are commonly used as basis functions in the finite element method.

An alternative route to interpolation, called Hermite interpolation uses an expansion in terms of function values as well as its derivative. This is of the form

$$f(x) = \sum_{i=1}^{N} L_i f(x_i) + \sum_{i=1}^{N} M_i f'(x_i)$$

Here $f' = df/dx$ and L_i and M_i are polynomials that satisfy the relationships

$$
\begin{aligned}
L_i(x_j) &= \delta_{ij} \\
\left.\frac{dL_i}{dx}\right|_{x=x_j} &= 0 \\
M_i(x_j) &= 0 \\
\left.\frac{dM_i}{dx}\right|_{x=x_j} &= \delta_{ij}
\end{aligned}
\tag{8.9}
$$

where δ_{ij} is Kronecker-delta which is 1 if $i = j$, and 0 otherwise. Polynomials satisfying these conditions are called Hermite polynomials. Explicit forms of L_i and M_i in terms of the Lagrange polynomial N_i are:

$$
L_i = (1 - 2(x - x_i) \left.\frac{dN_i}{dx}\right|_{x=x_i})[N_i(x)]^2
$$

$$
M_i = (x - x_i)[N_i(x)]^2
$$

As stated earlier, Lagrangian interpolation with N nodes is exact for polynomials upto order $N - 1$. In contrast, it can be shown that Hermite interpolation with N nodes is exact for polynomials of order upto $2N - 1$. For example one gets a cubic approximation with just 2 nodes instead of 4 that are required with Lagrangian interpolation. Hence fewer nodes are required for a given degree of approximation with Hermite interpolation. However it suffers from the drawback of requiring function derivatives at the nodal locations. For $f(x) = \sin \pi x$, $0 < x < 1$, the rms interpolation error defined as

$$
[\int_0^1 (f(x) - \sum_{i=1}^{N}(L_i f(x_i) + M_i f'(x_i)))^2 dx]^{1/2}
$$

for Hermite interpolation is summarized in Table 8.3.

Table 8.3 Hermitian Interpolation Error

N	x_i	E
2	0, 1	0.1357
3	0, 1/2, 1	0.057 0.0017
4	0, 1/3, 2/3, 1	0.000017

Explicit forms of N_i, L_i and M_i for values of N=2 and 3 are:

$$
N = 2 \qquad N_1 = (x - x_2)/(x_1 - x_2), \quad N_2 = (x - x_1)/x_2 - x_1)
$$

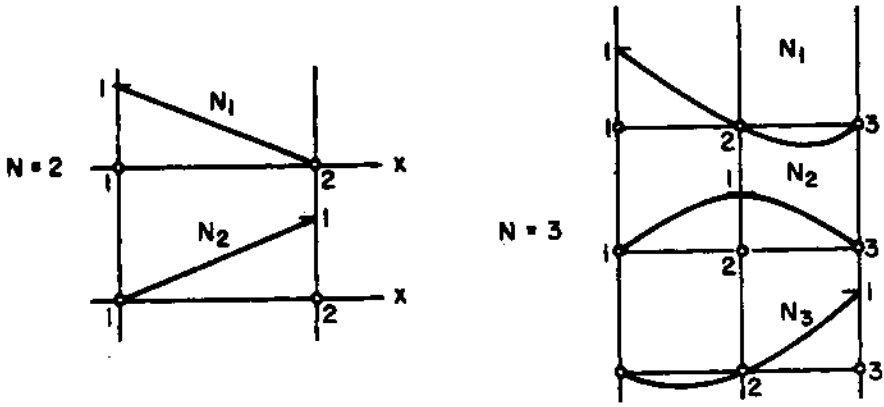


Figure 8.1: Lagrangian Interpolation Polynomials

$$
\begin{aligned}
N_1'(x_1) &= 1/(x_1 - x_2), N_2'(x_2) = 1/(x_2 - x_1) \\
L_1 &= (1 - 2(x - x_1)/(x_1 - x_2))((x - x_2)/(x_1 - x_2))^2 \\
L_2 &= (1 - 2(x - x_2)/(x_2 - x_1))((x - x_1)/(x_2 - x_1))^2 \\
M_1 &= (x - x_1)((x - x_2)/(x_1 - x_2))^2 \\
M_2 &= (x - x_2)((x - x_1)/(x_2 - x_1))^2
\end{aligned}
$$

$$
\begin{aligned}
N = 3 \qquad N_1 &= (x - x_2)/(x - x_3)/(x_1 - x_2)(x_1 - x_3) \\
N_2 &= (x - x_1)/(x - x_3)/(x_2 - x_1)(x_2 - x_3) \\
N_3 &= (x - x_1)/(x - x_2)/(x_3 - x_1)(x_3 - x_2) \\
N_1'(x_1) &= (2x_1 - x_2 - x_3)/(x_1 - x_2)(x_1 - x_3) \\
N_2'(x_2) &= (2x_2 - x_1 - x_3)/(x_2 - x_1)(x_2 - x_3) \\
N_3'(x_3) &= (2x_3 - x_1 - x_2)/(x_3 - x_1)(x_3 - x_2) \\
L_1 &= (1 - 2(x - x_1)N_1'(x_1))[(x - x_2)(x - x_3)/(x_1 - x_2)(x_1 - x_3)]^2 \\
L_2 &= (1 - 2(x - x_2)N_2'(x_2))[(x - x_1)(x - x_3)/(x_2 - x_1)(x_2 - x_3)]^2 \\
L_3 &= (1 - 2(x - x_3)N_3'(x_3))[(x - x_1)(x - x_2)/(x_3 - x_1)(x_3 - x_2)]^2 \\
M_1 &= (x - x_1)[(x - x_2)(x - x_3)/(x_1 - x_2)(x_1 - x_3)]^2 \\
M_2 &= (x - x_2)[(x - x_1)(x - x_3)/(x_2 - x_1)(x_2 - x_3)]^2 \\
M_3 &= (x - x_3)[(x - x_1)(x - x_2)/(x_3 - x_1)(x_3 - x_2)]^2
\end{aligned}
$$

Lagrangian interpolation polynomials N_i are exclusively used in the present chapter. The finite element method is developed using weighted integration over a subdomain of the physical region with polynomials N_i used as the test function. The shape of these polynomials with the number of nodes $N = 2$ and 3 are sketched in Figure 8.1.

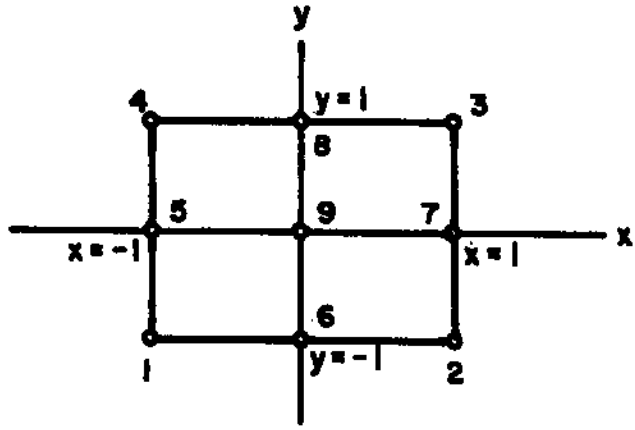


Figure 8.2: Normalized Two Dimensional Region

It is quite common to normalize the distances arising in interpolation, e.g., when $N = 2$ we set $x_1 = 0$ and $x_2 - x_1 = 1$; when $N = 3$ we set $x_2 = 0$ and $x_2 - x_1 = x_3 - x_2 = 1$. The expressions for N_i simplify under these conditions as:

$$\begin{aligned} N &= 2 \quad N_1 = 1 - x, \quad N_2 = x \\ N &= 3 \quad N_1 = -\frac{1}{2}x(1-x), \quad N_2 = 1 - x^2, \quad N_3 = \frac{1}{2}x(1+x) \end{aligned}$$

If ξ is the real physical coordinate with values ξ_i at the nodes, it can be related to the normalized coordinate x through the formula

$$\xi = \sum_{i=1}^{N} \xi_i N_i(x).$$

8.4.2 Two Dimensional Regions

Two dimensional interpolation is considered next. For convenience, start with a two dimensional square region with normalized distances as shown in Figure 8.2.

The nodes are shown by open circles in this figure. Lagrangian interpolation in two dimensions is obtained by simply taking the product of the respective one dimensional polynomials. Hence one can develop a constant interpolant with 1 node, a linear interpolant with 4 nodes and a quadratic interpolant with 9 nodes. The interpolation polynomials of order 0, 1 and 2 are linear combinations of $(1), (1, x, y, xy)$ and $(1, x, y, xy, x^2, x^2y, x^2y^2, xy^2$ and $y^2)$. These are shown compactly by the Pascal triangle in Figure 8.3. The interpolation error drops monotonically as the degree of the interpolating polynomial is raised.

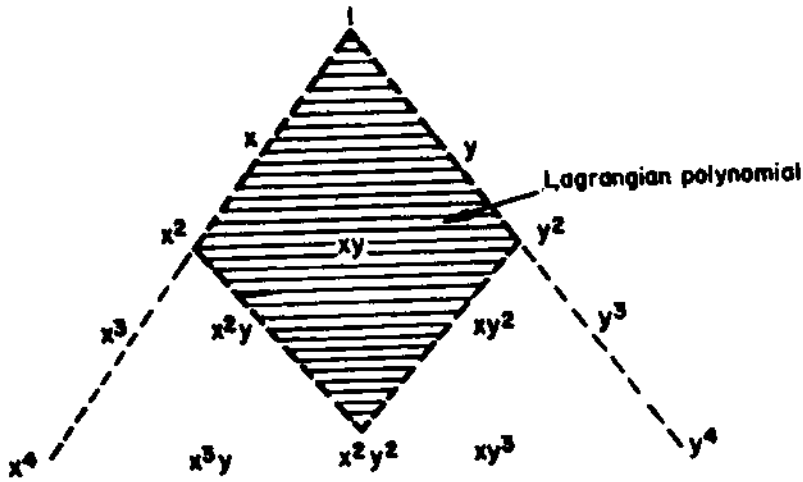


Figure 8.3: Pascal Triangle

This result however requires that the polynomial basis be complete. Hence in one dimension the Nth order interpolation must have $(N+1)$ coefficients corresponding to each power of the independent variable and none of them must be arbitrarily set to zero. In two dimensions, complete polynomials are obtained by considering all terms above and including the Nth row of the Pascal triangle but none below it. Hence the Lagrangian basis has terms over and above what is required to make a polynomial complete (shaded region in Figure 8.3).

The coefficients multiplying the individual terms $g(x,y)$ of the Pascal triangle for a shape function associated with any node of the normalized square region are obtained in a manner analogous to the one dimensional problem. Hence $N_i = \sum_{j=1}^{M} c_j g(x,y)$, where $M = 1$ for $N = 0$ (the interpolant order equal to 0), $M = 4$ for $N = 1$ and $M = 9$ for $N = 2$. The coefficients c_j are obtained by requiring that $N_i = 1$ when (x,y) falls on node i and $N_i = 0$ at the remaining $M-1$ nodes. For example for node 1 and $N = 2$ the coefficients c_j are obtained by solving the equation

$$\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} c_1 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ c_9 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ 0 \end{Bmatrix}$$

Complete forms of N_i for the standard element in Figure 8.2 are listed below.

$N = 0, M = 1$

$$N_1 = 1$$

$N = 1, M = 4$

$$\begin{aligned} N_1 &= \frac{1}{4}(1-x)(1-y) \\ N_2 &= \frac{1}{4}(1+x)(1-y) \\ N_3 &= \frac{1}{4}(1+x)(1+y) \\ N_4 &= \frac{1}{4}(1-x)(1+y) \end{aligned}$$

$N = 2, M = 9$

$$\begin{aligned} N_1 &= \frac{1}{4}xy(1-x)(1-y) \\ N_2 &= -\frac{1}{4}xy(1+x)(1-y) \\ N_3 &= \frac{1}{4}xy(1+x)(1+y) \\ N_4 &= -\frac{1}{4}xy(1-x)(1+y) \\ N_5 &= -\frac{1}{2}x(1-x)(1-y^2) \\ N_6 &= -\frac{1}{2}y(1-y)(1-x^2) \\ N_7 &= \frac{1}{2}x(1+x)(1-y^2) \\ N_8 &= \frac{1}{2}y(1+y)(1-x^2) \\ N_9 &= (1-x^2)(1-y^2) \end{aligned}$$

In the above expressions, the node numbers $1, 2...M$ are asssigned usually in a counter-clockwise manner.

As a rule Lagrangian interpolation of quadratic and higher orders will involve interior nodes such as node 9. A considerable amount of simplification can result in FEM calculations if interior nodes can be eliminated without sacrificing the order of interpolation. For example one may drop the term x^2y^2 in the Pascal triangle for $N = 2$ because the quadratic variations in x are accounted for in x^2 and x^2y while quadratic variations in y are accounted for in y^2 and xy^2. Hence node 9 (Figure 8.2) can be dropped from consideration and the interpolation function can be determined from a system of 8 algebraic equations. This result is summarized as:

$N = 2, M = 8$ (interior node dropped)

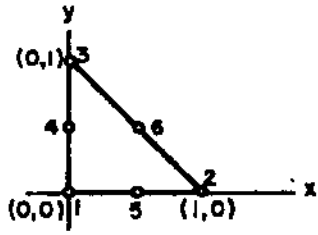


Figure 8.4: Standard Triangle

$$
\begin{aligned}
N_1 &= -\frac{1}{4}(1-x)(1-y)(x+y+1) \\
N_2 &= \frac{1}{4}(1+x)(1-y)(x-y-1) \\
N_3 &= \frac{1}{4}(1+x)(1+y)(x+y-1) \\
N_4 &= -\frac{1}{4}(1-x)(1+y)(x-y+1) \\
N_5 &= \frac{1}{2}(1-x)(1-y^2) \\
N_6 &= \frac{1}{2}(1-x^2)(1-y) \\
N_7 &= \frac{1}{2}(1+x)(1-y^2) \\
N_8 &= \frac{1}{2}(1-x^2)(1+y)
\end{aligned}
$$

These expressions for $N = 2$ can be written in compact form as (Figure 8.3):
Corner nodes $(i = 1, 2, 3, 4)$

$$N_i = \frac{1}{4}(1 + xx_i)(1 + yy_i)(xx_i + yy_i - 1)$$

Mid-side nodes $(i = 5, 6, 7, 8)$

$$
\begin{aligned}
x_i = 0; \quad N_i &= \frac{1}{2}(1 - x^2)(1 + yy_i) \\
y_i = 0; \quad N_i &= \frac{1}{2}(1 - y^2)(1 + xx_i)
\end{aligned}
$$

Figure 8.4 shows a six-noded triangular element with corner nodes (1,2,3) and mid-side nodes (4,5,6) with no interior node. The interpolation functions N_i for this geometry are:

$N = 1, M = 3$

$$N_1 = 1 - x - y$$

$$
\begin{aligned}
N_2 &= x \\
N_3 &= y
\end{aligned}
$$

$N = 2, M = 6$

$$
\begin{aligned}
N_1 &= (1 - x - y)\ 2(1 - x - y) - 1) \\
N_2 &= x(2x - 1) \\
N_3 &= y(2y - 1) \\
N_4 &= 4y(1 - x - y) \\
N_5 &= 4x(1 - x - y) \\
N_6 &= 4xy
\end{aligned}
$$

It is worth recapitulating that with function values prescribed at the boundary nodes as $f_1, ...f_M$, the function at any other point within the square or the triangular region enclosed by the boundary can be obtained as

$$f(\mathbf{x}) = \sum_{i=1}^{M} f_i N_i(\mathbf{x})$$

If f is specified as x or y coordinate of the nodes in the physical plane then the interpolation functions can be used as mapping functions between the physical plane and the normalized region. This mapping is quite important since finite element calculations involve integral evaluations. Numerical quadratures for integration are normally specified on a standard square or a triangular region. Examples of geometric transformation via interpolation functions are shown in Figure 8.5. Here, the physical coordinates X and Y are taken to be transformed as

$$X = \sum_{i=1}^{M} X_i N_i(x, y) \text{ and } Y = \sum_{i=1}^{M} Y_i N_i(x, y)$$

from a point (x, y) to (X, Y). For $N = 2$ the standard square and triangle in the x-y plane are equivalent to a curvilinear quadrilateral and triangle respectively in the X-Y plane with piecewise quadratics as their edges. This result is of great practical use in solving problems in a complex geometry.

8.4.3 Error Control

In one dimensional interpolation it was shown that interpolation error reduces as the order of interpolation (N) increases. This is commonly referred to as the p-version of controlling error. Alternatively, error can also be reduced by considering a smaller portion of the total physical domain. This is called the h-version of error control. Table 8.4 reconsiders the rms-error in interpolation for the function $f(x) = \sin \pi x$ over segments smaller than unity to illustrate this point. It can be seen that error diminishes as the region size is reduced for a given order N.

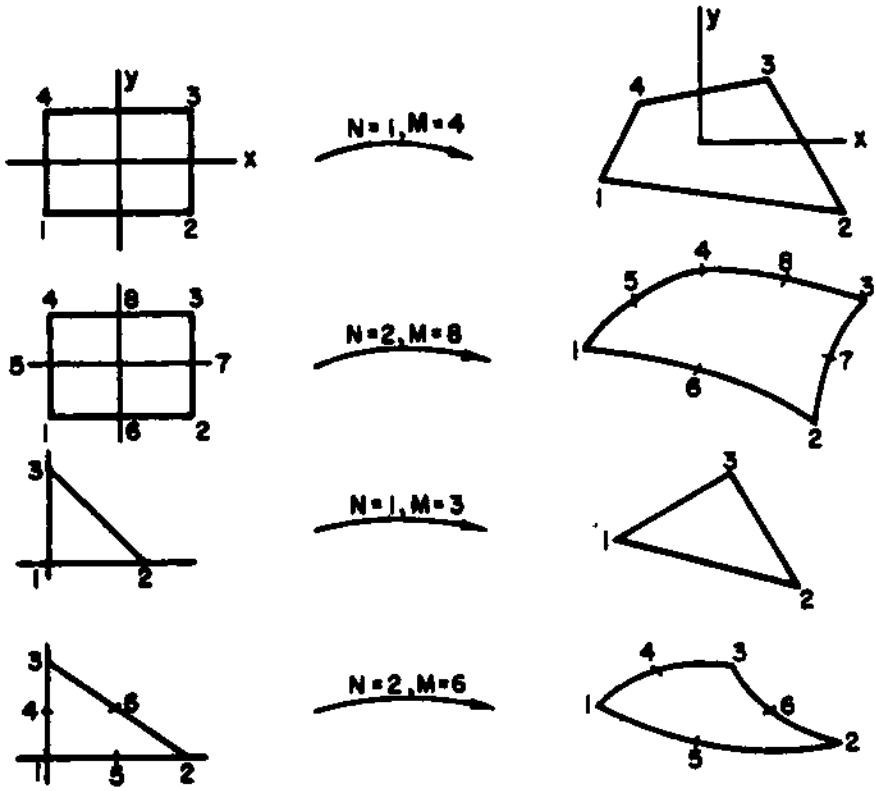


Figure 8.5: Coordinate Transformation Using Interpolation Functions

8.5 Numerical Integration

Integrals appearing in finite element calculations can be analytically evaluated where possible. However general purpose computer codes employ numerical integration via Gaussian quadrature (see Chapter 2). The quadrature is of the form

$$I = \int_{\Omega} f(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{M} W_i f(\mathbf{x}_i)$$

where W_i are weights and $\mathbf{x}_i$ the points at which the integrand is evaluated. In two dimensions, integration is required over straight and curvilinear triangles and quadrilaterals. Each geometry can however be mapped to a standard normalized element using interpolation polynomials. Hence numerical quadratures for the integration over a standard element alone required, Table 8.5.

With the background developed thus far, the solution of flow and heat transfer equations using the finite element method are discussed in the following sections.

Table 8.4 Error Control Using h-version. $(0 < x < x_d)$.
$f(x) \approx \sum_{i=1}^{N} a_i f(x_i)$

x_d	N	x_i	a_i	E
	2	1/3, 2/3	0.866, 0	0.384
1	3	0, 1/2, 1	0, 3.99, −3.99	0.034
	4	0, 1/3, 2/3, 1	0, 3.897, −3.897, 0	0.027
	2	1/6, 1/3	0.1339, 2.1961	0.079
0.5	3	0, 1/4, 1/2	0.0, 3.6568, −3.3137	0.0152
	4	0, 1/6, 1/3, 1/2	0.0, 3.2057, −0.6461, −3.5307	0.0013
	2	1/12, 1/6	0.0176, 2.894	0.0112
0.25	3	0, 1/8, 1/4	0.0, 3.2945, -1.8643	0.0025
	4	0, 1/12, 1/6, 1/4	0.0, 3.1459, -0.0865, -4.73359	0.000046

8.6 Application of FEM to One Dimensional Problems

8.6.1 Model Heat Conduction Problem

Consider the solution of the steady one dimensional heat conduction problem in a region with constant thermal properties. The governing equation is

$$T'' = \frac{d^2T}{dx^2} = 0, \quad 0 < x < 1 \tag{8.10}$$

with the boundary conditions $T(0) = 1$ and $T(1) = 0$. The analytical solution of this equation is $T = 1 - x$. The finite element solution proceeds along the following steps:

(i) Discretization of the physical region into elements.

Let the region of size unity be divided into L elements of equal length $(= 1/L)$.

(ii) Node numbering

Let $(k-1)$ be the order of interpolation of the dependent variable $T(x)$ over each element. Then the number of nodes per element is k. For a one dimensional

Table 8.5. Weights and Locations for Numerical Integration.

n	$W_i(i=1..n)$	$x_i(i=1..n)$	
3	0.55555	−0.77459	
	0.88888	0.0	
	0.55555	0.77459	
5	0.23692	−0.90617	
	0.47862	−0.53846	
	0.56888	0.0	
	0.47862	0.53846	
	0.23692	0.90617	
		x	y
4	1.0	0.57735	0.57735
	1.0	0.57735	−0.57735
	1.0	−0.57735	0.57735
	1.0	−0.57735	−0.57735
9	0.30864	−0.77459	0.77459
	0.493827	0.0	0.77459
	0.30864	0.77459	0.77459
	0.493827	−0.77459	0.0
	0.79012	0.0	0.0
	0.493827	0.77459	0.0
	0.30864	−0.77459	−0.77459
	0.493827	0.0	−0.77459
	0.30864	0.77459	−0.77459
3	1/3	0	1/2
	1/3	1/2	1/2
	1/3	1/2	0
7	1/ 40	0	0
	1/40	1	0
	1/ 40	0	1
	1/15	0	1/2
	1/15	1/2	0
	1/15	1/2	1/2
	3/40	1/3	1/3

region the total number of nodes can readily shown to be $(L(k-1)+1)$.

(iii) Weighted residual formulation

Let $N_i(x), i = 1..k$ be the element level interpolation function. N_i is also called a shape function in the literature. The temperature field is interpolated as $T\ (x) = \sum_{i=1}^{k} T_i N_i(x)$ over each element. The governing equation is recast in the integral form for each element as

$$\int_{\Omega_e} N_i(x) \frac{d^2T}{dx^2} dx = 0$$

Note that the weighting function in integration is the interpolation polynomial itself, a feature specific to the Bubnov-Galerkin finite element method. Clearly for each element the above integration produces one equation per shape function and hence k equations in all. The integral is evaluated as

$$\int_{\Omega_e} N_i(x) T''(x) dx = 0 \ = \ N_i T' \mid_{\partial\Omega_e} - \int_{\Omega e} N_i' T' dx.$$

The symbol $\partial\Omega_e$ is the boundary of Ω_e; in the one dimensional problem it refers to the corner nodes of each element. It is easy to understand that N_i has a unique value on $\partial\Omega_e$ and T' is equal but opposite in sign between continguous elements. Hence $N_i T'$ is not zero on $\partial\Omega_e$ but when the contributions of all the elements to an interior node are added, it cancels out. Anticipating this result the boundary term is dropped and we write

$$\int_{\Omega e} N_i' T' dx = 0. \tag{8.11}$$

Since $T(x) = \sum_{j=1}^{k} T_j N_j(x), T' = \sum_{j=1}^{k} T_j N_j'(x)$, Equation 8.11 becomes

$$\left[\sum_{j=1}^{k} \int_{\Omega_e} N_i' N_j' dx \right] \{T_j\} = 0, \qquad i = 1..k \tag{8.12}$$

This is a matrix equation at an element level and is of the form

$$AT = 0 \tag{8.13}$$

where matrix $A = [a_{ij}]$ and the entries a_{ij} are defined as $a_{ij} = \int_{\Omega_e} N_i' N_j' dx$. As mentioned earlier such integrations are analytically carried out where possible but numerical integration is more common. This is particularly true when the order of interpolation is high and in multi-dimensional problems. It should be realized that matrix A is singular at this stage and Equation 8.13 does not inidcate the solution T to be zero. The singularity is removed when the boundary conditions are included in the system of algebraic equations.

(iv) Assembly, boundary conditions and matrix inversion.

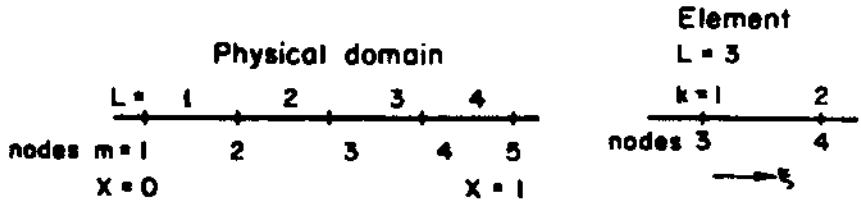


Figure 8.6: Discretization of a One Dimensional Region

These steps in a FEM calculation are unavoidable but are difficult to implement. The element-matrix equations are first assembled to form a global matrix equation so that contributions of all neighbouring elements to a common node are added. This step is best achieved by keeping an independent record of all the relationships between local and global node numbers on an element-by-element basis. Dirichlet boundary conditions are applied by replacing the appropriate row of the global matrix by zeroes and unity and the corresponding right hand entry by the function value. Matrix inversion is best accomplished by a direct banded-matrix solver. Steps (i)-(iv) are illustrated here through an example:

Let the number of elements L be 4 and interpolation on each element be linear ($k = 2$), Figure 8.6. On any element the interpolation functions N_i are, $N_1 = 1 - \xi$ and $N_2 = \xi$, where $0 < \xi < 1$ and ξ is the local independent variable over the element length. The relationship between x and ξ is $\xi = (x - x_m)/(x_{m+1} - x_m)$ where m is the global node number of the first node of the element under question. For the discretization shown in Figure 8.6, the local(LN)-global(GN) record for the nodes is:

Element L	Global Node Number		(GN)
	$k = 1$	$k = 2$	(LN $= k$)
1	1	2	
2	2	3	
3	3	4	
4	4	5	

Since global node 3 is shared by elements 2 and 3 the assembly process will produce a global matrix whose diagonal element at $i = j = 3$ is the sum of a_{11} of element 3 and a_{22} of element 2. Let a^e_{ij} be the element matrix of the eth element. For linear interpolation each of these matrices is square and of order two. The global matrix obtained after assembly is:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{e=3}$$

$$\begin{array}{c} \Longrightarrow \\ LN = 1 \rightarrow GN = 3 \\ LN = 2 \rightarrow GN = 4 \end{array} \quad \begin{bmatrix} a_{11}^1 & a_{12}^1 & 0 & 0 & 0 \\ a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & 0 \\ 0 & a_{21}^2 & a_{22}^2 + \underline{a_{11}^3} & \underline{a_{12}^3} & 0 \\ 0 & 0 & \underline{a_{21}^3} & \underline{a_{22}^3} + a_{11}^4 & a_{12}^4 \\ 0 & 0 & 0 & a_{21}^4 & a_{22}^4 \end{bmatrix}$$

For the diffusion problem governed by Equation 8.10 the matrix $[a_{ij}]^e$ is the same for all elements. When linear interpolation is used N_i' is a constant and equal to 1 if $i = 1$ and -1 if $i = 2$. Hence the element matrix is easily computed as

$$[a_{ij}]^e = \frac{1}{\Delta x^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

where Δx^e is the length of the element e.

The global matrix equation for the heat conduction problem with 4 elements is

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Here $T_1, ...T_5$ refer to the temperature values at the global node numbers 1..5. At this stage the matrix assembly is complete and one proceeds to the application of boundary conditions. These conditions are $T_1 = 1$ and $T_5 = 0$ and are implemented by replacing the appropriate rows (1 and 5) of the global matrix with zeros and unity as:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

The nodal values of temperature T are now obtained by inverting the global matrix as, $(T_1, ...T_5) = (\ 1, 0.75, 0.5\ 0.25, 0\ )$. This is identical to the analytical solution, $T(x) = 1 - x$ and is not surprising since the interpolation polynomial between any pair of nodes is of the same order as the analytical solution. Higher order elements will continue to reproduce the exact solution of Equation 8.10.

The model problem in this section can alternatively be solved by finite differences. Using central differencing one can approximate

$$T'' = \frac{(T_{i+1} - 2T_i + T_{i-1})}{\Delta x^2}$$

for a node spacing of Δx. The matrix equation for 5 nodes using finite differences can be seen to be equivalent to the global matrix arising in FEM that employs linear interpolation between any two nodes. In this respect finite differences can be thought of as a special case of FEM though it does not use explicitly the notion of interpolation and weighted integration. It is natural in FEM to construct a matrix to represent the governing differential equation as is natural in finite differences to form a successive substitution formula for the dependent variable at a given node. For this reason FEM employs direct matrix solvers for the system of algebraic equations while iterative methods are popular with finite differences.

8.6.2 Gradient Boundary Conditions

The specific advantage of using FEM when gradient boundary conditions are prescribed on a boundary is illustrated here. Suppose $T' = -g$ at $x = 0$ instead of $T = 1$. Then for the first element weighted integration yields

$$\int_{\Omega_e} N_i T'' dx = N_i T' \mid_{\partial\Omega_e} - \sum_j \int_{\Omega_e} N_i' N_j' T_j dx = 0$$

In this equation, the value of T' can be built in directly (equal to $-g$) and we get

$$(\sum_{j=1}^{k} \int_{\Omega_e} N_i' N_j' dx) T_j = g N_i(x = 0), \quad i = 1..k$$

Hence the element-level matrix for the first element using linear shape functions is

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix} = \begin{Bmatrix} g\Delta x \\ 0 \end{Bmatrix}$$

since $N_1(x = 0) = 1$ and $N_2(x = 0) = 0$. Here $\Delta x = x_2 - x_1$, the size of the first element. In particular if $g = 0$, the right side is a zero vector and the gradient condition need not be applied explicitly. This is an important point. In FEM not applying any boundary condition on a part of the flow domain is equivalent to applying $\partial T/\partial n = 0$ on that surface. This is in contrast with finite differences where the discretized version of every boundary condition must be applied.

For $g = 1$, the number of elements is equal to 4 and linear interpolation over each element the global matrix equation takes the form

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{Bmatrix} = \begin{Bmatrix} 0.25 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

The analytical solution for this problem is again $T = 1 - x$ and the numerical solution $T = (1, 0.75, 0.5, 0.25, 0)$ matches it exactly.

Let us briefly return to the computation of the element matrix $[a^e_{ij}]$. We have

$$a_{ij} = \int_{\Omega_e} N'_i(x) N'_j(x) dx,$$

defined over the element length $\Delta x^e = (x_2 - x_1)^e = (x_{m+1} - x_m)$. Here the index m is used to represent the global node number. The element length can be normalized to unity in terms of a coordinate $\xi = (x - x_m)/(x_{m+1} - x_m)$ and we get

$$\begin{aligned} a_{ij} &= \int_0^1 \left(\frac{dN_i}{d\xi}\frac{d\xi}{dx}\right)\left(\frac{dN_j}{d\xi}\frac{d\xi}{dx}\right)\frac{d\xi}{(d\xi/dx)} \\ &= \frac{1}{\Delta x^e}\int_0^1 N_{i\xi} N_{j\xi} d\xi \end{aligned}$$

Here we use the notation $N_{i\xi} = dN_i/d\xi$. In terms of ξ the linear shape functions are $N_1 = 1 - \xi$ and $N_2 = \xi$. The result for $[a_{ij}]^e$ used earlier can now be readily derived.

8.6.3 Extensions of One Dimensional Heat Conduction Equation

Consider a two special cases involving a variation in thermal conductivity. For an inhomogeneous region the conduction problem is governed by the equation

$$(k(x)T')' = 0, \qquad 0 < x < X \tag{8.14}$$

and for definiteness we assume the boundary conditions to be $T(0) = 1$ and $T(X) = 0$. The primes denote differentiation with respect to x. The procedure of discretizing the physical region, construction of an element matrix, matrix assembly, application of boundary conditions and matrix inversion carries over to this problem unchanged. The element-level matrix gets altered owing to the thermal conductivity function $k(x)$ and is given as

$$a^e_{ij} = \int_{\Omega^e} k(x) N'_i(x) N'_j(x) dx$$

This integral can be evaluated once $k(x)$ is prescribed. In practice variations in thermal conductivity may be small and one can assume a uniform value of $k(= k^e)$ over any element. Alternatively one use interpolation for $k(x)$ as $k(x) = \sum k_i G_i(x)$ where k_i are values of k at prescribed points and G_i an interpolation function distinct from N_i. Orders of interpolation of k and T

have to be chosen on grounds of physical reasoning. These remarks carry over to the steady diffusion-equation with source terms, namely

$$(kT')' = Q(x), \quad 0 < x < X \tag{8.15}$$

with $T(0) = 1$ and $T(X) = 0$. The element matrix equation is

$$\left[\int_{\Omega^e} k(x)N_i'(x)N_j'(x)dx\right]\{T_j\} = -\int_{\Omega^e} N_i(x)Q(x)dx, \quad i = 1..k$$

Functions $k(x)$ and $Q(x)$ can be prescribed analytically over the region $0 < x < X$, can be piecewise constant over each element Ω^e or be interpolated using polynomials of appropriate orders.

Consider the problem

$$(k(T)T')' = 0, \quad 0 < x < X \tag{8.16}$$

with $T(0) = 1$ and $T(X) = 0$, where thermal conductivity depends on temperature. This equation can be linearized to the form $U'' = 0$ using a new dependent variable U defined as

$$U = \int_0^T \frac{k(u)}{k_0}\, du \text{ where } k_0 = k(0)$$

The finite element approximations now carry over to the function U. If this linearizing procedure is not carried out the element level matrix is obtained as $[a_{ij}]^e$, where

$$a_{ij} = \int_{\Omega^e} k(T)N_i'(x)N_j'(x)dx \tag{8.17}$$

This integral cannot be evaluated since k depends on T-the dependent variable which is initially unknown. The solution of the problem must then be iteratively obtained. We may assume $k(T) = k_0$ or $k(T = 1)$ or an average value and solve for T_i at the nodes. This is only an intermediate solution $(= \tilde{T}_i)$. The quantities a_{ij} are now replaced by the new estimate

$$a_{ij} = \int_{\Omega^e} (\sum_I k_I(\tilde{T}_I)N_I(x))N_i'(x)N_j'(x)dx$$

and the temperature field is recomputed. The interpolation function for k and T can be identical in this problem since temperatures required for evaluation of k_I are available at the nodes of the physical region. The iterations continue till convergence is reached in the temperature field. This sequence of calculation is commonly referred to as *Picard* iteration.

The rate of convergence of the temperature field can be accelerated using a *Newton-Raphson* scheme. This scheme however requires that variation of k with T be reasonably smooth. The root of a function $f(x)(= X$, say) is

computed iteratively using Newton-Raphson scheme by successive application of the formula

$$X^{n+1} = X^n - \left.\frac{f(x)}{f'(x)}\right|_{x=X^n}$$

Here n is the iteration number; iterations continue till $f(X)$ is sufficiently close to zero and the difference $|X^{n+1} - X^n|$ is small. This can also be written as

$$[f'(X)]^n \delta X^{n+1} = -f(x)^n \text{where } \delta X^{n+1} = X^{n+1} - X^n$$

One can view $T_i(x)$ to be the root of the equation

$$\left[\int_{\Omega^e} k(T) N_i'(x) N_j'(x) dx\right] T_j = 0 \tag{8.18}$$

assembled over all the elements of the physical domain. Hence the Newton-Raphson scheme is generated by repeated application of the formula

$$[b_{ij}]^n \{\delta T_j^{n+1}\} = -\{R_i^n\} \tag{8.19}$$

Here R_i, the residue of the integral equation is

$$\begin{aligned} R_i &= \sum_{j=1}^{k} \left[\int_{\Omega^e} k(T^n) N_i'(x) N_j'(x) dx\right] \{T_j^n\}, \\ \delta T_j^{n+1} &= T_j^{n+1} - T_j^n \end{aligned}$$

at any node j and

$$\begin{aligned} b_{ij} &= \frac{\partial R_i}{\partial T_j} = \int_{\Omega^e} k(T^n) N_i'(x) N_j'(x) dx \\ &+ \int_{\Omega^e} (dk/dT)^n N_j N_i'(x) (dT/dx)^n dx \end{aligned}$$

Here the term dk/dT_j is evaluated as

$$\frac{dk}{dT}\frac{dT}{dT_j} = \frac{dk}{dT} N_j$$

Equation 8.19 constitutes the element level matrix equation of the problem. The initial temperature profile used for computing b_{ij} and R_i must satisfy the Dirichlet boundary conditions. In subsequent iterations boundary conditions applied to δT, the temperature correction is of the form δT (boundary nodes) $= 0$.

For completeness we illustrate the evaluation of integrals appearing in the definitions of R_i and b_{ij}. At the start of an iteration the following quantities are known:

$$
\begin{aligned}
N_i, N_j, N_i', N_j' &\quad \text{(analytical)} \\
k(T) &\quad \text{(analytical)} \\
\frac{dk}{dT} &\quad \text{(analytical)} \\
T &\quad \text{(at nodes)} \\
\frac{dT}{dx} = \sum T_i N_i' &\quad \text{(analytical)}
\end{aligned}
$$

We map the integral $\int \bar{G}(x)dx$ over to the standard element as $\int (G(\xi)) \frac{d\xi}{d\xi/dx}$. Writing $x(\xi) = \sum x_i N_i(\xi)$, $dx/d\xi$ is evaluated as $\sum x_i N_i'$ and is also a known quantity. At this stage the integral is determined using the quadrature formula

$$\int G(\xi)dx/d\xi(\xi)d\xi = \sum W_n G(\xi_n)dx(\xi_n)/d\xi$$

with as many integration points as may be necessary. At any point ξ_n, N_i and N_i' can be determined since they are analytical functions of ξ; T and dT/dx are determined as $\sum T_i N_i(\xi_n)$ and $\sum T_i N_i'(\xi_n)$ respectively. k and dk/dT are subsequently evaluated using the value of T obtained at ξ_n. Newton-Raphson iterations show faster convergence compared to Picard iterations only if k is a well-behaved function of T. However each iteration is computationally more intensive since Picard iterations require calculation of only a_{ij} while Newton-Raphson iterations require calculation of b_{ij} and R_i. Experience shows that the latter does afford an overall computing time advantage.

As an example, consider the property variation $k(T) = 1 - 0.4\,T$ in Equation 8.16. Dividing the region into four elements, using linear interpolation over each element and an initial guess $T = 1 - x$ for the temperature field, the element matrix $[b_{ij}]$ and the right hand vector R_i are obtained as:

$$[b_{ij}^1] = \begin{bmatrix} 3.6 & -4 \\ -3.6 & 4 \end{bmatrix}, \qquad R_i^1 = \begin{Bmatrix} -0.95 \\ 0.95 \end{Bmatrix}$$

$$[b_{ij}^2] = \begin{bmatrix} 3.2 & -3.6 \\ -3.2 & 3.6 \end{bmatrix}, \qquad R_i^2 = \begin{Bmatrix} -0.85 \\ 0.85 \end{Bmatrix}$$

$$[b_{ij}^3] = \begin{bmatrix} 2.8 & -3.2 \\ -2.8 & 3.2 \end{bmatrix}, \qquad R_i^3 = \begin{Bmatrix} -0.75 \\ 0.75 \end{Bmatrix}$$

$$[b_{ij}^4] = \begin{bmatrix} 2.4 & -2.8 \\ -2.4 & 2.8 \end{bmatrix}, \qquad R_i^4 = \begin{Bmatrix} -0.65 \\ 0.65 \end{Bmatrix}$$

The global matrix equation takes the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -3.6 & 7.2 & -3.6 & 0 & 0 \\ 0 & -3.2 & 6.4 & -3.2 & 0 \\ 0 & 0 & -2.8 & 5.6 & -2.8 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \delta T_1 \\ \delta T_2 \\ \delta T_3 \\ \delta T_4 \\ \delta T_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -0.1 \\ -0.1 \\ -0.1 \\ 0 \end{Bmatrix}$$

The temperature computed after one iteration is compared below with the exact solution $T(x) = (5 - \sqrt{25 - 16(1-x)})/2$.

T	(initial)	1	0.75	0.5	0.25	0
T	(iteration)	1	0.704	0.437	0.201	0
T	(analytical)	1	0.697	0.438	0.208	0

After the second iteration FEM and the analytical solution are practically indistinguishable.

8.6.4 Transient One Dimensional Heat Conduction.

Consider the finite element approximation of one dimensional diffusion equation

$$\rho C_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) \tag{8.20}$$

with suitable boundary conditions at $x = 0$ and L, and a prescribed initial condition for $T(x,t)$ at $t = 0$. The Galerkin formulation proceeds along the following lines. We expand T over a line element as $T = \sum T_j N_j$. Multiplying the governing equation by N_i and integrating over the element we get

$$\left[\int kN_i' N_j' dx\right] \{T_j\} + \left[\rho C_p \int N_i N_j dx\right] \{\frac{dT_j}{dt}\} = 0 \tag{8.21}$$

This is of the form $A_{ij} dT_j/dt = -B_{ij} T_j$ for each $i = 1..k$. Matrices A and B can be written down by inspection as

$$A_{ij} = \rho C_p \int N_i N_j dx$$

$$B_{ij} = \int kN_i' N_j' dx$$

The ordinary differential equation for T_j can be solved by a variety of techniques. In particular the ODE can be discretized as

$$A_{ij} \frac{T_j^{n+1} - T_j^n}{\Delta t} = -(p(B_{ij}T_j)^{n+1} + (1-p)(B_{ij}T_j)^n) \tag{8.22}$$

where p is a parameter between 0 and 1, n and $n+1$ are the current and new time levels, respectively. The discretization scheme is called ***explicit*** when $p = 0$, ***implicit*** when $p = 1$ and ***Crank-Nicolson*** when $p = 1/2$. In an explicit marching procedure the element matrix equation is obtained as

$$A_{ij}T_j^{n+1} = (A_{ij} - \Delta t B_{ij})T_j^n$$

The global matrix equation can now be set up by assembling the element matrices. Note that A_{ij} is a constant matrix and can be inverted once-and-for-all. When k is a function of x the integral appearing in the definition of B_{ij} must be numerically evaluated. When k depends on T as well as x, it is evaluated as $k(T^n, x)$, i.e. with current values of the dependent variable T. In either case the explicit scheme generates a linear system of equations of the form

$$\begin{aligned} T_i^{n+1} &= C_{ij}T_j^n \\ \text{where} \qquad C_{ij} &= A_{ij}^{-1}(A_{ij} - \Delta t B_{ij}) \end{aligned} \tag{8.23}$$

However the method is only conditionally stable and a small time step must be used. The stability properties of an explicit scheme are discussed in Chapter 4 in the context of finite differences.

In an implicit scheme the element matrix equation is

$$(A_{ij} + \Delta t B_{ij})T_i^{n+1} = A_{ij}T_i^n$$

If k depends on T then B_{ij} is determined with $k = k(T^{n+1}, x)$. Hence the matrix equation is non-linear and iterations are required to obtain a converged answer. During iterations k is determined with improved values of T^{n+1}. The time marching procedure is obtained as

$$T_i^{n+1} = C_{ij}T_j^n \tag{8.24}$$

where $C_{ij} = (A_{ij} + \Delta t B_{ij})^{-1}A_{ij}$. For an assembled system of equations valid for the entire region C_{ij} can be determined once-and-for-all if k is independent of T. In an implicit scheme the time step is limited by accuracy but not by stability. The discretization error in both explicit and implicit schemes is of order $O(\Delta t, \Delta x^2)$ when a quadratic basis is used. For a Crank-Nicolson method this is of order $O(\Delta t^2, \Delta x^2)$. The C-matrix for this scheme is, $C_{ij} = (A_{ij} + 0.5\Delta t B_{ij})^{-1}(A_{ij} - 0.5\Delta t B_{ij})$. Matrix C_{ij} (explicit) can be determined analytically without any matrix inversion, in both linear and non-linear problems, if A^{-1} is available. C_{ij}'s (implicit and Crank-Nicolson) however do require matrix inversion, once in linear and repeatedly in non-linear problems. Hence the advantage of stability is accompanied by the disadvantage of intensive computation.

Convergence of iterations of an implicit scheme applied to the non-linear problem when k depends on T can be accelerated once again using the Newton-Raphson method.

Let the residual $R_i = A_{ij}(T_j^{n+1}(*) - T_j^n) + B_{ij}\Delta t T_j^{n+1}$, for each $i = 1..k$. Here $T_j^{n+1}(*)$ is the current guess of the dependent variable that is used to construct k, B_{ij} as well as R_i. The new value of T_i^{n+1} is $\Delta T_i + T_i^{n+1}(*)$. The correction term is determined as

$$\left[\frac{\partial R_i}{\partial T_j^{n+1}}\right]\{\Delta T_j\} = -\{R_i\}$$

repeatedly till ΔT_j becomes quite small. The derivative $\partial R_i/\partial T_j$ can be evaluated as

$$\frac{\partial R_i}{\partial T_j^{n+1}} = A_{ij} + \Delta t\left(B_{ij} + \int_{\Omega^e}\frac{dk}{dT}N_i N_{jx}\frac{\partial T}{\partial x}dx\right)$$

as in the steady state problem.

8.7 Two Dimensional Problems

Consider the application of FEM to two dimensional problems. While the steps arising in a one dimensional calculation carry over to two dimensions unchanged, each intermediate step such as element generation and node numbering, numerical integration and matrix inversion is considerably more intense. A two dimensional element on a plane need not be rectangular. It is a finite region with n corners, m mid-side nodes and l interior nodes. The interpolant for a two dimensional function $f(x, y)$ over this element is built using the value of f at the nodes. For linear interpolation only corner nodes are involved; more nodes are required as the order of interpolation increases. In practice, interpolation functions are chosen in such a way that interior nodes are not involved in the calculations. Let ξ and η be the normalized coordinates on a standard element. The value of $f(x, y)$ anywhere within the element is obtained as

$$f(x, y) = \sum_{i=1}^{k} f_i N_i(\xi, \eta).$$

We prefer to use the notation x and y for coordinates in the physical plane and ξ and η as coordinates for the standard element. Expressions for N_i given in Section 8.4.2 can now be used with (x, y) replaced by (ξ, η), respectively. For a triangular element $k = 3$ (linear) or 6 (quadratic) and $0 \le \xi \le 1$ and $0 \le \eta \le 1$. For a square element $k = 4$ (linear) or 8 (quadratic) and the coordinate values are restricted to $-1 \le \xi \le 1$ and $-1 \le \eta \le 1$. Explicit expressions for N_i are given in Section 8.4.2. Mapping of the standard to the curvilinear element in the physical plane is accomplished using the transformations $x = \sum x_i N_i$ and $y = \sum y_i N_i$.

As an example consider the solution of the two dimensional steady conduction equation

$$T_{xx} + T_{yy} = 0 \tag{8.25}$$

subject to prescribed temperature and prescribed heat flux on portions of the boundary of the region. With the element shape and order of interpolation suitably chosen the interpolation functions $N_i(\xi, \eta)$ can be explicitly written down. The element-level matrix is obtained by weighted integration of the governing equation over any element. Hence

$$\begin{aligned}\int_{\Omega^e} N_i \nabla^2 T d\mathbf{x} = & \quad 0 \\ = & \quad -\int_{\Omega^e} \nabla N_i \cdot \nabla T d\mathbf{x} + \int_{\partial\Omega^e} N_i \frac{\partial T}{\partial n} ds\end{aligned}$$

with s being a coordinate parallel to the boundary $\partial\Omega^e$. The boundary terms for interior elements cancel after assembly since $\partial T/\partial n$ will be equal but opposite in sign. On the external boundary of the entire physical region the boundary terms can be dropped again if T is prescribed or $\partial T/\partial n = 0$. If the heat flux is prescribed as non-zero these terms must be retained in the calculation.

Substituting $T = \sum T_j N_j$, the element matrix equation is obtained as

$$[K_1]_{ij}\{T_j\} = \text{Boundary terms corresponding to prescribed heat flux.} \quad (8.26)$$

Indices i and j vary over $1..k$ and

$$\begin{aligned}K_{1ij} &= \int_{\Omega^e} \nabla N_i \cdot \nabla N_j d\mathbf{x} \\ &= \int_{\Omega^e} \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) dxdy\end{aligned}$$

Since N_i and N_j are defined on a standard element the integral appearing in the definition of K_{1ij} must be suitably transformed. Using the notation $N_{i\xi} = \partial N_i/\partial\xi$, $x_\xi = \partial x/\partial\xi$ and so on, we have the results:

$$\begin{aligned}\frac{\partial N_i}{\partial x} &= N_{i\xi}\xi_x + N_{i\eta}\eta_x, \quad & \frac{\partial N_j}{\partial x} &= N_{j\xi}\xi_x + N_{j\eta}\eta_x \\ \frac{\partial N_i}{\partial y} &= N_{i\xi}\xi_y + N_{i\eta}\eta_y, \quad & \frac{\partial N_j}{\partial y} &= N_{j\xi}\xi_y + N_{j\eta}\eta_y\end{aligned}$$

It can be shown that $\xi_x = y_\eta/J$, $\xi_y = -x_\eta/J$, $\eta_x = -y_\xi/J$, $\eta_y = x_\xi/J$, (Chapter 13). Here J, the Jakobian of the transformation is $x_\xi y_\eta - x_\eta y_\xi$ and $dxdy = Jd\xi d\eta$. Further

$$\begin{aligned}x_\xi &= \sum x_i N_{i\xi} \\ y_\xi &= \sum y_i N_{i\xi} \\ x_\eta &= \sum x_i N_{i\eta} \\ y_\eta &= \sum y_i N_{i\eta}\end{aligned}$$

The Jakobian is interpreted as the ratio of elemental areas in the physical and the transformed plane. For higher order elements J is a function of position. For linear elements it is a constant and equal to (element area in physical plane)/4. With these results the integrand in the element matrix K_1 can be completely determined and the integral evaluated using quadrature.

When heat flux is prescribed on a portion of the boundary, integrals such as $\int N_i ds$ must necessarily be evaluated. If triangular elements are used with second order interpolation then the side of the element that coincides with the heat flux boundary will have 3 nodes (1, 2, 3 counted in order) at which this boundary condition must be applied. Let S be the true length of the straight or curvilinear segment on the heat flux boundary. The two dimensional shape function for a six-noded triangle will reduce to quadratic interpolation on any one of its sides and the appropriate shape functions from the one dimensional problem can be used. Hence we get

$$\int N_i ds = \int_{-1}^{1} N_i s_\xi d\xi = \int_{-1}^{1} (N_i \sum_{j=1}^{3} s_j N_{j\xi}) d\xi$$

$$\begin{aligned}
\text{where } N_1 &= -\frac{1}{2}\xi(1-\xi) \\
N_2 &= 1-\xi^2 \\
N_3 &= \frac{1}{2}\xi(1+\xi) \\
s_1 &= 0 \\
s_2 &= S/2 \\
\text{and } s_3 &= S
\end{aligned}$$

It can now be shown that

$$\begin{aligned}
\int N_1 ds &= S/6 \\
\int N_2 ds &= 2S/3 \\
\int N_3 ds &= S/6
\end{aligned}$$

8.7.1 Example of Two Dimensional Heat Conduction

Consider the solution of the following problem using FEM:

$$T_{xx} + T_{yy} = 0, \quad 0 < x < 2, \quad 0 < y < 2 \tag{8.27}$$

<u>Case 1</u> $\quad T(0,y) = 0,\ T(x,0) = 0,\ T(2,y) = 0$ and $T(x,2) = 1$

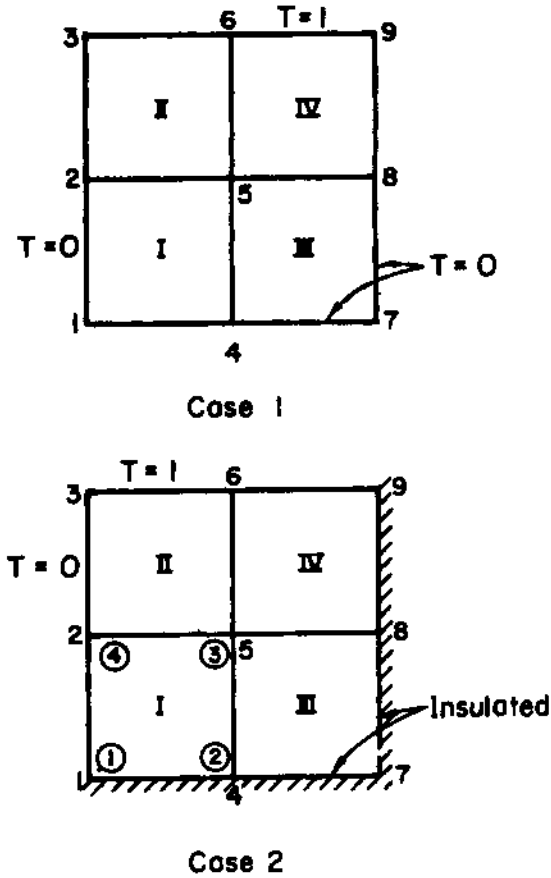


Figure 8.7: Two Dimensional Heat Conduction in a Square Domain. Local Node Numbers Encircled.

<u>Case 2</u> $\quad T(0,y) = 0,\ T_y(x,0) = 0$ and $T_x(2,y) = 0,\ T(x,2) = 1$

The discretized domain with linear quadrilateral elements and the boundary conditions are shown in Figure 8.7.

The element matrix $[K_1]^e$ is identical for all elements since thermal properties are taken to be uniform and independent of temperature. For clarity, the expressions for N_i are repeated here:

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta)$$
$$N_2 = \frac{1}{4}(1+\xi)(1-\eta)$$
$$N_3 = \frac{1}{4}(1+\xi)(1+\eta)$$

Table 8.6 Local-Global Node Record

Element	Local Node Number	Global Node Number
I	1 2 3 4	1 4 5 2
II	1 2 3 4	2 5 6 3
III	1 2 3 4	4 7 8 5
IV	1 2 3 4	5 8 9 6

$$N_4 = \frac{1}{4}(1-\xi)(1+\eta)$$

Hence

$$[K_1]^e = \begin{bmatrix} \frac{2}{3} & -\frac{1}{6} & -\frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{2}{3} & -\frac{1}{6} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{6} & \frac{2}{3} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{3} & -\frac{1}{6} & \frac{2}{3} \end{bmatrix}$$

The local-global node record is given in Table 8.6. Based on considerations of bandwidth minimization of the matrix to be inverted node numbering is initiated in the direction in which the linear dimension is smallest. For square regions there is no preferred direction (Figure 8.7).

The global (9×9) matrix is symmetric and when assembled using $[K_1]^e$ it can be shown to be as follows:

$$[K_1] = \begin{bmatrix} \frac{2}{3} & -\frac{1}{6} & 0 & -\frac{1}{6} & -\frac{1}{3} & 0 & 0 & 0 & 0 \\ -\frac{1}{6} & \frac{4}{3} & -\frac{1}{6} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ & & \frac{2}{3} & 0 & -\frac{1}{3} & -\frac{1}{6} & 0 & 0 & 0 \\ & & & \frac{4}{3} & -\frac{1}{3} & 0 & -\frac{1}{6} & -\frac{1}{3} & 0 \\ & & & & \frac{8}{3} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ & & & & & \frac{4}{3} & 0 & -\frac{1}{3} & -\frac{1}{6} \\ & & & & & & \frac{2}{3} & -\frac{1}{6} & 0 \\ & & & & & & & \frac{4}{3} & -\frac{1}{6} \\ 0 & & & & & & & & \frac{2}{3} \end{bmatrix}$$

The half-bandwidth of the matrix is 4; it corresponds to the difference, largest global node number−smallest global node number, for any element. For example for element I this difference is $5-1 = 4$.

Boundary conditions for nodes falling on insulated surfaces need not be specially applied ; for prescribed temperature boundaries the rows in the global matrix corresponding to the appropriate nodes are replaced by zeroes and unity. This procedure is described in Section 8.6.2. The right hand sides obtained

Table 8.7 Comparison of FEM against the Analytical Solution.

Case 1									
Node	1	2	3	4	5	6	7	8	9
T(analytical)	0	0	0	0	.25	1	0	0	0
T(FEM)	0	0	0	0	.25	1	0	0	1
Case 2									
Node	1	2	3	4	5	6	7	8	9
T(analytical)	0	0	0	0.364	0.5	1	0.5	0.636	1
T(FEM)	0	0	0	0.314	0.421	1	0.443	0.614	1

at this stage for cases 1 and 2 are (0,0,0,0,0,1,0,0,0,) and (0,0,0,0,0,1,0,0,1) respectively. The global matrix can be inverted to solve for the temperatures at the nodes. Analytical solutions of Equation 8.27 for the two cases are

<u>Case 1</u> $$T(x,y) = \sum_{n=1}^{\infty} \frac{2}{n\pi \sinh n\pi} \sin\frac{n\pi}{2} \; x \sinh \frac{n\pi}{2} y$$

<u>Case 2</u> $$T(x,y) = \sum_{n=1}^{\infty} \frac{4}{(2n-1)\pi \cosh \frac{2n-1}{2}\pi} \sin(\frac{2n-1}{4}\pi x) \cosh(\frac{2n-1}{4}\pi y)$$

The comparison between the FEM and the analytical solution is summarized in Table 8.7.

Errors decrease quite rapidly as the number of elements is increased. Boundary conditions at corner points tend to be ambiguous. While average values can be prescribed at such points (e.g., node 9, case 1) no treatment is available for node 1 (case 2) where an insulated surface intersects an isothermal surface. Corner points have only a limited range of influence in diffusion problems and the problem is circumvented through grid refinement. These must however be treated with care in flow problems.

As a second application of FEM to diffusion problems consider steady heat conduction in a region containing insulating fibres distributed uniformly within it. This configuration models a metal-matrix composite with low conductivity fibres embedded in it to improve its mechanical strength. It is of interest to determine the changes in the equivalent thermal conductivity of the region as a whole as a function of the fibre volume fraction. The geometry, coordinate system and boundary conditions for this problem are shown in Figure 8.8. The minimum and maximum temperatures have been scaled to the range 0 to 1 for the present discussion. The region is taken as square with a distribution of circular inhomogeneities whose conductivity is approximated as zero. The matrix conductivity is finite and the conductivity of the composite is normalized with respect to it. The effective conductivity of the composite is defined as that value for a homogeneous region which for a given overall temperature difference

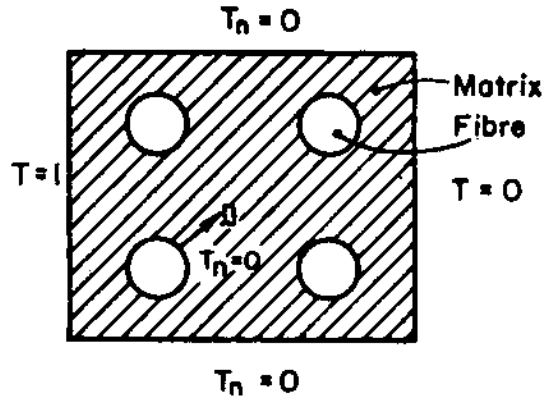


Figure 8.8: Physical Domain, Coordinate System and Boundary Conditions for Heat Conduction in a Composite Region

will permit the same amount of energy transfer through it. Since the fibres and the side walls ($y = 0, L$) are taken as insulating surfaces the energy supplied to the composite at $x = 0$ is equal to the energy leaving it at $x = L$. The transverse equivalent conductivity denoted by K_0 is then defined as

$$K_0 = -L \left.\frac{\partial T}{\partial x}\right|_{x=0} = -L \left.\frac{\partial T}{\partial x}\right|_{x=L}$$

averaged over $y = 0 - L$ and normalized with respect to the matrix conductivity. Here the temperature field is determined by solving the equation

$$\nabla^2 T = T_{xx} + T_{yy} = 0$$

in the matrix region excluded by the fibres. A coarse finite element mesh using triangular elements is shown in Figure 8.9. The possible patterns of distribution of the fibres are shown in Figure 8.10.

Typical dimensions used in the study are, region size $L = 6$ and fibre diameter $= 1$. For N, number of fibres equal to 9 this corresponds to an area fraction A of around 20 percent. The level of discretization used for this problem is such that each fibre surface is represented by 16 nodes. A larger number of nodes is seen to affect only marginally the value of K_0, the equivalent thermal conductivity. Figure 8.11 shows a plot of K_0 as a function of N and the empirical engineering formulae used in the literature for comparison. For $0 < A < 0.2$ the best line fit through the numerical data is, $K_0 = 1 - 1.63A$. For a longer discussion on the subject, see Muralidhar (1990).

8.7.2 Transient Heat Conduction with Variable Thermal Conductivity

The governing equation for transient conduction in two dimensions is

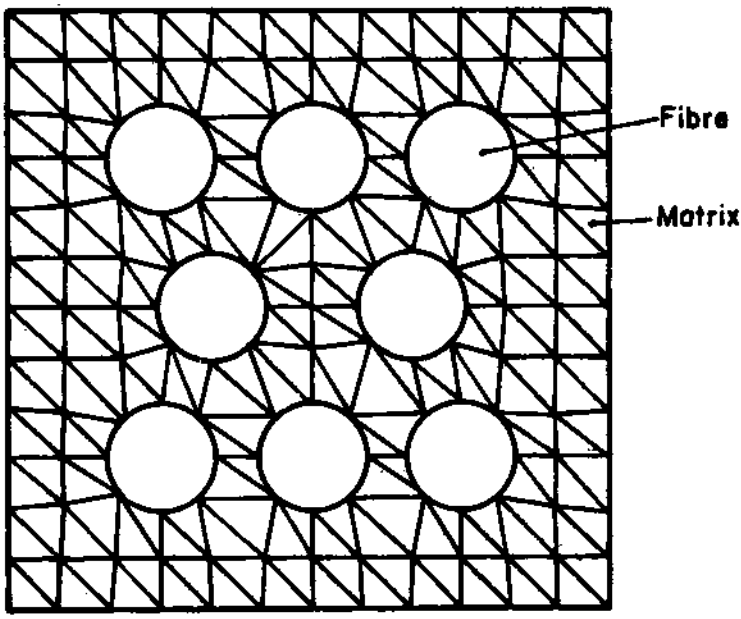


Figure 8.9: Example of FEM Mesh

$$\nabla \cdot (k\nabla T) = \rho C_p \frac{\partial T}{\partial t} \tag{8.28}$$

where for generality we take $k = k(T, x)$. The procedure for generating the element matrix is very similar to the one dimensional problem and only the final results are given below. As stated earlier n and $n+1$ represent the current and the new time levels separated by a time difference Δt. The marching algorithm is of the form $T_i^{n+1} = C_{ij} T_j^n$ where matrix C is defined as:

$$\begin{aligned}
\text{Explicit Scheme} \quad &: \quad C_{ij} = A_{ij}^{-1}(A_{ij} - \Delta t B_{ij}) \\
\text{Implicit Scheme} \quad &: \quad C_{ij} = (A_{ij} + \Delta t B_{ij})^{-1} A_{ij} \\
\text{Crank-Nicolson Scheme} \quad &: \quad C_{ij} = (A_{ij} + 0.5\Delta t B_{ij})^{-1}(A_{ij} - 0.5\Delta t B_{ij})
\end{aligned}$$

Here $A_{ij} = \rho C_p \int_{\Omega^e} N_i N_j d\mathbf{x}$ and $B_{ij} = \int_{\Omega^e} k(T, x) \nabla N_i \cdot \nabla N_j d\mathbf{x}$.

In an explicit scheme k is evaluated using the temperature field at time level n and is hence non-iterative. In the implicit and Crank-Nicolson methods k is evaluated using the temperature at the new time level $(n+1)$ and hence the process of moving from time level n to $n+1$ is itself iterative. Expressions for matrices A and B given above are identical to their counter parts in one dimension. However the integration over Ω^e is over a two dimensional region and N_i represents a two dimensional shape function. Boundary conditions in the transient problem are analogous to the steady problem. These are applied

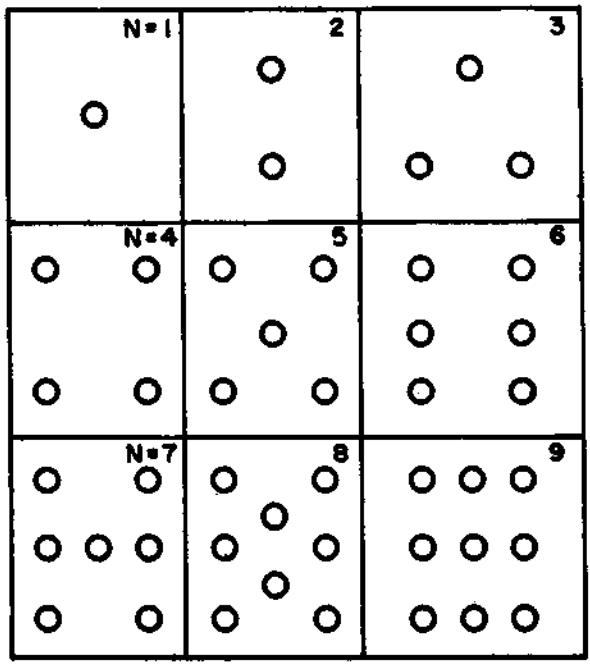


Figure 8.10: Patterns of Distribution of Fibres in a Composite.

via boundary terms associated with matrix B for Neumann conditions and by re-defining elements of matrix C for Dirichlet conditions.

Iterations can be set up for time marching from level n to $n+1$ using a Newton-Raphson scheme. This scheme for the implicit case can be easily shown to be:

$$
\begin{aligned}
T_j^{n+1} \text{ (new)} &= T_j^{n+1}(*) \text{ (guessed)} + \Delta T_j \text{ (correction)} \\
\left[\frac{\partial R_i}{\partial T_j^{n+1}}\right]\{\Delta T_j\} &= -\{R_i\}
\end{aligned}
$$

where $R_i = A_{ij}(T_j^{n+1}(*) - T_j^n) + B_{ij}^* \Delta t T_j^{n+1}(*)$ and

$$
\frac{\partial R_i}{\partial T_j^{n+1}} = A_{ij} + \Delta t \left(B_{ij}^* + \int_{\Omega^e} (\frac{dk}{dT}) N_i \nabla N_j \cdot \nabla T d\mathbf{x} \right)
$$

As in the one dimensional problem $k, dk/dT$ and B_{ij} are computed using the guessed temperature field $T_j^{n+1}(*)$. Iterations within a time step continue till ΔT_j becomes quite small.

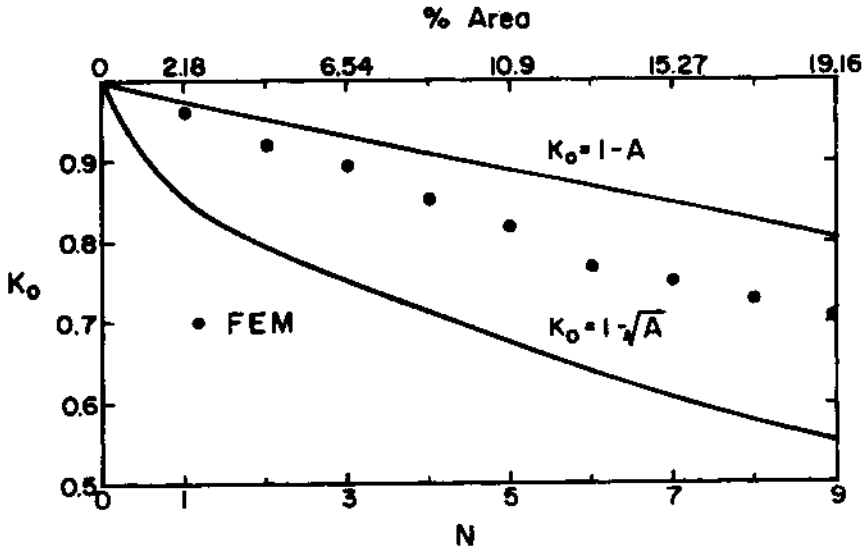


Figure 8.11: Plot of K_0 as a Function of Fibre Area Fraction

8.8 Advective Heat Transfer

The dimensionless form of the equation governing steady advective heat transfer in the presence of a prescribed flow field $\mathbf{u} = (u, v)$ is given as

$$uT_x + vT_y = \frac{1}{\mathrm{Pe}}(T_{xx} + T_{yy}) \tag{8.29}$$

with suitable boundary conditions for the temperature field $T(x, y)$. We proceed to solve for T as:

(i) Discretize the flow domain into triangular or rectangular elements with shape functions $N_i, i = 1..k$ where k is the number of nodes per element.

(ii) Expand $T(x, y)$ as $T = \sum_{i=1}^{k} N_i T_i$, where T_i are the temperature values at the nodes of the element.

(iii) Integrate Equation 8.29 over an element as

$$\left[\int_{\Omega^e} uT_x + vT_y - \frac{1}{\mathrm{Pe}}(T_{xx} + T_{yy})N_i(\mathbf{x})d\mathbf{x}\right] = 0$$

(iv) Recast the integral equation in matrix form as

$$[K_{ij}]\{T_j\} = 0$$

where

$$K_{ij} = \int_{\Omega^e} ((\mathbf{u} \cdot \nabla N_j) N_i + \frac{1}{\mathrm{Pe}} (\nabla N_i \cdot \nabla N_j)) d\mathbf{x}$$
$$= \int_{\Omega^e} \left\{ (u N_i N_{jx} + v N_i N_{jy}) + \frac{1}{\mathrm{Pe}} (N_{ix} N_{jx} + N_{iy} N_{jy}) \right\} d\mathbf{x}$$

and as before $N_{ix} = \partial N_i / \partial x$ and $N_{jy} = \partial N_j / \partial y$. Also u and v are assumed to be known quantities at every node. Element matrices K_{ij} must be assembled to form a global matrix valid for the entire region.

(v) The form of the matrix equation given above includes boundary conditions of the form $\partial T / \partial n = 0$, where n is the unit outward normal. Dirichlet boundary conditions are applied to the global matrix directly by replacing the appropriate nodal equation by the statement, T = a prescribed value. The heat flux boundary condition is imposed by retaining the boundary terms arising from integration by parts. For a boundary condition $-\partial T / \partial n = 1$ the element-level equation is

$$\int_{\Omega^e} \left((\mathbf{u} \cdot \nabla N_j) N_i + \frac{1}{\mathrm{Pe}} (\nabla N_i \cdot \nabla N_j) \right) d\mathbf{x} = -\frac{1}{\mathrm{Pe}} \int_{\partial \Omega^e} N_i ds$$

(vi) The last step in FEM is the inversion of the matrix equation using a suitable sparse matrix solver.

The matrix equation arising from 8.29 can become ill-conditioned at large values of Pe, the Peclet number in the sense that off-diagonal terms become more prominent than the diagonal. Conventional sparse matrix solvers can introduce substantial errors when such matrices are encountered. The conditioning of the matrix can be improved by well-designed preconditioners and certainly by grid refinement.

The solution of transient advection equation

$$T_t + u T_x + v T_y = \frac{1}{\mathrm{Pe}} (T_{xx} + Tyy) \tag{8.30}$$

proceeds along the steps given in Section 8.6.4 for a pure diffusion problem except that the matrix B must be modified to account for the presence of a velocity field. The formulas given in Section 8.6.4 for explicit, implicit and Crank-Nicolson schemes carry over with matrix B defined as

$$B_{ij} = \int_{\Omega^e} \left[(u N_i N_{jx} + v N_i N_{jy}) + \frac{1}{\mathrm{Pe}} (N_{ix} N_{jx} + N_{iy} N_{jy}) \right] d\mathbf{x}$$

For the non-linear problem where the thermal conductivity depends on temperature, the Newton-Raphson scheme described in Section 8.5.2 also carries over unchanged with the new definition of matrix B given above.

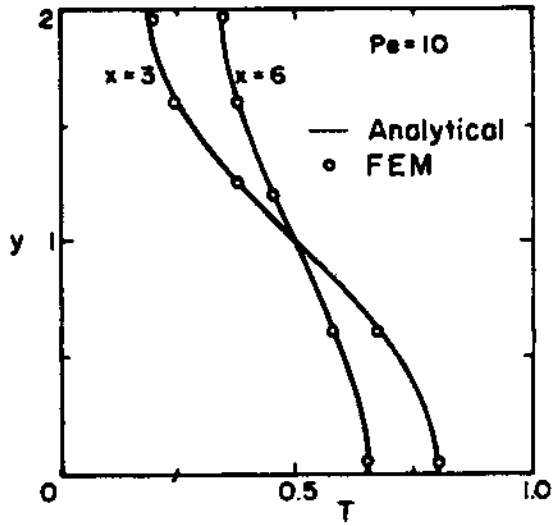


Figure 8.12: Comparison between FEM and Analytical Solution for Steady Advective Heat Transfer

The formulation given in this section has been tested against two model problems. The first problem is steady advection with a unidirectional velocity $\mathbf{u} = (1,\ 0)$ and an inflow thermal boundary condition $T(0,y) = 0$ if $y > 1, T(0,y) = 1$ if $y < 1$ and $T(0,1) = 0.5$. This represents two parallel streams of equal velocity but unequal temperature mixing along their interface. The side walls confining the flow are taken to be insulating. The outflow condition is assumed to be $\mathbf{u} \cdot \nabla T = T_x = 0$. Computations employ a region size of (6×2), six-noded triangular elements and a total of around 300 nodes. For Pe$\gg 1$ Equation 8.29 can be simpified to

$$T_x = \frac{1}{\mathrm{Pe}} T_{yy}$$

with inflow and boundary conditions as given for the full problem. The analytical solution of this simplified equation is

$$T(x,y) = \frac{1}{2} - \sum_{m=1}^{\infty} \frac{2}{(2m-1)\pi} \sin \frac{2m-1}{2}\pi \cos \frac{(2m-1)\pi y}{2} \exp(-\frac{(2m-1)\pi^2 x}{4\mathrm{Pe}})$$

A comparison between analytical and FEM solutions is shown in Figure 8.12 for Pe $= 10$; the comparison is seen to be good.

The second model problem involves unsteady one dimensional transport and is given as

$$T_t + T_x = \frac{1}{\mathrm{Pe}} T_{xx}$$

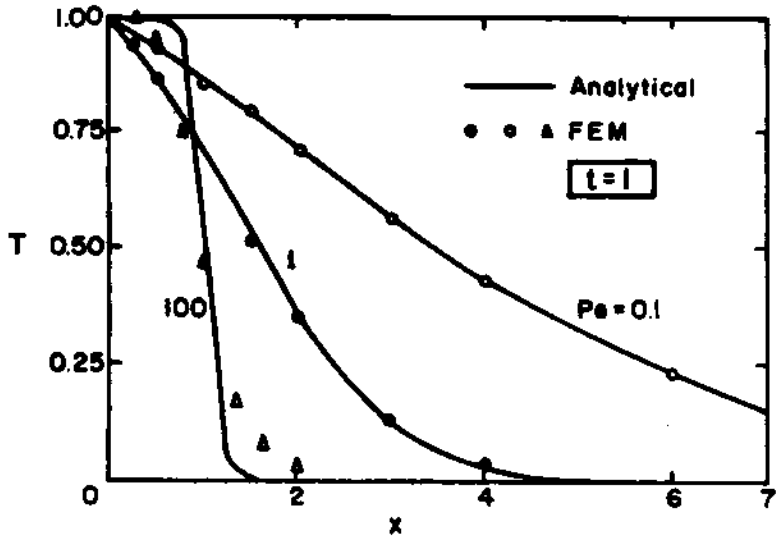


Figure 8.13: Comparison between FEM and Analytical Solution in Transient One Dimensional Advective Heat Transfer.

subject to the initial conditon $T(x,0) = 0$ and boundary conditions $T(0,t) = 1$ and $T(\infty,t) = 0$. Grid requirement is typically 100 nodes with a time step of 0.02. The analytical solution of this equation is

$$T(x,t) = \frac{2}{\sqrt{\pi}} \int_{x/\sqrt{\frac{4t}{\mathrm{Pe}}}}^{\infty} e^{-\eta^2} \exp\left[\mathrm{Pe}\frac{x}{2} - \frac{x^2}{16\eta^2}\mathrm{Pe}\right] d\eta$$

A comparison between FEM and the analytical solution at time $t = 1$ for three different Peclet numbers is shown in Figure 8.13. Even at as high a Peclet number as 100 the comparison is seen to be satisfactory.

8.9 Incompressible Steady Flows

In two dimensions, the continuity and momentum equations (namely, the *Navier-Stokes equations*) applicable for incompressible steady flow are given in dimensionless form as

$$u_x + v_y = 0 \tag{8.31}$$

$$uu_x + vu_y = -p_x + \frac{1}{\mathrm{Re}}(u_{xx} + u_{yy}) \tag{8.32}$$

$$uv_x + vv_y = -p_y + \frac{1}{\mathrm{Re}}(v_{xx} + v_{yy}) \tag{8.33}$$

We view the momentum Equations 8.32 and 8.33 as determining the velocity field $\mathbf{u} = (u, v)$ and the continuity Equation 8.31 as determining pressure. Clearly the three equations are coupled and must be simultaneously solved to extract both $\mathbf{u}$ and p. The extension of FEM to flow problems results in an element matrix structure that is different from examples considered in the earlier sections. The first difference arises from the need to solve for three unknowns at a given node and second arises for the following reason (Baker, 1983):

We can expand u and v over a six-noded triangle using quadratic interpolation but p only over the 3 corner nodes with a drop of one order of interpolation. This is because the highest derivative in p is one order smaller than the highest derivative in $\mathbf{u}$ in the governing equations. This is called unequal order interpolation and is crucial for the stability of this method. If $\mathbf{u}$ is interpolated over the three corner nodes of the triangle then p can be determined at only one point, normally the centroid of the triangle. This procedure is quite similar to the staggered grid approach used in finite difference algorithms such as MAC and SIMPLE where velocity nodes are placed on the cell face and pressure node at the cell-centre (Chapter 7).

For definiteness let us expand (u, v) over a six-noded triangle as

$$(u, v) = \sum_{i=1}^{6} (u_i, v_i) N_i$$

and pressure using the corner nodes as

$$p = \sum_{i=1}^{3} p_i L_i$$

L and N are linear and quadratic shape functions, respectively. For clarity we repeat their expressions here (see Figure 8.4 for node numbering over a standard triangle).

$$\begin{aligned}
L_1 &= 1 - \xi - \eta, \quad L_2 = \xi, \quad L_3 = \eta. \\
N_1 &= L_1(2L_1 - 1), \quad N_2 = L_2(2L_2 - 1), \quad N_3 = L_3(2L_3 - 1) \\
N_4 &= 4L_1L_3, \quad N_5 = 4L_1L_2, \quad N_6 = 4L_2L_3
\end{aligned}$$

Since the continuity equation determines pressure, the corresponding weighting function for integration is L_i. For momentum equations this function is N_i. Hence the element matrix equation is obtained by evaluating each term of the following equations

$$Rp_i = \int_{\Omega^e} L_i(u_x + v_y) d\mathbf{x} = 0, \quad i = 1..3 \tag{8.34}$$

$$
\begin{aligned}
Ru_i &= \int_{\Omega^e} N_i(uu_x + vu_y \\
&+ p_x - \frac{1}{\mathrm{Re}}(u_{xx} + u_{yy}))d\mathbf{x} = 0, \quad i = 1..6 \qquad (8.35)\\
Rv_i &= \int_{\Omega^e} N_i(uv_x + vv_y \\
&+ p_y - \frac{1}{\mathrm{Re}}(v_{xx} + v_{yy}))d\mathbf{x} = 0, \quad i = 1..6 \qquad (8.36)
\end{aligned}
$$

In each integral u, v and p are represented in terms of the applicable shape functions. The element matrix equation derived from these equations has a structure as:

$$
\begin{bmatrix}
\underset{6\times 6}{[K_1]} & & \underset{6\times 6}{0} & & \underset{6\times 3}{[K_2]} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\underset{6\times 6}{0} & & \underset{6\times 6}{[K_1]} & & \underset{6\times 3}{[K_3]} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\underset{3\times 6}{[K_2]^T} & & \underset{3\times 6}{[K_3]^T} & & \underset{3\times 3}{0}
\end{bmatrix}
\begin{Bmatrix} u_i \\ \cdots \\ v_i \\ \cdots \\ p_i \end{Bmatrix}
=
\begin{Bmatrix} Bu_i \\ \cdots \\ Bv_i \\ \cdots \\ 0 \end{Bmatrix}
$$

The notation $[\ ]^T$ refers to the transpose of a matrix. As in the examples considered earlier the right hand side vector consists of boundary integrals arising from integration by parts of terms such as $N_i\nabla^2 u$ and $N_i\nabla p$. These cancel out between contiguous elements and can be dropped altogether when Dirichlet boundary conditions such as $\mathbf{u} = 0$ on a solid boundary are prescribed. Complete forms of the entries in the element-level matrix are:

$$
\begin{aligned}
K_{1ij} &= \int_\Omega \left(\frac{1}{\mathrm{Re}}(\nabla N_i \cdot \nabla N_j) + uN_i\frac{\partial N_j}{\partial x} + vN_i\frac{\partial N_j}{\partial y}\right) dxdy, \quad i,j = 1..6\\
K_{2ij} &= \int_\Omega \frac{\partial N_i}{\partial x} L_i dxdy, \quad i = 1..6, j = 1..3\\
K_{3ij} &= \int_\Omega \frac{\partial N_i}{\partial y} L_j dxdy, \quad i = 1..6, j = 1..3\\
Bu_i &= \int_{\partial\Omega} N_i\left(\frac{1}{\mathrm{Re}}\frac{\partial u}{\partial n} - pn_x\right) ds, \quad i = 1..6\\
Bv_i &= \int_{\partial\Omega} N_i\left(\frac{1}{\mathrm{Re}}\frac{\partial v}{\partial n} - pn_y\right) ds, \quad i = 1..6
\end{aligned}
$$

Here $\mathbf{n}$ is a unit outward drawn normal on the boundary of the element under question and has components n_x and n_y.

The above matrix differs from those encountered earlier in that some of the diagonal terms are zero. This can lead to difficulties in applying direct solvers to

the global matrix. Such a problem can be circumvented by first interchanging rows of the matrix till all diagonal entries are non-zero. If the matrix needs to be repeatedly inverted (due to non-linearity or in transient problems) the sequence of operations which preconditions the matrix structure can also be stored so that successive row-manipulations are less time consuming compared to the first. In fact successful use of FEM for flow problems depends quite crucially on the availability of a powerful sparse matrix solver that can accommodate zeroes as well as numerically small numbers along the diagonal.

Another difference between the matrices of the flow and the diffusion problems is that some of the elements in the former contain $\mathbf{u}$, i.e., the dependent variable itself. This is of course due to the non-linearity of the governing equations. It can be treated easily by replacing $\mathbf{u}$ in K_{1ij} by $\bar{\mathbf{u}}$, a guessed velocity profile or one calculated at the previous iteration.

Note that the pressure boundary condition need be specified at only one point in the flow domain since only its first derivatives appear in the governing equations. Typically p is specified as zero at the outflow plane where flow is either uniform or fully developed so that $\partial u/\partial n$ is also zero on this boundary. When these conditions are met, the boundary terms may be dropped once again in the right hand side vector.

An alternative approach to constructing the stiffness matrix using the *penalty* formulation has been described by Hughes (1987), in the context of incompressible elasticity.

As an example of node numbering in flow problems consider the example shown in Figure 8.14. The region is square and is divided into 8 triangular elements. At corner nodes both velocity and pressure are unknowns; at mid-side nodes only velocity is determined. The local-global record of node numbers for two typical elements I and VIII is shown in Figure 8.15. At a corner node numbers such as (29, 30, 31) stand for u, v and p while at mid-side node numbers such as (16,17) represent u and v. The local-global record of node numbers for all the 8 elements in Figure 8.14 is summarized in Table 8.7. Such a record is essential for constructing the global matrix through assembly of the element matrices. For example when element III is assembled into the global matrix, entry (6, 3) at the element level is added to the number at (26, 30) of the global matrix. Similarly (4, 11) and (13, 14) at the element level are combined with entries at (16, 15) and (3, 26) respectively.

Tables such as 8.7 as well as information regarding x and y coordinates of each node and a list of boundary nodes are routinely developed through mesh generation (computer) programs.

The Navier-Stokes equations are non-linear and as a result the matrix equation arising from a finite element aproximation can only be iteratively solved. The strategy for iteration described earlier involves using the current guesses of the dependent variables ($\mathbf{u}$ and p) during the construction of the matrix. These guesses are updated when the matrix equation is solved. Iterations continue till the dependent variables converge to a predetermined criterion, e.g. 0.01 percent maximum change in the flow field.

Table 8.7 Generating Element-wise Node Record.

Element	Variable	Global Node Numbers					
I	u	1	24	29	16	14	27
	v	2	25	30	17	15	28
	p	3	26	31			
II	u	29	6	1	16	18	4
	v	30	7	2	17	19	5
	p	31	8	36			
III	u	6	29	34	20	18	32
	v	7	30	35	21	19	33
	p	8	31	36			
IV	u	34	11	6	20	22	9
	v	35	12	7	21	23	10
	p	36	13	8			
V	u	24	47	52	39	37	50
	v	25	48	53	40	38	51
	p	26	49	54			
VI	u	52	29	24	39	41	27
	v	53	30	25	40	42	28
	p	54	31	26			
VII	u	29	52	57	43	41	55
	v	30	53	58	44	42	56
	p	31	54	59			
VIII	u	57	34	29	43	45	32
	v	58	35	30	44	46	33
	p	59	36	31			

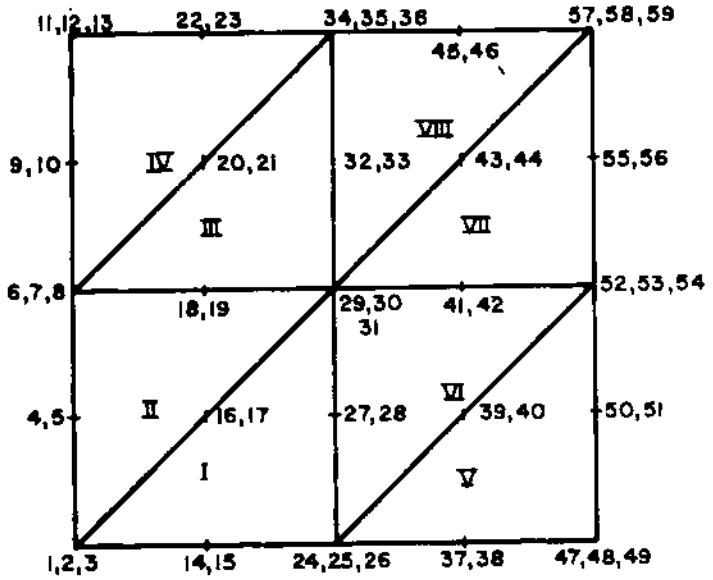


Figure 8.14: Node Numbering in Flow Problems

Starting from an initial guess which lies within its radius of converence, a Newton-Raphson scheme provides a quadratic convergence rate for the iterated quantities. Hence it requires fewer iterations to obtain a converged solution. It requires more intermediate calculations per iteration; however as stated earlier, in most problems of practical interest it provides an overall computing advantage.

The roots of a system of algebraic equations $f_i(x_i) = 0, i = 1..N$ are found iteratively using the Newton-Raphson method by solving the matrix equation

$$\left[\frac{\partial f_i}{\partial x_j}\right]\{\delta x_j\} = -\{f_i\} \tag{8.37}$$

where $\delta x_j = x_j$ (new) $- x_j$ (old). Iterations continue till $\mid \delta x_j \mid$ and $\mid f_i(x_j) \mid$ for each i and j become quite small. We view u, v and p as the roots of the system of simultaneous Equations 8.34-8.36. Hence the corrections $\Delta u, \Delta v$ and Δp to an assumed velocity and pressure field are computed from the following matrix equation

$$\begin{bmatrix} \partial Ru_i/\partial u_j & \partial Ru_i/\partial v_j & \partial Ru_i/\partial p_j \\ \partial Rv_i/\partial u_j & \partial Rv_i/\partial v_j & \partial Rv_i/\partial p_j \\ \partial Rp_i/\partial u_j & \partial Rp_i/\partial v_j & 0 \end{bmatrix} \begin{Bmatrix} \Delta u_j \\ \Delta v_j \\ \Delta p_j \end{Bmatrix} = -\begin{Bmatrix} Ru_i \\ Rv_i \\ Rp_i \end{Bmatrix}$$

Iterations continue till the corrections ($\Delta u, \Delta v$ and Δp) become quite small. Based on Equations 8.34-8.36, the residuals and their derivatives can be obtained as:

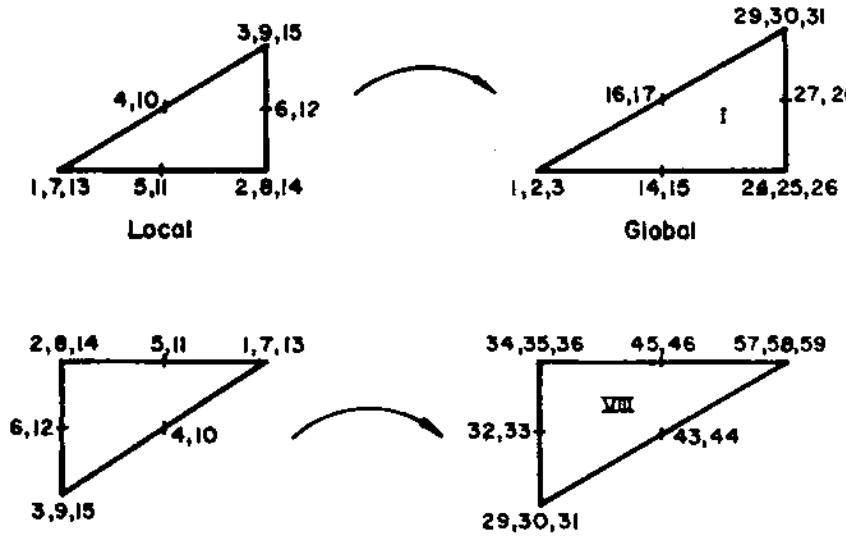


Figure 8.15: Generation of Local-Global Record of Nodes.

$$Ru_i = \int \left[\frac{1}{\mathrm{Re}}(N_{ix}N_{jx} + N_{iy}N_{jy}) + N_i(uN_{jx} + vN_{jy})\right]\{u_j\}d\mathbf{x}$$

$$- \int [N_{ix}L_j]\{p_j\}d\mathbf{x}, \quad i = 1..6$$

$$Rv_i = \int \left[\frac{1}{\mathrm{Re}}(N_{ix}N_{jx} + N_{iy}N_{jy}) + N_i(uN_{jx} + vN_{jy})\right]\{v_j\}d\mathbf{x}$$

$$- \int [N_{iy}L_j]\{p_j\}d\mathbf{x}, \quad i = 1..6$$

$$Rp_i = \int L_i\,[N_{jx}u_j + N_{jy}v_j]\,d\mathbf{x}, \quad i = 1..3$$

In the integrals summation over $j = 1..6$ is implied. Entries in the matrix equation 8.37 can be determined from the expressions for residues given above. Derivatives such as $\partial Ru_i/\partial u_j$ are computed as $\partial Ru_i/\partial u\ \partial u/\partial u_j = N_j\ \partial Ru_i/\partial u$. Complete form of the derivatives are given as:

$$\begin{aligned}
\frac{\partial Ru_i}{\partial u_j} &= \int \left[\frac{1}{\mathrm{Re}}(N_{ix}N_{jx} + N_{iy}N_{jy}) + N_i((uN_j)_x + vN_{jy}))\right] d\mathbf{x} \\
\frac{\partial Ru_i}{\partial v_j} &= \int N_i N_j \frac{\partial u}{\partial y} d\mathbf{x} \\
\frac{\partial Ru_i}{\partial p_j} &= -\int L_j N_{ix} d\mathbf{x}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial Rv_i}{\partial u_j} &= \int N_i N_j \frac{\partial v}{\partial x} d\mathbf{x} \\
\frac{\partial Rv_i}{\partial v_j} &= \int \left[\frac{1}{\mathrm{Re}} (N_{ix} N_{jx} + N_{iy} N_{jy}) + N_i (u N_{jx}) + \frac{\partial}{\partial y} (v N_j) \right] d\mathbf{x} \\
\frac{\partial Rv_i}{\partial p_j} &= \int L_j N_{iy} d\mathbf{x} \\
\frac{\partial Rp_i}{\partial u_j} &= \int L_j N_{jx} d\mathbf{x} \\
\frac{\partial Rp_i}{\partial v_j} &= \int L_i N_{jy} d\mathbf{x} \\
\frac{\partial Rp_i}{\partial p_j} &= 0
\end{aligned}
$$

Since the corrections Δu, Δv, and Δp are treated here as the dependent variables, the boundary conditions must be imposed in terms of these corrections. When a Dirichlet boundary condition is prescribed, the velocities and pressure are provided along with the initial guess. Hence it is sufficient that the individual corrections for u, v or p be set equal to zero at the appropriate nodes.

8.9.1 Example of Flow in an Expanding Channel

The formulation given in Section 8.7 for computing the velocity and pressure fields is applied to the problem of flow in a gradually expanding channel (Figure 8.16). The inflow velocity profile is parabolic. The channel shape is taken as the curve $y = (\tanh(2 - 3x) - \tanh(2))/2$. Benchmark results for the velocity vectror $\mathbf{u}$ and pressure p at Re = 10 are given by Napolitano and Orlandi (1985) and these are used as a reference solution for the finite element calculation. Two grids, orthogonal and non-orthogonal are considered here and shown in Figure 8.17. Results given below employ triangular elements, 441 nodes each for u and v and 121 nodes for p. Figure 8.18 shows the velocity profile in the channel at Re = 10 and 100. We see incipient separation at Re = 10 but this is quite clear at Re = 100.

Figure 8.19 shows the wall pressure profiles at Re = 10 and 100 determined using both orthogonal as well as the non-orthogonal grids. The reference solution is shown at Re = 10 ; the pressure values for an orthogonal grid match this solution quite closely especially where the expansion rate of the channel is large. At Re = 100 the wall pressure variations computed using the two grids are seen to be close to each other.

Figure 8.20 shows wall vorticity distribution at Re = 10 and 100. Vorticity ω is computed as $v_x - u_y$, i.e., $\omega = \sum_{i=1}^{6} (v_i N_{ix} - u_i N_{iy})$ at any point of interest. At Re =10 the wall vorticity on an orthogonal grid matches the reference solution better than the non-orthogonal grid. At Re = 100 the vorticity values for both grids are close to each other. This example is quite useful for benchmarking

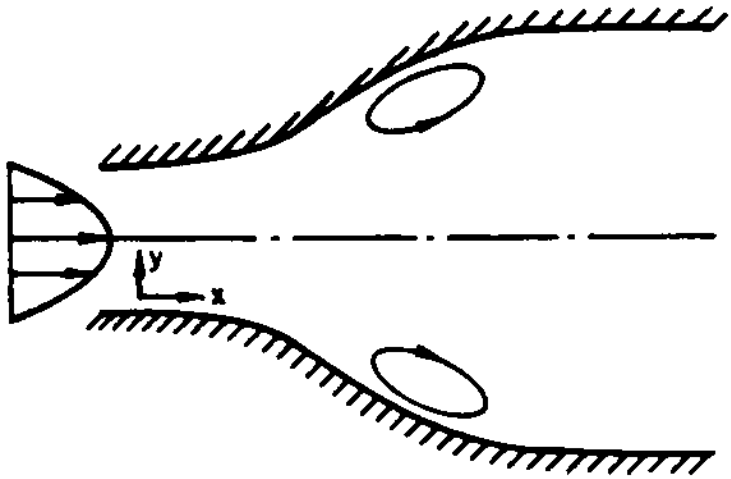


Figure 8.16: Schematic Drawing of an Expanding Channel.

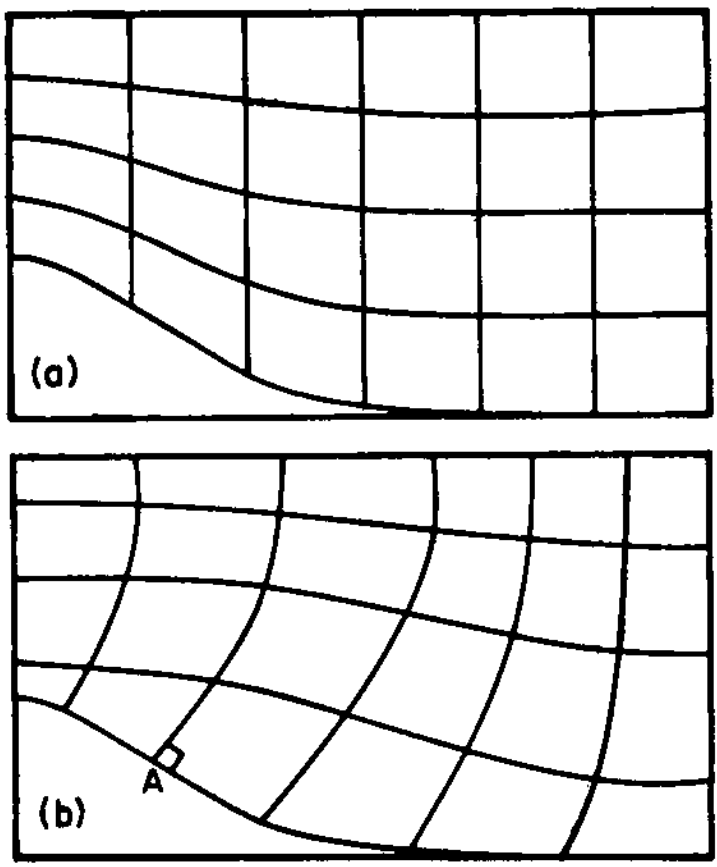


Figure 8.17: Examples of Grids Used for Flow Calculations: a. Nonorthogonal, b. Orthogonal.

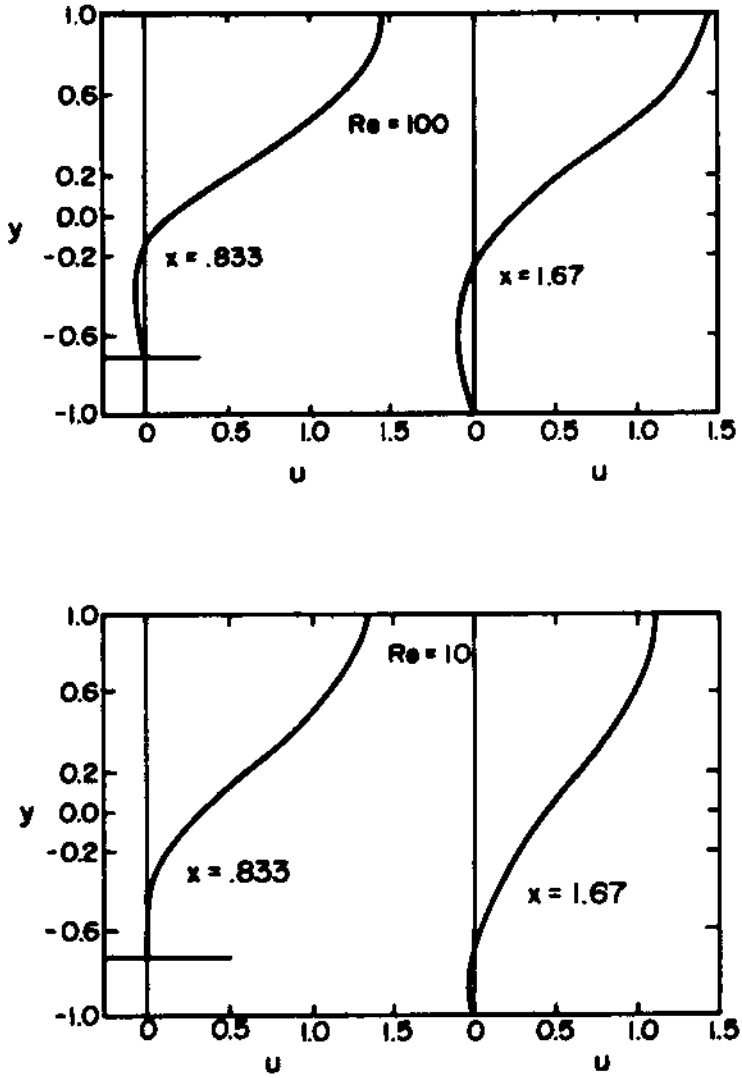


Figure 8.18: Velocity Profiles in an Expanding Channel.

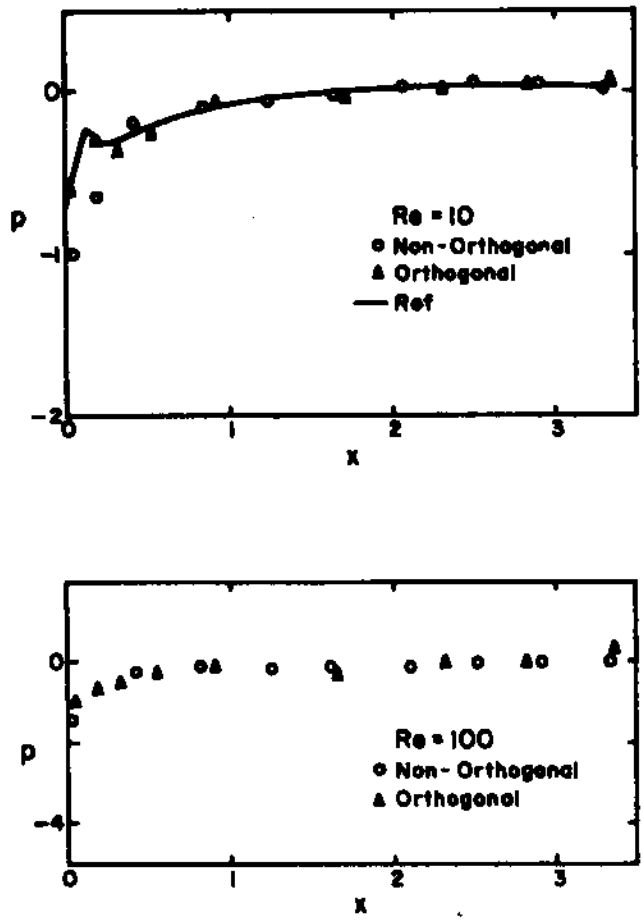


Figure 8.19: Wall Pressure Profiles in an Expanding Channel.

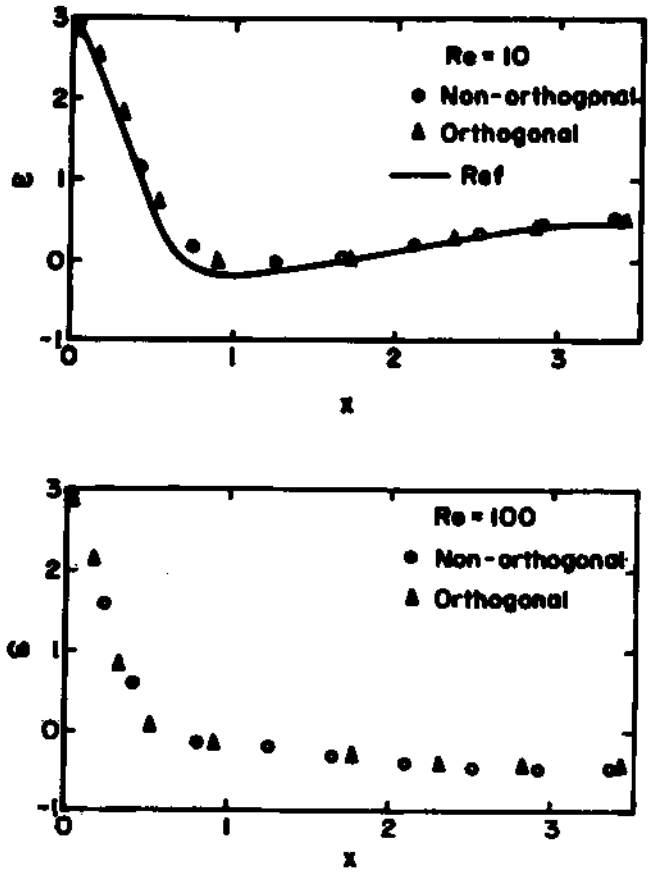


Figure 8.20: Wall Vorticity Profiles in an Expanding Channel.

FEM codes. At higher Reynolds numbers, a clear steady state may not be attained. Time-averaged velocity, pressure and vorticity fields may however be defined. In this respect, the data given in the review paper of Rodi *et al.* (1997) should be consulted.

References

Baker, A. J., *Finite Element Computational Fluid Mechanics,* McGraw Hill,New York, 1983.

Carey, G.F. and Oden, J.T., *Finite Elements:* Volumes 1-6, Prentice-Hall, New Jersey, 1983.

Chawla, K. K., *Composite Materials, Springer,* New York, 1987.

Gebhart B., Jaluria Y., Mahajan R.L., and Sammakia B., *Buoyancy-Induced Flows and Transport,* Hemisphere Publishing Corporation, New York, 1988.

Hughes, T.J.R., *The Finite Element Method,* Prentice-Hall, New Jersey, 1987.

Muralidhar, K., Equivalent Conductivity of a Hetrogeneous Medium, *Int J Heat and Mass Transfer,* Vol 33, pp 1759 - 1766, 1990.

Muralidhar, K., Pulsatile Flow Past a Thermally Participating Cylinder: A Regenerator Model, *Numerical Heat Transfer* Part A, Vol. 33, pp 653-671, 1998.

Napolitano, M. and Orlandi, P., Laminar Flow in a Complex Geometery: A Comparison, *Int J Num Methods Fluids,* Vol 5, pp 667 - 683, 1985.

Rodi, W., Ferziger, J.H., Breuer, M., and Pourquie, M., Status of Large Eddy Simulation: results of a Workshop, *Trans. ASME: J. Fluids Engg.*, Vol. 119, pp. 248-270, 1997.

Zienkiewicz, O.C. and Taylor, R.L., *The Finite Element Method,* fourth edition, Volumes 1-2, McGraw-Hill, New York, 1991.

Problems

1. Develop computer programs for the examples considered in this chapter. A recommended sequence for this exercise is

   (a) Steady heat conduction in a rectangular geometry with Dirichlet boundary conditions.

   (b) Steady nonlinear heat conduction in a complex geometry with Dirichlet boundary conditions.

   (c) Unsteady nonlinear heat conduction in a complex geometry with Dirichlet and Neumann boundary conditions.

   (d) Unsteady advective heat transfer in a channel geometry.

   (e) Steady incompressible fluid flow in a channel geometry.

(f) Unsteady incompressible fluid flow in a channel geometry.

(g) Flow past a circular cylinder confined in a parallel plates channel, including unsteady effects and heat transfer.

2. Combine the PCG algorithm of matrix inversion described in Chapter 2 with the FEM calculations for heat transfer as well as fluid flow. Elementary problems such as conduction in a square region and flow between parallel plates can be initially studied.

3. Determine the transient temperature profiles in a composite region consisting of circular fibres if the region is initially cold. Determine the equivalent conductivity K_0 of the composite subjected to a periodic thermal loading. What is the loading frequency at which K_0 becomes less than 0.01.

4. Consider steady flow past a cylinder of unit diameter. The cylinder is heated with a uniform heat flux q_w. Determine the temperature distribution on the cylinder surface as a function of position and time.

5. Solve the equation

$$\frac{\partial T}{\partial t} + T_x = \frac{1}{\mathrm{Pe}} T_{xx} - \lambda T, \quad \lambda > 0$$

in a region $0 < x < \infty$ with the initial condition $T(x, 0) = 0$ and the inflow condition $T(0, t) = 1$. Find the time required to reach steady state as a function of λ.

6. Solve the Navier-Stokes equations for flow past a circular cylinder for a Reynolds number $\mathrm{Re} < 20$. Plot the size of the separation bubble as a function of Re. Identify the onset of unsteadiness in the wake by increasing Reynolds number upto 40. Repeat the problem for a cylinder of square cross-section, and for a symmetric Joukowsky aerofoil cross-section.

7. Repeat the above problem for a pair of cylinders placed in-line with respect to the flow direction.

8. Develop the finite element equations, initial aad boundary conditions for unsteady fluid flow using the Newton-Raphson iterations. Suggest an extension of upwinding (Chapter 4) to FEM to stabilize flow calculations at large Reynolds number.

9. Consider purely oscillatory flow in a parallel plates channel, driven by a sinusoidal pressure gradient. Explain the application of boundary conditions, specifically at the outflow plane, for the finite element discretization of the governing equations.

   In many applications, the flow path is spatially periodic, for example a rod bundle in heat exchangers, or internally finned tubes. It is computationally economical in such geometries to solve the flow equations in a single repeating module, instead of the system as a whole. In this connection, discuss the application of a *periodicity* boundary condition in FEM.

10. Generalize the formulation given in this chapter to solve jointly for velocity, pressure and temperature. A variety of buoyancy-driven convection problems can be addressed with this code. For benchmark results of laminar natural convection in square and annular geometries, see Gebhart *et al.* (1984).

9
Panel Methods

9.1 Introduction

Panel methods were originally developed for solving the potential equation $\nabla^2\phi = 0$ for inviscid, incompressible, irrotational flow past objects of arbitrary shape. Here ϕ is the velocity potential defined as $\mathbf{u} = \nabla\phi$. It satisfies the impermeability condition $\mathbf{u} \cdot \mathbf{n} = \partial\phi/\partial n = 0$ on the surface of the solid object which is characterized by the unit normal $\mathbf{n}$. As a rule the approach flow far away from the solid object is assumed to be uniform. In two dimensions such problems can be solved by the method of complex potentials. However there is no systematic manner in which these potentials can be developed for complex geometries. The panel method does not experience this difficulty and is hence popularly used in practice.

The velocity field obtained via potential theory does not show separation in an adverse pressure gradient and the transition of laminar flow to turbulence. Further, the forces acting on objects placed in the flow are predicted as zero. Despite these drawbacks a potential velocity field does reveal qualitatively the overall effect of a disturbance on flow. The potential solution, viewed as a result for infinite Reynolds number is also essential for calculating higher order (boundary-layer) velocity profiles at finite Reynolds numbers.

Though panel methods came into existence in the context of potential flow calculations these have now been extended to include important features of real flow fields. This extension is possible when panel methods are coupled with approximations that are backed by experiment. The flow calculation is then closer to a simulation where the important features of the physical problem are reproduced with the barest minimum amount of measured data. Simulation of vortex shedding from bluff bodies, transonic flows including shock capturing and computation of lift and drag are now possible with panel methods. Chapter 10 deals with the solution of the full potential equation for transonic flows using the panel method. The present chapter primarily deals with inviscid, incompressible, irrotational flows for which ϕ satisfies Laplace equation. Extension towards determining quantities of engineering interest is described

where appropriate.

While the stream function ψ can be used as a potential for the flow field, panel methods normally employ the velocity potential ϕ as a dependent variable because of the following reasons:

On the surface of a solid object ψ = constant. This constant can be taken as zero if only one obstacle is present. If many objects are present the values of the constants are not known before hand and must be obtained as a part of the solution. In practice this is quite cumbersome. In contrast the boundary condition for ϕ can be uniformly applied for all obstacle surfaces. Secondly a stream function is defined only for two dimensional flows. The velocity potential concept can however be extended readily to three dimensions. In two dimensions the stream function can be extracted from ϕ by integrating the Cauchy-Riemann conditions

$$\psi_y = \phi_x \quad \text{and} \quad \psi_x = -\phi_y$$

When a panel method is used the sequence of calculations for the flow quantities is:

(a) Determine ϕ by solving

$$\nabla^2 \phi = 0$$

with the impermeability boundary condition

$$\frac{\partial \phi}{\partial n} = 0$$

on solid surfaces. Additional conditions may be imposed in the simulation of real flow fields.

(b) Determine the velocity field using the definition $\mathbf{u} = \nabla \phi$.

(c) Recover the pressure field from Bernoulli's equation

$$\frac{p}{\rho} + \frac{\mathbf{u}^2}{2} = \text{constant for steady flows}$$

$$\frac{\partial \phi}{\partial t} + \frac{p}{\rho} + \frac{\mathbf{u}^2}{2} = \text{constant for unsteady flows in which the approach flow at infinity continues to be steady}$$

$$\frac{\partial \phi}{\partial t} + \frac{p}{\rho} + \frac{\mathbf{u}^2}{2} = \text{function of time when the approach flow is itself unsteady}$$

(d) Integrate the pressure field suitably on the object surface to determine lift and drag.

(e) In practice the stream function is required primarily for drawing streamlines and visualize the flow pattern. Once the function $\psi(x, y)$ is known, the streamlines can be drawn as contours with ψ = constant on each

curve. Computation of ψ is normally initiated at the far-field undisturbed streamline with a value equal to ψ_1 that can be taken as zero. At a neighbouring point ψ_2 is determined as

$$\psi_2 - \psi_1 = \int_1^2 (udy - vdx)$$

By integrating in the y direction alone, i.e. by keeping the x-coordinate a constant for points 1 and 2 this integral reduces to

$$\psi_2 - \psi_1 = \int_1^2 udy$$

Here u is determined from step (b). The integral can be approximated as $(y_2 - y_1)(u_2 + u_1)/2$ when points 1 and 2 are sufficiently close. Two checks on the accuracy of this approximation are: constancy of ψ on a solid surface and the recovery of a second undisturbed streamline away from the obstacles placed in flow.

The panel method finds a theoretical justification in the Green's function method of solving partial differential equations (Roach, 1986). The singular potentials corresponding to sources, vortices and doublets are interpreted as the free space Green's functions whose spatial distribution and strengths are adjusted to satisfy boundary conditions. The influence of these singularities is localized in general and will vanish at infinity. Hence it is adequate to distribute the singularities over or within the obstacle surface and ensure simultaneously the application of the boundary conditions. In practice the surface of the object is approximated by a collection of straight panels, not necessarily of equal lengths. The impermeability boundary condition is exactly satisfied at a finite number of points on each panel. The accuracy of the method increases with the number of panels per body. In this respect the panel method is one of the several numerical methods available in the literature.

We briefly review the Green's function technique below. Consider the solution of the two dimensional problem

$$\begin{aligned} \nabla^2\phi &= F \quad \text{in a domain } D \\ \text{and} \quad \phi &= f \quad \text{on } C, \text{ the boundary of } D. \end{aligned}$$

The Green's function G for this problem is the solution of the equation $\nabla^2 G = \delta(x - \xi, y - \eta)$ subject to the boundary condition $G = 0$ on C. Here δ is the Dirac-delta function centered at $x = \xi$ and $y = \eta$. In a general setting, the differential operator acting on G is the adjoint of the operator acting on ϕ. Since the Laplace operator is self-adjoint, the equation governing G is also the Laplace equation. The boundary conditions on G are chosen in such a way that boundary terms arising from integration by parts during the construction of the adjoint operator reduce to zero. Multiplying the governing equation by G

and integrating over D we get

$$\int_D G\nabla^2\phi d\mathbf{x} = \int_D GF d\mathbf{x}$$

The left side can be integrated by parts to yield

$$\int_D G\nabla^2\phi d\mathbf{x} = \int_D \phi\nabla^2 G d\mathbf{x} + \int_C (G\phi_n - fG_n) ds$$

Since G satisfies the governing equation and boundary conditions stated above the solution for ϕ is obtained as

$$\phi(x,y) = \int_D GF d\mathbf{x} + \int_C fG_n ds$$

Here n is a unit outward drawn normal on C and $G_n = \partial G/\partial n$. The integral is evaluated by treating ξ and η as independent variables and $d\mathbf{x} = d\xi\, d\eta$. The explicit form of the Green's function is usually determined as the sum

$$G(\xi,\eta;x,y) = U(\xi,\eta;x,y) + g(\xi,\eta;x,y)$$

where U satisfies

$$\nabla^2 U = \delta(x-\eta, y-\eta)$$

with no boundary conditions and g satisfies

$$\nabla^2 g = 0$$

with boundary conditions that make $G = U + g = 0$ on C. The function U is called a freespace Green's function and can be easily determined for a variety of two and three dimensional operators. In contrast, solving for g is quite tedious and techniques that can be routinely used are not available. The panel method can be thought of as a procedure for solving for g and subsequently for ϕ using a linear superposition of the free-space Green's function U evaluated at distinct points (x,y) in the physical domain.

The form of U for the Laplace operator is determined as follows. Since U is not constrained by boundary conditions we may treat it as radially symmetric and solve

$$U_{rr} + \frac{1}{r}U_r = \delta$$

where r is a radial coordinate centred at a point (x,y) at which the solution ϕ is required. Hence

$$r = \left((x-\xi)^2 + (y-\eta)^2\right)^{1/2}$$

For $r > 0$, $\delta = 0$ and $U = A\ln r + B$. Here A and B are constants of integration. To determine A we integrate the governing equation on a closed disk, of radius R around $r = 0$. This yields

$$\int_{r\le R} \nabla^2 U ds = \int_{r\le R} \delta ds = 1$$

The left side is

$$\int_{r=R} \mathbf{n} \cdot \nabla U ds = \int_0^{2\pi} \left. \frac{\partial U}{\partial r} \right|_R R\, d\theta = 2\pi A$$

Hence $A = 1/2\pi$. The second constant B can be set equal to zero without loss of generality. Hence $U = 1/2\pi \ln r$. This can be identified with the velocity potential ϕ for a line source of unit strength located at the origin. For a source of strength m the velocity potential is $(m/2\pi) \ln r$. The physical interpretation of m is that it is the volume rate of flow generated by the source. Other singularities encountered in potential theory such as a point vortex and a doublet are also valid Green's functions and can be used individually or in a combination to build a panel method.

It is worth noting that the numerical algorithm for the panel method is built using solutions of the governing equations suitably combined to satisfy the boundary conditions. Hence the discretization pertains to that of the boundary of the flow region and not the flow field itself. In this respect the panel method falls in the class of boundary-element methods used for solving second order partial differential equations (Banerjee and Morino, 1990).

9.2 Source Panel Method

In two dimensions a panel is segment of a line or a curve with a distribution of singularities over its length. In three dimensions panels are rectangular or curvilinear patches having a distribution of singularities over their areas. The distribution could be uniform or non-uniform; in practice the form of the distribution function as a constant, linear or quadratic is assumed before hand. This section describes the panel method for two dimensional geometries with singularities chosen as sources (when the source strength m is positive) and sinks (when m is negative). A finite number of panels are laid out on the surface of the object and this constitutes the discretization of the boundary. The far field in most applications is undisturbed and need not be discretized. When several objects are placed in the flow their surfaces are individually discretized. It is not necessary to establish connectivity among these objects. This information is contained in the coordinates of the end points of the panels.

The complex potential for a line source of strength m per unit length is $F(z) = m/2\pi \ln z$, where the complex variable $z = x + iy = re^{i\theta}$. Since $F = \phi + i\psi$, the velocity potential is recovered as $\phi = m/2\pi \ln r$, where r is the radial coordinate measured with respect to the source location. Consider a panel AB of length $2L$ as shown in Figure 9.1 . The coordinate measured parallel to the panel is s. The panel has a continuous distribution of sources on it. Hence the source strength m computed per unit length parallel to the panel is a function of s. The variable r is now interpreted as the distance of any point on the panel AB to a general point P in the flow field. The velocity potential at P due to

the combined influences of all the sources on AB is

$$\phi(P) = \int_{-L}^{L} \frac{m(s)}{2\pi} \ln r ds \tag{9.1}$$

If (x, y) are coordinates of points on the panel, then in Equation 9.1

$$r = \left((x-\xi)^2 + (y-\eta)^2\right)^{1/2} \quad \text{and} \quad ds = (dx^2 + dy^2)^{1/2}$$

The net potential at P due to the influence of N panels is

$$\phi_p = \sum_{j=1}^{N} \int_{-L_j}^{L_j} \frac{m_j}{2\pi} \ln r_{pj} ds_j \tag{9.2}$$

Here j is the index number for the panels, $2L_j$ the panel length and r_{pj} the distance of any point on the jth panel to point P. m_j is the source strength per unit length for the jth panel and can vary with the jth wall coordinate s_j. In general it will vary from panel to panel. The potential at a point due to N panels and a superimposed flow of magnitude U and an angle of attack α is

$$\phi_P = U(x\cos\alpha + y\sin\alpha) + \sum_{j=1}^{N} \int_{-L_j}^{L_j} \frac{m_j}{2\pi} \ln r_{pj} ds_j \tag{9.3}$$

The form of ϕ given in Equation 13.2 automatically satisfies the governing equation $\nabla^2\phi = 0$ except at the source location. In the panel method the sources are located on the object surface and hence the governing equation is satisfied throughout the flow field. The source strengths m_j are the primary unknowns of the problem and are determined from the boundary condition $\partial\phi/\partial n = 0$. The number of points, called *control points* at which this condition is applied is equal to the number of unknowns. This can be estimated as follows:

Let N be the number of panels. If m_j is assumed to be constant on each panel the number of unknowns $(= M)$ is clearly N. This is called the zeroth order panel method. If m_j is assumed to vary linearly on each panel, it is characterized by two parameters and the number of unknowns increases to $2N$. However if the function $m(s)$ is required to be continuous going from one panel

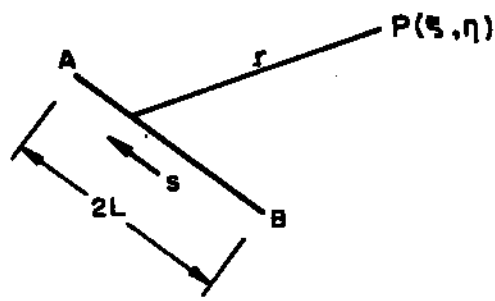


Figure 9.1: Nomenclature for a Panel

to the next, the corner values of m between neighbouring panels must be equal. The number of unknowns reduces in this case to $N+1$. This is called the first order panel method. For quadratic variation of m on each panel with inter-panel continuity the number of unknowns is $2N+1$. This is second order panel method. The potential at point P owing to N panels, each panel being of kth order as well as superimposed flow is written as

$$\phi_P = U(x\cos\alpha + y\sin\alpha) + \bigcup_{i=1\ldots N} \sum_{j=1}^{k} \int_{-L_i}^{L_i} \frac{m_{ij}N_j}{2\pi} \ln r_{pi} ds_i \tag{9.4}$$

Here $\bigcup$ symbolizes union over all N panels, m_{ij} is the jth source parameter on the ith panel, N_j a kth order polynomial and s_i the wall coordinate along the ith panel. When the source strength is taken constant for each panel we have $k=1$. Further $N_j = 1$ and $m_{ij} = m_i$; Equation 9.4 then reduces in form to Equation 9.3. For $k=2$, the unknowns are m_{i1} and m_{i2} for the ith panel and, $N_1 = 1-((s-s_1)/2L_i)$, $N_2 = (s-s_1)/2L_i$. For $k=3$, the unknowns are m_{i1}, m_{i2} and m_{i3}; the interpolation polynomials are

$$N_1 = -\frac{1}{2}((s-s_1)/L_i)(1-(s-s_1)/L_i)$$

$$N_2 = 1-((s-s_1)/L_i)^2$$

and

$$N_3 = \frac{1}{2}((s-s_1)/L_i)(1+(s-s_1)/L_i)$$

Further, $\bigcup$ represents addition of contributions from neighbouring panels to the corner shared by them. Curvilinear panels are handled by interpolating the local independent variable s using the interpolation polynomials N_j in a manner analogous to the treatment of curvilinear elements in Chapter 8.

A distribution of sources on an impermeable object surface will satisfy the equation $\nabla^2\phi = 0$ throughout the flow field and hence will continue to give lift and drag as zero. This can be established as:

Consider a collection of N sources of strength m_j and location z_j, $j = 1\ldots N$ placed in uniform flow. Here $z = x+iy$ is the complex variable. We need to show that the complex force

$$X - iY = \frac{i\rho}{2}\oint_C \left(\frac{dF}{dz}\right)^2 dz = 0 \quad \text{when} \quad \sum_j m_j = 0$$

for a curve C that encloses all the sources. The complex potential F for this problem is

$$F = Uz + \sum_j \frac{m_j}{2\pi}\ln(z-z_j)$$

the complex velocity

$$\frac{dF}{dz} = U + \sum_j \frac{m_j}{2\pi} \frac{1}{z - z_j} \quad \text{and} \quad \left(\frac{dF}{dz}\right)^2 = U^2 + \frac{U}{\pi} \sum_j m_j \frac{1}{z - z_j}$$
$$+ \left(\frac{1}{2\pi^2}\right) \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \frac{m_i m_j}{(z - z_i)(z - z_j)} + \sum_{j=1}^{N} \left[\frac{m_j}{2\pi(z - z_j)}\right]^2$$

The individual integrals are

$$\oint_C U^2 dz = 0, \oint_C \frac{U}{\pi} \sum m_j \frac{1}{z - z_j} dz = 2iU \sum_j m_j = 0, \oint_C \frac{dz}{(z - z_i)^2} = 0$$

and

$$\sum_j \sum_i \oint_C \frac{m_i m_j}{(z - z_i)(z - z_j)} dz = \sum_j \sum_i m_i m_j \oint_C \left(\frac{1}{z - z_i} - \frac{1}{z - z_j}\right) dz = 0$$

Hence $X - iY = 0$. A distribution of sources can be used to generate a non-zero lift or drag only when combined with some experimental data.

To keep the discussion at an introductory level we consider zeroth order panels characterized by a single unknown m_j for the jth panel. These unknowns are determined using the boundary condition

$$\frac{\partial \phi}{\partial n} = 0$$

applied at $M(= N)$ points. For the zeroth order panel method the control points are located at the centre of each panel. Let n_i be the unit normal drawn on the object surface at the ith control point. The equations that solve for m_j are of the form

$$\frac{\partial \phi}{\partial n_i} = 0 = \mathbf{U} \cdot \mathbf{n}_i + \sum_{j=1}^{N} \frac{m_j}{2\pi} \int_{-L_j}^{L_j} \frac{\partial}{\partial n_i} \ln r_{ij} \, ds_j, \quad i = 1 \ldots, M(= N) \tag{9.5}$$

Here

$$\mathbf{U} = U(\cos\alpha, \sin\alpha)$$

and $\mathbf{U} \cdot \mathbf{n}_i = U \cos(\beta_i - \alpha)$ (Figure 9.2).

The system of equations resulting from Equation 9.5 is

$$\frac{1}{2\pi}[C]\{m_j\} = -\{\mathbf{U} \cdot \mathbf{n}_i\} \tag{9.6}$$

where C_{ij}, an element of C is

$$C_{ij} = \int_{-L_j}^{L_j} \frac{\partial}{\partial n_i} \ln r_{ij} \, ds_j = \int_{-L_j}^{L_j} \frac{1}{r_{ij}} \frac{\partial r_{ij}}{\partial n_i} ds_j$$

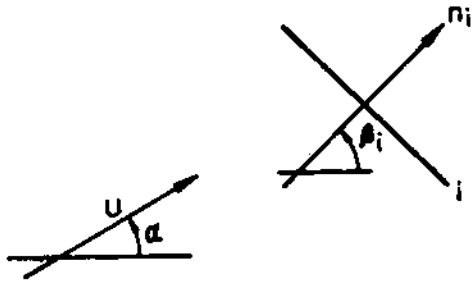


Figure 9.2: Orientation of Flow with Respect to a Panel

Determination of C_{ij} and the inverse of C form the core of a panel computation. Note once again that n_i is the unit outward drawn normal at the ith control point, r_{ij} is the distance of any point on the jth panel to the ith control point, s_j is a local coordinate on the jth panel and integration is performed over the length of the jth panel. The integration is commonly performed numerically using equally spaced points over the panel length followed by the application of Simpson's rule. In most applications, 7-9 points are sufficient for numerical integration.

The derivative $\partial r_{ij}/\partial n_i$ can be evaluated for a pair of panels i and j as follows (Figure 9.3). The coordinates of the control point Q are (x_i, y_i) and $P(x_j, y_j)$ is a generic point on the jth panel. Let the coordinates of n_i be n_x and n_y; these are known once the coordinates of the end points of the ith panel are known. Writing

$$r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2\right)^{1/2}$$

and

$$\partial r/\partial n = \nabla r \cdot \mathbf{n}$$

it can be easily established that

$$\frac{\partial r_{ij}}{\partial n_i} = \frac{((x_i - x_j)n_x + (y_i - y_j)n_y)}{r_{ij}}$$

The integral in the definition of C_{ij} can be now be evaluated using distinct values of x_j and y_j along the jth panel.

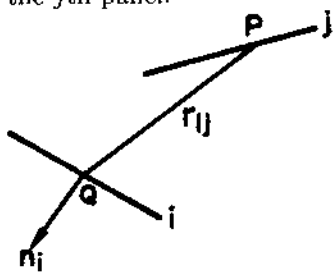


Figure 9.3: Computation of the Normal Derivative

The entries C_{ij} in matrix C are called influence coefficients and have the following physical meaning: $m_j C_{ij}/2\pi$ is the velocity induced at a point i by a uniform source distribution of strength m_j. There is some difficulty in calculating C_{ii}, i.e. the influence of a panel at its own control point. This is because r_{ii} is zero at one point (the control point itself) and the integrand becomes singular at this location. The integral however exists and can be obtained analytically as described below.

Consider a panel of length $2L$ and a source strength m (Figure 9.4). The velocity components induced by the panel at point P are:

$$u = \frac{m}{2\pi}\int_{-L}^{L}\frac{\partial}{\partial x}\ln r dt \tag{9.7}$$

$$v = \frac{m}{2\pi}\int_{-L}^{L}\frac{\partial}{\partial y}\ln r dt \tag{9.8}$$

where $r = [(x-t)^2 + y^2]^{1/2}$. We are interested in the value of the normal velocity v as (x, y) approaches the origin $(0, 0)$, the control point of the panel. Equation 9.8 can be evaluated as:

$$\begin{aligned} v &= \frac{m}{2\pi}\int_{-L}^{L}\frac{y dt}{(x-t)^2+y^2} = \frac{m}{2\pi}\left[\tan^{-1}\frac{t-x}{y}\right]_{-L}^{L} \\ &= \frac{m}{2\pi}\left[\tan^{-1}\frac{L-x}{y} - \tan^{-1}\frac{-L-x}{y}\right], \\ &= \frac{m}{2\pi}(\nu_1+\nu_2) \end{aligned}$$

where ν_1 and ν_2 are angles as shown in Figure 9.5.

As P approaches 0, $\nu_2 \to \pi/2$ and $\nu_1 \to \pi/2$. Thus

$$\begin{aligned} v &= (m/2\pi)\, C_{ii} = (m/2\pi)\,\pi \\ \text{and} \quad C_{ii} &= \pi \end{aligned} \tag{9.9}$$

A solid object cannot be a net mass source. Hence

$$\sum_{j=1}^{N} m_j L_j = 0$$

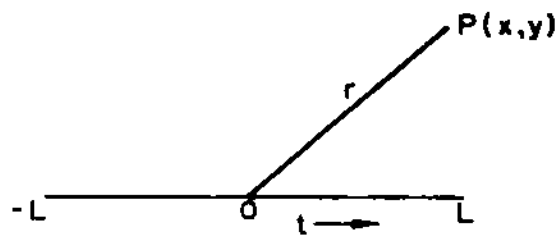


Figure 9.4: Computation of Influence Coefficient C_{ii}

and this can be used as a check on discretization, the numerical calculations involved in constructing matrix C and its inversion. The source panel method can be illustrated using the following examples:

Example 9.1: Flow Past a Cylinder

Potential flow past a circular cylinder at zero angle of attack is considered here. Eight panels, specifically their mid-points define the cylinder surface (Figure 9.6).

Clearly as the number of panels increases the representation of the geometry improves. The velocity potential at any of the 8 control points is the sum of the contributions of the 8 panels and uniform flow. At a control point the total potential is given as

$$\phi_i = Ux_i + \sum_{j=1}^{8} \frac{m_j}{2\pi} \int_j \ln r_{ij}\, ds_j, \quad i = 1 \ldots 8$$

This choice of ϕ automatically satisfies the equation $\nabla^2\phi = 0$ in the flow field. Applying the boundary condition $\partial\phi_i/\partial n_i = 0$ at each control point, the matrix equation is derived as

$$[C]\{m_j\} = -U\ \{\mathbf{n}_i \cdot \mathbf{i}\}$$

where $\mathbf{i}$ is a unit vector in the x-direction. The elements of C are

$$C_{ij} = \int_j \frac{1}{r_{ij}} \frac{\partial r_{ij}}{\partial n_i}\, ds_j$$

For complex shapes C_{ij} is evaluated by numerical integration. It can be analytically determined only on a few occasions. As an example consider the evaluation of the coefficient C_{53} (Figure 9.7).

Noting that

$$\frac{\partial r_{53}}{\partial n_5} = \nabla_5 r_{53} \cdot n_5 = \left(\mathbf{i}\frac{\partial}{\partial x_5} + \mathbf{j}\frac{\partial}{\partial y_5}\right) \left((x_5 - x_3)^2 + (y_5 - y_3)^2\right)^{1/2} \cdot (-\mathbf{j}),$$

$$ds_3 = dy_3, \quad (x_5, y_5) = (0, -1) \quad \text{and} \quad x_3 = -1$$

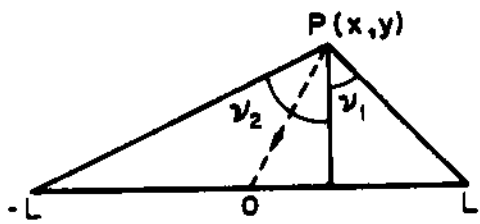


Figure 9.5: Limiting Process $P \to O$

we get

$$C_{53} = \int_a^b (1 + y_3)/(1 + (1 + y_3)^2)\, dy_3$$

Here a and b are the y-coordinates of the end points of panel 3. This integral is

$$C_{53} = \ln \left[(1 + (1 + b)^2)/(1 + (1 + a)^2)\right]^{1/2}$$

In any panel calculation the data input consists of the angle of attack, number of panels and the coordinates of the end points of the panels. The coordinate data for the cylinder is:

Panel Number	Coordinates			
j	x_1	y_1	x_2	y_2
1	0.4142	1.0	−0.4142	1.0
2	−0.4142	1.0	−1.0	0.4142
3	−1.0	0.4142	−1.0	−0.4142
4	−1.0	−0.4142	-0.4142	−1.0
5	−0.4142	−1.0	0.4142	−1.0
6	0.4142	−1.0	1.0	−0.4142
7	1.0	−0.4142	1.0	0.4142
8	1.0	0.4142	0.4142	1.0

The complete matrix equation for this problem is:

$$\frac{1}{2\pi}\begin{bmatrix} \pi & 0.351 & 0.404 & 0.404 & 0.404 & 0.404 & 0.404 & 0.351 \\ & \pi & 0.351 & 0.404 & 0.404 & 0.404 & 0.404 & 0.404 \\ & & \pi & 0.351 & 0.404 & 0.404 & 0.404 & 0.404 \\ & & & \pi & 0.351 & 0.404 & 0.404 & 0.404 \\ & \text{Symmetric} & & & \pi & 0.351 & 0.404 & 0.404 \\ & & & & & \pi & 0.351 & 0.404 \\ & & & & & & \pi & 0.351 \\ & & & & & & & \pi \end{bmatrix} \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \end{Bmatrix}$$

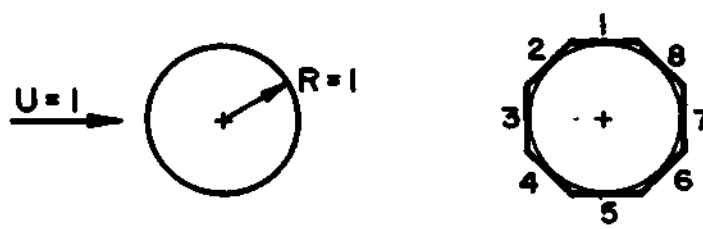


Figure 9.6: Example of Flow past a Cylinder

$$= \begin{Bmatrix} 0 \\ 0.707 \\ 1.0 \\ 0.707 \\ 0.0 \\ -0.707 \\ -1.0 \\ -0.707 \end{Bmatrix}$$

Solving this equation, the unknown source strengths are determined as

$$\frac{m_j}{2\pi} = (0, 0.266, 0.376, 0.266, 0, -0.266, -0.376, -0.266)$$

Clearly $\sum m_j L_j = 0$.

As a rule matrices encountered in panel methods are considerably smaller than those seen in methods that discretize the governing equation over the flow field. This is related to the fact that panel methods require discretization of only the object surface. However these smaller matrices are dense; in particular they are neither sparse nor banded. If the object shape is non-symmetric, the C-matrix will fail to show symmetry. The condition of diagonal dominance will also not be satisfied and iterative methods for solving the system of equations will not be applicable. When other forms of singularities are used it is quite possible that the diagonal entries of C become zero. Hence matrix inverters that are used with panel methods must be quite general and include preconditioning to account for the possibility of zeros along the diagonal. A preconditioned matrix can be inverted using the Gauss-Jordon method. The problem of matrix inversion is quite severe in three dimensional geometries since numerical errors owing to round-off on a computer can be amplified.

Since the impermeability condition is applied only at a finite number of points it is to be expected that the normal component of velocity will be non-zero at all other locations. However the fact that $\sum m_j L_j = 0$ shows that the normal velocity changes sign and the average value over the cylinder is zero. A measure of the magnitude of the normal velocity (also called *leakage*) is the rms value defined as

$$u_L = \left[\frac{1}{2\pi} \int_0^{2\pi} (\mathbf{u} \cdot \mathbf{n})^2 d\theta \right]^{1/2}$$

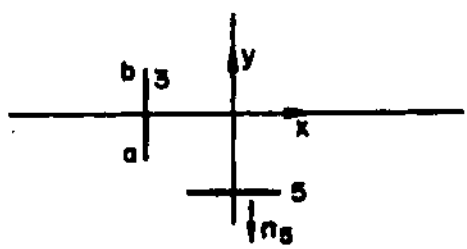


Figure 9.7: Calculation of Influence Coefficient C_{53}

and evaluated on a contour enclosing the body.

It can be shown that u_L decreases monotonically with an increasing number of panels. For $N = 8$, $u_L = 0.576$; $N = 16$, $u_L = 0.0677$ and $N = 32$, $u_L = 0.0369$ for the example of flow past a cylinder. Here the contour diameter is taken to be 5 percent larger than the cylinder diameter itself. It can be seen that 8 panels are quite inadequate to prevent leakage.

The velocity at any point on or away from the solid surface is calculated as

$$\mathbf{u} = \nabla\phi = U(\cos\alpha, \sin\alpha) + \sum_{j=1}^{N} \frac{m_j}{2\pi} \int_{-L_j}^{L_j} \nabla \ln r_{ij}\, ds_j$$

The integral appearing here can be evaluated by Simpson's rule. At the control points on the surface of the object the non-zero component of velocity is tangential to the surface and is computed as

$$U_t = \mathbf{u} \cdot \mathbf{t} = \frac{\partial\phi}{\partial s} = U\sin(\beta - \alpha) + \sum_{j=1}^{N} \frac{m_j}{2\pi} \int_{-L_j}^{L_j} (\nabla \ln r_{ij} \cdot \mathbf{t}) ds_j$$

Here $\nabla \ln r_{ij}$ is computed as described earlier and the components of the tangent vector $\mathbf{t}$ can be determined from the panel orientation. In particular $\mathbf{t} = (\sin\beta, -\cos\beta)$ where β is defined in Figure 9.2. The values of U_t at the control points of panels 1-8 for the present example are (2.0, 1.414, 0, 1.414, 2.0, 1.414, 0, 1.414). These closely match the exact solution $U_t = 2\sin\theta$.

The pressure distribution on the surface of the object is obtained using Bernoulli's equation $p/\rho + U_t^2/2 =$ constant. Hence the dimensionless pressure coefficient $C_p = (p - p_\infty)/\frac{1}{2}\rho U^2$ is $1 - (U_t/U)^2$. Figure 9.8 shows a plot of C_p as a function of θ for flow past a cylinder. This matches with the analytical solution ($= 1 - 4\sin^2\theta$) quite well.

The lift and drag on the cylinder computed using this pressure distribution are $L = \int_0^{2\pi} p(R,\theta)\, R\sin\theta d\theta$ and $D = \int_0^{2\pi} p(R,\theta)\, R\cos\theta d\theta$. Both L and D are zero for the pressure distribution shown in Figure 9.8. A method of estimating finite values of lift and form drag is described later in this chapter.

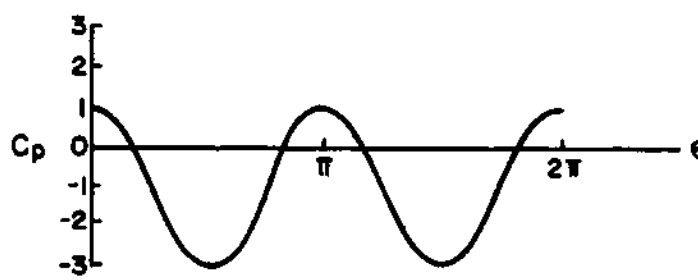


Figure 9.8: Pressure Coefficient for Flow past a Cylinder

Example 9.2: Flow Past an Oscillating Cylinder

A circular cylinder of diameter d placed in steady uniform approach flow ($= U$) oscillates parallel to the flow direction with a frequency ω. Assuming that ω is large in comparison to the frequency scale U/d (i.e. $\omega d/U > 1$) meaningful values of drag can be computed using potential theory. The origin of drag in this problem lies in the acceleration of the bulk of the fluid surrounding the cylinder. The flow continues to be governed by the equation $\nabla^2\phi = 0$ applied to the instantaneous cylinder position and the permeability boundary condition $\partial\phi/\partial n = \mathbf{U_c} \cdot \mathbf{n}$, where $\mathbf{U_c}$ is the cylinder velocity. If $\mathbf{U_c}$ changes with time, the panel strengths will also change with time. The instantaneous pressure distribution on the cylinder surface is determined from Bernoulli's equation

$$\frac{\partial\phi}{\partial t} + \frac{p}{\rho} + \frac{u^2}{2} = \text{constant}$$

u being the total velocity of the fluid on the cylinder surface. The Bernoulli constant is independent of time because the approach velocity is steady. It is equal to $p_\infty/\rho + U^2/2$. The calculations are simplified by using a coordinate system that moves with the cylinder. The governing equation does not contain time and hence remains unchanged. The panel strengths m_j are obtained at any instant by solving the system of equations

$$\sum_{j=1}^{N} \frac{m_j}{2\pi} \int_{-L_j}^{L_j} \frac{\partial}{\partial n_i} \ln r_{ij}\, ds_j = -U\cos(\alpha - \beta_i) + \mathbf{U_c} \cdot \mathbf{n_i}$$

The coefficient matrix for m_j is independent of time and can be inverted permanently. The total velocity u at any point P on the cylinder is

$$\mathbf{u} = \mathbf{U} + \sum_{j=1}^{N} \frac{m_j}{2\pi} \int_{-L_j}^{L_j} \nabla \ln r_{pj} ds_j$$

and $u = |\mathbf{u}|$, the velocity amplitude in Bernoulli's equation. The time derivative $\partial\phi/\partial t$ must be evaluated as

$$\frac{\partial\phi}{\partial t} = \frac{\partial\phi}{\partial U_c}\frac{\partial U_c}{\partial t} - U_c\frac{\partial\phi}{\partial x}$$

to account for the fact that the coordinate system moves with a time-dependent velocity U_c in the x-direction. In the present example we specify $U_c = A \sin(2\pi\omega t)$, where ω is the oscillation frequency in Hz. The derivative $\partial\phi/\partial U_c$ is evaluated with respect to the current value of U_c during a cycle of cylinder motion. In any computer calculation this is implemented by altering the instantaneous cylinder velocity by ε and computing the derivative as

$$\frac{\partial\phi_i}{\partial U_c} = \sum_{j=1}^{n} \frac{1}{2\pi} \frac{m_j(\text{new}) - m_j(\text{old})}{\varepsilon} \int_{-L_j}^{L_j} \ln r_{ij} ds_j$$

Table 9.1: Drag Coefficient for an Oscillating Cylinder placed in Uniform Flow; Example 9.2

t	0.00	0.05	0.10	0.125	0.15	0.20	0.25
C_D	7.89	6.38	2.44	0.00	−2.44	−6.38	−7.89

t	0.30	0.35	0.375	0.40	0.45	0.50
C_D	−6.38	−2.44	0.00	2.44	6.38	7.89

Once the pressure distribution is known the drag as a function of time can be calculated as described in Example 9.1.

The analytical solution of this problem for a cylinder of radius R can be shown to be:

$$\phi = Ur\cos\theta + \frac{R^2}{r}(U - U_c)\cos\theta,$$

where (r, θ) is measured from the instantaneous cylinder centre. The total velocity u on the cylinder surface ($u = \mid \nabla\phi \mid$) is

$$u = (\phi_r^2 + \frac{1}{r^2}\phi_\theta^2)^{1/2} = (U_c^2\cos^2\theta + (-2U + U_c)^2\sin^2\theta)^{1/2}.$$

The time derivative of velocity potential on the cylinder surface is

$$\begin{aligned}\left.\frac{\partial\phi}{\partial t}\right|_{r=R} &= \frac{\partial U_c}{\partial t}\frac{\partial\phi}{\partial U_c} - U_c\frac{\partial\phi}{\partial x} \\ &= -2\pi\omega AR\cos\theta\cos(2\pi\omega t) - U_c\left(U - (U - U_c)(1 - 2\cos^2\theta)\right)\end{aligned}$$

The pressure coefficient C_p is

$$\begin{aligned} C_p &= \frac{p - p_\infty}{\frac{1}{2}\rho U^2} = (1 - u^2) - \frac{1}{\frac{1}{2}U^2}\frac{\partial\phi}{\partial t} \\ \text{and} \quad C_D &= \int_0^\pi \frac{(p - p_\infty)}{\frac{1}{2}\rho U^2}\cos\theta\, d\theta \end{aligned}$$

For dimensionless values of $U = 1$, $R = 1$, an amplitude of velocity fluctuation $A = 0.2$ and $\omega = 2$, the drag coefficient as a function of time is given in Table 9.1 for a duration of one time period. The time period of drag (=0.5) is seen to be equal of that of the cylinder oscillation ($= 1/\omega$). When the superimposed flow is zero the two time periods are again equal. It can be shown that the drag amplitude increases linearly with the frequency ω.

Example 9.3: Flow past an Array of Two Cylinders.

Consider flow past an array of two cylinders of equal diameter, in configurations A and B shown in Figure 9.9. The cylinders in A will tend to attract each other

Table 9.2: Drag and Lift as a Function of L/d.

L/d	1.25	1.5	2.0	3.0	4.0	6.0	∞
$A(C_L)$	3.25	1.39	0.483	0.129	0.0527	0.015	0
$B(C_D)$	1.01	0.704	0.347	0.113	0.049	0.015	0

due to a drop in pressure at the throat while in B will repel each other due to the formation of a high pressure stagnant region between them. The magnitude of the forces increase as the separation between the cylinders (and hence d/L) decreases. The discretization of the cylinders for a panel calculation is straight forward and consists of setting up panels individually on the cylinders without any need for a connection between them (Figure 9.10).

The lift coefficient for A and the drag coefficient for B, for any one cylinder is summarized as a function L/d in Table 9.2.

9.3 Calculation of Drag

In high Reynolds number flows the fluid separates from the surface of the object in regions of adverse pressure gradient. The resulting flow field is time-dependent. Once the flow has separated from the solid boundary, the time-averaged pressure variation over the rest of the object surface is nearly uniform. Thus, the pressure variation over the object is determined by potential theory upto the point of separation and is subsequently a constant. This fact can be used for practical form drag calculations.

To determine the point of separation one must acknowledge the formation of boundary-layers on the face of the object exposed to the approach flow. Viscous effects are confined to the interior of the boundary-layer and the flow exterior to it is irrotational (Figure 9.11). In particular the fluid adjacent to the wall

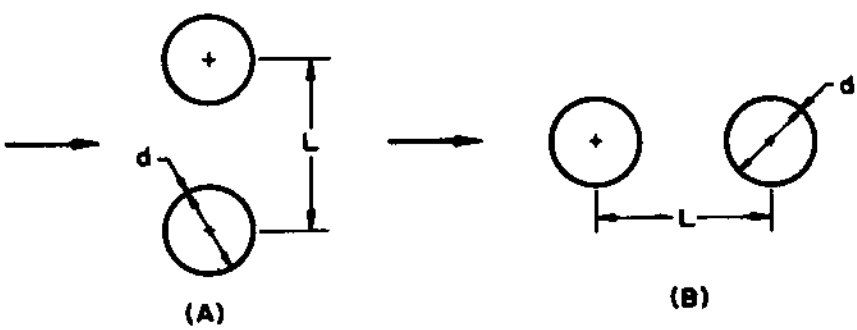


Figure 9.9: Array of Two Cylinders

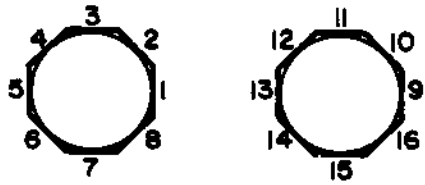


Figure 9.10: Discretization for an Array of Cylinders

sticks to it; for a stationary surface the fluid velocity components at the wall are zero. For large Reynolds number $\delta/R << 1$, i.e., the boundary-layer is thin. The Navier-Stokes equations can be simplified under these conditions to the form

$$uu_x + vu_y = UU_x + \nu u_{yy} \tag{9.10}$$

along with the incompressibility constraint

$$u_x + v_y = 0 \tag{9.11}$$

Here U is the potential velocity evaluated on the object surface and can be determined in general using the source panel method. The local coordinate directions shown in Figure 9.11 continue to be indicated by x and y.

For the computation of form drag one requires the surface pressure distribution. As stated earlier, the potential pressure distribution can be considered valid upto the point of separation and subsequently a constant. Hence determination of this point is crucial for drag calculations. The criterion for separation on the surface of a stationary object is

$$\tau_{\text{wall}} = \mu \left.\frac{\partial u}{\partial n}\right|_{\text{wall}} = 0$$

i.e. a zero wall shear stress. The Karman-Pohlhausen method for locating the point of separation is now described (Schlichting and Gersten, 2000).

The starting point for this method is the Momentum-Integral (MI) equation obtained by integrating Equations 9.10 and 9.11 over the boundary-layer thickness. The MI equation can be shown to be

$$\frac{\tau_w}{\rho} = \frac{d}{dx}\left(U^2\delta_2\right) + U\delta_1\frac{dU}{dx} \tag{9.12}$$

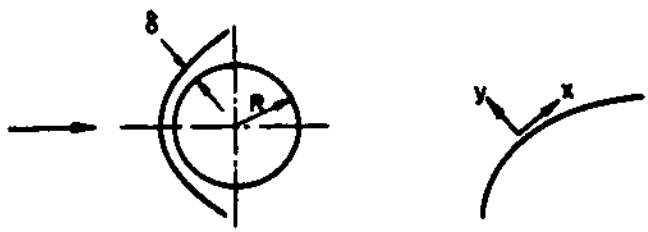


Figure 9.11: Schematic of Boundary Layer

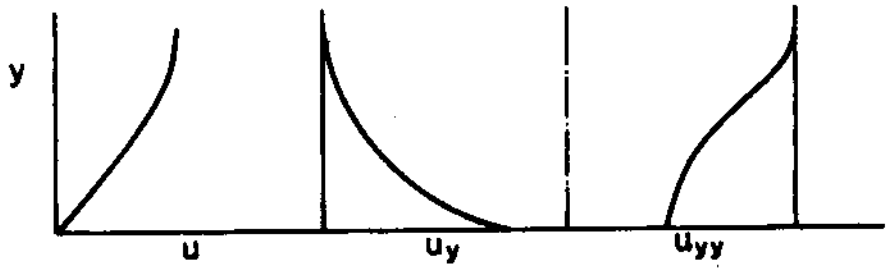


Figure 9.12: Sketch of Assumed Velocity Profile ($p_x < 0$).

Here τ_w is the wall shear stress, U the known potential velocity at the wall and δ_1 and δ_2 are the displacement and momentum thicknesses defined as

$$\delta_1 = \int_0^\delta (1 - \frac{u}{U})dy \quad \text{and} \quad \delta_2 = \int_0^\delta \frac{u}{U}(1 - \frac{u}{U})dy$$

The coordinates x and y are parallel and perpendicular to the object surface respectively. Equation 9.12 can be viewed as an ordinary differential equation for $\delta(x)$ that must be integrated in the x- direction till the wall shear stress reduces to zero.

The quantities δ_1, δ_2 and τ_w can be expressed in terms of δ by adopting the following procedure. With $\eta = y/\delta$ the velocity profile is first chosen as

$$\frac{u}{U} = f(\eta) = a\eta + b\eta^2 + c\eta^3 + d\eta^4, \quad 0 \le \eta \le 1 \tag{9.13}$$

This profile automatically satisfies the no-slip condition $\eta = 0, u = 0$. The profile parameters a, b, c and d are determined from the boundary conditions

$$\begin{aligned} \eta &= 1, \quad u = U, u_y = u_{yy} = 0 \\ \eta &= 0, \quad \nu u_{yy} = \frac{1}{\rho}\frac{dp}{dx} = -UU_x \end{aligned}$$

Again U is the potential velocity evaluated at the solid surface. The wall boundary condition ensures that in a favourable pressure gradient flow where $p_x < 0, u_{yy}$ is negative at the wall. Since u_y is a maximum at the wall and zero at the edge of the boundary-layer a negative u_{yy} uniformly reduces u_y without the formation of inflexion points. In an adverse pressure gradient, $p_x > 0$ and the wall boundary condition will give rise to velocity profiles with inflexion points that represent separation. This is illustrated in Figures 9.12 and 9.13. The profile parameters can now be shown to be, $a = 2 + s/6$, $b = -s/2$, $c = -2 + s/2$ and $d = 1 - s/6$, where s, called a shape factor is

$$s = \delta^2/\nu\, dU/dx$$

The velocity profile then becomes

$$\begin{aligned} \frac{u}{U} &= F(\eta) + \frac{s}{6}G(\eta) \quad \text{where} \\ F &= 2\eta - 2\eta^3 + \eta^4 \quad \text{and} \quad G = \eta - 3\eta^2 + 3\eta^3 - \eta^4. \end{aligned}$$

The funcions F and G are sketched in Figure 9.14. At $s = -12$, $u_y \mid_{y=0} = 0$ and the boundary-layer starts to separate. Hence the integral analysis is not valid for values of $s < -12$. For $s > +12$ the velocity profile takes on a value greater than unity for $\eta < 1$. This is unrealistic and so the shape factor is restricted to the range $-12 \leq s \leq 12$.

The fourth order velocity profile given above is substituted in the MI equation to obtain an ordinary differential equation in terms of δ. This equation can then be integrated, say numerically by a Runge-Kutta method. Integration proceeds in favourable pressure gradient flows till s just exceeds $+12$ and in adverse pressure gradient flows till incipient separation, i.e. $s \leq -12$. One can compute

$$\begin{aligned}
\delta_1 &= \int_0^\delta \left(1 - \frac{u}{U}\right) dy = \left(\frac{3}{10} - \frac{s}{120}\right) \delta, \\
\delta_2 &= \int_0^\delta \frac{u}{U}\left(1 - \frac{u}{U}\right) dy = \frac{1}{63}\left(\frac{37}{5} - \frac{s}{15} - \frac{s^2}{144}\right) \delta \\
\text{and} \quad \frac{\tau_w}{\rho} &= \nu \left.\frac{\partial u}{\partial y}\right|_{y=0} = \frac{\nu U}{\rho}\left(2 + \frac{s}{6}\right)
\end{aligned}$$

The equation for δ is set up once the potential flow $U(x)$ is prescribed. The initial condition for δ at $x = 0$ is $\delta(0) = 0$ for cusp shaped objects and $\delta(0)/R = \sqrt{3.025/\mathrm{Re}}$ where $\mathrm{Re} = U_\infty R/\nu$ for rounded objects whose upstream radius of curvature is R (Figure 9.15).

The integration of the MI equation can also be accomplished by manual calculation. This is most conveniently done by defining additional shape parameters (besides s) as $Z = \delta_2^2/\nu$ and $K = ZdU/dx$. The MI equation is written as

$$\frac{\tau_w}{\rho U^2} = \frac{d\delta_2}{dx} + (2\delta_2 + \delta_1)\frac{1}{U}\frac{dU}{dx}$$

In terms of K and Z thus becomes

$$U\frac{dZ}{dx} = H = 2\left\{\frac{\tau_w \delta_2}{\mu U} - (2 + \frac{\delta_1}{\delta_2})K\right\}$$

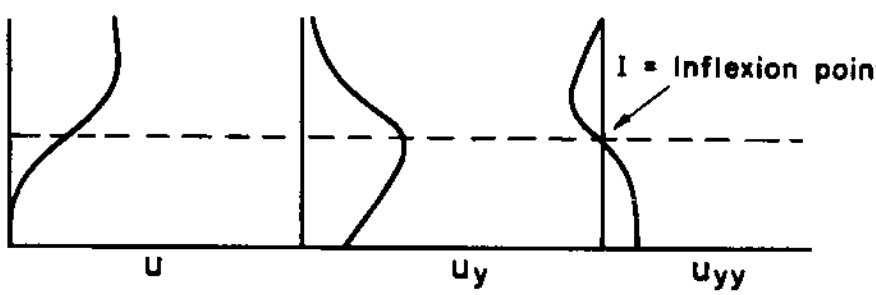


Figure 9.13: Sketch of Assumed Velocity Profile ($p_x > 0$).

where H is a unique function of s. The explicit form of H is obtained by noting that

$$\begin{aligned}
\frac{\tau_w \delta_2}{\mu U} &= \left\{(2+\frac{s}{6})\left(\frac{37}{315}-\frac{s}{945}-\frac{s^2}{9072}\right)\right\} \\
\frac{\delta_1}{\delta_2} &= \frac{63\left(\frac{3}{10}-\frac{s}{120}\right)}{37/5-s/15-s^2/144} \quad \text{and} \\
K=\frac{\delta_2^2}{\nu}\frac{dU}{dx} &= \left(\frac{\delta_2}{\delta}\right)^2 s = \left(\frac{1}{63}\left(\frac{37}{5}-\frac{s}{15}-\frac{s^2}{144}\right)\right)^2 s
\end{aligned}$$

Hence the MI equation reduces to the system of equations

$$U\frac{dZ}{dx} = H(s) \quad \text{and} \quad K(s) = Z\frac{dU}{dx} \tag{9.14}$$

which are solved subject to suitable initial conditions. For cusp-shaped objects the initial boundary-layer thickness is zero, hence $\delta(0) = \delta_2(0) = s(0) = K(0) = Z(0) = 0$. For rounded objects, $x = 0$ is a stagnation point at which $U = 0$. Hence H at $x = 0$ is zero. The value of s that makes $H = 0$ (and also satisfies the constraint $-12 \le s \le 12$) is $s_o = 7.052$. The corresponding value of K is $K_o = 0.077$. With these initial values the pair of equations given above can be integrated. For rounded objects the initial boundary-layer thickness is obtained from the definition of $s_o = (\delta_o^2/\nu)\,(dU/dx)$. Hence

$$\delta_o = \delta(0) = \sqrt{7.052/(1/\nu)\,(dU/dx)}$$

Near $x = 0$ most rounded objects can be approximated as a cylinder of a suitable radius R and $U = 2U_\infty \sin(x/R)$ near $x = 0$. Hence

$$dU/dx\,|_{x=0} = 2U_\infty/R$$

and $\delta(0) = \sqrt{3.025/\mathrm{Re}}$ as stated earlier. The integration steps in the x-direction are:

$$Z^{n+1} \quad : \quad \frac{Z^{n+1}-Z^n}{\Delta x} = \left(\frac{H}{U}\right)^n$$

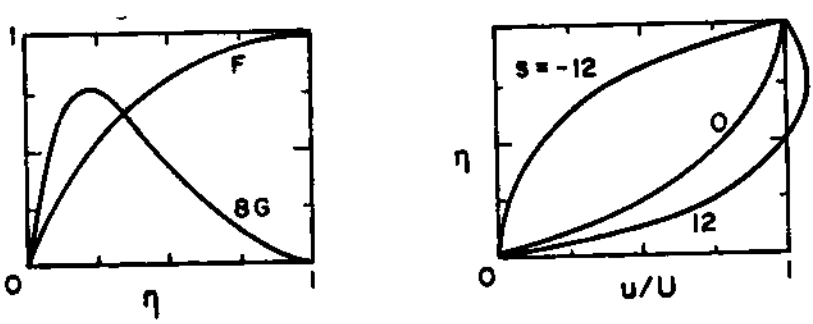


Figure 9.14: Plots of Functions F and G and Velocity Profile u/U

$$
\begin{aligned}
K^{n+1} &: \quad K^{n+1} = Z^{n+1}\left(\frac{dU}{dx}\right)^{n+1} \\
s^{n+1} &: \quad s^{n+1} \text{ is root of } K^{n+1} = K(s^{n+1}) \\
H^{n+1} &: \quad H^{n+1} = H(s^{n+1}).
\end{aligned}
$$

The wall shear stress $\tau_w/\mu = U/\delta\,(2 + s/6)$. The s and H calculations are independent of the object shape and can be carried out using tables or elementary computer programs. For $n = 1$ both H and U are zero and the ratio H/U must be determined as $\left[\frac{dH}{dx}/\frac{dU}{dx}\right]|_{x=0}$, using L'Hospital's rule. For flow past a cylinder this ratio is zero.

The sequence of calculations given above is now applied to the problem of flow past a cylinder of unit radius and unit approach velocity. The MI equations written in dimensionless form in terms of the angular coordinate are

$$U\frac{dZ}{d\phi} = H(s) \quad \text{and} \quad K(s) = Z\frac{dU}{d\phi}$$

where ϕ is measured from the forward stagnation point. The dimensionless wall shear stress can be shown to be $(\tau_w\sqrt{\mathrm{Re}})/(\rho U_\infty^2) = U\sqrt{(dU/d\phi)/s}\;(2 + s/6)$. For a step size of 1 degree the values of Z, K, s and τ_w, H are given in Table 9.3.

At $\phi = 90$ degrees both $dU/d\phi$ and s are zero; the ratio $(dU/d\phi)/s$ can be evaluated as $(d^2U/d\phi^2)/ds/d\phi$ which for a cylinder works out to be 2/17.52. Hence $\sqrt{(dU/d\phi)/s}$ at $\phi = 90$ degrees is 0.388.

Table 9.3 shows that $s < -12$ for $\phi > 108$ degrees and this can be taken as the point of separation. Hence the true pressure distribution on the cylinder can be taken as,

$$
\begin{aligned}
\frac{p(R,\phi) - p_\infty}{\frac{1}{2}\rho U_\infty^2} &= 1 - 4\sin^2\phi, \quad 0 < \phi < 108 \text{ (degrees)} \\
&= -2.618 \qquad 108 < \phi < 180 \text{ (degrees)}
\end{aligned}
$$

Using this distribution the drag coefficient for the cylinder can be calculated as 2.29. This value is consistent withwhat is experimentally realized in the

Figure 9.15: Sketch of Objects with (a) Cusp-shaped and (b) Rounded Leading Edges

Table 9.3: Karman-Pohlhausen Calculations for Flow Past a Cylinder

ϕ	U	$dU/d\phi$	Z	K	s	H	$\tau_w\sqrt{\mathrm{Re}}/\rho U_\infty^2$
0	0	2	0.0385	0.077	7.052	0	0
10	0.347	1.969	0.0385	0.0767	7.003	0.0019	0.583
20	0.684	1.879	0.0428	0.0757	6.87	0.0073	1.125
30	1.0	1.732	0.0427	0.0738	6.62	0.0176	1.587
40	1.285	1.532	0.0463	0.0710	6.26	0.0336	1.935
50	1.532	1.285	0.0518	0.0666	5.73	0.0587	2.144
60	1.732	1.0	0.0598	0.0598	4.99	0.098	2.195
70	1.879	0.684	0.0718	0.0491	3.94	0.161	2.079
80	1.969	0.347	0.0904	0.0314	2.405	0.269	1.796
90	2.0	0	0.1209	0	0	0.4698	1.350
100	1.969	−0.347	0.1765	−0.0612	−4.25	0.897	0.727
105	1.932	−0.518	0.223	−0.1156	−8.206	1.33	0.307
107	1.913	−0.585	0.248	−0.1454	−10.79	1.605	0.0898
108	1.902	−0.618	0.263	−0.1629	−12.75	1.795	< 0

Reynolds number range of 40-200, when the separation bubble over the cylinder is steady. Experiments show a point of separation of 81 degrees and a value of $C_D = 1$ to 1.4 in the laminar separated regime of flow ($10^3 < \mathrm{Re} < 5 \times 10^5$). The difference arises from the fact that the point of separation moves owing to disturbed flow conditions downstream. Further, the shedding of vortices makes the drag coefficient for a cylinder a function of time.

9.4 Calculation of Lift

As an example of calculation of lift we study flow past a thin symmetric Joukowski aerofoil at an angle of attack α. The aerofoil is generated using conformal transformation applied to an eccentrically located cylinder (Figure 9.16). This transformation is $z = \zeta + c^2/\zeta$, where $z = x + iy$ and $\zeta = \xi + i\eta$. The availability of this transformation can be exploited to generate the panel coordinates for the aerofoil in terms of the polar angle ν. This relationship can be shown to be (Currie, 1993)

$$x = 2c\cos\nu \quad \text{and} \quad y = 2c\varepsilon\sin\nu(1-\cos\nu) \quad \text{where} \quad c = L/4$$

$$\varepsilon = \frac{m}{c} = \frac{4}{3\sqrt{3}}\frac{t}{L} \quad \text{and the cylinder radius is} \quad a = m + c$$

These equations are valid only for small values of t/L, usually less than 10 percent. With t and L specified, m and c can be readily calculated. The (x, y) coordinates can be obtained by varying ν over 0 to 2π. In any calculation involving aerofoils the region containing the forward stagnation point (B in

Figure 9.16) must be resolved with a large number of panels. This can be accomplished by using smaller step sizes in ν near point B.

The application of the straightforward source panel method will give zero drag and lift for the aerofoil. For determination of lift we use the experimentally observed result known as *Kutta condition.* It is stated as: For bodies with sharp trailing edges which are at a small angle of attack to the free stream the flow will adjust itself in such a way that the rear stagnation point coincides with trailing edge. This means that the flow must leave the trainling edge smoothly. With this condition fluid particles do not have to turn through a large angle at the trailing edge of the object. Kutta condition fails when the angle of attack is so large that the aerofoil is no longer a streamlined object. Consequently flow separates on the surface of the aerofoil.

With reference to the aerofoil shown in Figure 9.16 we use the Kutta condition as follows. The application of the source panel method will produce a non-zero positive y-component of velocity at point A. To facilitate computation this velocity is calculated about 1% chord length away from point A, i.e. at P. Let this velocity be V. Since in reality, the y-component of velocity at P is zero (the Kutta condition) one can visualize the existence of a clockwise vortex of strength Γ at the origin that suppresses the velocity component V. The magnitude of the vortex strength that suppresses the velocity V is given by potential theory as $\Gamma = 2\pi V\overline{OP}$. Hence the lift on the aerofoil is $\rho U\Gamma$ and the lift coefficient $C_L = (4\pi\overline{OP}V)/L$. For a Joukowski aerofoil the analytical expression for lift is

$$C_L = 2\pi\left(1 + (4t/3\sqrt{3}L)\right)\sin\alpha$$

upto the angle of stall.

Values of C_L determined using the procedure given above are compared below with the analytical result for the lift coefficient. The panel solution has been obtained using 16 panels using a uniform step size of $\Delta\nu = 22.5$ degrees in the transformed plane. A substantially large number of panels is required for better accuracy.

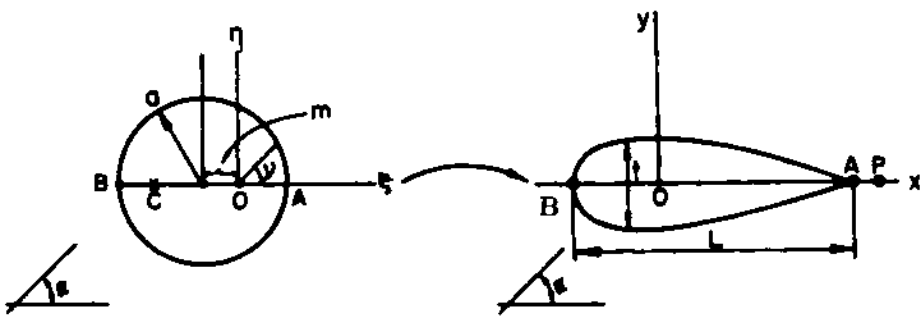


Figure 9.16: Flow Past a Joukowski Aerofoil

Table 9.4: Lift Coefficient for an Aerofoil

$t/L = 0.05$

αdegrees	0	5	10	15	20
C_L (panel)	0	0.442	0.88	1.313	1.735
C_l (analytical)	0	0.568	1.133	1.688	2.23

$t/L = 0.10$

αdegrees	0	5	10	15	20
C_L (panel)	0	0.49	0.98	1.46	1.93
C_l (analytical)	0	0.589	1.175	1.751	2.314

9.5 Vortex Panel Method

The source panel method is applicable to the problem of flow past non-lifting surfaces in general, since distribution of sources cannot be used to generate lift. However, flow with circulation yields a non-zero lift (since $L = \rho U \Gamma$) and a numerical method based on panels with a vortex distribution on them is more natural for the study of flow past lifting surfaces.

A vortex panel which has a distribution of circulation Γ per unit length on it will induce a velocity potential ϕ at a point $P(x, y)$ given as

$$\phi = -\frac{1}{2\pi} \int_{-L}^{L} \theta(s)\Gamma(s) ds \tag{9.15}$$

based on the complex potential $F(z) = \frac{i\Gamma}{2\pi} \ln z$ for a clockwise vortex of strength Γ; see Figure 9.17 for the definition of angle θ. As in the source panel method we assume $\Gamma(s)$ to be a constant on each panel and equal to Γ_j on the jth panel. Hence the velocity potential at a point P due to the jth panel is

$$\phi_{Pj} = -\frac{\Gamma_j}{2\pi} \int_{-L_j}^{L_j} \theta_j(s)\, ds_j$$

The total potential due to N panels is

$$\phi_{Pj} = -\sum_{j=1}^{N} \frac{\Gamma_j}{2\pi} \int_{-L_j}^{L_j} \theta_j(s)\, ds_j \tag{9.16}$$

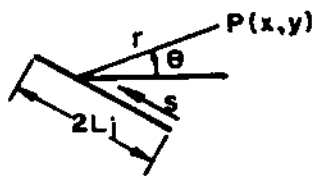


Figure 9.17: Definition of Geometry in Vortex Panel Method

The vortex strengths Γ_j are now determined from the condition that the surface of the object is a streamline and hence impermeable. This condition is enforced at N control points which are located at the centres of the panels themselves. The total potential at a point P due to a distribution of vortices and superimposed uniform flow is

$$\phi_P = U(x\cos\alpha + y\sin\alpha) - \frac{1}{2\pi}\sum_{j=1}^{N}\Gamma_j\int_{-L_j}^{L_j}\theta_j(s)\,ds \tag{9.17}$$

The constraint that the surface of the object is a streamline is equivalent to requiring that the normal velocity $\partial\phi/\partial n = 0$ at the control points. On the control point of the ith panel we have

$$\frac{\partial\phi_i}{\partial n} = \mathbf{U}\cdot\mathbf{n}_i - \frac{1}{2\pi}\sum_{j=1}^{N}\Gamma_j\int_{-L_j}^{L_j}\frac{\partial\theta_{ij}}{\partial n_i}\,ds_j = 0 \tag{9.18}$$

where θ_{ij} is the angle made at the control point i by any point on the jth panel. Clearly $\theta_{ii} = 0$. The above equation can be cast in a matrix form as

$$\frac{1}{2\pi}[D]\{\Gamma_j\} = \{\mathbf{U}\cdot\mathbf{n}_i\} \tag{9.19}$$

where the elements of matrix D are

$$D_{ij} = \int_{-L_j}^{L_j}\frac{\partial\theta_{ij}}{\partial n_i}\,ds_j$$

The quantities s and n are shown in Figure 9.18.

For any point $P(x_j, y_j)$ on the jth panel

$$\theta_{ij} = \tan^{-1}\frac{y_i - y_j}{x_i - x_j}, \quad -L_j < s_j < L_j$$

$$\frac{\partial\theta_{ij}}{\partial n_i} = \nabla\theta_{ij}\cdot\mathbf{n}_i, \quad \text{where} \quad \nabla = \left(\frac{\partial}{\partial x_i}, \frac{\partial}{\partial y_i}\right)$$

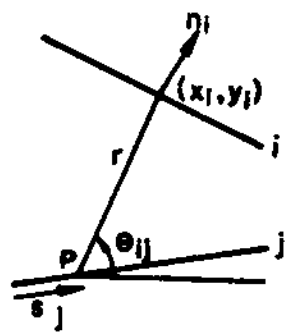


Figure 9.18: Description of Geometrical Quantities for Vortex Panel Method

For a panel of length $2L_j$ the unit normal is calculated as

$$\mathbf{n}_i = -\frac{(y_b - y_a)}{2L_j}\mathbf{i} + \frac{(x_b - x_a)}{2L_j}\mathbf{j}$$

The length $2L_j$ is given (from Figure 9.18) as

$$2L_j = \left[(x_a - x_a)^2 + (y_b - y_a)^2\right]^{1/2}$$

Finally

$$ds_j = \sqrt{1 + \left(\frac{dy}{dx}\right)_j^2}\, dx_j$$

where

$$\left(\frac{dy}{dx}\right)_j = \frac{y_c - y_d}{x_c - x_d} \quad \text{for the} \quad j\text{th panel.}$$

Combining these expressions, we get the entries of the matrix D in Equation 9.19 as

$$D_{ij} = \int_{x_c}^{x_d} F(x_j)\sqrt{1 + \left(\frac{dy}{dx}\right)_j^2}\, dx_j \tag{9.20}$$

where

$$\begin{aligned} F(x_j) &= \frac{1}{r^2}\left[(y_b - y_a)(y_i - y_j) + (x_i - x_j)(x_b - x_a)\right]\frac{1}{2L_j} \\ \text{and} \quad r^2 &= (x_i - x_j)^2 + (y_i - y_j)^2 \end{aligned}$$

Here (x_i, y_i) are the coordinates of the control point on the ith panel. The integral appearing in the definition of D_{ij} is best calculated numerically.

As with the source panel method $\mathbf{U}\cdot\mathbf{n_i} = \mathbf{U}\cos(\beta_i - \alpha)$, where β_i the angle made by the normal of the ith panel with x axis and α is the angle of attack. Following the procedure outlined in the source panel method, it is shown below that the diagonal elements of matrix D are

$$D_{ii} = 0$$

Consider a vortex panel of length $2L$ and a point P away from it (Figure 9.19). The potential induced at P is $\phi(P) = -\frac{\Gamma}{2\pi}\int_{-L}^{L}\theta(t)dt$ and the normal velocity $v = \partial\phi/\partial y = -\Gamma/(2\pi)\int_{-L}^{L}\partial\theta/(\partial y)\,dt$. We need the limiting value of v as P approaches the origin O. Since $\theta = \tan^{-1} y/(x - t)$, we have

$$\frac{\partial\theta}{\partial y} = \frac{x - t}{(x - t)^2 + y^2}$$

and

$$v = \frac{\Gamma}{4\pi} \ln\left[(x-t)^2 + y^2\right]\Big|_{-L}^{L}$$

This limit (and hence D_{ii}) is zero as x and y approach zero. The diagonal terms in higher order vortex panel methods are non-zero but small. Hence special matrix inverters that are capable of accommodating small (or zero) diagonal terms must be used while solving for the vortex strengths.

The vortex panel method will give a non-zero lift only when the Kutta condition is applied at the trailing edge of the object i.e., the trailing edge be a stagnation point. Hence flow leaves this edge smoothly and the net circulation here is zero. For the discretization shown in Figure 9.20, it is equivalent to writing

$$\Gamma_1 + \Gamma_N = 0 \tag{9.21}$$

Along with the Kutta condition, the vortex panel method generates $N+1$ equations for N unknowns. The solution is either obtained in the sense of least squares or by artificially ignoring one of the equations.

Example 9.4: Lift Generated by a Flat Plate

Consider the calculation of lift on a flat plate of length L placed in flow at an angle of attack α. We discretize the plate with N vortex panels, each of strength Γ_j, $j = 1 \ldots, N$. Since all panels lie along the x axis, their y coordinates are zero (Figure 9.21).

The unknown vorticities Γ_j are computed by requiring that the plate be impermeable at N control points. This results in the system of equations

$$\sum_{j=1}^{N} \Gamma_j D_{ij} = 2\pi U \sin\alpha$$

where

$$D_{ij} = \ln\left(\frac{a_j - x_i}{b_j - x_i}\right) \quad \text{and} \quad D_{ii} = 0$$

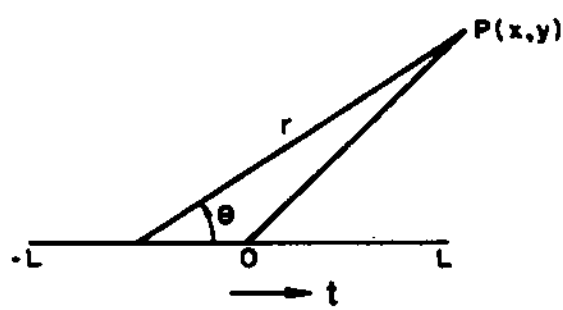


Figure 9.19: Computation of Diagonal Terms of Matrix D

a_j and b_j are the x-coordinates of the end points of the jth panel and x_i is the x-coordinate of the ith control point, i.e. the mid-point of the ith panel.

The Kutta condition is normally imposed at an exterior point C requiring that the total normal velocity at C be zero, i.e., the fluid does not turn around at C. This point is usually located at a distance 1.01 L from the leading edge of the plate. The velocity induced by a vortex panel at its end points is infinity and for this reason C is not located on the end of the plate itself. Hence at C,

$$U \sin\alpha - \sum_{j=1}^{N} \frac{\Gamma_j}{2\pi} \ln\left(\frac{a_j - x_c}{b_j - x_c}\right) = 0$$

We now have $N+1$ equations for N unknowns Γ_j. Ignoring the first equation (for the first panel) usually leads to the least error in the full solution.

For $N = 8$, the D-matrix has the following form:

$$[D_{ij}] = \begin{bmatrix} 0.133 & 0.154 & 0.182 & 0.222 & 0.287 & 0.404 & 0.689 & 4.83 \\ 1.090 & 0.000 & -1.090 & -0.510 & -0.336 & -0.250 & -0.200 & -0.167 \\ 0.510 & 1.090 & 0.000 & -1.090 & -0.510 & -0.336 & -0.250 & -0.200 \\ 0.336 & 0.510 & 1.090 & 0.000 & -1.090 & -0.410 & -0.336 & -0.250 \\ 0.250 & 0.336 & 0.510 & 1.090 & 0.0 & -1.090 & -0.510 & -0.336 \\ 0.200 & 0.250 & 0.336 & 0.510 & 1.090 & 0.000 & -1.090 & 0.510 \\ 0.167 & 0.200 & 0.250 & 0.336 & 0.510 & 1.090 & 0.000 & -1.090 \\ 0.143 & 0.167 & 0.200 & 0.250 & 0.336 & 0.510 & 1.090 & 0.000 \end{bmatrix}$$

The total vortex associated with the flat plate of length L is

$$\Gamma = \int_0^L \Gamma(x)dx$$

and the lift, from Kutta-Joukowski law is $\rho U\Gamma$. The lift coefficient is

$$C_L = \frac{\rho U\Gamma}{\frac{1}{2}\rho U^2 L} = \frac{2\Gamma}{UL}$$

The analytical solution for the problem described here under the assumption of potential flow is

$$\frac{\Gamma(x)}{2\pi U \sin\alpha} = \frac{1+\cos\theta}{\pi \sin\theta}$$

where θ and x are related as

$$\frac{x}{L} = \frac{1}{2}(1-\cos\theta), \quad 0 \le \theta \le \pi$$

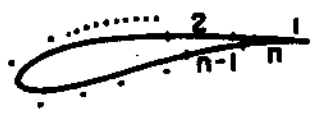


Figure 9.20: Kutta Condition

Table 9.5: Vortex Panel Method Applied to Flow Past a Flat Plate

x/L	0.10	0.20	0.40	0.60	0.80	1.00
$\bar{\Gamma}$(numerical)*	0.90	0.64	0.42	0.28	0.18	0.0
$\bar{\Gamma}$(analytical)	0.90	0.65	0.39	0.25	0.16	0.0

(* First panel ignored).

For this solution $\Gamma = \int_0^L \Gamma(x)dx = \pi \sin\alpha UL$ and $C_L = 2\pi \sin\alpha$. Let us non-dimensionalize $\Gamma(x)$ by $2\pi UL \sin\alpha$, and call it $\bar{\Gamma}(x)$. Then the matrix problem becomes

$$[D_{ij}]\,[\bar{\Gamma}] = 1$$

valid for all angles of attack. For $N = 41$, Table 9.5 compares the analytical and numerical solutions of $\bar{\Gamma}$. The average vortex strength $\int_0^1 \bar{\Gamma}(x)dx$ is 0.5062 and 0.5 based on the numerical and analytical solutions. These results are quite close to each other.

9.6 Inviscid Simulation of Real Flows

Real flows past solid objects at high Reynolds numbers are characterized by the formation of boundary-layers, separation, transition and turbulence. Such problems can be computed in principle through the use of the full unsteady Navier-Stokes equations. Numerical solutions of the full Navier-Stokes equations using extremely fine meshes and powerful computers continue to handle only modest Reynolds numbers. A typical upper limit for the Reynolds number for model-free computation of flow past a circular cylinder at this time of writing is of the order of a few thousands.

A wealth of literature that addresses real flows using the potential flow equation exists (Sarpkaya, 1989). The present section describes approaches employed in this direction. As stated earlier, these methods must be viewed as simulators that must be progressively fine-tuned to improve the theoretical predictions against laboratory experiments. The central idea that permits simulation of real flows using potential theory is the determination of the vorticity

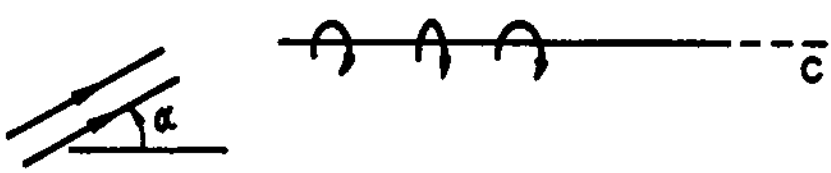


Figure 9.21: Flow Past a Flat Plate

generation rate. In potential flows this is zero; in real flows vorticity is generated in the boundary-layers formed on the solid surface and ejected into the wake at the point of separation. Once vorticity is created, it cannot be destroyed and in the absence of viscosity, it will be advected to infinity with the mean flow. However it can distort the flow sufficiently to produce an asymmetric time-dependent surface pressure distribution and hence a non-zero lift and drag. Present-day simulations using potential flow equations can predict oscillations of the separation and stagnation points with time, frequency of vortex shedding and the rate of growth of the wake size with distance.

For objects with sharp edges such as an aerofoil or a flat plate the points of separation will coincide with these sharp edges. The position at which the vortex is located at its birth is determined by requiring that the no-slip condition be satisfied at the point of separation. For bluff objects the initial condition before vortices are shed may be taken as attached flow or prescribed from experimental data. For objects with sharp trailing edges the Kutta condition is imposed at time $t = 0$.

The vorticity generation rate in the boundary-layer is given by the equation

$$\frac{\partial \Gamma}{\partial t} = \int_0^{\delta} u\omega \, dy \tag{9.22}$$

where Γ is the circulation of the nascent vortex, δ the boundary-layer thickness, u the velocity component parallel to the solid and ω, the vorticity $v_x - u_y$. For computation of quantities of interest in engineering it is necessary to model the wake with care. Hence Equation 9.22 is normally applied at the point of separation. At large Reynolds numbers δ is small and Equation 9.22 simplifies to

$$\frac{\partial \Gamma}{\partial t} = \frac{1}{2} U_s^2 \tag{9.23}$$

Here U_s is the free stream velocity prevailing at the edge of the boundary-layer around the point of separation. Since Equation 9.23 is only approximate, the point at which vorticity is produced may not coincide with the point of separation but be located at a distance m along the normal to the solid surface passing through this point. The introduction of a point vortex of strength Γ at a point z_i alters the complex potential for the flow field by an amount $i\Gamma/(2\pi)\ln(z - z_i)$. It also alters the fact that the solid surface is a streamline. For this reason it is necessary to introduce an image system of vortices to ensure that the normal velocity component on the surface is zero. For a circular cylinder each vortex in the fluid domain will introduce two vortices within, one of which will be at the cylinder centre. Since vortices shed from the points of separation are equal but opposite in strength the contribution of vortices at the centre will be zero at any instant of time. The second vortex within the cylinder is at z_i'. The location m of the generated vortex is determined on the basis of the fact that the no-slip condition is satisfied at the point of separation at the precise moment when the vortex is generated.

For a circular cylinder of unit radius and at a point of separation θ_s the expressions for z_i and z_i' in the complex plane are (Lamb, 1945; Churchill and Brown, 1990):

$$z_i = (1 + m_i)\exp\left(i(\pi - \theta_s)\right), \quad i = \sqrt{-1}$$

$$z_i' = 1/\bar{z}_i,$$

where bar denotes a complex conjugate and

$$1 + m_i = \frac{(1+ \mid \Gamma \mid /2\pi U_s)}{(1- \mid \Gamma \mid /2\pi U_s)} \tag{9.24}$$

In any given simulation Γ is estimated as $\frac{1}{2}U_s^2\Delta t$, where Δt is the time step. The time step is a parameter of the simulation and is usually chosen in such a way that the computation reproduces certain features observed in experiments.

Vortices once introduced in the flow will be carried forward by the fluid. A typical displacement of a vortex over a time step is computed as $\mathbf{u}\Delta t$ in the direction of the local velocity. For vortices created near the object surface, called nascent vortices, numerical experiments show that the characteristic velocity used for moving them should be the average of the velocities at the locations of the four earlier ones. This strategy improves the influence of the wake oscillations on the point of separation. The characteristic velocity used for moving the nascent vortices can also be used in the vorticity generation rate (Equation 9.23).

With the background given, an algorithm for simulating real flow past a bluff object can be developed along the following lines:

1. Potential flow past the object is initially solved using an analytical or a numerical method. Data for the point of separation at small time must come from analytical solutions or laboratory experiments. For example it is known that the initial point of separation for a circular cylinder is the rear stagnation point $\theta_s = \pi$.

2. A vortex along with its mirror image is generated in the vicinity of the point of separation. The strength and location of the nascent vortex are determined using Equations 9.23 and 9.24 respectively.

3. All vortices generated earlier are convected forward using the local velocity or an average of velocities prevailing at four vortex positions that fall on the same streak line.

4. Using the singularities that describe the solid surface, those generated at the current time level and in the past, the velocity field can be computed. For flow past a cylinder of unit radius with N vortices and N vortex images, the complex potential is

$$F(z) = U\left(z + \frac{1}{z}\right) + \sum_{j=1}^{N} \frac{i\Gamma_j}{2\pi}\left[\ln(z - z_j) - \ln\left(z - \frac{1}{\bar{z}_j}\right)\right]$$

and the complex velocity $u - iv$ is dF/dz. When a panel method is used the velocity at a point is the superposition of velocities induced by the panels and the discrete vortices in the flow field. There is no need for using an image system in this formulation since the panel strengths are determined on the basis that the surface of the cylinder, in the presence of vortices in the flow field is a streamline. In two dimensions it is possible to solve the stream function equation $\nabla^2\psi = -\omega$ on a grid for flow past a cylinder. Here values of ω at the nodes are obtained from the vortex positions using interpolation. This approach determines the fluid velocity and is useful when streamlines must be drawn to visualize the flow field.

5. The surface pressure distribution is calculated using the unsteady Bernoulli equation. Instantaneous lift and drag are subsequently computed.

6. With the pressure distribution known the points of separation are determined using Karman-Pohlhausen method. The KP method is not applicable in principle for unsteady problems with moving separation points. However numerical experiments show that the error incurred owing to this step is not large.

7. Vortices are generated at the new points of separation and the entire calculation starting with Step 2 is repeated. Calculations continue till the point of separation stabilizes or moves in a periodic manner.

References

Anderson, Jr., J.D., *Fundamentals of Aerodynamics*, McGraw-Hill, NY.,1985.

Banerjee, P.K. and Morino, L., *Boundary Element Methods in Nonlinear Fluid Dynamics*, Elsevier Applied Science, New York, 1990.

Churchill, R.V. and Brown, J.W., *Complex Variables and Applications*, McGraw-Hill, New York, 1990.

Currie, I.G., *Fundamental Mechanics of Fluids*, second edition, Mc-Grawh-Hill, New York, 1993.

Inamuro, T., Adachi, T. and Sakata, H., A Numerical analysis of Unsteady Separated Flow by Vortex Shedding Model, JSME Bull., Vol 26, pp 2106-2112, 1983.

Lamb, H., *Hydrodynamics*, Dover Publications, New York, 1945.

Lewis, R.I., *Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems*, Cambridge University Press, Cambridge, UK., 1991.

Moran, J., *An Introduction to Theoretical and Computational Aerodynamics*, John Wiley, NY., 1980.

Roach, G.F., *Green's Functions*, second edition, Cambridge University Press, Cambridge, 1986.

Sarpkaya, T., Computational Methods With Vortices - The 1988 Freeman Scholar Lecture, ASME J Fluids Engg, Vol 111, pp 5-21, 1989.

Schlichting, H. and Gersten, K., *Boundary-layer Theory*, Springer, New York, 2000.

Problems

1. Write computer programs for the examples considered in the present chapter using the source panel method. Discuss issues such as convergence of the solution with increasing number of panels and their order.
2. Develop a panel method using a combination of sources and vortices. Include Kutta condition for aerofoil-type objects.
3. Determine the drag acting on a Joukowski aerofoil as a function of thickness and the angle of attack.
4. Consider an array of three cylinders of equal diameter d placed on the vertex of an equilateral triangle of edge L. The cylinders will individually experience both lift and drag. Determine the value of L/d at which the force coefficient $C_F = \sqrt{C_L^2 + C_D^2}$ is a maximum for each of the cylinders.
5. Develop expressions for drag acting on a circular cylinder placed in oscillatory flow.
6. Develop the panel algorithm using a doublet distribution. Compare this method with the source and the vortex panel methods. [The velocity potential induced by a doublet is $\frac{1}{r}\cos\theta$.]
7. Consider the solution for flow past a flat plate using the vortex panel method. This is discussed in the text. Examine the accuracy of the method by ignoring successively the equations generated by the N panels. Examine the accuracy by solving the $(N + 1 \times N)$ system of equations using the method of least squares.
8. Formulate the source panel equations for N panels by minimizing the mean square leakage velocity on the object surface computed using $M(> N)$ points. Extend the procedure to vortex and doublet panel methods.
9. Develop the source panel method, vortex panel method and the doublet panel method to compute potential flow in a two dimensional channel with a sudden expansion. Identify the region of adverse pressure gradient and the point of separation. Study the sensitivity of the solution to the number of panels, order and the number of control points. Using the procedure developed by Sarpkaya (1989), extend the simulation to predict the size and strength of the recirculation zone in the channel.
10. Develop a computational model for determining the time-dependent lift and drag coefficients for a circular cylinder using a source panel method. The vorticity generation rate may be computed using Equation 9.23. A phase difference of $\pi/2$ is required between the events of the upper and lower halves of the cylinder to predict a non-zero instantaneous lift.
11. Solve for the potential velocity distribution and wall pressure in a two dimensional channel with a sudden expansion. Compare the numerical solution of panel method with finite differences applied to Laplace equation $\nabla^2\phi = 0$ and the analytical solution based on Schwarz-Christoffel transformation (Currie, 1993).

10
Three Dimensional Panel Method for Compressible Flow

10.1 Introduction

Although Navier-Stokes and Euler codes are able to handle increasingly complex geometries as a result of improvement in the grid generation process, they are still unable to model true complete configurations in a routine manner. Each new geometry requires a substantial time for the generation of flow field grid, and there is need to tune various new parameters within the code in order to adjust them for each new geometry and flow conditions. The ability to quickly model and analyze a new configuration in the design process of a new aircraft, or to evaluate the effect of adding or removing a piece of geometry from the existing model is of great importance. The present chapter describes a technique that can be used for aerodynamic design of aircraft at transonic speeds.

To overcome difficulties associated with the surface-conforming field grid generation for numerical solutions by finite difference techniques, computational schemes have been developed recently based on *integral equation* (IE) formulation for solving the *full potential equation* (FPE) for steady inviscid transonic flow. For example, see Johnson *et al.* (1982), Ravichandran *et al.* (1984), Oskam (1985), Erickson and Strande (1985), Kandil and Yates (1986), Sinclair (1986), Erickson *et al.* (1987), Samant *et al.* (1987), Sinclair (1988), Kandil and Hu (1988(a) and (b)), Samant *et al.* (1989), Sinclair (1990), Madson and Erickson (1990), Arora and Agarwal (1990), and Bhattacharya and Arora (1992, 1994).

The integral equation formulation uses the concept of zonal modelling, in which a field method is coupled with a surface panel method, the latter being employed to handle complex geometrical shapes. This procedure looks attractive for extending the practical scope of surface panel methods into the transonic

regime, *via* the field integrals. However some of these solutions are too complex, while others are hampered by the requirement for large memory and time to compute and store the field influences.

An efficient hybrid scheme which uses the integral method of solution, is described in this chapter. In the present work the field potentials are computed by a finite volume method using Cartesian grid, and the object (to be called *configuration*) is represented by flat panels. This yields a practical method for the analysis of complex configurations at transonic speeds, with moderate requirements on memory and time.

In the present method, the Green's third identity is first applied to the full potential equation. The free stream being subsonic, this procedure yields an integral equation for the disturbance velocity potential in terms of surface source and doublet singularity distributions, and the nonlinear field source distribution, that surround the configuration surface, representing compressibility. The solution thus constructed automatically satisfies the far field boundary condition at infinity. Since the field sources die rapidly away from the body only a small computational domain around the body would then suffice to solve the flow problem.

To obtain a numerical solution of the integral equation, surface panels are used in conjunction with the field volume elements in a coupled manner. The configuration surface and the wake are represented by a number of flat quadrilateral panels and the singularity distribution is assumed piecewise constant over each panel. A Cartesian three-dimensional grid is generated within a finite volume around the configuration and the wake, which encompasses and penetrates the body. The cells formed by the rectangular grid are used as finite elements, wherein the field source strengths are assumed to be uniform. Thus the field grid is independent of the surface geometry.

The surface panels are used to enforce the surface boundary conditions and finite volume cells to compute the field compressibility influences. The field potentials are evaluated by a finite volume successive line over relaxation (SLOR) scheme sweeping outwards from the surface boundary. The surface source distribution is determined by the external Neumann boundary condition and the Kutta condition is enforced for the determination of the doublet distribution. Application of the Kutta condition is simple and no extra equation or trailing edge velocity potential is required. This gives rise to a Poisson type iterative procedure for the numerical solution of the integral equation. To obtain the surface velocity and hence the pressure distribution, the surface potentials are numerically differentiated.

For supercritical flows the nonlinear source term in the FPE is augmented by the addition of artificial viscosity in the supersonic region which produces an upwind bias and satisfies the proper domain of dependence. Following Osher, Hafez and Whitlow (1985), mass flux biasing is used for modelling the artificial viscosity, which lends an excellent shock capture capability to the method as evidenced by test results reported later (Section 10.4).

Boeing TRANAIR method is a similar effort, but far more complex though

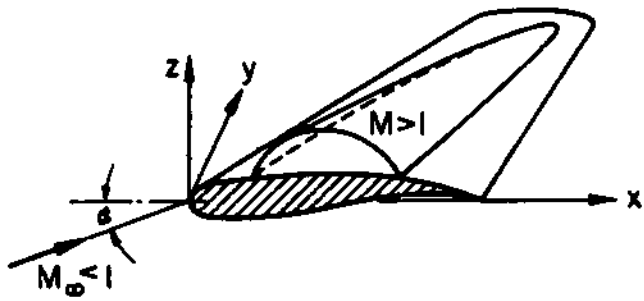


Figure 10.1: Steady Flow Problem: Reference Coordinates

extremely versatile, has been developed over a long period of time by a large team of scientific workers (Samant *et al.*, 1987, Erickson *et al.*, 1987). Another effort in the similar category is that of British Aerospace (Sinclair, 1990). Both these efforts differ from our approach in many significant details. The general formulation of the integral equation for the FPE presented below is similar to the work of Erickson and Strande (1985).

10.2 Formulation

10.2.1 Governing Equations

The full potential equation for steady inviscid compressible flow past a three dimensional aerodynamic configuration (Figure 10.1) can be expressed in terms of the velocity potential ϕ as

$$\nabla \cdot (\rho \nabla \phi) = 0 \tag{10.1}$$

where ρ is density given by isentropic relation

$$\rho = \left[1 + \frac{1}{2}(\gamma - 1)\mathrm{M}_\infty^2 (1 - \nabla\phi \cdot \nabla\phi)\right]^{1/(\gamma-1)}$$

The subscript ∞ refers to the undisturbed free stream condition. The velocity $\boldsymbol{q} = \nabla\phi$ and density ρ have been normalized by their free stream values. For subsonic free stream Mach numbers ($\mathrm{M}_\infty < 1$), Equation 10.1 may be expressed as the Poisson equation

$$\nabla^2 \phi(x, y, z) = G \tag{10.2}$$

where $G = (-1/\rho)(\nabla\rho \cdot \nabla\phi)$ represent the density variation in the flow field. Then application of Green's third identity (Kellog, 1967, Moulden, 1984) to Equation 10.2 yields an integral equation, which for the perturbation potential $\varphi = \phi - \phi_\infty$, at the field point $P(x, y, z)$ can be expressed as

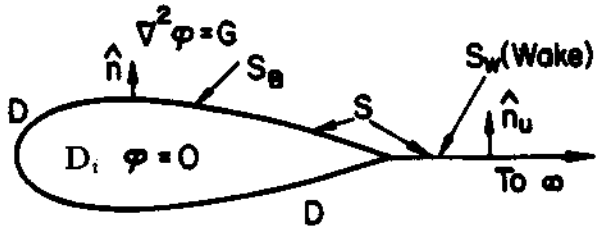


Figure 10.2: Body Configuration, Fluid Domain and Boundary

$$\varphi = \iint_S \{\sigma(Q)K_\sigma + \mu(Q)\,K_\mu\}\,dS + \iiint_D G(Q)K_\sigma dV \tag{10.3}$$

where

$$\begin{aligned} K_\sigma &= -1/(4\pi R), \quad K_\mu = \hat{\mathbf{n}}\cdot\nabla_Q\{-K_\sigma\} \\ \boldsymbol{R} &= \boldsymbol{Q} - \boldsymbol{P} \equiv (\xi - x, \eta - y, \zeta - z) \\ |\,\mathbf{R}\,| &= \left[(\xi - x)^2 + (\eta - y)^2 + (\zeta - z)^2\right]^{1/2} \\ \sigma &= \Delta\,(\partial\varphi/\partial n)\,, \quad \text{and} \quad \mu = \Delta\varphi \end{aligned}$$

Here Δ represents jump across the surface S that bounds D and D_i (Figure 10.2). In the surface integral (Equation 10.3), Q represents a variable point of surface S, whereas in the volume integral it represents a variable point of domain D; $\mathbf{R}$ is the distance from the field point P to the variable point Q; $\hat{\mathbf{n}}\cdot\nabla = \partial/\partial n$ represents the derivative normal to the surface S, directed into domain D. Integral equation 10.3 states that the velocity potential at a particular point P can be computed from the source and doublet distribution $\sigma(Q)$ and $\mu(Q)$ on S (taken to be the configuration surface and the wake), and from the spatial distribution of field sources G, representing the compressibility in the domain D around S.

Since $G = (-1/\rho)\,(\nabla\rho\cdot\nabla\phi)$ in Equation 10.1 is a nonlinear function of $\nabla\varphi$, $\varphi(P)$ cannot be constructed directly. This is possible with the solution of the linear equation

$$\nabla^2\varphi = 0$$

which is represented by

$$\varphi(P) = \iint_S (\sigma K_\sigma + \mu K_\sigma)\,dS$$

Thus the distribution of $\varphi(x, y, z)$ that satisfies the non-linear problem, Equation 10.3 and the boundary conditions on S, must be determined by some iterative procedure. If this is done, it follows from Equation 10.2 that

$$\iiint_D \left(\nabla^2\varphi - G\right)\,K_\sigma dV = 0$$

i.e. the full potential equation has been satisfied throughout D.

Further in domain D for a finite volume cell D_c enclosed by surface area S_c, the governing Equation 10.1 may be expressed as

$$\iiint_{D_c} \nabla \cdot (\rho \nabla \varphi)\, dV = \iint_{S_c} \rho\, (\hat{\mathbf{n}} \cdot \nabla \phi)\, dS = 0 \tag{10.4}$$

Here, $\hat{\mathbf{n}}$ is the unit outward drawn normal on the domain D pointing into the fluid region (Figure 10.2). Equation 10.4 may be used to evaluate the field potentials in the domain D through a finite volume procedure and thereby evaluate the field sources.

10.2.2 Boundary Conditions

On the body surface, the total normal component of velocity must vanish, i.e.

$$\hat{\mathbf{n}} \cdot \nabla \phi = \hat{\mathbf{n}} \cdot (\mathbf{V}_\infty + \nabla \varphi) = 0 \tag{10.5}$$

Further one prescribes the perturbation potential underneath the body surface S to be zero, i.e.

$$\varphi = 0 \quad \text{in} \quad D_i \tag{10.6}$$

the interior Dirichlet boundary condition.

With the boundary condition (Equation 10.6) the source and doublet distribution on S become

$$\sigma = \partial \varphi / \partial n \quad \text{and} \quad \mu = \varphi \tag{10.7}$$

where φ is now the perturbation potential on the exterior surface of S facing the fluid domain D.

Using the boundary condition (Equation 10.5) together with Equation 10.7, the source distribution is obtained as

$$\sigma = -\hat{\mathbf{n}} \cdot \mathbf{V}_\infty \tag{10.8}$$

Since for a specified problem, the geometry of the configuration $\hat{\mathbf{n}}$ and flow geometry $\mathbf{V}_\infty$ are known, the source distribution σ is determined a priori. Further, from Equation 10.7 we note that the doublet distribution μ is the perturbation potential φ, which is the unknown of the flow problem.

In addition, the Kutta condition of zero pressure difference is specified at the trailing edges.

Finally, note that the far field boundary condition of vanishing perturbation potential is automatically satisfied by the integral Equation 10.3.

10.2.3 Integral Equation

With source strength σ specified through Equation 10.8, Equation 10.3 gives for the potential at the field point P

$$\begin{aligned} E^*\varphi(P) &= \iint_{S_B} \sigma(Q)\, K_\sigma dS + \iint_{S_{B-P}} \varphi(Q)\ \hat{\mathbf{n}} \cdot \nabla_Q(-K_\sigma) dS \\ &+ \iint_{S_W} \Delta\varphi_w(Q)\, \hat{\mathbf{n}}_u \cdot \nabla_Q(-K_\sigma) dS \\ &+ \iiint_D G(Q) K_\sigma dS \end{aligned} \tag{10.9}$$

$$\begin{aligned} \text{with} \quad E^* &= 1/2, \quad \text{for points on the body surface} \\ &= 1, \quad \text{for points in the domain } D \\ \text{and} \quad \varphi &= 0, \quad \text{for points inside the body, the domain } D_i. \end{aligned}$$

Here S_B and S_W indicate the body and the wake surfaces respectively; S_{B-P} indicates the portion of S_B that does not contain point P; $\hat{\mathbf{n}}_u$ represents the unit outward normal drawn on the upper surface.

For present discussion, we assume that the wake has vanishing thickness and it has zero entrainment, which leads to the jump condition:

$$\Delta\left(\hat{\mathbf{n}} \cdot \nabla\varphi\right) = 0 \quad \text{on} \quad S_W$$

Since the pressure is continuous across the wake, we conclude that the difference in potential ϕ across the wake S_W, denoted by $\Delta\varphi_w$, is constant along a streamline in the wake. For the disturbance potential this may be expressed as

$$\Delta\varphi_w = \varphi_u - \varphi_l = \text{constant on } S_W \text{ along a streamline}$$

This must be equal to the value at the trailing edge (te), at any spanwise location, i.e.

$$\Delta\varphi_w = \Delta\varphi_{te}$$

This ensures that the Kutta condition of zero pressure difference is satisfied at the trailing edge.

10.2.4 Artificial Viscosity

The full potential Equation 10.1 for transonic flow has mixed character (see Chapter 1, Section 1.7); it behaves as an elliptic equation in subsonic regions and hyperbolic in supersonic regions. The central difference approximation for this equation mimics an elliptic behaviour and hence is unable to ensure stability in the supersonic zones or to capture shocks. For supercritical flows, the

hyperbolicity of the equation should be ensured at supersonic points. To simulate this requirement, upwind differencing can be accomplished by the addition of the artificial viscosity vector **A** and components (P, Q, R) to Equation 10.1. This step in conservative form can be expressed as

$$\nabla \cdot (\rho \nabla \phi + \boldsymbol{A}) = 0 \tag{10.10}$$

The artificial viscosity **A** can be modelled either by using the density biasing scheme following Holst (1980) or by using the mass flux biasing scheme (Osher *et al.*, 1985).

Here we consider modelling the artificial viscosity following Osher *et al.* (1985) referred to as Hafez-Osher (H-O) scheme. This scheme uses upwinding of the mass flux rather than the density and yields the following form for the derivatives of the component of the artificial viscosity:

$$\begin{aligned}
\frac{\partial P}{\partial x} &= -\Delta x \overrightarrow{\delta_x} \left[\frac{\left(\overleftarrow{\delta_x}\phi\right)^+}{q} \overleftarrow{\delta_x} (\rho q)_- \right], \quad \overleftarrow{\delta_x}\phi > 0 \\
\frac{\partial P}{\partial x} &= \Delta x \overleftarrow{\delta_x} \left[\frac{\left(\overrightarrow{\delta_x}\phi\right)^-}{q} \overrightarrow{\delta_x} (\rho q)_- \right], \quad \overrightarrow{\delta_x}\phi < 0
\end{aligned} \tag{10.11}$$

where $\overleftarrow{\delta_x}$ and $\overrightarrow{\delta_x}$ refer to backward and forward differencing respectively, and the operators

$$g^+ = \max(g, 0), \quad g^- = \min(g, 0)$$

The term $(\rho q)_-$ is expressed as

$$\begin{aligned}
(\rho q)_- &= \rho q - \rho^* q^*, \quad \text{if} \quad q > q^* <=> M > 1 \\
(\rho q)_- &= 0, \quad \text{if} \quad q \leq q^* <=> M \leq 1
\end{aligned}$$

The quantities ρ^* and q^* represent the sonic condition given by

$$(q^*)^2 = \frac{2 + (\gamma - 1)\mathrm{M}_\infty^2}{(\gamma + 1)\mathrm{M}_\infty^2}, \quad \rho^* = (q^* \mathrm{M}_\infty)^{2/(\gamma-1)}$$

Corresponding expressions can be written for the derivatives of artificial viscosity in the y and z directions.

The H-O scheme admits physically correct limit solutions to the FPE, and hence is more appropriate as a model for artificial viscosity. The other advantage of this scheme is that there is no need for the user defined parameters to control the artificial dissipation, as are usually required for the density biasing schemes, see for example, Baker (1983). This scheme is applied in evaluating the field sources.

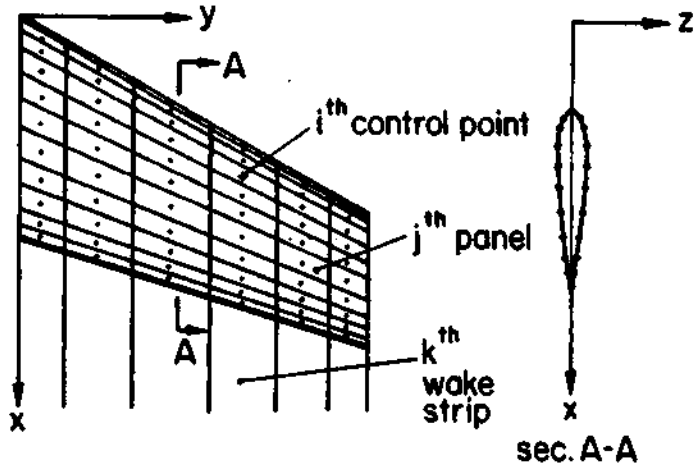


Figure 10.3: Illustration of Surface Panelling

With the addition of artificial viscosity, Equation 10.10 for velocity potential can be expressed in Poisson form as

$$\nabla^2\phi = -\frac{1}{\rho}\left(\nabla\rho\cdot\nabla\phi + \nabla\cdot\mathbf{A}\right) = G^* \tag{10.12}$$

where G^* now represents the modified field sources related to G and $\mathbf{A}$ as

$$G^* = G - \frac{1}{\rho}\nabla\cdot\mathbf{A}$$

This ensures the upwind biasing necessary to simulate the hyperbolic behaviour at supersonic points and for shock capture.

The density biasing scheme, on the other hand is simpler and faster and is used for the computation of field potentials by a finite volume technique. With the introduction of artificial viscosity, Equation 10.4 may be expressed as

$$\iint_{S_c} \tilde{\rho}\left(\hat{\mathbf{n}}\cdot\nabla\phi\right) dS = 0 \tag{10.13}$$

where $\tilde{\rho}$ is retarded density. The algebraic expression for $\tilde{\rho}$ depends on the direction of retardation, e.g. retarding density in the x-direction is expressed as

$$\begin{aligned} \tilde{\rho}_x &= \rho - \Delta x\,\nu\,\partial\rho/\partial x, \quad \delta_x\phi > 0 \\ &= \rho + \Delta x\,\nu\,\partial\rho/\partial x, \quad \delta_x\phi < 0 \end{aligned} \tag{10.14}$$

where ν is the switching function which takes the value zero for subsonic points.

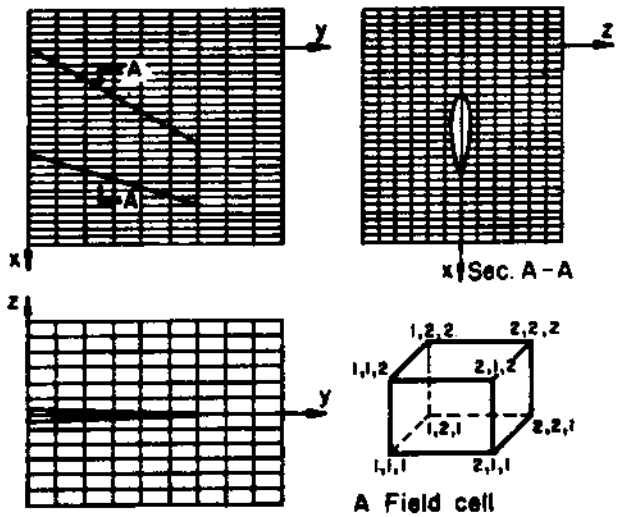


Figure 10.4: Field Grid for Finite Volume Calculations

10.3 Numerical Solution

10.3.1 Discretization

For the numerical solution of integral Equation 10.9, the body surface and the wake are represented by a number of flat quadrilateral panels (Figure 10.3) and a constant singularity distribution is assumed on each panel. For geometry definition and flat panel approximation, see Johnson (1980).

Although D is the complete domain external to the configuration, the field source strengths G decay rapidly away from the surface. Hence in practice only a limited region around the configuration would be sufficient to solve the flow problem. The region of space including the configuration and its immediate neighbourhood is subdivided into finite volume cells with Cartesian grid, which penetrates the configuration (Figure 10.4). The field source strengths are assumed to be uniform in each cell. Thus we have the standard surface panelling as in a subsonic panel method, along with the field cells and two set of control points (Figure 10.5). The surface control points are taken to be the centroids of the surface panels where the boundary conditions on the body surface are specified. The field control points are defined as the centroids of the cells, where the field source distributions are evaluated.

Assigning index i to each control point at which Equation 10.9 with modified field sources G through Equation 10.12 is to be satisfied, we obtain

$$E^*\varphi_i = \sum_{j=1}^{Ns} \sigma_j B_{ij} + \sum_{j=1, i\neq j}^{Ns} \varphi_j C_{ij} +$$

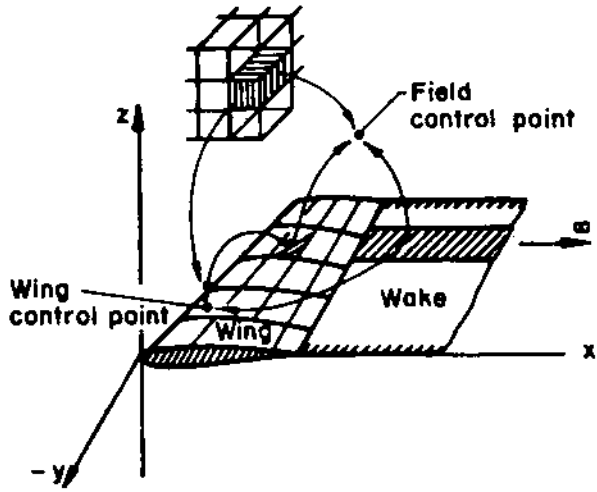


Figure 10.5: Discretization and Influences among the Surface Panels, Wake and the Field Elements

$$\sum_{k=1}^{Nw} (\Delta\varphi)_k \, Cw_{ik} + \sum_{n=1}^{Ng} G_n^* D_{in} \tag{10.15}$$

$$\text{where} \quad E^* = 1/2, \quad \text{for} \quad i = 1, 2 \ldots Ns, \quad \text{surface control points (CP)}$$
$$= 1, \quad \text{for} \quad i = 1, 2 \ldots N_g, \quad \text{field points (GP)}$$

N_s is the total number of surface panels on the body, N_w the total number of streamwise wake columns and N_g the total number of field cells in domains D and D_i. Further

$$B_{ij} = \iint_{\sum_j} K_\sigma \, dS, \quad C_{ij} = \iint_{\sum_j} \hat{\mathbf{n}} \cdot \nabla_Q \left(-K_\sigma\right) dS$$

$$Cw_{ik} = \iint_{\sum_k} \hat{\mathbf{n}}_u \cdot \nabla_Q \left(-K\sigma\right) dS, \quad D_{in} = \iiint_{\Delta V_n} K_\sigma \, d\xi d\eta d\zeta$$

B_{ij} and C_{ij} represent the potentials induced at the ith control point due to a source and doublet distribution of unit strength respectively on the jth surface panel. D_{in} is the potential induced at the ith control point due to a uniform unit source distribution in the nth field cell. The surface integrations are indicated on flat quadrilateral jth panel $\sum_j$, kth wake column $\sum_k$ and volume integration on nth field cell ΔV_n. In Equation 10.15

$$\Delta\varphi_k = (\Delta\varphi_{te})_k = \varphi_{Nu} - \varphi_{Nl}$$

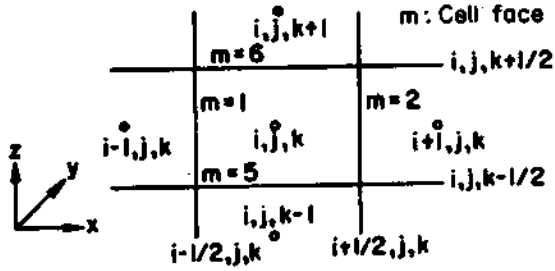


Figure 10.6: Field Cell - Index Notation

where φ_{Nu} and φ_{Nl} are taken as the values of the potential on the panels adjacent to the trailing edge on the upper and lower surface respectively at the start of the kth wake column. Hence Kutta condition at the trailing edge is satisfied implicitly and no extra equation is needed.

Equation 10.15 with $G_n^* = 0$ gives rise to a low order surface panel method for incompressible flow problem. This approach has been pioneered by Morino *et al.* (1975) and forms a subset of the general method of Johnson (1980) for incompressible flows. It has been further advanced by Maskew (1982) and others.

For the field solution, the finite volume relation follows from Equation 10.13 and may be written as

$$\sum_{m=1}^{6} W_m \tilde{\rho}_m \boldsymbol{V}_m \cdot \hat{\mathbf{n}}_m = 0 \tag{10.16}$$

where $m = 1 \ldots 6$ represents the 6 faces of a cell (Figure 10.6). The quantity $\boldsymbol{V}_m \cdot \hat{\mathbf{n}}_m$ may be expressed as

$$\boldsymbol{V}_m \cdot \hat{\mathbf{n}}_m = (\varphi_m - \varphi_0)/d_m + \boldsymbol{V}_\infty \cdot \hat{\mathbf{n}}_m \tag{10.17}$$

Here φ_0 represents potential inside the cell, φ_m the potential in adjacent cell across face m of area W_m, and d_m is the distance between centroids of cells lying on two sides of face m. Using Equation 10.17, Equation 10.16 may be represented as

$$\sum_{m=1}^{6} \left[a_m (\varphi_m - \varphi_0) + b_m \boldsymbol{V}_\infty \cdot \hat{\mathbf{n}}_m\right] = 0 \tag{10.18}$$

$$\text{where} \quad a_m = W_m \tilde{\rho}_m / d_m \quad \text{and} \quad b_m = W_m \tilde{\rho}_m$$

10.3.2 Numerical Implementation of Artificial Viscosity

The upwind density biasing, Equation 10.14 provides for $\tilde{\rho}_m$ in discretized form as

$$\begin{aligned}\tilde{\rho}_{i+1/2,j,k} &= [(1-\nu)\,\rho]_{i+1/2,j,k} + \nu_{i+1/2,j,k}\,\rho_{i+r+1/2,j,k} \\ \tilde{\rho}_{i-1/2,j,k} &= [(1-\nu)\,\rho]_{i-1/2,j,k} + \nu_{i-1/2,j,k}\,\rho_{i+r-1/2,j,k} \qquad (10.19)\end{aligned}$$

$$\text{with} \quad r = \pm 1 \quad \text{when} \quad u_{i,j,k} = \phi_{x_{i,j,k}} < \text{or} > 0$$

and ν is a weighting factor to the artificial viscosity. It also serves as a switching function, defined as

$$\nu_{i+1/2,j,k} = \begin{cases} \left[(M^2_{i,j,k} - 1)C\right]^+ & \text{for } u_{i+1/2,j,k} > 0 \\ \left[(M^2_{i+1,j,k} - 1)C\right]^+ & \text{for } u_{i+1/2,j,k} < 0 \end{cases}$$

The relationship between the cell faces denoted by $m = 1 \ldots 6$ and the indices used in the above expressions are explained in Figure 10.6. The cell centroid has the index i, j, k. The faces $m = 1$ and 2 are parallel to the x-plane, and are denoted by the indices $i - 1/2, j, k$ and $i + 1/2, j, k$. Similarly, faces 3 and 4 which are parallel to the y-plane, and 5 and 6 to the z-plane, are denoted appropriately. Upwind biasing of density in the y- and z- directions may be written by expressions similar to the above. Equation 10.19 is a modified form of the expressions used by Holst (1980) for upwind density biasing, where the indices are used to represent grid points rather than cell centroids.

It is obvious that ν switches to '0' in subsonic regions. The parameter C is a user specified constant affecting the amount of artificial viscosity added and is usually set between 1.0 and 2.0.

The differentiation of field potentials to obtain the field sources is accompanied by upwind biasing of mass-flux, wherein the discretized form of the Equation 10.12 may be written as (Arora and Agarwal, 1990)

$$\begin{aligned}\left[\frac{\partial P}{\partial x}\right]_i &= -\frac{1}{q_{i+1}}(UD)_{i+1/2} + \frac{1}{q_i}(UD)_{i-1/2}, \quad \text{for} \quad u_{i-1/2} > 0 \\ &= \frac{1}{q_i}(UD)_{i+1/2} - \frac{1}{q_{i-1}}(UD)_{i-1/2}, \quad \text{for} \quad u_{i+1/2} > 0 \quad (10.20)\end{aligned}$$

where

$$(UD)_{i+1/2} = \frac{(\rho_{i+1}q_{i+1})_- - (\rho_i q_i)}{x_{i+1} - x_i} - u_{i+1/2}, \quad \text{and} \quad u = \phi_x.$$

Similar expressions are obtained for $\partial Q/\partial y$ and $\partial R/\partial z$.

10.3.3 Evaluation of Influence Coefficients

The surface influence coefficients B_{ij}, C_{ij} and Cw_{ik} can be easily evaluated in the panel coordinate system $(\xi, \eta, \zeta = 0)$ attached with the influencing panel j.

They are then expressible in terms of the standard integrals (Johnson, 1980) as

$$B_{ij} = -\frac{1}{4\pi} H(1,1,1), \quad C_{ij} = \frac{1}{4\pi} zH(1,1,3)$$

$$\text{with} \quad H(M,N,K) = \iint_{\Sigma} \frac{(\xi - x)^{M-1} (\zeta - y)^{N-1}}{\rho^k} d\xi d\zeta$$

$$\rho = R(\zeta = 0), \quad R = \left[(\xi - x)^2 + (\eta - y)^2 + (\zeta - z)^2\right]^{1/2}$$

Since φ is a scalar, the summations indicated over the body surface and the wake in Equation 10.15 are then straightforward, as no further coordinate transformation is required.

The field source influence coefficient D_{in} for each of the field cells, external to the body in domain D, can be evaluated analytically in closed form in terms of the grid points forming the field cells in the shape of cuboids (Figure 10.4). Hence

$$\begin{aligned} D_{in} = 1/(4\pi)[&I(2,2,2,) + I(1,1,2) + I(1,2,1) + I(2,1,1) \\ &-I(1,2,2) - I(2,1,2) - I(2,2,1) - I(1,1,1)] \end{aligned}$$

where

$$\begin{aligned} I &= \iiint \left(-\frac{1}{|\mathbf{R}|}\right) d\xi\, d\eta\, d\zeta \\ &= AB \ln\{(H+C)/D\} + AC \ln\{(H+B)/E\} \\ &+ BC \ln\{(H+A)/F\} + \frac{1}{2}\left[B^2 \tan^{-1}\{BH/(AC)\}\right. \\ &+ \left. C^2 \tan^{-1}\{CH/(AE)\} + A^2 \tan^{-1}\{AH/(BC)\}\right] \end{aligned}$$

$$\text{with} \quad A = x - \xi, \quad B = y - \eta, \quad C = z - \zeta, \quad D^2 = A^2 + B^2$$

$$E^2 = A^2 + C^2, \quad F^2 = B^2 + C^2 \quad \text{and} \quad H^2 = A^2 + B^2 + C^2$$

It may be observed that the identical field cells ($n = 1, 2, \ldots Ng$) with their centroids located at the same axiswise distance from the field control point i will have the same value of D_{in}. This symmetry property is exploited in the evaluation of D_{in} which reduces considerably the number of calculations. More importantly a saving in the overall storage can be achieved.

The body surface slices through some of the field cells, which alters the finite volume discretization in the vicinity of the boundary surface (Figure 10.7). An exact evaluation of field source integrals for such cells, which are of trapezoidal shape, can be easily achieved by the use of Gauss divergence theorem. It can be shown that the field integral for such a cell is the same as the field integral over the complete cuboid which partly lies inside the body, i.e. domain D_i, with the source strength evaluated on the body surface from the external field. Since

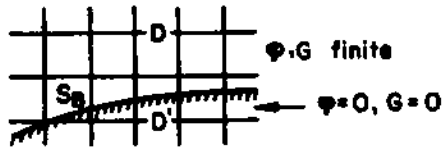


Figure 10.7: Finite Volume Discretization on the Body Surface

$\varphi(P) = 0$ inside the domain D_i, the value of the volume integral over the cells completely lying inside the body will be zero. Evaluation of the field integral is thus a simple summation over all the field cells.

10.3.4 Iteration Scheme

The construction given by the discretized form of the integral Equation 10.15, can be solved by the following iterative procedure.

Step I

The iteration starts with an initial guess of G^*. The simplest choice would be $G^* = 0$. Equation 10.15 then corresponds to the solution of incompressible flow and gives N_s linear algebraic equations for the value of φ at the surface control points, $i = 1, ...N_s$, i.e. φ^o(CP).

Step II

With the values of surface potentials $\{\varphi^o(\text{CP})\}$ as internal Dirichlet boundary condition, compute $\{\varphi^o(\text{GP})\}$ at the field cell-centroids, using finite volume formulation provided by Equation 10.18.

The finite volume cells, formed by the Cartesian grid, are categorised into those that lie within the configuration, those that pierce the surface and those that lie in the field. In the internal Dirichlet formulation that has been adopted, the cells of the first category do not have any influence on the rest of the field, while those of the second category take on that value of potential which is held by the portion of the configuration surface penetrated by that cell and is implemented through an appropriate interpolation scheme (Figure 10.8). Here φ_{gc}, the value of potential on the centroid of a cell intersecting the surface, is given in terms of φ_{pc_1} and φ_{pc_2}, the potentials on the two adjacent surface panels in the wraparound direction, as

$$\varphi_{gc} = w_1\varphi_{pc_1} + w_2\varphi_{pc_2}$$

where w_1, w_2 are linear weight functions. These cells are then treated as boundary cells for evaluation of potentials in the field.

The potential values φ(GP) for the field cells are evaluated by iteratively sweeping outwards from the surface boundary and updating the potential at a

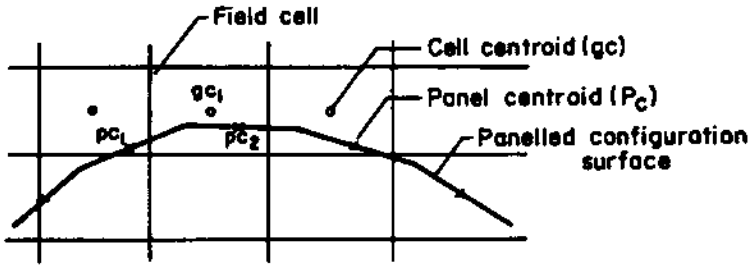


Figure 10.8: Transfer of Potentials from Surface onto Field $\phi_{gc} = f(\phi_{pc_1}, \phi_{pc_2})$

cell centroid in terms of the potentials held by the neighbouring cell centroids by an implicit finite volume successive line over-relaxation (SLOR) scheme. An implicit SLOR scheme for solving in the z-direction and sweeping in the x and y-directions may be formed by writing Equation 10.18 as

$$-a_5\varphi_5 + A\varphi_0 + a_6\varphi_6 = \sum_{m=1}^{4} a_m\varphi_m + \sum_{m=1}^{6} b_m \boldsymbol{V}_\infty \cdot \hat{\mathbf{n}}_m \tag{10.21}$$

with $A = \sum_{m=1}^{6} a_m$, and $m = 5$ and 6 denote the lower and upper cell faces in the z-direction (Figure 10.6).

Equation 10.21 shows that the latest available values of φ across x- and y-faces of the cell are treated as known, and are sent to the RHS. The resulting tridiagonal system of equations is then solved by the Thomas algorithm (Chapter 2). For the first cell in the field on a z-column, the value of φ_5 in Equation 10.21 is known as it belongs to the cell which intersects the boundary surface S_B. If the first cell intersects the wake surface, the Neumann condition of zero mass flux through the cell face oriented along the wake is enforced. For the far field boundary, it is assumed that φ no longer changes in the z-direction, and hence for the last cell in the field $\varphi_6 = \varphi_0$.

Once the field potentials are known, the gradients of the potential, the density and its gradients, and $\nabla \cdot \mathbf{A}$ are computed in the field by finite differences, to provide the first estimate of $(G_n^*)^1$ over all the field cells through Equation 10.12. Using these values, the field integral on the body surface $f(\mathrm{CP})$, is obtained as

$$f^1(\mathrm{CP}) = \sum_{n=1}^{N_g} (G_n^*)^1 \; D_{in} \tag{10.22}$$

Step III

With $\{f^1(\mathrm{CP})\}$ known, Equation 10.15 is again solved for $\{\varphi^1(\mathrm{CP})\}$ as the solution of N_s linear algebraic equation for the surface control points. This completes one iteration cycle. The convergence criterion is specified in terms of the surface potentials to a set tolerance.

During every iteration cycle we are led to a set of N_s algebraic equations for the surface alone. These are solved by the Gauss-Siedel iterative procedure with successive over-relaxation, providing N_s disturbance potential values on the body surface. The over-relaxation factor is optimized by numerical experimentation.

10.3.5 Computation of Velocity and Pressure Distribution

The iterative solution yields N_s disturbance potential φ values on the body surface. A local linear distribution of φ is then assumed using values at five panel centres, that is a central panel and its four immediate neighbours. The surface potentials are differentiated using the least squares procedure. The distribution is suitably modified for the points in the neighbourhood of the wing leading and trailing edges and the wing root and tip. Thus the component of disturbance velocity tangential to the body surface is obtained at the N_s control points.

The local total velocity at the body surface is then obtained as

$$q = | \hat{\mathbf{e}}_\infty + \nabla\varphi | \tag{10.23}$$

Finally the pressure on the body surface is computed using the isentropic relation

$$C_p = -\frac{2}{\gamma M_\infty^2}\left[\left\{1 + \frac{\gamma-1}{2} M_\infty^2 (1-q^2)\right\}^{\frac{\gamma}{\gamma-1}} - 1\right] \tag{10.24}$$

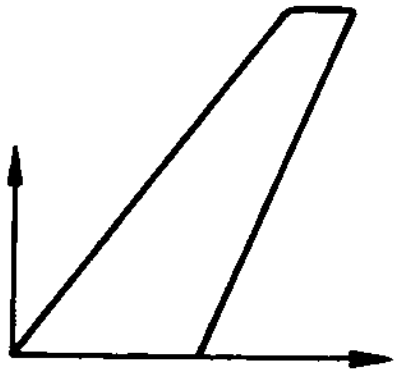
10.4 Applications and Discussion

As application of the technique discussed in this chapter, computations have been performed on an axisymmetric launch vehicle, M-6 wing, MTA wing, RAE wing-body configuration and Lockheed Wing C at supercritical speeds. The computed pressure distributions have been compared with other known computations or with experiments or both. Examples of complex configurations include a MTA wing-body and a multi-body launch vehicle configuration at subcritical speeds. Figure 10.9 shows the geometric features of M-6 wing, Lockheed wing C and RAE wing A used in the RAE wing-body combination and Figure 10.10 shows the surface panelling on the RAE wing-body. MTA wing and wing-body configurations are represented in Figure 10.11.

Subcritical Surface Panel Solutions

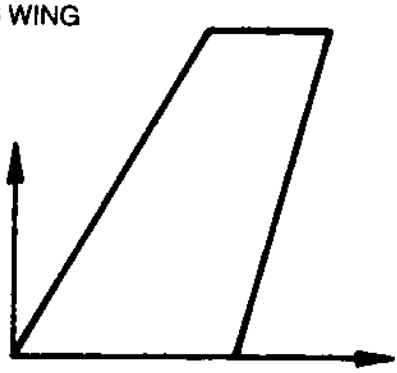
Subcritical surface panel solutions by the present method have been well validated against AURORE, an industry code at the Aeronautical Development

RAE Wing A



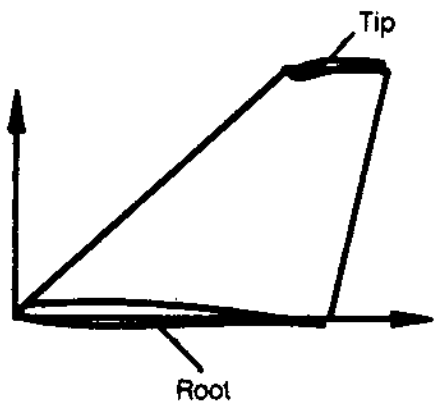
AR = 6
Λ = 36.65°
ATE = 22.34°
λ = 0.3
t/C = 0.09
Symmetric RAE 101 section
No twist

ONERA M - 6 WING



AR = 3.8
Λ_{LE} = 30°
Λ_{TE} = 15.8°
λ = 0.562
t/C = 0.1
Symmetric section
No twist

LOCKHEED Wing C



AR = 2.6
Λ_{LE} = 45°
$(t/C)_{root}$ = 0.06
$\alpha_{0,R}$ = 2.38°
λ = 0.3
Λ_{TE} = 9.7°
$(t/C)_{tip}$ = 0.1
$\alpha_{0,T}$ = -579°

Figure 10.9: Geometric Features of RAE Wing A, Onera M-6 Wing and Lockheed Wing C

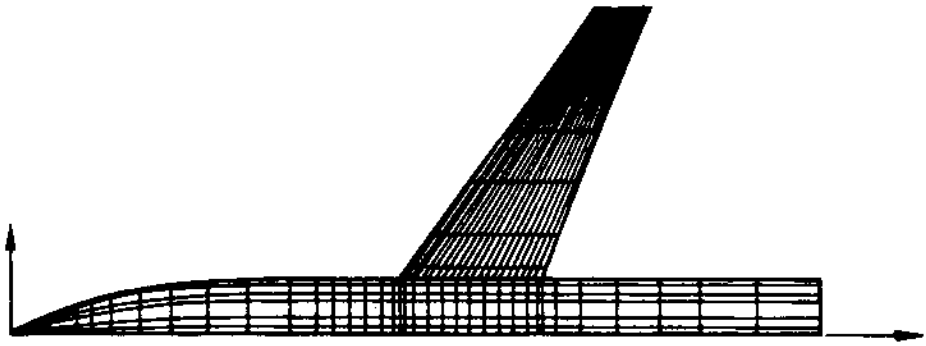


Figure 10.10: Surface Panelling on RAE Wing-Body Combination

Agency (ADA), Bangalore, for various configurations. Here we present results for two such computations.

As a first example we consider wing-body combination of a Medium Transport Aircraft (MTA), a geometry supplied by ADA, at free stream Mach number of 0.5 and zero incidence. This configuration uses a total of 1210 surface panels of which 594 are on the wing. Figure 10.12 shows the pressure distribution on the wing in the presence of the body, and for the wing alone case. Figure 10.13 shows pressure distribution on the body in combination with the wing and for body alone.

Figure 10.14 shows results of surface pressure distribution on the wing sample multi-body configuration - a launch vehicle with strap-on boosters, at free stream Mach number of 0.4 and zero incidence. This configuration uses typically 1192 surface panels.

Supercritical Transonic Solutions

As a first example of transonic flow computation, axisymmetric flow over PSLV body, a launch vehicle, is considered at free stream Mach numbers of 0.68 and 0.80. The surface pressure distributions are shown in Figure 10.15; also shown are the comparisons with experimental data (Ahmed, 1984). The comparisons show a good agreement. This example uses 66×9 surface panels and $41 \times 14 \times 28$ field cells for the half model.

Next we present results of computation on M-6 wing at $M_\infty = 0.84$, $\alpha = 3.06^o$, and at $M_\infty = 0.92$, $\alpha = 0^o$, (Bhattacharya and Arora, 1992). Computations were carried out with 78×13 surface panels, and $50 \times 14 \times 20$ field cells. The surface pressure distributions are presented in Figures 10.16 and 10.17, and compared with experiments (Schmitt and Charpin, 1979) and an Euler solution (Sacher, 1985). The major flow features show good agreement with experimental data as well as the Euler solution. The ability of the present scheme to capture strong shock is well demonstrated. The leading edge peaks appear not to be fully resolved, a feature common to other integral field solutions as well, see Erickson *et al.* (1987) and Sinclair (1988). Euler computations when

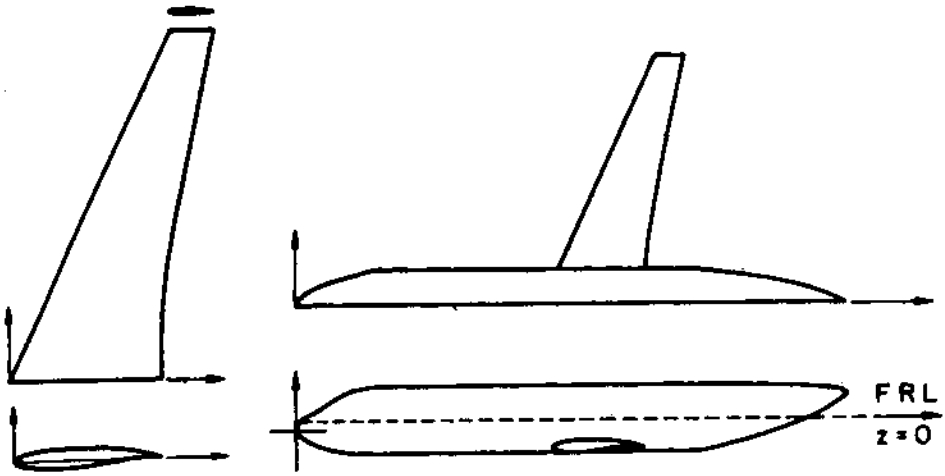


Figure 10.11: MTA Wing and Wing-Body Configuration

carried out on a medium grid suffer from similar problems, see Sacher (1985). The cell size used in our field computations may be described as medium at best, tending to coarse.

The observed agreement in location and strength of the Euler shock may be attributed to the quasi-conservative nature of the present hybrid scheme. The finite volume evaluation of the field potentials is a conservative process, which tends to locate the shock downstream of the Euler shock, whereas the integral evaluation of the surface potentials by the application of Green's identity to the Poisson-equation form of the FPE is a non-conservative process, which has the effect of shifting the shock location upstream.

As a further example flow over MTA wing, designated as TW 010, is analyzed at $M_\infty = 0.76$ and $\alpha = 0.5^o$, using 66×13 surface panels and $55 \times 16 \times 20$ field cells. The results for pressure distribution are shown in Figure 10.18, along with an Euler solution obtained at NAL Bangalore. Since the surface wraparound panels between the two spanwise stations considered for the present computation is meagre, the solution does not appear to have completely developed. However the shock location is well marked, though somewhat smeared due to the field cell size being coarse.

Transonic flow over RAE wing-body configuration is considered at $M_\infty = 0.8$, $\alpha = 2^o$, and $M_\infty = 0.9$, $\alpha = 0^o$, using 50×8 surface panels on wing, 74×7 surface panels on body, and $55 \times 14 \times 20$ field cells. The results for pressure distribution are shown in Figures 10.19 and 10.20 for the wing, and

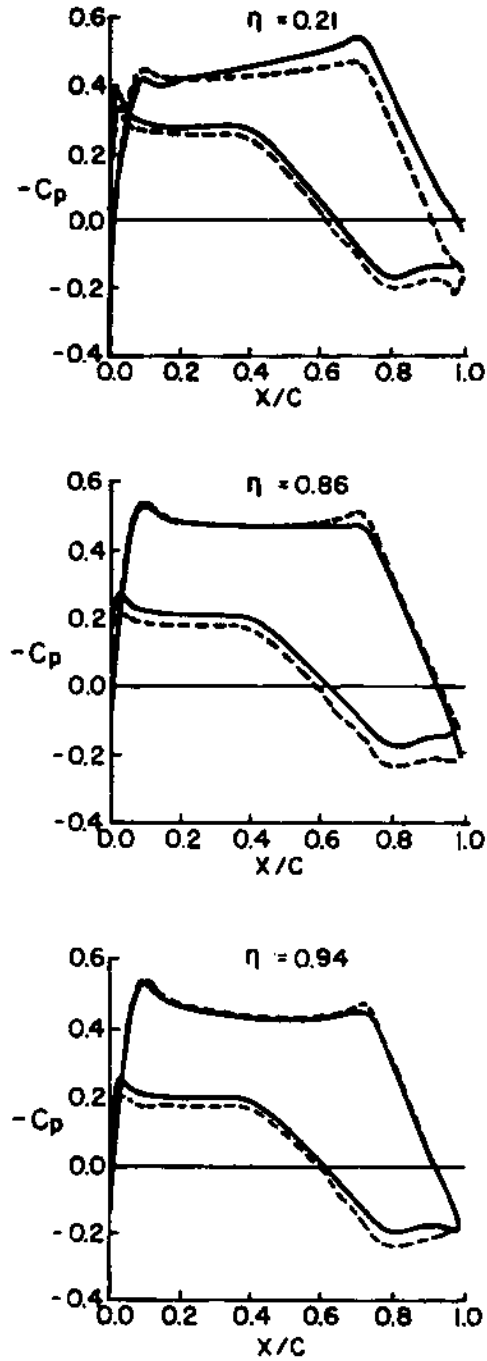


Figure 10.12: Pressure Distribution on the Wing (MTA) $M_\infty = 0.5, \alpha = 0°$; The dashed line corresponds to wing alone and the solid line corresponds to wing-body combination.

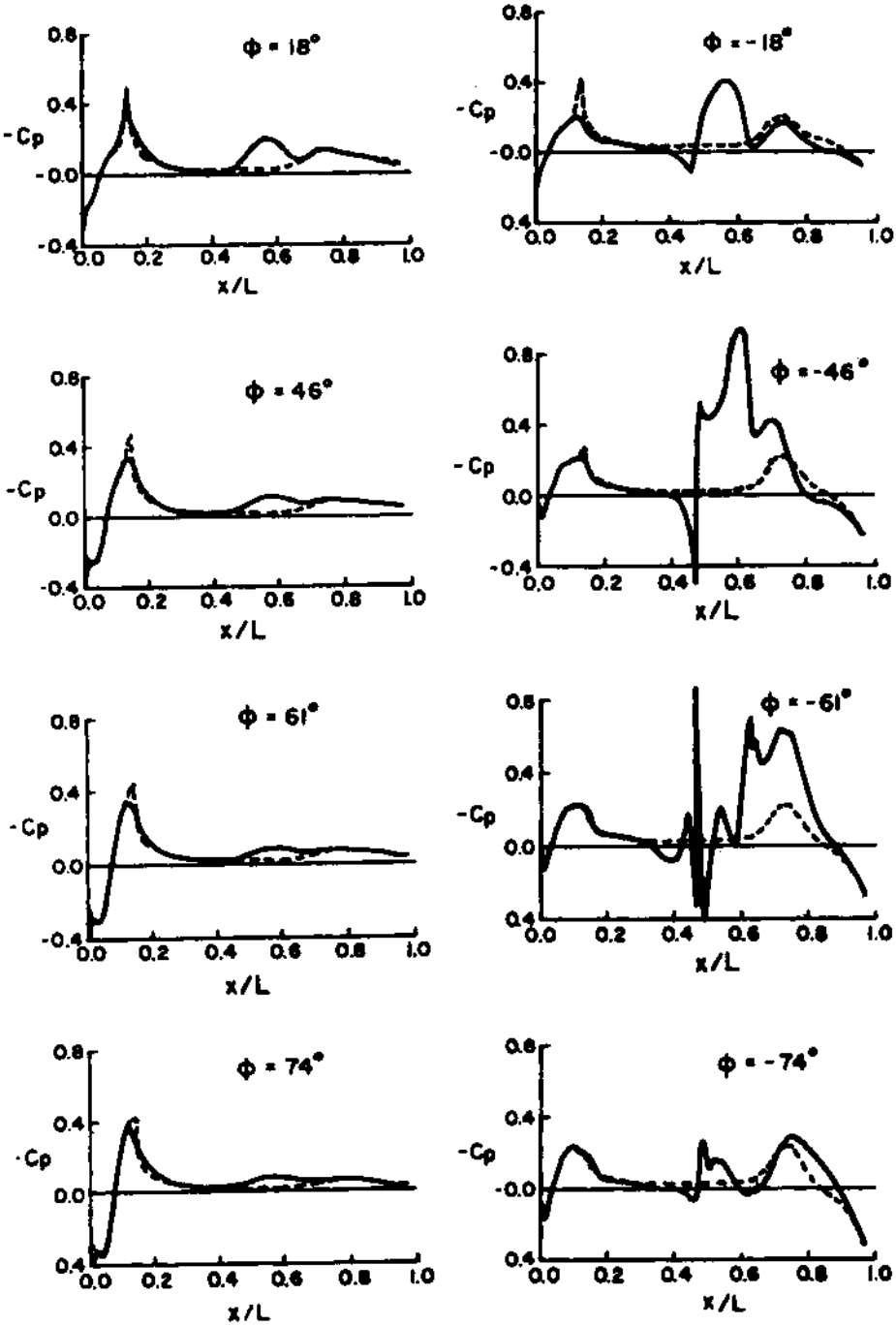


Figure 10.13: Pressure Distribution Along the Body (MTA) $M_\infty = 0.5, \alpha = 0°$; The solid line corresponds to wing-body combination and the dashed line to the body alone.

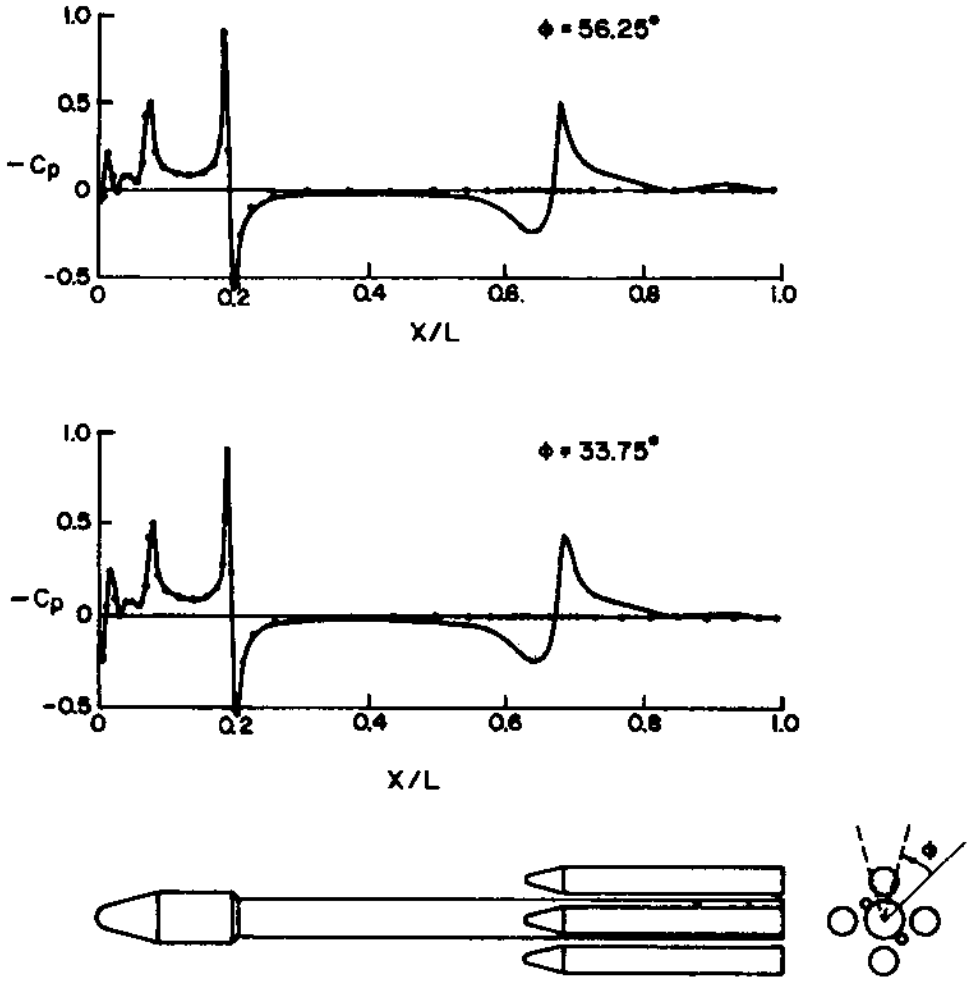


Figure 10.14: Pressure Distribution on Multibody Configuration $M_\infty = 0.4, \alpha = 0°$; The solid line corresponds to multibody and the crosses to core body alone.

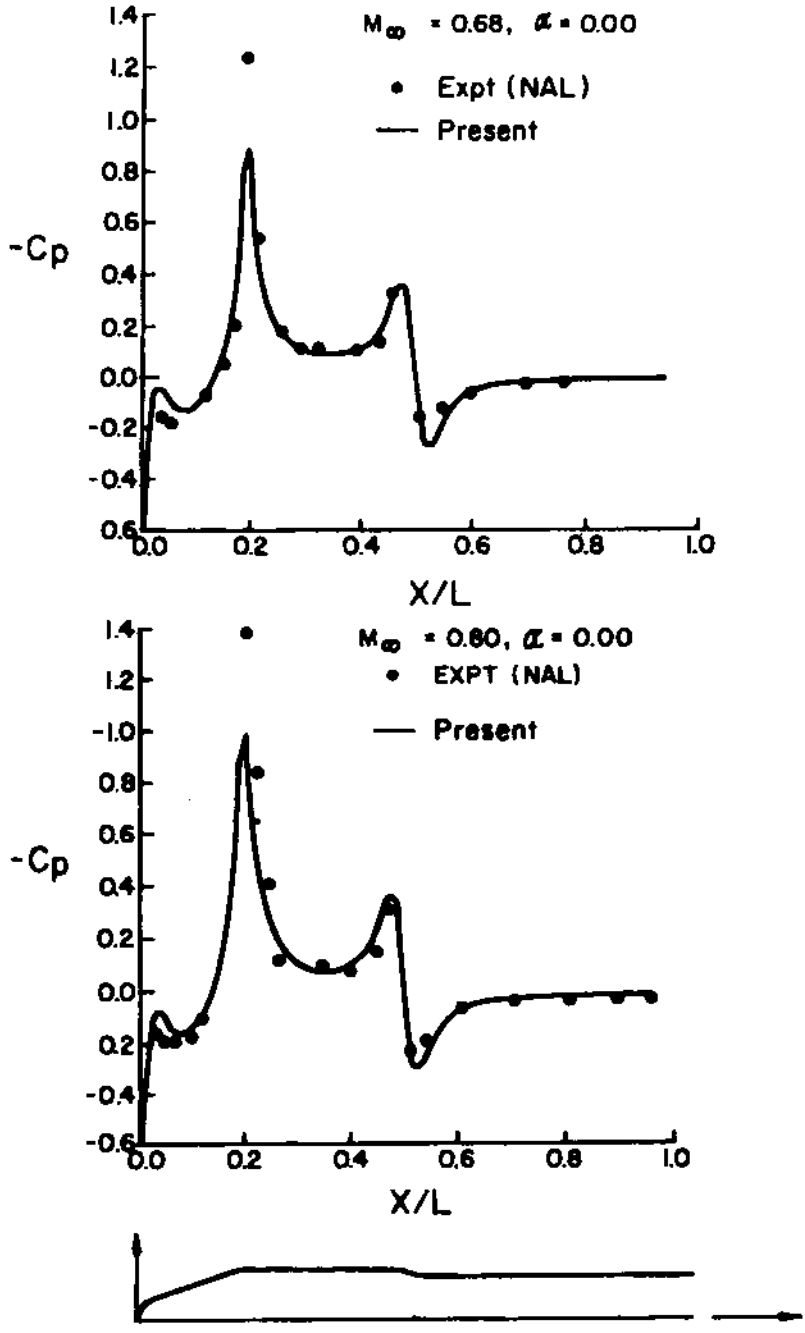


Figure 10.15: Pressure Distribution over Launch Vehicle $M_\infty = 0.68, \alpha = 0^o$

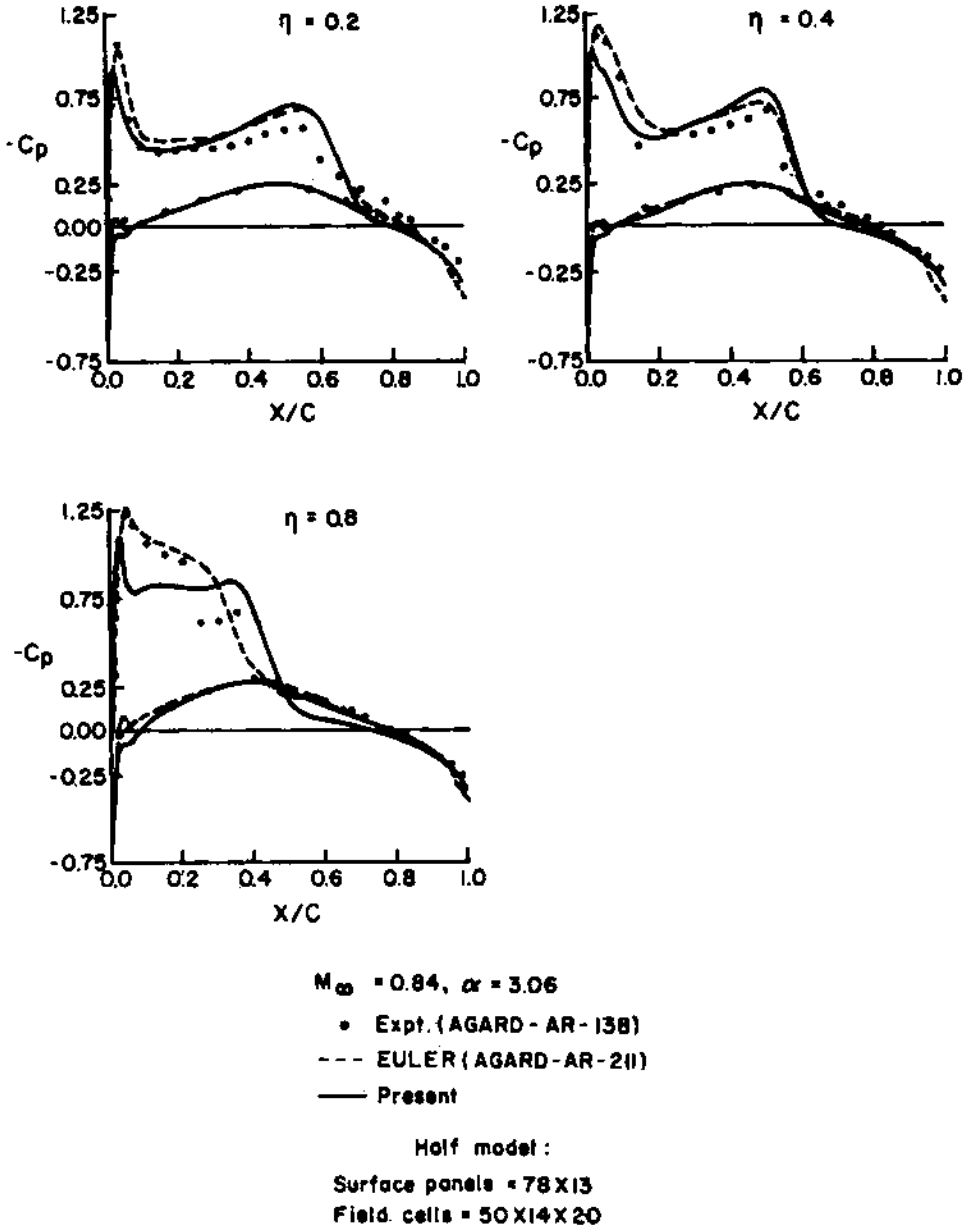


Figure 10.16: Pressure Distribution over M6 Wing

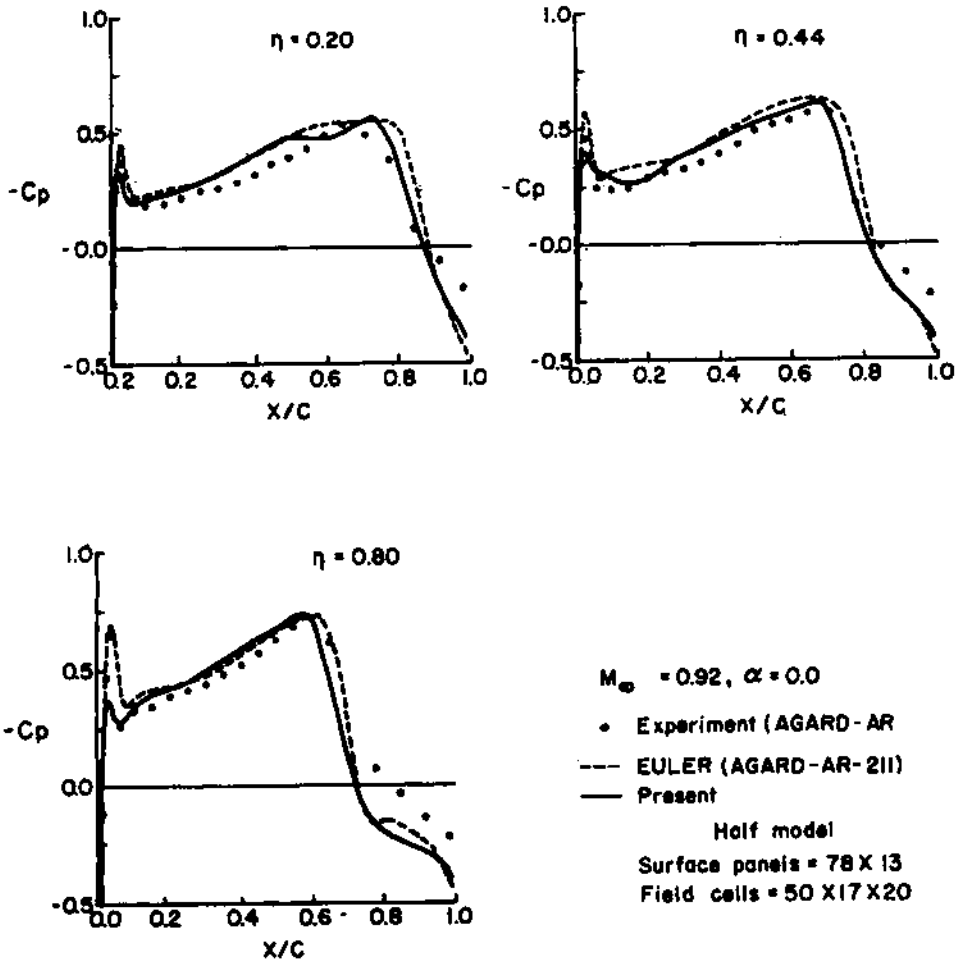


Figure 10.17: Pressure Distribution over M6 Wing

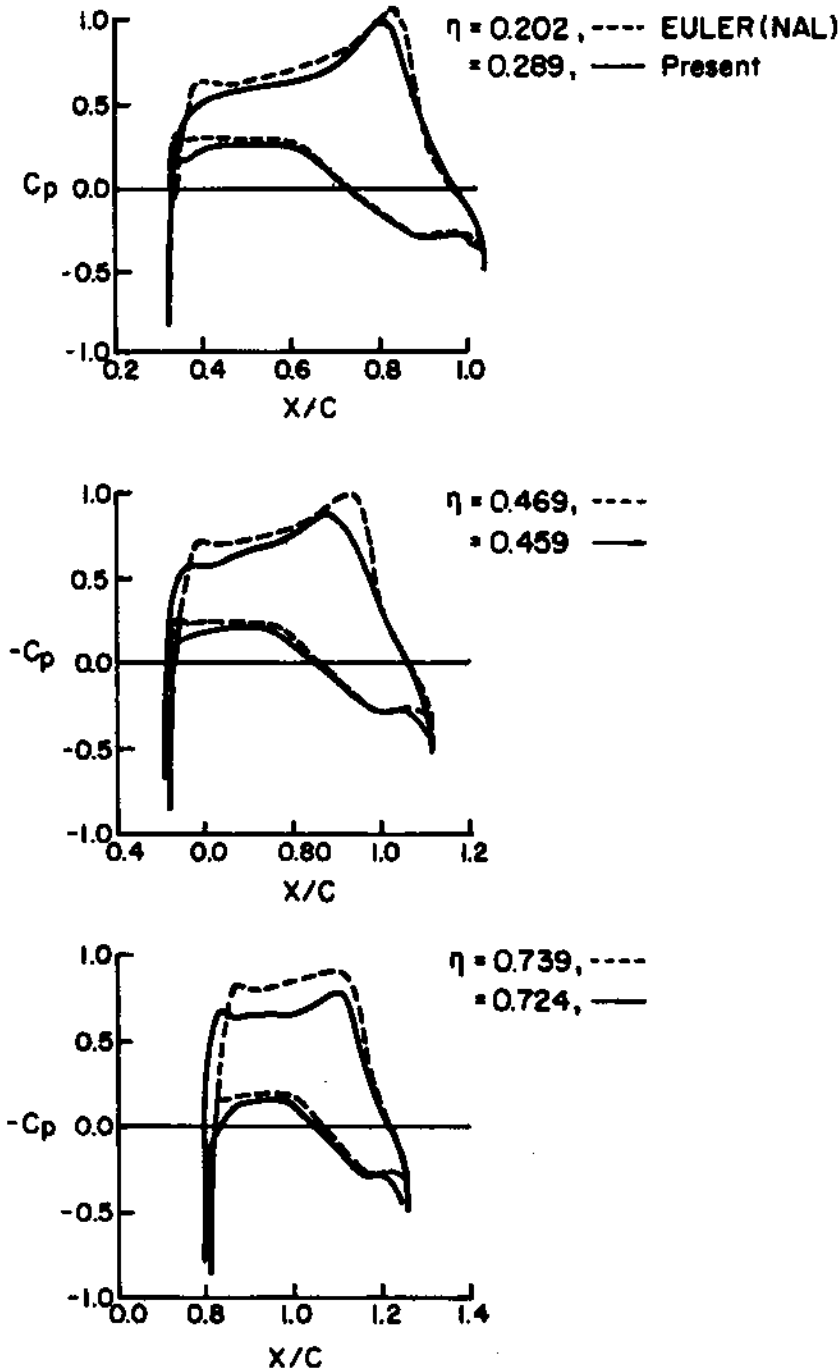


Figure 10.18: Pressure Distribution over MTA Wing $M_\infty = 0.76, \alpha = 0.5^o$

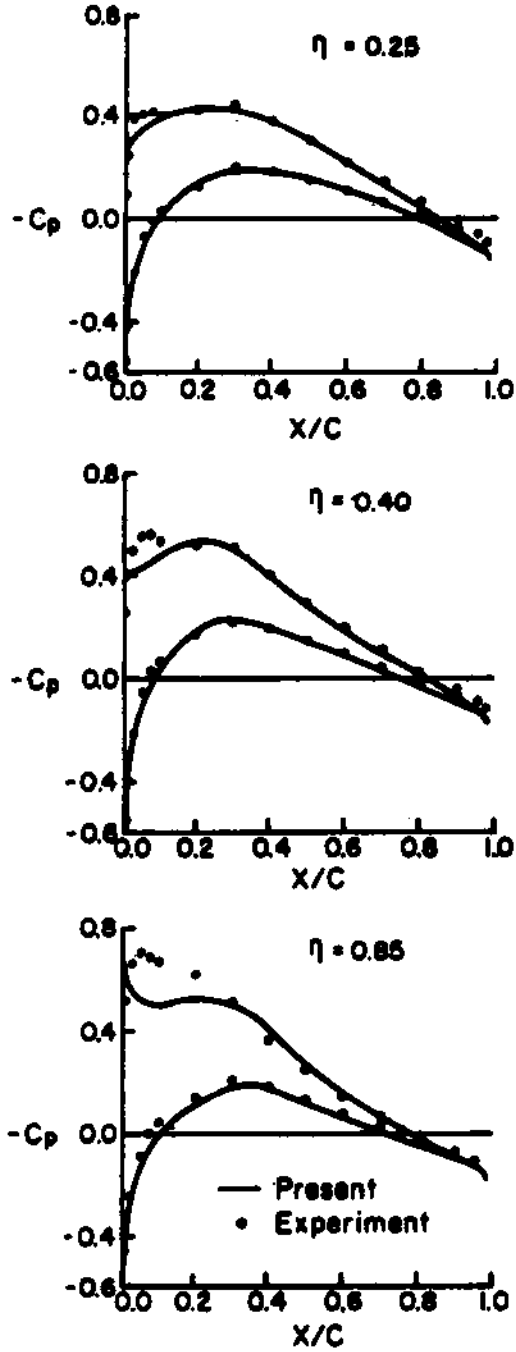


Figure 10.19: Pressure Distribution over RAE Wing for Wing-Body Combination $M_\infty = 0.84, \alpha = 3.06^o$

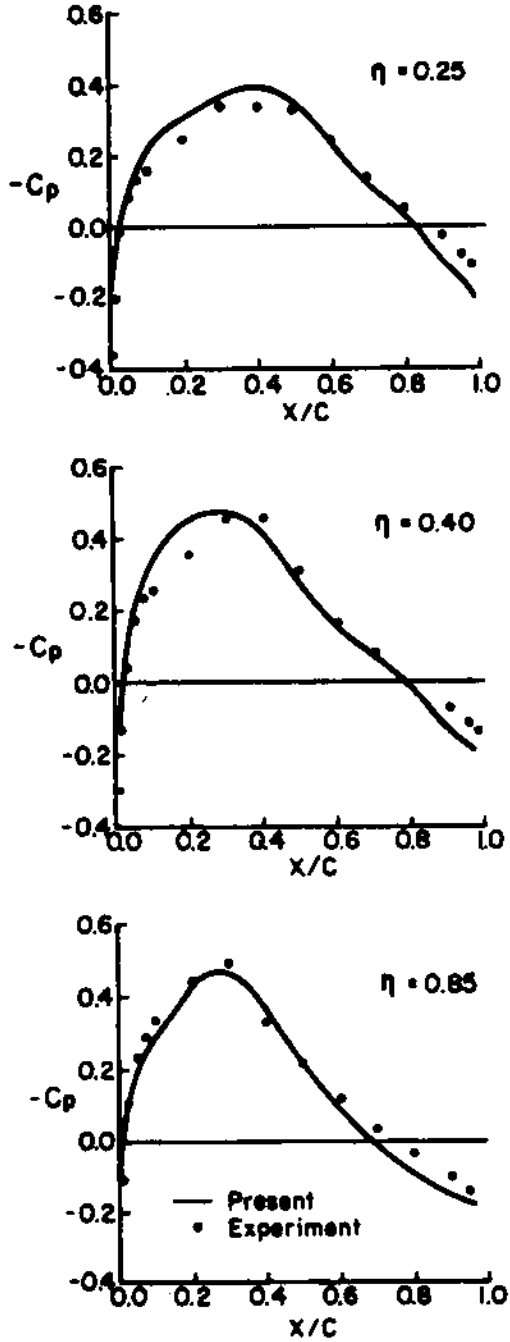


Figure 10.20: Pressure Distribution over RAE Wing for Wing-Body Combination $M_\infty = 0.92, \alpha = 0^o$

Figures 10.21 and 10.22 for the body, in comparison with experiment (Schmitt and Charpin, 1979). The major flow features show good comparison, though the loss in leading edge peak resolution in the lifting case is again observed.

Finally to test the versatility of the scheme, a highly cambered and twisted wing, Lockheed Wing C, was considered at $M_\infty = 0.85$ and $\alpha = 5^o$, with 12×66 surface panels and $55 \times 15 \times 22$ field cells. The results are presented in Figure 10.23, and compared with large scale results of Keener (1985). Here again the major flow features are reproduced well. The loss in leading edge region and near the wing tip are noticeable as in the other cases discussed above. The surface pressures from the present computations appear to compare more favourably with experiment (Keener, 1985) than those of FLO-22 and FLO-27 potential computations, as presented in Hinson and Burgess (1980).

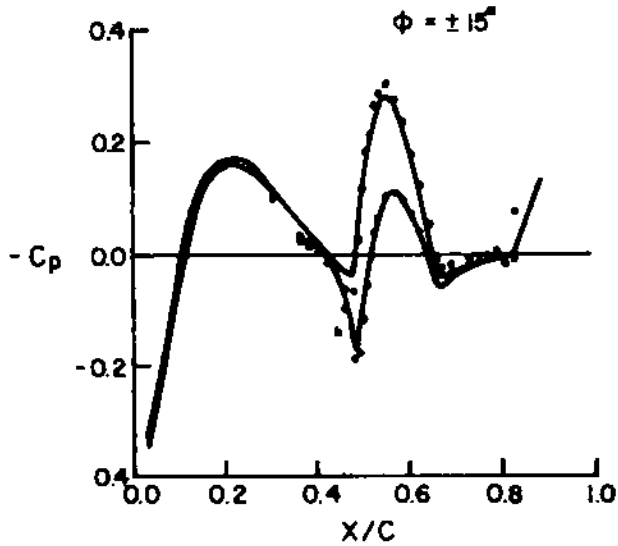
A common noticeable feature in the above computations is, a gradual deterioration of the solution as the wing tip is approached. This may be attributed to the reduction in field resolution on the wing surface in the spanwise outboard direction, due to rectangular field grid alignment on a tapered wing. Further it may be noted that for the computations of the present work, the wing tip has been left open.

10.5 Closure

The computational scheme presented above, is based on the integral equation (IE) method. It is capable of treating steady transonic flow past three dimensional configurations, for a subsonic free stream. The ability of the scheme, with a linear panel method-like surface definition and with almost automatic field grid generation, makes it a suitable method for analyzing transonic flow over complex configurations. It uses the same configuration format as that of a panel method. However, since the nature of transonic flow makes it much more sensitive to both surface curvature and slope than a subcritical flow, the present scheme will require a significantly large number of panels for surface discretization than a typical panel method for more accurate flow resolution. Also to resolve the leading edge suction peaks and the shocks, much smaller field cells are needed. This can make the calculations computationally intensive.

The scheme presented is limited to subsonic freestream Mach numbers $M_\infty < 1$, though local regions of supersonic flow are allowed. The formulation is of shock capturing type and the contribution of shock, as a source of disturbance, is embedded in the volume integral term.

The IE method has several advantages in comparison to the finite difference (FD) technique. In the finite difference/volume schemes, the computation of transonic flow problems usually require fine grids and large computational domains, whose outer boundary extend many chord lengths away from the configuration. Special treatment is required to satisfy the far-field conditions. The far field boundary condition, on the other hand, is automatically satisfied in the IE approach, and only a small computational domain is needed around the



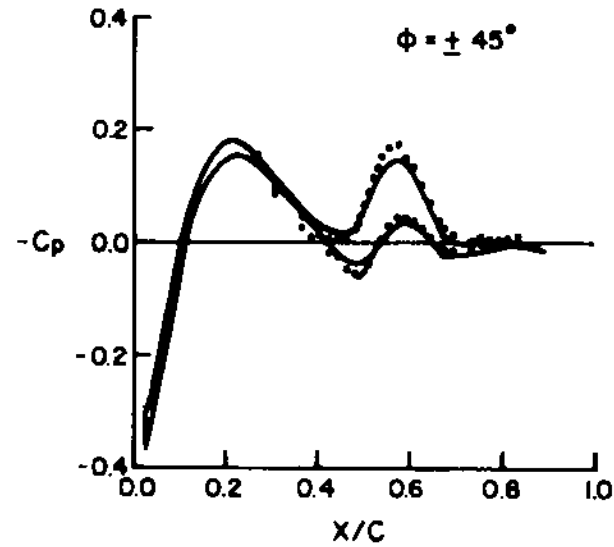
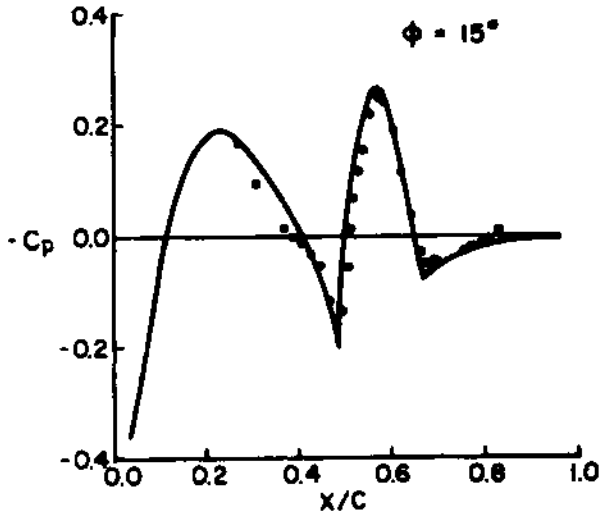


Figure 10.21: Pressure Distribution over RAE Wing for Wing-Body Combination $M_\infty = 0.84, \alpha = 3.06^o$



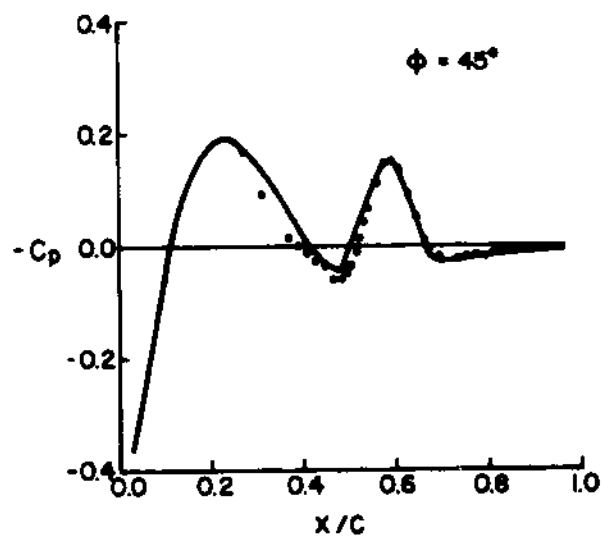


Figure 10.22: Pressure Distribution over RAE Wing for Wing-Body Combination $M_\infty = 0.92, \alpha = 0^o$

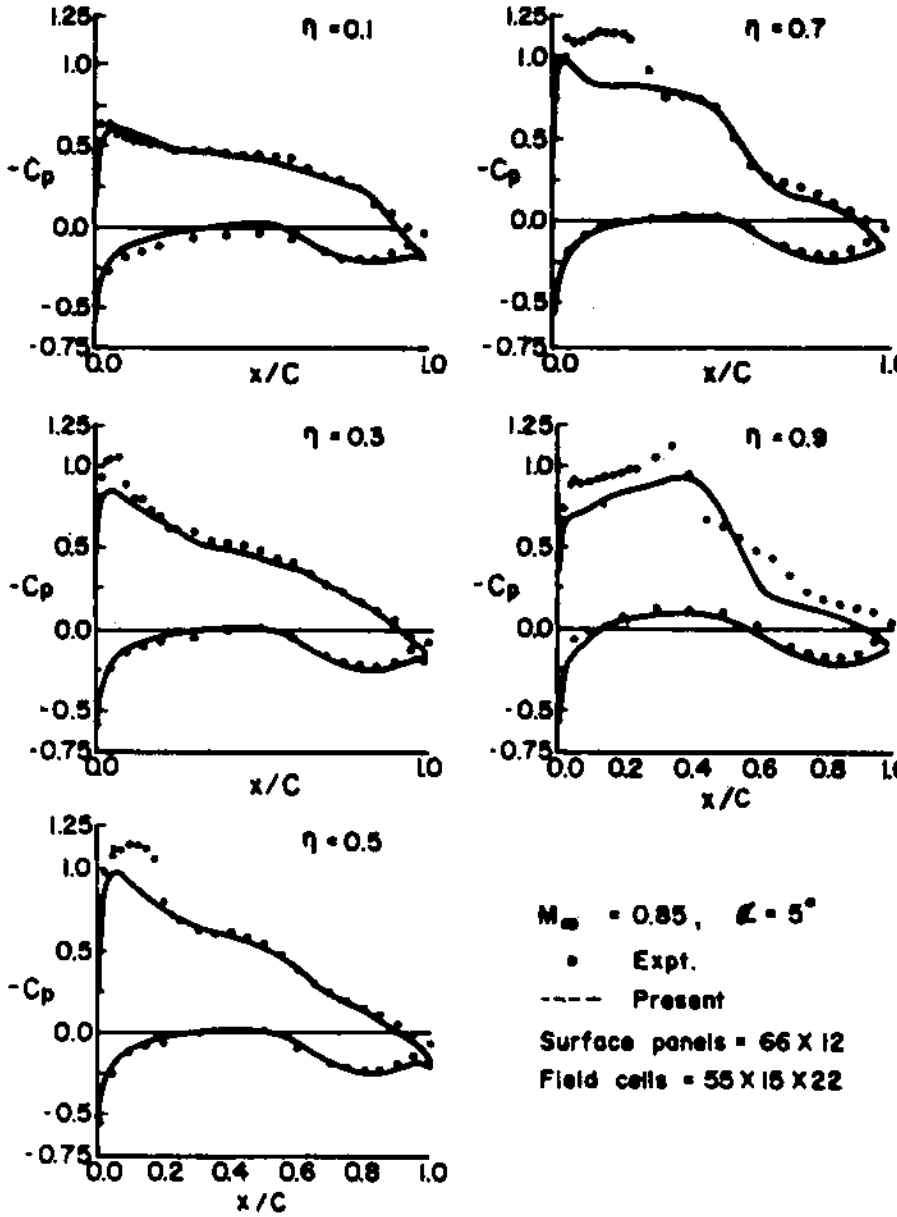


Figure 10.23: Pressure Distribution over Lockheed Wing C

source of disturbance. The accuracy of the method depends on the evaluation of integrals rather than the derivatives, and hence coarser grids can be adopted.

Finally in the IE approach the surface grid (panelling) is independent of the field grid, and the entire problem of surface conforming grid is avoided. The simplicity of piecewise constant singularity panels offers great flexibility for application to complex configurations; since continuity of singularity distribution is not enforced from panel to panel, the assembly of panels representing a complicated surface is fairly straightforward. Further the low computing cost of this method makes it attractive for applications to nonlinear flow problems requiring an iterative procedure.

The computational scheme developed appears efficient in terms of the overall iteration count as compared with the other existing schemes that use finite difference/finite volume methods throughout the large computational domain on a fine grid. However the major advantage of the scheme attempted here is its inherent capability to treat the complex configurations with the same ease as a simple 3-D wing.

In addition to applications to more complex situations, the scheme can be effectively used as a computational tool to analyse the various candidate wind tunnel test section models. The boundary condition generality necessary to model this problem (for example, confined boundaries) can be easily supplied.

References

Ahmed, S., Pressure Measurements in the Heat shield Region of PSLV Model at transonic Speeds, NAL TWT-1-33-84, August 1984.

Arora, N.L. and Agarwal, J.P., Three Dimensional Transonic Full Potential Solution by an Integral Equation Method, ICAS Paper -90-7.10.4. Proceedings of the 17th Congress of the International Council of the Aeronautical Sciences, Stockholm, Sweden, Sept. 1990, pp. 2055-2064.

Baker, T.J., Computations of Transonic Potential Flow, in Computational Methods for Turbulent, Transonic, and Viscous Flows, (Ed.) J.A. Essar, Hemisphere Pub. 1983, pp. 213-289.

Bhattacharya, A.K. and Arora, N.L., A Hybrid Integral Equation Finite Volume Scheme for Transonic Potential Flow Past Aribtrary Configurations, Proceedings of the 5th Asian Congress of Fluid Mechanics, Taejon, Korea, August 1992, pp. 618-621.

Bhattacharya, A.K. and Arora, N.L., A Hybrid Integral Equation Finite Volume Scheme for Transonic Potential Flow Past Complex Configurations, Aeronautical Journal of the Royal Aeronautical Society, London, Vol 98, No 972, pp 35-48, 1994.

Erickson, L.L., Madson, M.D., and Woo, A. C., Application of the TranAir Full Potential Code to the F-16A. AIAA Journal, Vol. 24, No. 8, August 1987, pp. 540-545.

Erickson, L.L. and Strande, S.M., A Theoretical Basis For Extending Surface-Panelling Methods to Trasonic Flow. AIAA Journal, Vol. 23, No. 12, Dec. 1985, pp. 1860-1867.

Hinson, B.L. and Burgess, K.P.: Acquisition and Application of Transonic Wing and Far-Field Test Data for Three Dimensional Computational Method Evaluation. AFOSR TR-80-0421, March 1980.

Holst, T.L., Fast, Conservative Algorithm for solving the Transonic Full Potential Equation. AIAA Journal Vol. 18, No. 12, pp. 1431-1439, 1980.

Johnson, F.T., A General Method for the Analysis and Design of Arbitrary Configurations in Incompressible Flows. NASA CR 3079, 1980.

Johnson, F.T., James, R.M., Bussoletti, J.E., Woo, A.C. and Young, D.P.: A Transonic Rectangular Grid Embedded Panel Method. AIAA Paper 82-0953, 1982.

Kandil, O.A. and Yates, E.C., Jr., Computation of Transonic Flow past Delta Wings - Integral Equation Approach. AIAA Paper 85-1582, 1985; also AIAA Journal, Vol 24, pp 1729-1736, 1986.

Kandil, O.A. and Hu, H., Integral Equation Solution for Transonic and Subsonic Aerodynamics, in Notes on Numerical Fluid Mechanics, Vol.21, (Eds.) Ballman, J., Eppler, R. and Hackbusch, W., Vieweg, pp. 101-109, 1988(a).

Kandil, O.A. and Hu, H., Full Potential Integral Solution for Transonic Flows with and without Embedded Euler Domains. AIAA Journal, Vol 26, No 9, pp 1079-1086, 1988(b).

Keener, E.R.: Pressure Distribution Measurements on Transonic Low Aspect Ratio Wing. NASA TM-86683, Sept 1985.

Kellog, O.D., *Foundations of Potential Theory*, Springer Verlag, Berlin 1967.

Madson, M.D. and Erickson, L.L., Towards the Routine Aerodynamic Analysis of Complex Configurations, in Applied Computational Aerodynamics (Ed. Henne, P.A.), Vol. 125, Progress in Astronautics and Aeronautics, pp. 701-751, 1990.

Maskew, B., Prediction of Subsonic Aerodynamic Characteristics: A Case for Low-Order Panel Methods, J. Aircraft, Vol.19, No. 2, Feb. 1982, pp. 157-163.

Morino, L., Chen. L.T., and Suciu, E.O., Steady and Oscillatory Subsonic and Supersonic Aerodynamics around Complex Configurations. AIAA Journal, Vol. 13, March 1975, pp. 368-374.

Moulden, T.H., *Fundamentals of Transonic Flow*, John Wiley & Sons, N.Y., 1984.

Osher, S., Hafez, M., and Whitlow, W. Jr., Entropy Condition Satisfying Approximations for the Full Potential Equation of Transonic Flow, Mathematics of Computation, Vol. 44, No. 169, Jan. 1985, pp. 1-29.

Oskam, B., Transonic Panel Method For Full Potential Equation Applied to Multi-Component Airfoils, AIAA paper 83-1855, 1983; also AIAA Journal, Vol. 23, No. 9, September 1985, pp. 1324-13.

Ravichandran, K.S., Arora, N.L., and Singh, R., Transonic Full Potential Solutions by an Integral Equation Method, AIAA Journal, Vol. 22, No. 7, July 1984, pp. 882-888.

Sacher, P., Numerical Solutions for Three-Dimensional Cases -Swept Wings, Sec. 7 in *Test Cases for Inviscid Field Methods*, AGARD-AR-211, 1985.

Samant, S.S., Bussoletti, J.E., Johnson, F.T., Burkhart,R.H., Everson, B.L., Melvin, R.G., Young, D.P., Erickson, L.L., Madson, M.D., and Woo, A.C., TRANAIR: A Computer Code for Transonic Analyses of Arbitrary Configurations. AIAA Paper 87-0034, 1987.

Samant, S.S., Bussoletti, J.E., Johnson, F.T., Melvin, R.G., and Young, D.P.: Transonic Analysis of Arbitrary Configurations using Locally Refined Grids, in Lecture Notes in Physics, Vol 323, (Eds.) Dwoyer, D.L., Hussaini, M.Y. and Voigt, R.G., Springer-Verlag, pp. 518-522, 1989.

Schmitt, V. and Charpin, F., Pressure Distribution Measured on RAE Wing 'A' in combination with an Axi-symmetric Body, Sec 4 in 'Experimental Data Base for Computer Program Assessment', AGARD-AR-138, 1979.

Sinclair, P.M., An Exact Integral (Field Panel) Method for the calculation of Two-dimensional Transonic Potential Flow Around Complex Configurations. Aeronautical Journal , June/ July, Vol. 90, pp. 227-236, 1986.

Sinclair, P.M., A Three-dimensional Field-Integral Method for the Calculation of Transonic Flow on Configurations- Theory and Preliminary Results. Aeronautical Journal, June/ July, Vol. 92, pp. 235-241, 1988.

Sinclair P.M., *Integral Methods at British Aerospace Brough*, in Computational Methods in Aeronautical Fluid Dynamics (Ed.) Stow P., Oxford, pp. 23-37, 1990.

11
Compressible Flows

11.1 Introduction

In this chapter the principles governing compressible flows and their calculation will be highlighted. Attention will be focussed mainly on computation of transonic and supersonic gas flows. However, general viscous flow computations involving turbulence modeling is not covered in the present chapter. It is to be realized that the subject area of turbulence modeling in the presence of shock wave is an area of current active research.

The amount of material in the field of computational fluid dynamics applied to compressible flows is ever increasing. During the last decade there has been a spurt of activity in developing new and robust methods and their simultaneous application from transonic to hypersonic flow regimes, especially in the context of external aerodynamics.

The equations governing compressible flows are different in character than the ones for incompressible flows (Chapter 1, Section 1.6.10). The spatial derivatives and the vanishing flow Mach numbers in incompressible flows make the governing partial differential equation of elliptic type. Whether a general point in the flow domain is elliptic or not is determined by the local Mach number. In the supercritical part of the flow the equations are hyperbolic. In this part of the flow the computations can have serious difficulty owing to the presence of shock waves, strong rarefactions, sonic lines and contact discontinuities.

Using standard finite difference schemes with a fixed stencil (as in Chapters 4 and 5), a shock can give rise to spurious oscillations near the discontinuities. This is called *Gibbs' phenomenon,* (Greenberg, 1998). The sonic line in a steady flow introduces a saddle point type singularity in the flow equations. Similarly strong rarefaction and contact surfaces with their associated density jumps cause similar problems.

For inviscid flows the presence of a sonic line within the flow field also implies the change from a subsonic elliptic differential equation to a hyperbolic differential equation. Although the transition in physical properties are smooth across this line, it is the change of equation type which can give rise to spurious

mass addition and other loss of conservation properties. The mixed character of the flow equations and its numerical handling was the cause of many failed attempts in computing transonic flows for quite some time.

Viscous compressible flows exhibit a large inviscid zone and a very narrow region where viscous effects are directly important. However, computing parameters such as the time steps are determined by this small region, thereby making the overall computation quite inefficient, if proper care is not exercised. One of the ways to alleviate this problem is to divide the flow regime into smaller subdomains, and for each subdomain different computational parameters can be chosen.

11.2 Basic Equations of Compressible Flows

The basic equations are the conservation laws for mass, momentum and energy. Additionally when a linear relationship between the stress tensor and strain rates is postulated we get the Navier Stokes equations. These are given by

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} + \frac{\partial \boldsymbol{H}}{\partial z} = \frac{\partial \boldsymbol{R}}{\partial s} + \frac{\partial \boldsymbol{S}}{\partial y} + \frac{\partial \boldsymbol{T}}{\partial z} \tag{11.1}$$

where

$$\begin{aligned}
\boldsymbol{Q} &= (\rho,\, \rho u,\, \rho v,\, \rho w,\, e) \\
\boldsymbol{F}(\boldsymbol{Q}) &= u\boldsymbol{Q} + p\,(0,\, 1,\, 0,\, 0,\, u) \\
\boldsymbol{G}(\boldsymbol{Q}) &= v\boldsymbol{Q} + p\,(0,\, 0,\, 1,\, 0,\, v) \\
\boldsymbol{H}(\boldsymbol{Q}) &= w\boldsymbol{Q} + p\,(0,\, 0,\, 0,\, 1,\, w) \\
\boldsymbol{R}(\boldsymbol{Q}) &= \mu\,(0,\, \tau_{11},\, \tau_{12},\, \tau_{13},\, \sigma_1) \\
\boldsymbol{S}(\boldsymbol{Q}) &= \mu\,(0,\, \tau_{21},\, \tau_{22},\, \tau_{23},\, \sigma_2) \\
\boldsymbol{T}(\boldsymbol{Q}) &= \mu\,(0,\, \tau_{31},\, \tau_{32},\, \tau_{33},\, \sigma_3)
\end{aligned} \tag{11.2}$$

with

$$\tau_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \lambda \nabla \cdot \mathbf{V} \tag{11.3}$$

$$\text{and} \quad \sigma_i = u_j \tau_{ij} + \frac{1}{(\gamma - 1)\, Pr} \frac{\partial}{\partial x_i} (c^2).$$

In Equation 11.2, ρ is the density, $\mathbf{V} = (u, v, w)$, is the velocity, p is pressure and e is internal energy. These are the unknowns of the flow. μ is the coefficient of viscosity, γ is the ratio of specific heats, Pr is the Prandtl number and c is the speed of sound. Working with perfect gases, one can use

$$p = (\gamma - 1) \left[e - \frac{\rho}{2} \left(u^2 + v^2 + w^2 \right) \right] \tag{11.4}$$

The above equations are valid for perfect gases and hence these equations are not valid for hypersonic flows with real gas effects.

Equation 11.1 is said to be in strong conservation form since it is in a form where the time rate of the density (unknowns) terms are related to the divergence of the flux terms. For Equation 11.1 the following homogeneity property of the flux vector holds good:

$$\begin{aligned}
\partial \boldsymbol{F} &= \frac{d\boldsymbol{F}}{d\boldsymbol{Q}}\,\partial \boldsymbol{Q} = A\,\partial \boldsymbol{Q} \\
\partial \boldsymbol{G} &= B\,\partial \boldsymbol{Q} \\
\partial \boldsymbol{H} &= C\,\partial \boldsymbol{Q} \\
\partial \boldsymbol{R} &= D\,\partial \boldsymbol{Q} \\
\partial \boldsymbol{S} &= E\,\partial \boldsymbol{Q} \\
\partial \boldsymbol{T} &= F\,\partial \boldsymbol{Q}
\end{aligned} \tag{11.5}$$

where $A, B, \ldots F$ are the Jacobian matrices. Equation 11.1 then reduces to

$$\frac{\partial \boldsymbol{Q}}{\partial t} + A\,\frac{\partial \boldsymbol{Q}}{\partial x} + B\,\frac{\partial \boldsymbol{Q}}{\partial y} + C\,\frac{\partial \boldsymbol{Q}}{\partial z} = D\,\frac{\partial \boldsymbol{Q}}{\partial x} + E\,\frac{\partial \boldsymbol{Q}}{\partial y} + F\,\frac{\partial \boldsymbol{Q}}{\partial z} \tag{11.6}$$

This nonconservative form is quite often used since it allows easy estimation of eigenvalues and eigen vectors (see Steger and Warming, 1981).

Strong conservation laws of the form (Equation 11.1) are often preferred since it is easy to construct fully conservative numerical schemes. Discontinuous solutions emerge automatically without any special treatment if the governing differential equation is in conservative form and the numerical scheme follows the conservative property (see Lax and Wendroff, 1960).

11.2.1 Flow Equations in Transformed Coordinates

In general, Equations 11.1 and 11.6 may not be solved in the Cartesian frame of reference . Due to the need to solve the flow equations around a complex shaped body, the above equations may have to be transformed to body-conforming non-uniform grids. We show a general transformed equation for two dimensional problems for illustration purposes for which the transformation is given by

$$x = x\,(\xi,\,\eta) \quad \text{and} \quad y = y(\xi,\,\eta) \tag{11.7}$$

In the transformed plane, the Navier-Stokes equations take the form

$$\frac{\partial \boldsymbol{Q}'}{\partial t} + \frac{\partial \boldsymbol{G}'}{\partial \xi} + \frac{\partial \boldsymbol{G}'}{\partial \eta} = \frac{\partial \boldsymbol{R}'}{\partial \xi} + \frac{\partial \boldsymbol{S}'}{\partial \eta} \tag{11.8}$$

where

$$\begin{aligned}
\boldsymbol{Q}' &= \boldsymbol{Q}/J \\
\boldsymbol{F}' &= \{U\boldsymbol{Q} + p\,(0,\,\xi_x,\,\xi_y,\,U)\}\,/J
\end{aligned}$$

$$
\begin{aligned}
\boldsymbol{G}' &= \{V\boldsymbol{Q} + p\,(0,\,\eta_x,\,\eta_y,\,V)\}\,/J \\
\boldsymbol{R}' &= \{\xi_x\,\boldsymbol{R} + \xi_y\,\mathbf{S}\}\,/J \\
\boldsymbol{S}' &= \{\eta_x\,\boldsymbol{R} + \eta_y\,\boldsymbol{S}\}\,/J
\end{aligned}
\tag{11.9}
$$

with

$$J = (\xi_x\,\eta_y - \xi_y\,\eta_x)$$

and

$$
\begin{aligned}
U &= (\xi_x\,u - \xi_y\,v) \\
V &= (\eta_x\,u + \eta_y\,v)
\end{aligned}
$$

Notice that Equation 11.8 is also in strong conservation form. In most applications, loss of strong conservation form is due to the appearance of source terms associated with a specific coordinate transformation, for example in the transformation to a cylindrical coordinate system.

Equations 11.1, 11.6 and 11.8 represent inviscid flows when the viscous right hand side terms are dropped. These are the corresponding Euler equations of the problem. A method based on integral forms, e.g., finite volume method is found superior because (a) it works on the physical plane and hence the problems associated with truncation errors are minimized and (b) it captures the solution with discontinuities without being affected by the nonuniqueness problem.

11.2.2 Integral Form of Governing Equations

The governing equations can be cast in integral form by integrating Equation 11.1 in the region Ω_{ijk} (with a boundary $\delta\Omega_{ijk}$) from the time step t_n to t_{n+1} as

$$
\begin{aligned}
&\int_{t_n}^{t_{n+1}} \int_{\Omega_{ijk}} \int\int \left\{ \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} \right\} dx\,dy\,dz\,dt \\
= &\int_{t_n}^{t_{n+1}} \int_{\Omega_{ijk}} \int\int \left\{ \frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{S}}{\partial y} + \frac{\partial \mathbf{T}}{\partial z} \right\} dx\,dy\,dz\,dt
\end{aligned}
\tag{11.10}
$$

Applying Gauss divergence theorem, the above simplifies to

$$
\begin{aligned}
&\int_{t_n}^{t_{n+1}} \int_{\Omega_{ijk}} \int\int \frac{\partial Q}{\partial t} dx\,dy\,dz\,dt \\
+ &\int_{t_n}^{t_{n+1}} \int_{\delta\,\Omega_{ijk}} \int \{F n_x + G n_y + H n_z\}\; ds\,dt \\
= &\int_{t_n}^{t_{n+1}} \int_{\delta\,\Omega_{ijk}} \int \{R n_x + S n_y + T n_z\}\; ds\,dt
\end{aligned}
\tag{11.11}
$$

Here (n_x, n_y, n_z) are the outward normal unit vector of $\delta\,\Omega_{ijk}$ and ds its surface element. Equation 11.11 forms the starting point for various numerical schemes, the finite volume method, for example.

11.2.3 Full Potential Equation

It is possible to simplify the Navier-Stokes equations to a form that is easy to solve and at the same time generates sufficient details of the flow field. One such simplification in which the viscous terms are omitted results in the Euler equations. This equation finds application in aerodynamic flows and also in various vortex dominated flows. For transonic flow past straight and moderate aspect ratio wings, it is possible to solve a simpler set of equations provided that the local Mach number does not exceeds 1.2 anywhere in the fluid region. The transonic regime is one of the most efficient speed regimes for sustained cruise flight of aircrafts. It can produce very high ratios of lift to drag. At the same time, flight at transonic speeds adds a new problem of wave drag which is not encountered in subsonic flight. A poor design can lead to operation with very strong shocks present which not only increase drag but also introduce the problem of unsteady load on the main lifting surfaces and control surfaces.

In steady flows past solid surfaces viscous effects are confined to the narrow boundary layer and may be neglected while calculating the lifting load and pitching moment. One can further simplify the governing equations by assuming the flow to be irrotational and adiabatic. For a two dimensional problem, the last assumption of irrotationality reduces the number of unknowns from six (ρ, u, v, p, e, c) to two (ρ, ϕ). Here ϕ is the velocity potential. Thus the number of equations to be solved is also reduced from four to one. Transonic flows that are of interest are adiabatic and hence such an assumption is quite useful.

The condition of irrotationality implies zero vorticity. In compressible flows vorticity arises from two sources:

(a) **Viscous effect** The no-slip boundary condition introduces vorticity in the field. Also the wake is another source of vorticity generation.

(b) **Strong shocks** If the shocks produced in the flow are strong, curved and vary rapidly in strength along their length, the flow behind the shock will be rotational, even if the flow ahead of it is irrotational. This is a direct consequence of Crocco's theorem (Anderson, 1986).

In transonic flows the shocks are usually curved and they vary in strength along their length. However if the maximum Mach number of the flow ahead of these shocks is less than 1.2 or so, the vorticity is small behind the shock and the flow can be assumed irrotational for all practical purposes. Thus one can introduce a velocity potential ϕ defined as $\mathbf{V} = \nabla\phi$. Substituting this is the continuity equation $\nabla\cdot\mathbf{V}$ gives

$$(\rho\,\phi_x)_x + (\rho\,\phi_y)_y = 0 \tag{11.12}$$

The above equation needs to be solved along with the energy equation for steady adiabatic flow for which

$$(\rho\,u\,h)_x + (\rho\,v\,h)_y = 0 \tag{11.13}$$

Here $h = e + p/\rho + 1/2\,(u^2 + v^2)$, is the enthalpy. Equation 11.13 implies that h is a constant along streamlines. If all streamlines originate from the same upstream reservoir, h cannot vary from one streamline to another. Thus, the energy equation for a perfect gas becomes

$$\frac{c^2}{\gamma - 1} + \frac{1}{2}\left(u^2 + v^2\right) = \frac{c_\infty^2}{\gamma - 1} + \frac{1}{2} q_\infty^2 \tag{11.14}$$

Since ρ and c are related as

$$c^2 = \frac{\rho^{\gamma - 1}}{\mathrm{M}_\infty^2}$$

where M_∞ is the Mach number of the free stream. The energy equation can be alternately written as

$$\rho = \left[1 + \frac{\gamma - 1}{2}\,\mathrm{M}_\infty\,\left\{1 - u^2 - v^2\right\}\right]^{1/(\gamma - 1)} \tag{11.15}$$

Equations 11.12 and 11.15 are two equations for the two unknowns ρ and ϕ. These are full potential equations in conservative form. It is possible to eliminate ρ from the continuity equation. One then obtains the quasi-linear form of the full potential equation, in nonconservative form as

$$\left(u^2 - c^2\right)\phi_{xx} + 2uv\,\phi_{xy} + \left(v^2 - c^2\right)\phi_{yy} = 0 \tag{11.16}$$

For both strong conservation form and quasi-linear form of the governing equation, we have used the continuity and energy equation only along with the thermodynamic relations for perfect gases. The question that automatically arises is, how is momentum conserved. We state here that in the smooth regions of flow field momentum is conserved. However, across shocks, the isentropic flow does not conserve momentum.

The momentum deficit across shocks leads to a mechanism for wave drag in potential flows. Otherwise note that the potential flow being isentropic and reversible cannot have drag.

It is known that the Navier Stokes and Euler equations admit shocks as valid solutions (Peyret and Taylor, 1983). The jump in flow properties in a steady inviscid flow may be computed using Rankine-Hugoniot relationships. The question that arises is that whether the shocks computed by solving potential flow resembles to those calculated from Rankine- Hugoniot relation? Potential flow being isentropic and hence reversible admit both compression and expansion shocks while the Rankine-Hugonoit relations do not admit expansion waves since they lead to decrease of entropy. Numerical experience shows that potential flow shows loss of momentum across a shock and these are close to real shock losses provided the Mach number ahead of shock is less than 1.2.

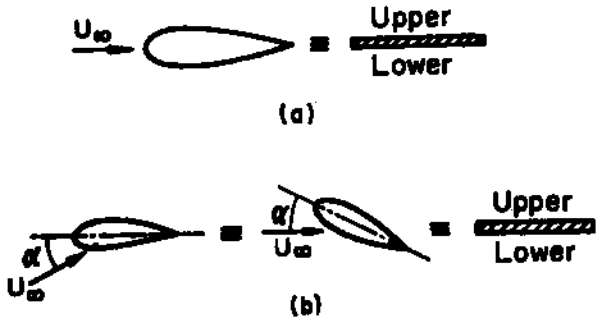


Figure 11.1: Mapping of aerofoil into a slit (a) Zero angle of attack (b) Arbitrary angle of attack

11.2.4 Transonic Small Perturbation (TSP) Equation

In this subsection, we further simplify the full potential equation for transonic flows. We make the additional assumption that the presence of the physical body in a flow field produces effects which could be termed as small perturbations. This will indeed be the case for transonic flow past thin aerofoils and bodies of revolution of small fineness.

Let us study planar flows for which the governing full potential equation is given by Equation 11.15 and either Equation 11.12 in conservative form or Equation 11.16 in nonconservative form. Since we are interested in an exterior flow problem, let the disturbance be felt upto a characteristic distance from the body. There are two ways of treating a complicated body shapes.

(a) Work with a grid system where the body itself is a grid line.

(b) Replace the body by a slit and transfer the boundary conditions on the body on the slit itself. This is shown in Figure 11.1 for the zero and non zero angle of attack cases respectively.

In the development of the *transonic small perturbation* equation the second strategy is adopted. The velocity potential ϕ has contributions from the free stream (undisturbed value) and a disturbance potential ϕ, such that

$$\phi = q_\infty x + \phi$$

when the flow is along the x-axis. The velocity components (nondimensionalised with respect to the free stream velocity) are given by, $u = 1 + \phi_x$ and $v = \phi_y$.

The required boundary condition on the body is that the flow is tangential to the boundary, i.e.

$$\frac{v}{u} = \frac{dy}{dx}$$

In terms of the disturbance potential this becomes

$$\phi_y = (1 + \phi_x)\, dy/dx$$

In line with the previous assumption of small perturbation, the disturbance velocities can be taken as negligible in comparison to the free stream velocity, such that $|\phi_x|, |\phi_y| << 1$. Thus, the boundary condition becomes

$$\phi_y \approx \frac{dy}{dx}$$

The right hand introduces the effect of angle of attack.

We now proceed to the derivation of the transonic small perturbation (TSP) equation. We define the new variables

$$x' = x, \quad y' = \delta^{1/3} y \quad \text{and} \quad \phi' = \delta^{-2/3} \phi \tag{11.17}$$

where δ is a characteristic length proportional to body thickness. With the help of these new coordinates, one can simplify Equation 11.16 to get

$$\left[\frac{1 - M_\infty^2}{\delta^{2/3}} - (\gamma + 1)\, M_\infty^2\, \phi'_{x'}\right] \phi'_{x'x'} + \phi'_{y'y'} = 0 \tag{11.18}$$

The factor $K = \left(1 - M_\infty^2\right) / \left(\delta^{2/3}\right)$ is called the transonic similarity parameter. After dropping primes, Equation 11.18 can be rewritten in the physical plane in the following form:

$$\left[1 - M_\infty^2 - (\gamma + 1)\, M_\infty^2\, \phi_x\right] \phi_{xx} + \phi_{yy} = 0 \tag{11.19}$$

This is called the TSP equation. The TSP equation defines two characteristics given by

$$(dy/dx)_{I,II} = \pm \left[-1 + M_\infty^2 + (\gamma + 1)\, M_\infty^2\, \phi_x\right]^{-1/2} \tag{11.20}$$

and we will have two real characteristics if and only if

$$\left(1 - M_\infty^2\right) < (\gamma + 1)\, M_\infty^2\, \phi_x. \tag{11.21}$$

Obviously this is satisfied in the supersonic region of the flow. Notice that in the supersonic region the characteristics are symmetric about the x-axis and any point can only be influenced by points upstream of it in the zone of influence (Courant and Friedrichs, 1967). While solving the TSP equation, one has to keep this in mind while incorporating the upstream influence in the supersonic region and correspondingly an equal upstream and downstream influence for the subsonic region.

11.3 Numerical Methods of Compressible Flow Equations

So far, we have discussed various forms of the governing differential equation for compressible flows. We are now in a position to discuss the solution procedure for each one of them. We will do it in a sequence that is roughly proportional to the computational effort needed in terms of hardware requirement.

11.3.1 Solution of TSP Equation by Finite Difference Methods

The governing equation in strong conservation form is as given by Equation 11.19 and which could be written as,

$$P_x + Q_y = 0$$

where

$$P = \left(1 - M_\infty^2\right) \phi_x - \left(\frac{\gamma + 1}{2}\right) M_\infty^2 \phi_x^2 \quad \text{and} \quad Q = \phi_y$$

Considering the node (i, j) and the eight surrounding nodes, we can write,

$$\left(\frac{\partial P}{\partial x}\right)_{ij} \cong \frac{2}{(\Delta x_{i+1} + \Delta x_i)} \left[P_{i+1/2j} - P_{i-1/2j}\right] + \text{error terms}$$

with $\Delta x_{i+1} = x_{i+1} - x_i$

The error terms are $O\left(\Delta x_i^2\right)$ and hence the above representation is of second order provided Δx_i and Δx_{i+1} are comparable in size. Thus it is important that the finite difference grid stretches smoothly and slowly. Any abrupt switch in grid spacing will result in a loss of accuracy. Similarly we can write

$$\left(\frac{\partial Q}{\partial y}\right)_{ij} = \frac{2}{(\Delta y_{j+1} + \Delta y_j)} \left[Q_{ij+1/2} - Q_{ij-1/2}\right] + \text{error terms.}$$

Thus the TSP equation in discretized form is

$$\frac{P_{i+1/2j} - P_{i-1/2j}}{\Delta x_i + \Delta x_{i+1}} + \frac{Q_{ij+1/2} - Q_{ij-1/2}}{\Delta y_j + \Delta y_{j+1}} = 0 \tag{11.22}$$

$$\text{where} \quad Q_{ij+1/2} = \frac{\phi_{ij+1} - \phi_{ij}}{\Delta y_{j+1}} \quad \text{and} \quad Q_{ij-1/2} = \frac{\phi_{ij} - \phi_{ij-1}}{\Delta y_j}$$

Similar expressions for $P_{i+1/2j}$ and $P_{i-1/2j}$ can also be written. The above discretization conserves mass in the numerical sense and is a property for all fully conservative schemes. For subsonic flow, the physical domain of influence and numerical domain of influence are identical, when we use Equation 11.22 for the solution of TSP equation.

11.3.2 Murman Cole Switching in Supersonic Regions

In a hyperbolic region the field values at any point depends on data within a domain bounded by the two characteristics which pass through that point. In this region, Equation 11.22 would be unsuitable. A new discretization must be used in place of Equation 11.22 to mimic the physical domain of influence. To simulate this in the finite difference scheme, Murman and Cole (1971) replace the centered difference approximation for ϕ_{xx} by an upwind formula of the following form:

$$\frac{P_{i-1/2j} - P_{i-3/2j}}{\Delta x_{i+1} + \Delta x_i} + \frac{Q_{ij+1/2j} - Q_{ij-1/2j}}{\Delta y_{j+1} + \Delta y_j} = 0 \tag{11.23}$$

In Equation 11.23 the P values have been shifted to the left by one node. Now the numerical domain of influence is compatible with the physical domain of influence in the supersonic region.

Equation 11.22 is used whenever the flow is subsonic and in supersonic region we switch over to Equation 11.23. Note that the discretization error for Equation 11.23 is proportional to Δx. This is a source of loss of accuracy in the solution.

Not only is there a loss of accuracy, there is loss of conservation of mass also while switching between the two discretized version of the TSP equation. This happens when we switch from Equation 11.22 to Equation 11.23 across sonic line and then across the shock. The error introduced at the sonic line is not large because the flow is smoothly accelerating through the sonic line. Across the shock, however, the switching back to Equation 11.22 results in a large spurious mass source. As a result, the numerical solution of TSP equation in the above form differs considerably from the numerical solution of the Euler equation even at a moderate Mach number. The TSP equation in nonconservative form usually produces a shock much weaker and upstream of the Euler shock. It is probably due to the fact that the mass production at the shock in the non-conservative formulation simulates the boundary layer thickening associated with adverse pressure gradient.

11.3.3 Murman-Cole Switching in Conservation Form

A remedy for the loss of conservation property of the TSP equation due to switching was proposed by Murman. The final form of the discretized equation is given below:

$$\begin{aligned} \text{Equation 11.22} \quad &+ \quad \frac{1}{(\Delta x_{i+1} + \Delta x_i)} \left[-\mu_{ij} \left(P_{i+1/2j} - P_{i-1/2j} \right) \right. \\ &+ \quad \left. \mu_{i+1j} \left(P_{i-1/2j} - P_{i-3/2j} \right) \right] = 0 \end{aligned} \tag{11.24}$$

In the above equation

$$\begin{aligned} \mu_{ij} &= 0 \quad \text{if} \quad \left[\left(1 - M_\infty^2\right) - (\gamma + 1)\, M_\infty^2\, \phi_x \right]_{ij} > 0 \\ &= 1 \quad \text{otherwise.} \end{aligned}$$

Note the following possibilities:

(i) It is easy to see that Equation 11.24 behaves identically to Equation 11.22 if points (i, j) and $(i-1, j)$ are subsonic since

$$\mu_{ij} = \mu_{i-1j} = 0$$

(ii) If points (i, j) and $(i-1, j)$ are supersonic, then

$$\mu_{ij} = \mu_{i-1j} = 1$$

and Equation 11.24 becomes identical to Equation 11.23.

(iii) Consider the case where (i, j) is supersonic and $(i-1, j)$ is subsonic. The point (i, j) is then a sonic point, and then

$$\mu_{ij} = 1 \quad \text{and} \quad \mu_{i-1j} = 0.$$

For this point the difference equation to be solved is

$$\frac{Q_{ij+1/2} - Q_{ij-1/2}}{\Delta y_{j+1} + \Delta y_j} = 0$$

(iv) Next consider the shock point where (i, j) is subsonic and $(i-1, j)$ is supersonic; then

$$\mu_{ij} = 0 \quad \text{and} \quad \mu_{i-1j} = 1$$

and the discretized equation is a sum of Equation 11.22 and Equation 11.24 as far as P - terms are concerned. The programmer all the while considers Equation 11.24 alone and all the special cases are considered automatically by appropriately defining μ_{ij}'s.

11.3.4 Region Downstream Of The Trailing Edge

If the flow to be studied is non-lifting, ϕ is continuous everywhere and Equation 11.24 is directly applied. If the flow includes a lifting object, the function ϕ becomes multivalued until a cut is introduced to make the fluid region simply-connected. We locate this cut from the trailing edge up to the far field boundary. Across the cut, it is improper to use finite differencing. Instead, we use the following finite difference representation for ϕ_y to localize the generation of vorticity:

$$\phi_y \mid_{i,\text{cut}} = \frac{\phi_{i,\text{cut}+1} - \phi_{i,\text{cut}-1} - \Gamma}{\Delta y \mid_{\text{cut}}}$$

where Γ, the circulation is

$$\Gamma = \int_{\text{body}} \phi_x dx$$

11.3.5 Far Field Boundary Conditions

Under the assumption of the small perturbation theory, the disturbance to the far field is small and decays rapidly away from the body in case of subsonic flows. If the free stream is supersonic, it is not correct to prescribe the boundary conditions at the far field. A proper approach would be to prescribe ϕ and ϕ_x at the inflow boundary (with no further specification at the outflow plane). At the outflow boundary, the application of boundary condition would amount to reflection of waves.

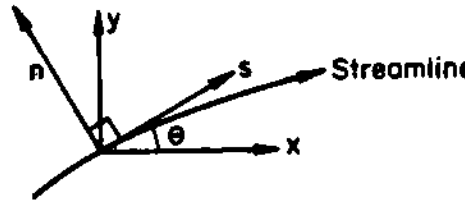


Figure 11.2: Rotated Coordinate System

11.4 Solution of Transonic Full Potential Equation (FPE)

In terms of computational difficulty, the full potential equation follows the TSP equation. The governing equation is given by Equation 11.16 in quasilinear form. The characteristics of this equation are

$$(dy/dx)_{I,II} = \frac{-uv \pm c\left[u^2 + v^2 - c^2\right]^{1/2}}{(c^2 - u^2)} \tag{11.25}$$

Thus real characteristics exist only when the local flow is supersonic, i.e. when $c^2 < u^2 + v^2$. It can be shown that the two characteristic lines in supersonic flow are oriented symmetrically about the local flow direction. Any point will then be influenced by what takes place in the upstream domain of dependence emanating from this point. Note that the characteristics are not symmetric about the reference axes, as was the case with the TSP equation. Thus any numerical scheme should be compatible with the domain of dependence in both the subsonic and supersonic region. Since the characteristics are no longer symmetric about the x-axis, a simple shift of the x-derivatives, as in the Murman-Cole scheme is not adequate. Clearly there can be a total mismatch of the physical and computational domain of dependence if the flow is not along x-axis e.g. when ϕ_y is large. In general both the x- and y-derivatives have to be upwinded while solving FPE. The precise manner in which this is done was first suggested by Jameson and is known as Jameson's rotated difference scheme (Jameson, 1974). It rotates the numerical domain of influence such that it is aligned with the physical domain of influence, regardless of the local flow direction.

For the purpose of discussing the above scheme we explain it with reference to the quasilinear form of FPE. Let us consider a new s-n coordinate system at any point P, as shown in Figure 11.2. The coordinate direction s is aligned in the local flow direction and n is the normal direction.

Here $q = \left(u^2 + v^2\right)^{1/2}$ and $\tan\theta = v/u$. Note that the new coordinate system is cartesian and not a curvilinear system and we associate it with the local flow direction. Thus θ is a constant and not a function of x and y. Since

$$s = x\cos\theta + y\sin\theta \quad \text{and} \quad n = y\cos\theta + x\sin\theta,$$

$$\phi_s = \phi_x\cos\theta + \phi_y\sin\theta$$

$$\begin{aligned} \text{and} \quad \phi_{ss} &= \phi_{xx}\cos^2\theta + 2\phi_{yx}\cos\theta\sin\theta + \phi_{yy}\sin^2\theta \\ &= u^2\phi_{xx}/q^2 + 2uv\phi_{xy}/q^2 + v^2\phi_{yy}/q^2 \\ \phi_{nn} &= v^2\phi_{xx}/q^2 - 2uv\phi_{xy}/q^2 + u^2\phi_{yy}/q^2 \end{aligned}$$

After some rearrangement, the quasilinear form of the FPE can be written as

$$\left(c^2 - q^2\right)\phi_{ss} + c^2\phi_{nn} = 0 \tag{11.26}$$

This equation has real characteristics when $\left(c^2 - q^2\right) < 0$ and has slopes of the characteristics given by

$$(ds/dn)_{I,II} = \pm c/\left[c^2 - q^2\right]^{1/2} \tag{11.27}$$

Following the same reasoning as that was used in solving the TSP-equation, Jameson recommended that the equation be symmetrically differenced about the point in subsonic regions. In supersonic regions, Jameson recommended that the terms contributing to ϕ_{ss} be shifted in the direction of the flow. The term ϕ_{nn} is still differenced symmetrically. If the flow is such that u and v are both positive, then the following finite difference formulae are used.

$$\phi_{xx} = \frac{\phi_{ij} - 2\phi_{i-1,j} + \phi_{i,j-2}}{\Delta x^2} + \bigcirc\left(\Delta x\phi_{xxx}\right)$$

$$\phi_{yy} = \frac{\phi_{ij} - 2\phi_{i,j-1} + \phi_{i,j-2}}{\Delta y^2} + \bigcirc\left(\Delta y\phi_{yyy}\right)$$

$$\phi_{xy} = \frac{\phi_{ij} - \phi_{i,j-1} + \phi_{i-1,j-1} + \phi_{i-1,j}}{\Delta x\Delta y} + \bigcirc\left(\frac{\Delta x}{2}\phi_{xxy} + \frac{\Delta y}{2}\phi_{xyy}\right)$$

The symbol $\bigcirc$ stands for an order-of-magnitude estimate. Once again these formulae result in loss of accuracy. For other combinations of flow directions similar first order expressions can be derived. Since the truncation error terms have resemblance to the viscous terms of Navier-Stokes equations, these truncation error terms are also called *artificial viscosity*. Note that

$$[\phi_{xx}]_{\text{upwind}} = [\phi_{xx}]_{\text{central}} - \Delta x\phi_{xxx}$$

$$[\phi_{xy}]_{\text{upwind}} = [\phi_{xy}]_{\text{central}} - (\Delta x/2)\,\phi_{xxy} - (\Delta y/2)\,\phi_{yyx}$$

Hence the upwind formulae for ϕ_{xx}, ϕ_{yy} and ϕ_{xy} can be written as central difference formulae plus certain first order terms. This makes it possible to write a single finite difference equation, applicable both in the subsonic and supersonic region as follows:

$$\left[\left(1-\frac{u^2}{c^2}\right)\phi_{xx}-\frac{2uv}{c^2}\phi_{xy}+\left(1-\frac{v^2}{2}\right)\phi_{yy}\right]_{\text{central}}+c^{-2}\left(1-\frac{c^2}{q^2}\right)$$
$$\times\left\{u^2\Delta x\phi_{xxx}+uv\Delta x\phi_{xxy}+uv\Delta y\phi_{xyy}+v^2\Delta y\phi_{yyy}\right\}=0 \qquad (11.28)$$

The first group of terms can be used in both the subsonic and supersonic regions, while the second group of terms should be used only for the supersonic region. The above discretization is in nonconservative form. The same equation can be cast in conservative form by multiplying Equation 11.28 by ρ and rearranging each term. First let us look at the artificial viscosity terms:

$$\frac{\mu\rho}{a^2}\left[\left(u^2\phi_{xxx}+uv\phi_{xxy}\right)\Delta x\right]+\frac{\mu\rho}{a^2}\left[\left(v^2\phi_{yyy}+uv\phi_{yyx}\right)\Delta y\right]$$

where $\mu=\left(1-c^2/q^2\right)$.

These are replaced by another set of terms in two alternate forms:

Artificial Viscosity Form 1

$$\rho\left[\mu\Delta x\left(\frac{u^2}{a^2}\phi_{xx}+\frac{uv}{a^2}\phi_{xy}\right)\right]_x+\rho\left[\mu\Delta y\left(\frac{v^2}{a^2}\phi_{yy}+\frac{uv}{a^2}\phi_{xy}\right)\right]_y$$

Artificial Viscosity Form 2

$$\left[\rho\frac{\mu\Delta x}{a^2}\left(u^2\phi_{xx}+uv\phi_{xy}\right)\right]_x+\left[\frac{\rho\mu\Delta y}{a^2}\left(v^2\phi_{xy}+uv\phi_{xy}\right)\right]_y$$

When the artificial viscosity form 1 is used in Equation 11.28 without any modifications to the first set of terms, one gets the following finite-difference equation for FPE:

$$\begin{aligned}\left[\left(1-\frac{u^2}{a^2}\right)\phi_{xx}\right. &- \left.\frac{2uv}{a^2}\phi_{xy}+\left(1-\frac{v^2}{a^2}\right)\phi_{yy}\right]_{\text{central}}\\ &+\left[\frac{\mu\Delta x}{a^2}\left(u^2\phi_{xx}+uv\phi_{xy}\right)\right]_x\\ &+\left[\frac{\mu\Delta y}{a^2}\left(v^2\phi_{yy}+uv\phi_{xy}\right)\right]_y=0 \qquad (11.29)\end{aligned}$$

The above equation is called the quasi-conservative form of the discretized full potential equation. The solution given by this equation gives extremely strong shocks and is hence not advised for use.

In Equation 11.28 multiplying the first set of terms by ρ and using the energy equation, one gets

$$\rho\left[\left(1-\frac{u^2}{a^2}\right)\phi_{xx}-\frac{2uv}{a^2}\phi_{xy}+\left(1-\frac{v^2}{a^2}\right)\phi_{yy}\right]=\left(\rho\phi_x\right)_x+\left(\rho\phi_y\right)_y$$

Along with the artificial viscosity form 2 gives the alternate equation in full conservative form

$$\begin{aligned}(\rho\phi_x)_x &+ (\rho\phi_y)_y + \left[\frac{\rho\mu\Delta x}{a^2}\left(u^2\phi_{xx} + uv\phi_{xy}\right)\right]_x \\ &+ \left[\frac{\rho\mu\Delta y}{a^2}\left(v^2\phi_{yy} + uv\phi_{xy}\right)\right]_y = 0 \end{aligned} \tag{11.30}$$

Since

$$\begin{aligned} \rho_x &= \frac{\rho}{a^2}\left(u\phi_{xx} + v\phi_{xy}\right) \\ \text{and} \quad \rho_y &= \frac{\rho}{a^2}\left(u\phi_{yy} + v\phi_{xy}\right) \end{aligned}$$

the artificial viscosity terms can be simplified to

$$-\left(\mu\Delta x u\rho_x\right)_x - \left(\mu\Delta y v\rho_y\right)_y$$

and Equation 11.30 can be written in the form

$$\left(\bar{\rho}\phi_x\right)_x + \left(\hat{\rho}\phi_y\right) = 0 \tag{11.31}$$

where $\bar{\rho} = \rho - \mu\Delta x\rho_x$ and $\hat{\rho} = \rho - \mu\Delta y\rho_y$.

Thus in supersonic regions, all that is required is to replace the physical density ρ with the artificial density values $\bar{\rho}$ and $\hat{\rho}$. Thus the method is also called the ***artificial density*** or ***artificial compressibility*** method. As long as Δx and Δy are not too large the solutions obtained by using artificial density will not be far away from the solution obtained by using the physical density.

The above forms are derived for equations written in a Cartesian frame. However the Cartesian coordinate system is not a very convenient system for treating problems involving arbitrary geometries. It is natural to use a body fitted curvilinear coordinate system such that the error in applying the boundary condition is minimized.

Let ξ and η be one such coordinate system in which $\eta = 0$ represents the body itself. Every point (i, j) in the physical plane has a corresponding point in the transformed plane except at the introduced cut, where for every point in the physical plane there are two points in the transformed plane.

Since (ξ, η) are both arbitrary and continuous functions of (x, y), we can obtain the various transformation derivatives as

$$\begin{aligned} \xi_x &= Jy_\eta \\ \xi_y &= -Jx_\eta \\ \eta_x &= -Jy_\xi \\ \eta_y &= Jx_\xi \end{aligned}$$

where $J = \xi_x\eta_y - \xi_y\eta_x$ is the Jacobian of the transformation. Let us further define the contravariant component of the velocities as

$$\begin{aligned} U &= u\xi_x + v\xi_y \\ \text{and} \quad V &= u\eta_x + v\eta_y \end{aligned}$$

In the transformed plane the continuity Equation 11.12 transforms to

$$\left(\frac{\rho U}{J}\right)_\xi + \left(\frac{\rho V}{J}\right)_\eta = 0$$

The boundary condition at the body surface is that of zero normal velocity

$$V = 0$$

which in terms of the velocity potential and its derivatives is

$$\phi_\eta = -\frac{\phi_\xi\left(\xi_x\eta_x + \xi_y\eta_y\right) + \eta_x}{\eta_x^2 + \eta_y^2}$$

Boundary Condition at the Far Field

As in most transonic flows, the conditions at the far stream is subsonic. We illustrate this condition with reference to closed two-dimensional geometries such as an aerofoil producing lift. Let the circulation generating the lift be Γ. Looked at from the far field the flow will be viewed as if a single vortex of strength Γ is located in the chordal plane within the physical geometry. Hence the far field boundary condition for lifting flows is

$$\phi = \frac{\Gamma}{2\pi}\tan^{-1}\left(\sqrt{1 - \mathrm{M}_\infty^2}\,\frac{y}{x}\right) \tag{11.32}$$

11.5 Unsteady Transonic Potential Flow

In this section, unsteady transonic flows are discussed. The governing equation are simplified by assuming the existence of a velocity potential. Even then the unsteady transonic potential flows are governed by nonlinear physical phenomena and are thus different from unsteady inviscid incompressible flows for which the governing differential equations are linear.

There are two major difficulties encountered in the study of unsteady transonic flows. Since the governing equations for unsteady transonic potential flow are nonlinear, the benefits of transform techniques applicable for linear problems can not be utilized. Every different situation has to be studied separately by some time marching technique (Chapters 4 and 5). Secondly the unsteady transonic flows are characterized by the long time lag between the motion of an aerodynamic surface and the response of flow around it.

To illustrate these difficulties, the governing equations for unsteady irrotational flows are first derived. Start with the continuity equation

$$\frac{\partial \rho}{\partial t} + (\rho u)_x + (\rho v)_y = 0 \tag{11.33}$$

On introducing the perturbation potential, it simplifies to

$$\rho_t + (\rho \phi_x)_x + (\rho \phi_y)_y = 0 \tag{11.34}$$

where suffixes such as t, x and y indicate partial differentiation, and the density is related to velocity through the energy equation as before. Let us obtain the energy equation first. The x momentum equation is

$$(\rho u)_t + \left(\rho u^2 + p\right)_x + (\rho u v)_y = 0$$

or

$$\rho\left(\phi_t\right)_x + \rho u u_x + u\left[(\rho u)_x + (\rho v)_y + \rho_t\right] + \rho v u_y + p_x = 0 \tag{11.35}$$

The terms within the square bracket are zero because of continuity. Then the above equation can then be written as

$$\left(\phi_t\right)_x + \left(u^2/2 + v^2/2\right)_x + p_x/\rho = 0 \tag{11.36}$$

where for irrotational inviscid flows

$$p_x/\rho = \frac{\gamma p_\infty^{1/\gamma}}{(\gamma - 1)\,\rho_\infty}\,\frac{d}{dx}\left(p^{1-1/\gamma}\right)$$

The momentum equation can now be integrated with respect to x, to obtain

$$\phi_t + \left(u^2 + v^2\right)/2 + \gamma p/\left(\gamma - 1\right)\rho = f(y) \tag{11.37}$$

Similarly the y momentum equation can be simplified to

$$\phi_t + \left(u^2 + v^2\right)/2 + \gamma p/\left(\gamma - 1\right)\rho = g(x) \tag{11.38}$$

Equations 11.37 and 11.38 can be combined to get

$$\phi_t + \frac{\left(u^2 + v^2\right)}{2} + \frac{\gamma p}{(\gamma - 1)\,\rho} = \text{constant} \tag{11.39}$$

The constant is easily obtained by noticing that the conditions at the free stream is steady and

$$\frac{\gamma p_\infty}{\rho_\infty} = \frac{1}{\mathrm{M}_\infty^2}$$

These simplifications lead to

$$\rho = \left[1 + \frac{\gamma - 1}{2}\,\mathrm{M}_\infty^2\left\{1 - 2\phi_t - \phi_x^2 - \phi_y^2\right\}\right]^{1/(\gamma-1)} \tag{11.40}$$

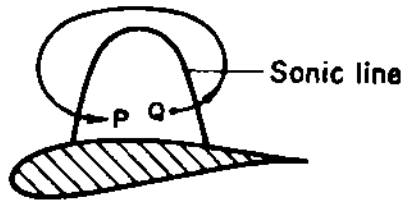


Figure 11.3: Information Transfer in Transonic Flows

This is the form of energy equation that is appropriate for the present study. Eliminating ρ from the continuity equation by using Equation 11.40 for ρ_t, ρ_x and ρ_y, the unsteady transonic full potential equation is finally written as

$$\phi_{tt} + 2u\phi_{xt} + 2v\phi_{yt} = \left(c^2 - u^2\right)\phi_{xx} + \left(c^2 - v^2\right)\phi_{yy} - 2uv\phi_{xy} \qquad (11.41)$$

This is once again in quasilinear form. To study the property of this equation, let us look at the simplified form of the above equation when ϕ only is a function of x and t. This equation is given by

$$\phi_{tt} + 2u\phi_{xt} = \left(c^2 - u^2\right)\phi_{xx}$$

whose characteristics are

$$dx/dt\,|_{I,II} = u \pm c$$

Thus the characteristics are real irrespective of whether the flow is subsonic or supersonic locally. The full equation also has the same property regardless of the local Mach number. Like any other hyperbolic equation, it needs to be solved by a time marching procedure. The above characteristics have the following physical significance. When a disturbance is created at any point, it moves in both upstream and downstream direction with reference to an observer moving with the convection speed u. These waves are called the *advancing* and *retarding* waves. While the advancing waves can move quite fast, the retarding waves move rather slowly since $u \approx c$ for transonic flows. Further both waves move downstream and there is a distinct domain of influence.

In two dimensional transonic flows, a downstream point Q can affect an upstream point only by taking an winding route, as shown in Figure 11.3. The wave first moves downstream in the supersonic pocket and once it crosses the sonic line the effects are felt everywhere as in a subsonic flow. Therefore the waves take long time to reach an upstream point causing a time lag between the imposed motion and its global effect.

11.6 Unsteady Transonic Small Disturbance Equation

In this section Equation 11.41 is simplified by using the small disturbance assumption. Under this the dimensionless velocity components have the order of

magnitude estimate

$$u = \Phi_x = 1 + \phi_x = \bigcirc(1) \quad \text{and} \quad v = \Phi_y = \phi_y \ll 1$$

The symbold Φ and ϕ have been used to distinguish respectively between the total velocity and its perturbation. Equation 11.41 has terms on the right hand side, which are not explicitly differentiated with respect to time. We note

$$\Phi_t = \phi_t; \quad \Phi_{xx} = \phi_{xx} \quad \text{and} \quad \Phi_{yy} = \phi_{yy}$$

From Equation 11.39

$$\Phi_t + u^2 + v^2 + \frac{c^2}{\gamma - 1} = \frac{1}{2} + \frac{1}{\mathrm{M}_\infty^2 (\gamma - 1)}.$$

Thus

$$c^2 = -(\gamma - 1)\,\Phi_t + (\gamma - 1)\,\phi_x + 1/\mathrm{M}_\infty^2 \tag{11.42}$$

For Equation 11.41 we can evaluate the terms under the small perturbation assumption as $u^2 \approx 1$ and v^2 as second order small quantity and uv as first order small quantity. Thus, in terms of the disturbance velocity potential, Equation 11.41 becomes

$$\begin{aligned} \mathrm{M}_\infty^2 \phi_{tt} + 2\mathrm{M}_\infty^2 \phi_{xt} &= \left[\left(1 - \mathrm{M}_\infty^2\right) - (\gamma + 1)\,\phi_x \mathrm{M}_\infty^2 \right. \\ &- \left. (\gamma - 1)\,\phi_t\right] \phi_{xx} + \phi_{yy} \end{aligned} \tag{11.43}$$

11.6.1 Boundary Conditions

At the far field, the disturbances vanish only for steady flows. For unsteady flows the disturbances travel undamped for large distances and eventually reach the far field boundary. Ideally, these disturbances should be transmitted through the computational boundary. Hence there is a need for non-reflecting type boundary condition at the far field.

On the solid boundary its own motion must be considered. Let $(\dot{x}, \dot{y})$ be the velocity components of the body at the point under consideration. Let (u, v) be the fluid velocity at the same point. Then, the slope of the body $dy/dx = (\phi_y - \dot{y}) / (1 + \phi_x - \dot{x})$. For a small velocity of the body, this leads to a small perturbation

$$\phi_y = \frac{dy}{dx} + \dot{y} \tag{11.44}$$

Finally consider the boundary conditions at the vortex sheet. In unsteady lifting flows, vorticity is continually being shed from the trailing edge and the shed wake is the vortex sheet. At the sheet, the pressure and hence density is

continuous, i.e. $\left[2\phi_t + (1+\phi_x)^2 + \phi_y^2\right]$ is continuous. Therefore neglecting the second order terms

$$[2\phi_t + 1 + 2\phi_x]_{\text{upper}} = [2\phi_t + 1 + 2\phi_x]_{\text{lower}}$$

or

$$(\phi_U - \phi_L)_t + (\phi_U - \phi_L)_x = 0$$

Since $\Gamma = (\phi_U - \phi_L)$, the above equation is simply the inviscid vorticity transport equation

$$\Gamma_t + \Gamma_x = 0 \tag{11.45}$$

The above equation states the fact that the shed vorticity is convected away at the free stream speed of unity, regardless of the local Mach number.

11.7 Nonunique Solution of Potential Equation

It has been observed that the solution obtained by solving the small perturbation and/or full potential equation gives nonunique results. It has been conjectured that this is inherent in the governing field equations themselves and certainly not due to any error associated with the solution methods (Steinhoff and Jameson, 1981), while the corresponding Euler solutions do not exhibit nonunqiueness. There is a clear need to undertake studies to see the difference of solutions obtained from potential equation and the Euler equation, specially for those parameters for which the former exhibits nonunique solutions.

For unsteady flows, the associated problem is that of very long transient decay time for specific values of Mach number, body thickness ratio and mean angle of attack.

Salas and Gumbert (1985) have computed steady flow results from potential flow theory for relatively thicker airfoil over a range of Mach number and have observed nonuniqueness only when angle of attack is sufficiently large. However, nonuniqueness is experienced at zero angle of attack only over a limited range of Mach number. They have compared solutions for flow past NACA 0012 aerofoils at $M_\infty = 0.67$ and 0.75 by both the potential and Euler equations. They conclude that that the nonuniqueness is an indicator of failure of potential theory in general. Williams *et al.* (1985) have related the nonuniqueness problem for two-dimensional flows with the instability of the computed solution. Euler equations do not have this drawback and is the topic of discussion of the next section.

11.8 Solution Of Euler Equation

In the present section we discuss methods of solving steady and unsteady Euler equation. For purely supersonic flow it is possible to solve the steady Euler

equation along the hyperbolic characteristic direction. For steady transonic flows with strong shocks it is necessary to integrate the unsteady Euler equations till the solution settles down to steady-state.

It is to be noted that for purely hyperbolic differential equations it is possible to use the method of characteristics. Even then the finite difference/finite volume methods are superior in terms of computational speed. The knowledge of characteristics however is useful in understanding many efficient numerical schemes.

11.8.1 Preliminary Considerations

To understand the above mentioned issues and related ideas, we return to the fundamentals. Let us first discuss the disturbance propagation principles in relation to the one dimensional Euler equation. For inviscid flow we use Equation 11.6 with $\mathbf{Q} = (\rho, \rho u, \rho e)^T$ and

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{3-\gamma}{2}u^2 & (3-\gamma)u & (\gamma-1) \\ -\gamma u e + (\gamma-1)u^3 & \gamma e - \frac{3}{2}(\gamma-1)u^2 & \gamma u \end{bmatrix} \tag{11.46}$$

The other Jacobian matrices are identically zero for the present one dimensional inviscid analysis. In Equation 11.46 we can replace e by the speed of sound using the perfect gas relation

$$e = \frac{c^2}{\gamma(\gamma-1)} + \frac{u^2}{2}$$

such that

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{3-\gamma}{2}u^2 & (3-\gamma)u & (\gamma-1) \\ -\frac{c^2 u}{\gamma-1} + \left(\frac{\gamma}{2}-1\right)u^3 & \frac{c^2}{\gamma-1} + \left(\frac{3}{2}-\gamma\right)u^2 & \gamma u \end{bmatrix} \tag{11.47}$$

The eigenvalues of matrix A are

$$\lambda = u, \quad u + c \quad \text{and} \quad u - c \tag{11.48}$$

Thus the Jacobian matrix A has three real eigenvalues implying Equation 11.6 to be a hyperbolic differential equation. Having established the real (distinct) eigenvalues of the Jacobian matrix A, it is easy to postulate the existence of a transformation matrix T, such that

$$T^{-1}AT = \Lambda \tag{11.49}$$

with Λ as the diagonal matrix whose entries are the real eigenvalues. Therefore one is now in a position to write down the relations along characteristics. The quasilinear form of the governing equation is

$$\frac{\partial \mathbf{Q}}{\partial t} + A\frac{\partial \mathbf{Q}}{\partial x} = 0$$

Using the similarity transformation relation 11.49, one gets

$$\frac{\partial \mathbf{Q}}{\partial t} + T\Lambda T^{-1}\frac{\partial \mathbf{Q}}{\partial x} = 0 \tag{11.50}$$

If a further simplification is made by noting that locally A is not a function of x and t, then Equation 11.50 can be alternatively written as

$$\frac{\partial}{\partial t}\left(T^{-1}\mathbf{Q}\right) + \Lambda\frac{\partial}{\partial x}\left(T^{-1}\mathbf{Q}\right) = 0 \tag{11.51}$$

Note that locally freezing the variable coefficients in A is an approach followed often for numerical solution of partial differential equation. As A is independent of x and t, so is T. Defining, $\mathbf{Q}'' = T^{-1}\mathbf{Q}$, Equation 11.51 reduces to a set of ordinary differential equation

$$\frac{d}{dz_i}q_i'' = 0 \tag{11.52}$$

where

$$\frac{d}{dz_i} = \frac{\partial}{\partial t} + \lambda_i\frac{\partial}{\partial x}$$

represents the derivative along the i-th characteristics direction and q_i'' is the characteristic variable. To obtain q_i'' we need to obtain the transformation matrix T. Since

$$T^{-1}AT = \Lambda, \quad AT = T\Lambda$$

Thus T is the left eigenvector matrix, which could be obtained easily for each λ_i. Having obtained the T matrix it is easy to construct

$$\mathbf{Q}'' = T^{-1}\mathbf{Q}$$

For the present case

$$\begin{aligned} q_1'' &= \frac{p}{\rho^\gamma} \\ q_2'' &= u + \frac{2c}{\gamma - 1} \\ q_3'' &= u - \frac{2c}{\gamma - 1} \end{aligned} \tag{11.53}$$

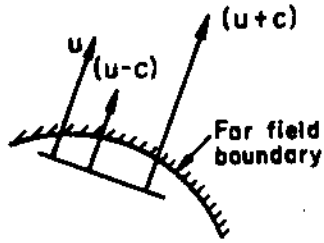


Figure 11.4: Far-field Boundary Condition ($u > c$)

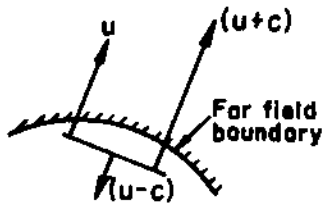


Figure 11.5: Far-field Boundary Condition ($u < c$)

These are called the Riemann invariants and Equation 11.52 simply states the fact that the Riemann invariants are constants along the respective characteristic curves.

Notice that the above procedure of *quasi-linearization* is an useful tool at the far field where the differential equation in one dimensional form can be used to set up the boundary conditions. For example, one can have the following conditions at the far field boundary.

(a) The flow is supersonic at the far field ($u > c$): Then all the signals will propagate outwards (Figure 11.4).

(b) The flow is subsonic at the far field ($u < c$): Here notice that out of three signal two are outgoing while one is propagated inwards (Figure 11.5).

A similar exercise can be performed for the two dimensional steady Euler equations. These are

$$\begin{aligned}
u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) &= 0 \\
u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} &= 0 \\
u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} &= 0 \\
u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y} + \rho c^2\frac{\partial u}{\partial x} + \rho c^2\frac{\partial v}{\partial y} &= 0 \qquad (11.54)
\end{aligned}$$

In obtaining the fourth equation of Equation 11.54, the fact that the entropy

is constant along a streamline is used, along with the equation of state

$$p = p_0 \left(p/p_0\right)^{\gamma} \exp\left(\frac{s - s_0}{c_v}\right)$$

with c_v, the specific heat at constant volume. Alternatively we can define $\mathbf{Q} = (u, v, p)$ and then eliminate ρ from Equation 11.54 to obtain the following equation

$$\frac{\partial \mathbf{Q}}{\partial x} + A\,\frac{\partial \mathbf{Q}}{\partial y} = 0 \tag{11.55}$$

where

$$A = \begin{bmatrix} uv/\left(u^2 - c^2\right) & -\rho v c^2/\left(u^2 - c^2\right) & \rho u c^2 \\ -v/\rho\left(u^2 - c^2\right) & uv/\left(u^2 - c^2\right) & -c^2/\left(u^2 - c^2\right) \\ u/\rho & 0 & v/u \end{bmatrix} \tag{11.56}$$

The eigenvalues of A can be obtained as $\lambda_{1,2,3} = (uv + c^2\beta)/(u^2 - c^2)$, v/u and $(uv - c^2\beta)/(u^2 - c^2)$ with $\beta = \sqrt{(q^2 - c^2)}/c$.

In the above, the second characteristic direction is the streamline while the other two are the Mach waves (Liepmann and Roshko, 1957). The Riemann invariants are the entropy and the left and right running Mach waves. It is worthwhile to mention at this point that neither three dimensional flows nor unsteady two dimensional flows can be reduced to characteristic form.

Currently all methods (e.g., Godunov's method (1959) and ENO schemes of Chakravarthy *et al.* (1986)) regard the multidimensional problem as consisting of that many one dimensional problems and solve one dimensional Riemann problem in each direction separately. These methods use upwinding to remove spurious oscillation where the flow variables are discontinuous. If the solution is dominated by shock and shear waves, the result would be contaminated when these waves are not aligned with the particular choice of grid. This not only makes the interpretation of the nature of waves difficult but also gives rise to spurious pressure disturbances.

Boundary Conditions at Solid Walls

The velocity boundary conditions at the solid wall are that of no penetration, namely

$$\mathbf{u} \cdot \boldsymbol{n} = 0 \tag{11.57}$$

To determine the pressure at solid walls we use the normal momentum equation approach. The governing equations in a Cartesian frame are

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0$$

where

$$\begin{aligned} \mathbf{Q} &= (\rho, \rho u, \rho v, e)^T \\ \mathbf{F} &= \left(\rho u, \rho u^2 + p, \rho uv, u(p+e)\right)^T \\ \text{and} \quad \mathbf{G} &= \left(\rho v, \rho uv, \rho v^2 + p, v(p+e)\right)^T \end{aligned}$$

The normal momentum equation is the inner product

$$(0,\, n_x,\, n_y,\, 0) \cdot \left(\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) = 0$$

Considering only the stationary wall cases, n_x and n_y will not depend on time and then the above equation can be simplified to

$$\frac{\partial p}{\partial n} = \rho \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial s} \tag{11.58}$$

11.8.2 Treatment of Convection Equation

We discuss below useful relations which are essential for developing numerical methods for the Euler equations. The Euler equations in one dimension are

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x}(\mathbf{Q}) = 0 \tag{11.59}$$

$$\begin{aligned} \text{with} \quad \mathbf{Q} &= (\rho, \rho u, \rho e)^T \\ \text{and} \quad \mathbf{F} &= \left(\rho u, \rho u^2 + p, \rho uh\right)^T \end{aligned}$$

We have already defined the Jacobian matrix A, as given by

$$A(\mathbf{Q}) = \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial \mathbf{Q}} \quad \text{with} \quad a_{ij} = \frac{\partial F_i}{\partial Q_j} \tag{11.60}$$

For hyperbolic conservation laws, a transformation matrix T was defined in Equation 11.49. We can also define a new state variable $\mathbf{V}$ by the following differential relation

$$T^{-1} d\mathbf{Q} \equiv d\mathbf{V}(x,t) \tag{11.61}$$

such that the relation 11.51 can be rewritten as a system of convection equations of the form

$$\frac{\partial V^{(k)}}{\partial t} + \lambda_k(\mathbf{Q}) \frac{\partial V^{(k)}}{\partial x} = 0, \quad k = 1, 2, 3 \tag{11.62}$$

These convection equations are coupled and nonlinear since λ_k and $V^{(k)}$ are dependent on $\mathbf{Q}$. For the corresponding linear system, we study the following scalar equation

$$\frac{\partial V}{\partial t} + c\frac{\partial V}{\partial x} = 0 \tag{11.63}$$

Let us now discuss a class of numerical schemes for Equation 11.63, which are collectively referred to as the q-scheme. The difference equation based on this scheme is written as

$$V_j^{n+1} = V_j^n - \frac{\nu}{2}\left(V_{j+1}^n - V_{j-1}^n\right) + \frac{q}{2}\left(V_{j+1}^n - 2V_j^n + V_{j-1}^n\right) \tag{11.64}$$

with $\nu = c\Delta t/\Delta x$ as the Courant number.

Following are the well known schemes in the order of increasing accuracy:

(a) Lax-Friedrichs scheme, $q = 1$

(b) Courant-Isaacson-Rees scheme, $q = \mid \nu \mid$

(c) Lax-Wendroff scheme, $q = \nu^2$

The scheme 11.64 is characterized by the following necessary condition for stability: $\nu^2 \leq q \leq 1$. Apart from the condition of stability one particular aspect of accuracy is also needed for flow calculation in the presence of shock. This property is called *monotonicity* and is related to second law of thermodynamics. We require any physical process to be associated with entropy either remaining constant or increasing with time. One is interested to know whether the numerical solution follows the exact solution for all time when the initial condition is characterized by monotonicity.

The actual solution of Equation 11.63 for $t > 0$ is just the initial value distribution shifted over a distance $c\Delta t$ (or $\nu\Delta x$). The numerical solution satisfies this condition for only integer values of ν. However the numerical solution must preserve the monotonicity of the initial distribution. This is illustrated with reference to an initial value distribution that is monotonically nondecreasing

$$V_j^n - V_{j-1}^n \geq 0 \quad \text{for all} j \tag{11.65}$$

Consider some explicit difference schemes written in compact form

$$V_j^{n+1} = \sum_{k=-l}^{+l} c_k V_{j+k}^n \tag{11.66}$$

For example, the q-schemes of Equation 11.64 gives rise to $l = 1$ and

$$\begin{aligned} c_{-1} &= \frac{\nu}{2} + \frac{q}{2} \\ c_0 &= 1 - q \\ c_1 &= -\frac{\nu}{2} + \frac{q}{2} \end{aligned}$$

One of the restriction on c_k involves satisfying the condition

$$\sum_{k=-l}^{l} c_k = 1$$

since V = constant also has to be one of the admissible solution. Using Equation 11.66 in Equation 11.65 one gets

$$\sum_{k=-l}^{l} c_k \left(V_{j+k}^n - V_{j+k-1}^n\right) \geq 0 \tag{11.67}$$

from which we derive the sufficiency condition as

$$c_k \geq 0 \quad \text{for all} \quad k \tag{11.68}$$

For the q-schemes this translates into

$$\mid \nu \mid \leq q \leq 1 \tag{11.69}$$

The upwind scheme, with $q = \mid \nu \mid$ is therefore the most accurate in terms of preserving monotonicity. It has been shown by Godunov (1959) that no convection scheme with an order of accuracy higher than one, can preserve monotonicity.

11.8.3 Riemann Problem

The model Riemann problem is used commonly in numerical simulation of unsteady inviscid flows. It was first introduced by Godunov (1959). We introduce the Riemann problem with reference to unsteady one dimensional flow of a perfect gas. The reader is advised to refer Gottlieb and Groth (1988) and the book by Courant and Friedrichs (1967) for a detailed and complete analysis.

For one dimensional unsteady flow the governing equations are given by the Euler equations, the solution of which fixes the state Q (p, u, ρ or c and the gas constants γ and R). In the Riemann problem one is interested in studying the evolution of a discontinuity prescribed as the initial condition located between two adjacent nodes of the grid as shown in Figure 11.6 As time increases, this discontinuity at $x_i + \alpha(x_{i+1} - x_i)$ with $0 < \alpha < 1$ will break into leftward and rightward moving waves which are separated by a contact surface. Across a contact surface the pressure and velocity are continuous but the density and temperature are not continuous. The waves could be either compression (shock) or rarefaction waves depending on the initial data. Four possible prototype configurations are shown below in x-t plane along with the attendant condition of initial data.

In Figure 11.7, S represents a shock, R a rarefaction wave and C the contact surface. The subscript l and r refers to left and right states respectively and the arrowhead indicates the direction of wave propagation. The first two

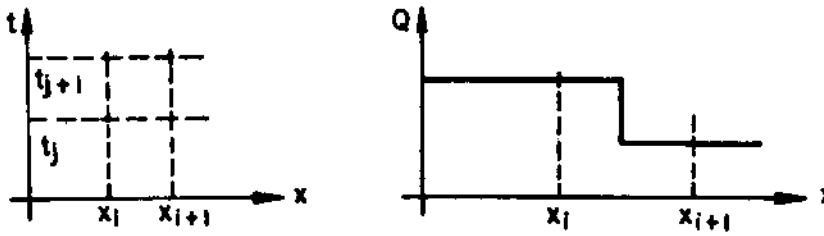


Figure 11.6: Grid Network and Initial Data $Q(x, t_j)$.

configurations are typical of shock tube flow situations. We have not indicated another wave pattern which is similar to Figure 11.7, but with two contact surfaces with vacuum between them. The Riemann problem is characterized by the absence of any characteristic length scale and hence solutions only depend on x/t. Such solutions are called *self-similar* and represented by

$$Q(x,t) = W\left(\frac{x}{t}, Q_L, Q_R\right) \tag{11.70}$$

As noted in Gottlieb and Groth (1988), "The problem of determining the types of waves, their strengths and the flow properties in each region between the waves and contact surface for some particular set of initial data is called a Riemann problem and the algorithm for determining the solution to this problem is called a Riemann solver." In Figure 11.7, there are four unknowns across the contact surface as a function of x/t. They are namely p^*, u^*, ρ_r^* and ρ_l^* with p^* and u^* the common pressure and velocity across the contact surface. These four unknowns are obtained as a solution to the three conservation laws for mass, momentum and energy supplemented by an equation of state for the perfect gas. These reduce to the Rankine-Hugoniot relations across the shock and isentropic characteristic equations across rarefaction waves.

Shock Relations

If we indicate by c_s, the shock speed, then the Rankine-Hugoniot relations are as follows:

$$\begin{aligned} \rho_r\left(u_r - c_s\right) &= \rho_l\left(u_l - c_s\right) \\ p_r + \rho_r u_r\left(u_r - c_r\right) &= p_l + \rho_l u_l\left(u_l - c_s\right) \\ h_r + \frac{1}{2}\left(u_r - c_s\right)^2 &= h_l + \frac{1}{2}\left(u_l - c_s\right)^2 \end{aligned} \tag{11.71}$$

Equations 11.71 can be manipulated to arrive at a relation between velocity and pressure as

$$u_r = u_l \pm \phi_l\left(p_r\right) \tag{11.72}$$

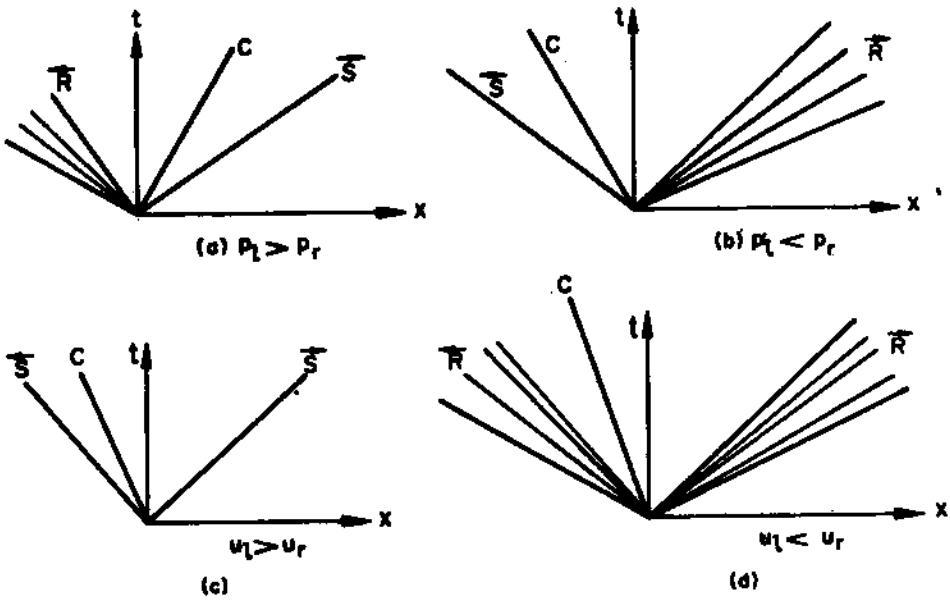


Figure 11.7: Realizable Wave Patterns as Solutions of the Riemann Problem

where

$$\phi_l(p) = (p - p_l)\left[\frac{(1-\mu^2)/\rho_l}{p + \mu^2 p_l}\right]^{1/2}$$

with

$$\mu = \left(\frac{\gamma - 1}{\gamma + 1}\right)^{1/2}$$

In Equation 11.72 the upper sign is for a backward facing shock ($p_r > p_l$) while the lower sign is for a forward facing shock ($p_l > p_r$). In either case $u_l > u_r$. Similarly we can also obtain

$$u_l = u_r \pm \phi_r(p_l) \tag{11.73}$$

where

$$\phi_r(p) = (p - p_r)\left[\frac{(1-\mu^2)/\rho_r}{p + \mu^2 p_r}\right]^{1/2}$$

Equations 11.72 and 11.73 are represented in the pressure-velocity diagram (Figure 11.8).

Rarefaction Relations

Rarefaction waves are isentropic, and the Riemann invariants can be used for the relations between left and right state

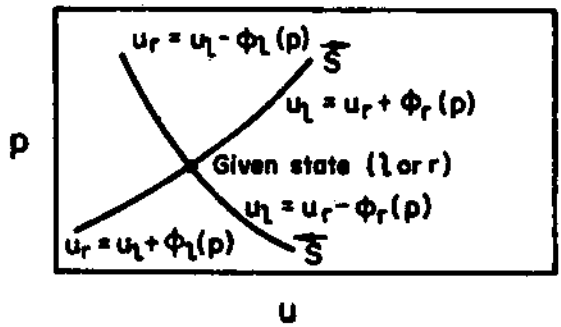


Figure 11.8: Pressure-Velocity Diagram for Shock Wave

$$u_l \pm \frac{2}{\gamma - 1} c_l = u_r \pm \frac{2}{\gamma - 1} c_r = \text{constant} \tag{11.74}$$

where the upper sign holds for forward facing waves $R \rightarrow$ and the lower sign holds for backward facing waves $R \leftarrow$. For an isentropic process

$$\frac{\rho_r}{\rho_l} = \left(\frac{p_r}{p_l}\right)^{1/\gamma}$$

Hence the Riemann invariants can be expressed as

$$\begin{aligned} u_r &= u_l \pm \frac{2}{\gamma - 1} (c_l - c_r) \\ &= u_l \pm \frac{2\sqrt{\gamma}}{\gamma - 1} \left(\sqrt{p_l/\rho_l} - \sqrt{p_r/\rho_r}\right) \end{aligned}$$

The lower sign is for forward facing rarefaction wave for which $p_r > p_l$ and the upper sign is for backward facing rarefaction wave for which $p_l > p_r$ and in either case $u_r > u_l$. The above relations can be written in compact form as

$$u_r = u_l \pm \psi_l(p) \quad \text{where} \quad \psi_l(p) = \frac{2\sqrt{\gamma}}{\gamma - 1} \frac{p_l^{1/2\gamma}}{\sqrt{\rho_l}} \left(p^{\gamma - 1/2\gamma} - p_l^{\gamma - 1/2\gamma}\right)$$

$$u_l = u_r \pm \psi_r(p) \quad \text{where} \quad \psi_r(p) = \frac{2\sqrt{\gamma}}{\gamma - 1} \frac{p_l^{1/2\gamma}}{\sqrt{\rho_r}} \left(p^{\gamma - 1/2\gamma} - p_r^{\gamma - 1/2\gamma}\right)$$

These are shown once again in the pressure-velocity diagram in Figure 11.9.

Referring to Figure 11.6, the solution of the Riemann problem involves the determination of the left state of a forward facing waves (which could itself be either a shock or a rarefaction wave) and the right state of a backward facing wave (again a rarefaction or shock wave), given the right state of the forward facing wave and left state of the backward facing wave.

Using Figures 11.8 and 11.9, we want to determine the locus of all possible left states associated with a forward facing rarefaction or shock wave with a

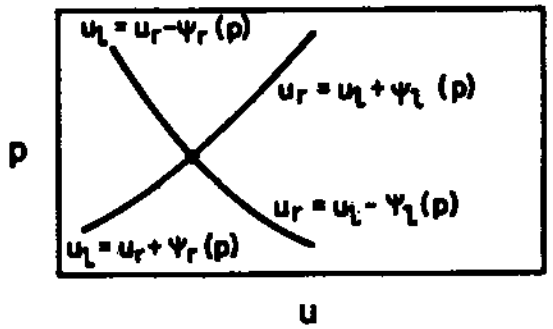


Figure 11.9: Pressure-Velocity Diagram for Rarefaction Waves

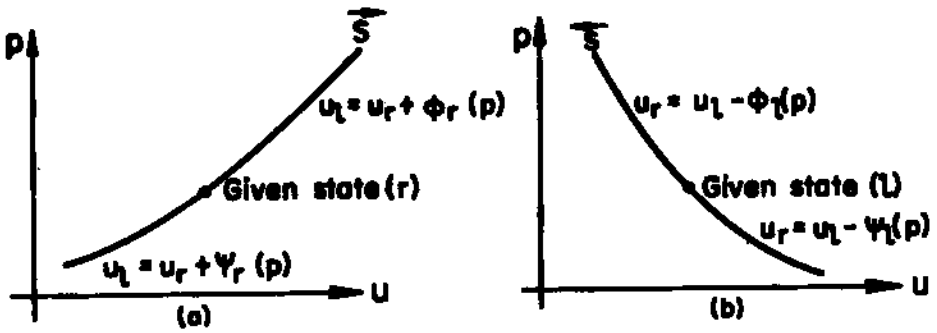


Figure 11.10: Pressure-Velocity Diagram Reconsidered

given state to the right of the forward facing wave. Similarly for a backward facing shock or rarefaction wave we would like to obtain the locus of all possible right states when the condition to the left of the backward facing wave is prescribed. The situation is depicted in the pressure-velocity diagram (Figure 11.10). Figure 11.10(a) consists of two parts depending on the following:

(a) if $p > p_r$, then the forward facing wave will be shock wave and

(b) if $p < p_r$, then we will have a rarefaction wave as the forward facing wave.

Similarly Figure 11.10(b) also consists of two parts namely,

(a) for $p > p_l$ the backward facing wave is a shock wave and

(b) for $p < p_l$ the backward facing wave is a rarefaction wave.

11.8.4 Riemann Solver for One Dimensional Euler Equation

The solution procedure for the Riemann problem is now discussed. Consider $(p_l, c_l, u_l, \gamma_l, R_l)$ and $(p_r, c_r, u_r, \gamma_r, R_r)$, the left and right states respectively at time $t = t_j$, as shown in Figure 11.6. With reference to the above discussions

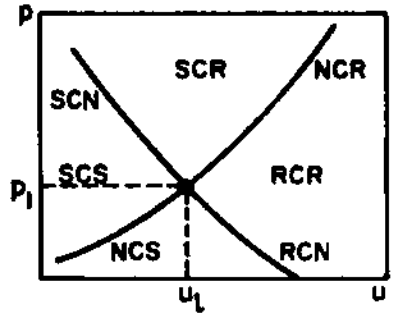


Figure 11.11: Boundaries for Riemann Problem

on shock and rarefaction relations, we represent the left state in the pressure-velocity diagram by the point (p_l, u_l) in Figure 11.11.

The right state could be anywhere else in the diagram. The label of the regions are self-explanatory, e.g. RCS represents a left moving rarefaction wave separated from the right moving shock wave by a contact surface. The curved lines are the boundaries of such regions. On the boundaries, one of the waves from the two adjacent patterns on either side of the boundary degenerates into a Mach wave. Here NCR represents a wave pattern with no wave or just a Mach wave travelling to the left and a shock travelling to the right with a contact surface between these waves. In the following, analytic expressions for the various boundaries of Figure 11.11 are obtained. See Figure 11.12(a-c).

(i) NCS boundary (Figure 11.12(a)): If we denote the intermediate state by quantities with asterisk, then $p^* = p_l$ and $u^* = u_l$. Noting that $\gamma_l \not\equiv \gamma_r$, we write down the Rankine-Hugoniot jump relations across the shock as

$$\begin{aligned} \rho_r (u_r - c_s) &= \rho^* (u_l - c_s), \\ \rho_r + \rho_r u_r (u_r - c_s) &= p_l + \rho^* u_l (u_l - c_s), \\ h_r + \frac{1}{2}(u_r - c_s)^2 &= h^* + \frac{1}{2}(u_l - c_s)^2 \end{aligned} \tag{11.75}$$

From the first two relations

$$u_r = u_l - \frac{c_r}{\gamma_r}\left(1 - \frac{p_l}{p_r}\right)\frac{c_r}{(u_r - c_s)},$$

where

$$c_r^2 = \frac{\gamma_n p_r}{\rho_r}.$$

All these relations can be further simplified to obtain

$$\frac{c_r}{(u_r - c_s)} = \pm\left\{\frac{p_l}{p_r}\frac{\gamma_r + 1}{2\gamma_r} + \frac{\gamma_r - 1}{2\gamma_r}\right\}^{-1/2}$$

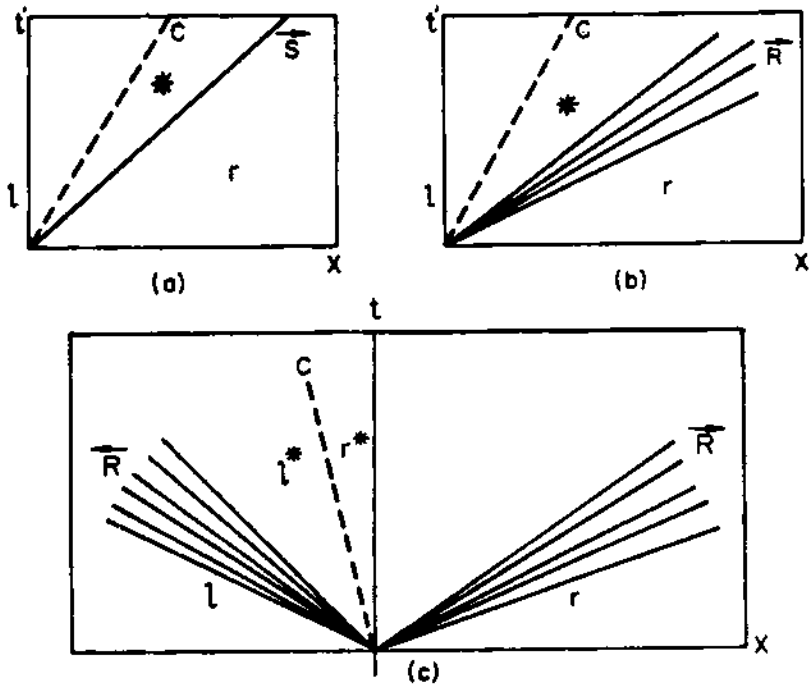


Figure 11.12: (a) NCS, (b) NCR and (c) RCR Boundaries in a Riemann Problem

The negative sign is chosen, so as to avoid a situation whereby a vacuum would have been formed when $u_r > c_s$. Using this information we obtain the velocity relation as

$$u_r = u_l - \frac{c_l}{\gamma_r}\left(\frac{p_l}{p_r} - 1\right)\left\{\frac{p_l}{p_r}\frac{\gamma_r + 1}{2\gamma_r} + \frac{\gamma_r - 1}{2\gamma_r}\right\}^{-1/2} \tag{11.76}$$

(ii) NCR boundary: The corresponding wave system is depicted in Figure 11.12(b) in the physical x-t plane. In this case also

$$p^* = p_l \quad \text{and} \quad u^* = u_l$$

Note additionally that $\gamma^* = \gamma_r$, since the region r and $*$ represents the same gas and they are connected by the Riemann invariant

$$\frac{2c^*}{\gamma_r - 1} - u^* = \frac{2c_r}{\gamma_r - 1} - u_r$$

Also the isentropic relations apply between the intermediate and right state

$$\frac{\rho^*}{\rho_r} = \left(\frac{p_r}{p^*}\right)^{-1/\gamma_r} = \left(\frac{p_r}{p_l}\right)^{-1/\gamma_r}$$

On simplification one obtains the following velocity relation

$$u_r = u_l + \frac{2c_r}{(\gamma_r - 1)}\left[1 - \left(\frac{p_l}{p_r}\right)^{(\gamma_r - 1/2\gamma_r)}\right] \tag{11.77}$$

(iii) SCN boundary: The analytic expression for the SCN boundary can be obtained in a manner similar to NCS boundary as

$$u_r = u_l - \frac{c_l}{\gamma_l}\left(\frac{p_r}{p_l} - 1\right)\left[\frac{\gamma_l + 1}{2\gamma_1}\frac{p_r}{p_l} + \frac{\gamma_l - 1}{2\gamma_l}\right]^{-1/2} \quad \text{for} \quad p_r \geq p_l \tag{11.78}$$

(iv) RCN boundary: The expression for the RCN boundary is

$$u_r = u_l + \frac{2c_l}{(\gamma_l - 1)}\left[1 - \left(\frac{p_r}{p_l}\right)^{(\gamma_l - 1/2\gamma_l)}\right] \tag{11.79}$$

(v) RCR region: By now it is apparent that the solution to most Riemann problems involves an iterative procedure. If we have a wave pattern with two rarefaction waves separated by a contact surface, we can solve for the intermediate state explicitly only when $\gamma_r = \gamma_l = \gamma$. The configuration of the wave system is indicated in Figure 11.12(c) in the physical x-t plane. The results are obtained as follows. States r and r^* are connected by the Riemann invariant

$$\frac{2c_r^*}{(\gamma - 1)} - u_r^* = \frac{2c_r}{(\gamma - 1)} - u_r \equiv -\tilde{u}_r \tag{11.80}$$

Similarly the l and l^* states are also connected by the Riemann invariant

$$\frac{2c_l^*}{(\gamma - 1)} + u_l^* = \frac{2c_l}{(\gamma - 1)} + u_l \equiv \tilde{u}_l \tag{11.81}$$

Note that

$$\frac{c_r^*}{c_l^*} = \left(\frac{\rho_l^*}{\rho_r^*}\right)^{1/2}$$

since

$$u_l^* = u_r^* = u^* \quad \text{and} \quad p_r^* = p_l^*$$

Also we can use the isentropic relations

$$\frac{p_l^*}{p_l^*} = \left(\frac{\rho_l}{\rho_l^*}\right)^{\gamma} \quad \text{and} \quad \frac{p_r}{p_r^*} = \left(\frac{\rho_r}{\rho_r^*}\right)^{\gamma}$$

Therefore

$$c_r^*/c_l^* = \frac{c_r}{c_l}\,(p^*/p_r)^{\gamma-1/2\gamma} = Z \text{ (say)} \tag{11.82}$$

and

$$c_l^* = c_l\,(p^*/p_r)^{\gamma-1/2\gamma} \tag{11.83}$$

From Equations 11.80 and 11.81

$$\tilde{u}_l - \tilde{u}_r = \frac{2}{(\gamma-1)} c_l^* \left(1 + \frac{c_r^*}{c_l^*}\right)$$

Using Equations 11.82 and 11.83 in the above

$$p^* = p_l \left[\frac{\gamma-1}{2}\,\frac{\tilde{u}_l - \tilde{u}_r}{c_l(1+Z)}\right]^{2\gamma/\gamma-1} \tag{11.84}$$

Once again utilizing Equations 11.80 and 11.81 one obtains an expression for u_l^* in the following manner. Note that

$$\tilde{u}_r c_l^* + \tilde{u}_l c_r^* = u^*\,(c_l + c_r^*)$$

Hence

$$u^* = \frac{\tilde{u}_r + \tilde{u}_1 Z}{1+Z} \tag{11.85}$$

The expressions 11.84 and 11.85 provide explicit expressions for p^* and u^*.

There is an increased usage of Riemann solvers in various finite difference and finite volume schemes for solving viscous and inviscid flows. Some of the exact Riemann solvers are described with reference to their computational efficiency by Gottlieb and Groth (1988). Also available are various approximate Riemann solvers which result in a large saving in computer time.

An approximate Riemann solver due to Jacobs (1991) is described below. It was used to simulate transient hypervelocity viscous flow. It uses a cell-centered time dependent finite volume formulation of axisymmetric Navier-Stokes equations. The inviscid component of fluxes are computed with the approximate Riemann solver while the viscous component of fluxes are calculated using the divergence theorem. The Riemann solver is applied in a locally rotated frame of reference in which one of the velocity component is normal to the cell interface.

The solver has three stages, the first of which uses the relation 11.84 and 11.85 to evaluate p^* and u^* assuming an RCR wave system. In following this procedure, one should avoid a situation where $(\tilde{u}_l - \tilde{u}_r) < 0$, i.e. the formation of vacuum at the interface. If the pressure jump across either wave is large the guess for the intermediate pressure is modified using the strong shock relations. For example, if p^*/p_l and p^*/p_r are greater than 10, one uses the alternate form

$$p^* = \frac{\gamma+1}{2}\,\rho_l \left[\frac{\sqrt{\rho_r}}{\sqrt{\rho_r}+\sqrt{\rho_l}}\,(u_l - u_r)\right]^2 \tag{11.86}$$

and

$$u^* = \frac{\sqrt{\rho_l}u_l + \sqrt{\rho_r}u_r}{\sqrt{\rho_l} + \sqrt{\rho_r}} \tag{11.87}$$

If p^* calculated using isentropic relation is greater than p_i and p_r, the pressure estimate of the intermediate state is obtained using Newton-Raphson method as

$$p^*_{i+1} + p^*_i - F_i / \left(\frac{dF_i}{dp^*}\right)$$

where

$$F_i = u^*_l - u^*_r$$

The expressions for u^*_l and u^*_r are obtained from normal shock relations in the limit of large pressure jump

$$\begin{aligned} u^*_l &= \tilde{u}_l - \frac{2c_l}{\gamma - 1}\left(\frac{p^*}{p_l}\right)^{\gamma-1/2\gamma} \quad \text{if} \quad p^* \leq 10p_l \\ &= u_l - \sqrt{\frac{2p^*}{\rho_l^{\gamma+1}}} \quad \text{if} \quad p^* > 10p_l \end{aligned}$$

and

$$\begin{aligned} u^*_r &= \tilde{u}_r + \frac{2c_r}{\gamma - 1}\left(\frac{p^*}{p_r}\right)^{\gamma-1/2\gamma} \quad \text{if} \quad p^* \leq 10p_r \\ &= u_r + \sqrt{\frac{2p^*}{\rho_r^{\gamma+1}}} \quad \text{if} \quad p^* > 10p_r \end{aligned}$$

In the third stage, other flow properties are evaluated depending on whether p^* is greater than or less than equal to p_l and p_r respectively.

Although Riemann solvers are used for solving Euler or Navier-Stokes equation for determining the interaction of adjacent fluid cells, some problems remain for multidimensional flows. In general, even for two dimensional problems one cannot prescribe a unique wave propagation direction as in one dimensional flows. For two dimensional Euler equations one has four waves, out of which there are two nonlinear acoustic waves, a linear entropy and a shear wave. The propagation velocity is different for each of the waves and they can also propagate in an infinite number of directions. This causes a nonunique definition of upwind directions in multidimensional flow. Development of general multidimensional schemes is one of the frontier areas of research for high speed flow computation. The finite volume method is one such approach, and is introduced in teh following section.

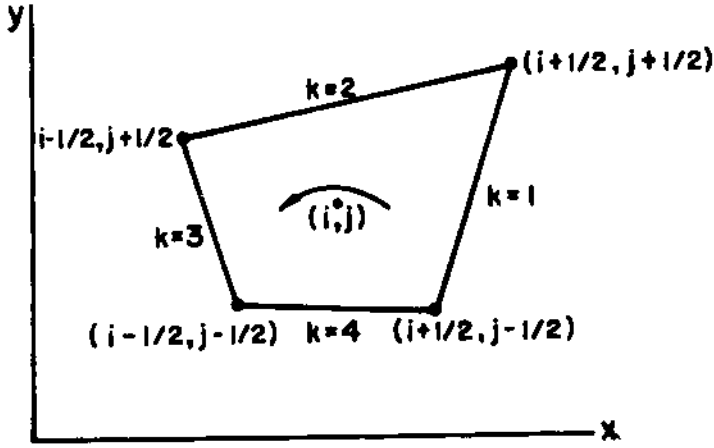


Figure 11.13: Cell Configuration in Finite Volume Method

11.8.5 The Finite Volume Method

Here the development of the finite volume method for unsteady two dimensional Euler equation in the cartesian frame of reference is illustrated. The starting point is the component form of Equation 11.10, written as

$$\int_{\Omega_{ij}} \int \frac{\partial Q}{\partial t} dV + \int_{\partial\Omega_{ij}} \int \left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} \right) dA = 0 \tag{11.88}$$

For two dimensional flows, the volume integral is replaced by an area integral. Similarly the second area integral is replaced by a line integral to yield

$$\frac{\partial}{\partial t} \int_{\Omega_{ij}} \int Q dA + \oint_i (F dy - G dx) = 0 \tag{11.89}$$

The above formulae are used for each cell. If we define a cell average by

$$\bar{Q}_{ij} = \frac{1}{\Omega_{ij}} \int_{\Omega_{ij}} \int Q \, dA$$

then Equation 11.89 can also be written as

$$\Omega_{ij} \frac{\partial \bar{Q}_{ij}}{\partial t} + \left\{ \sum_{k=1}^{n} (F\Delta y - G\Delta x)_k \right\}_{ij} = 0 \tag{11.90}$$

For an n-sided cell, (with $n = 4$ for a quadrilateral cell), the indexing scheme and the integration direction for the line integral are shown in Figure 11.13. Note

$$\Delta x_1 = x_{i+1/2,j+1/2} - x_{i+1/2,j-1/2}$$

$$\Delta y_1 = y_{i+1/2,j+1/2} - y_{i+1/2,j-1/2}$$

A positive flux term for the (i,j)th cell will have a negative counterpart in an adjacent cell, because of the fixed clockwise/anti clockwise sequence followed in evaluating the line integrals. The values F_k and G_k are the x and y direction flux averages on the kth cell face. Define

$$\Delta S_k = \left[\Delta x^2 + \Delta y^2\right]_k^{1/2}$$

The outward normal on the kth face is

$$\mathbf{n}_k = \left(\frac{\Delta y}{\Delta s}\right)_k \hat{\mathbf{i}} - \left(\frac{\Delta x}{\Delta s}\right)_k \hat{\mathbf{j}} = \sin\theta_k\,\hat{\mathbf{i}} - \cos\theta_k\,\hat{\mathbf{j}}$$

One can also define the velocity on each cell face by one component parallel to it and another normal to the cell face

$$(u_p)_k = u\sin\theta_k - v\cos\theta_k$$

$$(u_n)_k = u\cos\theta_k - v\sin\theta_k$$

where u and v are the Cartesian velocity components. Therefore the flux normal to the kth cell face is

$$(F\Delta y - G\Delta x)_k = (F\sin\theta_k - G\cos\theta_k)\ \Delta S_k$$

Returning to the vector form

$$\mathbf{F} = \left(\rho u, \rho u^2 + p, \rho uv, \rho uh\right)^T \qquad \mathbf{G} = \left(\rho v, \rho uv, \rho v^2 + p, \rho vh\right)^T$$

$$(\mathbf{F}\Delta y - \mathbf{G}\Delta x)_k \quad = \quad \Delta S_k\left(\rho u_n, \rho u_n u + p\sin\theta_k, \rho u_n v - p\cos\theta_k, \rho u_n h\right)^T \tag{11.91}$$

A flux vector splitting scheme for Equation 11.91 is described next.

Steger and Warming Flux Vector Splitting Scheme (1981)

As noted earlier, the flux functions $\mathbf{F}$ and $\mathbf{G}$ are homogeneous function of degree one with respect to $\mathbf{Q}$ and thus

$$\mathbf{F} = A\mathbf{Q} \quad \text{and} \quad \mathbf{G} = B\mathbf{Q} \quad \text{where} \quad A = \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \quad \textit{and} \quad B = \frac{\partial \mathbf{G}}{\partial \mathbf{Q}}.$$

Thus the left hand side of Equation 11.91 can also be written in the alternate form

$$\left(\mathbf{F}\frac{\Delta y}{\Delta S} - \mathbf{G}\frac{\Delta x}{\Delta S}\right)_k \Delta S_k = (A\sin\theta_k - B\cos\theta_k)\ \mathbf{Q}_k\,\Delta S_k$$

where $(A\sin\theta_k - B\cos\theta_k)$ has the following eigenvalues:

$$\begin{aligned} \lambda_1 &= \lambda_2 = u_n \\ \lambda_3 &= u_n + c \\ \text{and} \quad \lambda_4 &= u_n - c. \end{aligned} \tag{11.92}$$

The outward flux normal to the kth cell face can be expressed in terms of the above eigenvalues as

$$\begin{aligned} = \frac{\rho \Delta S_k}{2\gamma} \quad & \Big\{ 2(\gamma-1)\lambda_1 + \lambda_3 + \lambda_4, \\ & 2(\gamma-1)\lambda_1 u + \lambda_3 (u + c\sin\theta_k) + \lambda_4 (u - c\sin\theta_k), \\ & 2(\gamma-1)\lambda_1 v + \lambda_3 (v - c\cos\theta_k) + \lambda_4 (v + c\cos\theta_k), \\ & (\gamma-1)\lambda_1 (u^2+v^2) + \frac{\lambda_3}{2}\left[(u+c\sin\theta_k)^2 + (v-c\cos\theta_k)^2\right] \\ + \quad & \frac{\lambda_4}{2}\left[(u-c\sin\theta_k)^2 + (v+c\cos\theta_k)^2\right] + \frac{(3-\gamma)(\lambda_3+\lambda_4)c^2}{2(\gamma-1)} \Big\} \end{aligned}$$

For each $\lambda = \lambda_i$ we can compute

$$\lambda_i^+ = \frac{\lambda_i + |\lambda_i|}{2} \tag{11.93}$$

$$\lambda_i^- = \frac{\lambda_i - |\lambda_i|}{2} \tag{11.94}$$

Associated with $\lambda_i^{\pm}$ one can calculate the forward flux and the backward flux

$$\left(\mathbf{F}\frac{\Delta y}{\Delta S} - \mathbf{G}\frac{\Delta x}{\Delta S}\right)^+ \Delta S \quad \text{and} \quad \left(\mathbf{F}\frac{\Delta y}{\Delta S} - \mathbf{G}\frac{\Delta x}{\Delta S}\right)^- \quad \text{respectively}$$

In close analogy to the one dimensional case, the forward flux at the cell face is computed by evaluating it at Q_L state and the negative flux at Q_R state such that

$$\begin{aligned} \left(F\frac{\Delta y}{\Delta S} - G\frac{\Delta x}{\Delta S}\right) \Delta S &= \left(F\frac{\Delta y}{\Delta S} - G\frac{\Delta x}{\Delta S}\right)^+ \Bigg|_{Q=Q_L} \Delta S \\ &+ \left(F\frac{\Delta y}{\Delta S} - G\frac{\Delta x}{\Delta S}\right)^- \Bigg|_{Q=Q_R} \Delta S \end{aligned} \tag{11.95}$$

The fluxes at cell face as calculated by Equation 11.95 can now be coupled with any time stepping scheme.

11.8.6 Closing Remarks on Euler Equation Methods

In this section we have emphasized more on fundamentals of computational methods for Euler equation than on giving detailed description of various methods that are in use. This is more for keeping the size of this chapter limited. The discussions involved the ideas and principles of the Riemann problem which form the theoretical basis of characteristic theory for purely hyperbolic or mixed elliptic- hyperbolic system. In this sub-section, we therefore discuss other general schemes with their strengths and limitations.

We have given the recipe of only one simple scheme in the last sub-section, namely the Steger and Warming (1981) flux-vector splitting scheme. However the decomposition of fluxes into $\{F(\Delta y/\Delta S) - G(\Delta x/\Delta S)\}^{+}\Delta S$ and $\{F(\Delta y/\Delta S) - G(\Delta x/\Delta S)\}^{-}\Delta S$ raises a problem that these quantities are not differentiable when any of the eigenvalues λ_i changes sign. The inherent upwind scheme for the fluxes combine into a meaningless quantity at the sonic and shock point. Van Leer (1982) has proposed an equally simple flux vector splitting scheme which does not have the above mentioned problem near the sonic point. This is achieved by ensuring the split fluxes and their derivatives with respect to the unknown vector to be continuous and symmetric for all Mach numbers. Despite the improvement, this scheme as well as that of Steger and Warming (1981) have lower accuracy because of higher numerical dissipation. For the Van Leer scheme, the split fluxes deviate most from the true flux at $M = 0$, thereby broadening the boundary layer at the stagnation point.

In another class of methods, the neighbouring cells interact through discrete, finite amplitude waves whose strength and properties are determined by solving the Riemann's initial value problem for the discontinuity at the cell interface. These schemes are collectively called Flux Difference Schemes. They are very accurate and particularly well suited for explicit upwind formulation. The increased accuracy comes at the cost of enhanced complexity of coding and also increased computing time. One of such schemes is due to Roe (1981) and is used often. Despite its accuracy it fails to resolve strong bow shocks. For flow past blunt bodies it computes a shock structure which is either protruding or indentured.

At present, the research in this area is motivated by a desire to combine the simplicity and accuracy of Flux Vector Schemes and Flux Difference Schemes respectively. Liou and Steffen (1991) have come out with a new flux splitting scheme which can compute the blunt body problem without any difficulty. The method is easy to adopt and far more simpler in structure and is as accurate as the Roe scheme. This scheme is known as Advection Upstream Splitting Method (AUSM), in which the convective and pressure terms are treated separately. The convective terms are upstream- biased using the appropriately defined interface Mach number. The pressure term is split in terms of the polynomial expansion in the characteristic speed. This is the way the splitting of the Euler flux vector is performed. If one wants to extend the scheme for Navier Stokes equation, then the viscous terms can be treated by centrally averaging

at the interface. The results obtained by this scheme is indeed very promising and worthy of further studies in diverse flow situations.

However all the schemes that are currently used produce grid structure and orientation-dependent results because in multidimensional upwind differencing the direction of wave propagation is always assumed to be normal to the interface. This not only leads to smearing of shock and shear wave when these are not grid-aligned, but may also lead to misinterpretation. For example a grid-oblique shear wave can appear as a combination of grid-aligned compression and shear wave. Formulation of genuine multidimensional upwind schemes is one area where vigorous research is in progress.

Finally the current status of compressible flow computation for practical engineering application should be mentioned. Steady viscous flow past a complete aircraft is analyzed by solving Navier-Stokes equation using one million grid points (Takanashi, 1990). For the unsteady inviscid flows, a multizone Euler solver utilizing finite volume cell vertex method has been employed for flow past a fighter aircraft at transonic speeds (Creaseman and Agarwal, 1992). However for viscous unsteady three dimensional flows even the fastest supercomputer presently available is unable to provide a solution.

References

Anderson, Jr, J.D., *Fundamentals of Aerodynamics*, McGraw-Hill, New York, 1986.

Beam, R. M. and Warming, R. F., An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, AIAA Journal, Vol 16, pp. 393 - 402, 1978.

Bram Van Leer, Flux-Vector Splitting for the Euler Equations, In Lecture Notes in Physics, Vol 170, pp. 507 - 512, 1982.

Chakravarthy, S. K., Harten, A., and Osher, S., Essentially Non-oscillatory Shock Capturing Schemes of Arbitrarily High Accuracy, AIAA Paper number 86-0339, 1986.

Courant, R. and Friedrichs, K. O., *Supersonic Flow and Shock Waves*, Interscience Publishers Inc., New York, 1967.

Creaseman, F. and Agarwal, S., Japan CFD Journal, Vol 1, no. 3, pp. 295-306, 1992.

Dowell, E. H., Ueda, T., and Goorjian, P. M., Transient Decay Time and Mean Values of Unsteady Oscillations in Transonic Flow, AIAA Jour., Vol 21, no. 12, pp. 1762 - 1764, 1983.

Godunov, S. K., Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics, Math. Sbornik, Vol 47, pp. 271 - 306, 1959.

Godunov, S. K., Zabrodyn, A. W., and Prokopov, G. P., A Difference Scheme For Two-Dimensional Unsteady Problems Of Gas Dynamics And Computation Of Flow With Detached Shock Wave. Zhur. Vych. Mat. i Mat. Fyz., Vol 1, p 1020, 1961.

Gottlieb, J. J. and Groth, C. P. T., Assessment of Riemann Solvers for Unsteady One- Dimensional Inviscid Flows of Perfect Gas, Journal of Comp. Physics, Vol 78, pp. 437 -458, 1988.

Greenberg, M.D., *Advanced Engineering Mathematics*, Prentice-Hall International, New Jersey, 1998.

Jacobs, P. A., An Approximate Riemann Solver For Hypervelocity Flows, NASA Contractors Report number 187629, 1991.

Jameson, A., Iterative Solutions of Transonic Flows Over Aerofoils and Wings Including Flows at Mach 1, Commn in Pure and Applied Math, Vol 28, pp. 283-309, 1974.

Lax, P. D. and Wendroff, B., Systems of Conservation Laws, Comm. Pure Appl. Math., Vol 13, pp. 217-237, 1960.

Liepmann, H. W. and Roshko, A., *Elements of Gas Dynamics*, John Wiley, New York, 1957.

Liou, M-S and Steffen, C. J., A New Flux Splitting Scheme, NASA TM 104404, 1992.

Murman, E. M. and Cole, J. D., Calculation of Plane Steady Transonic Flows, AIAA Jour., Vol 9, pp. 114 - 121, 1971.

Peyret, R. and Taylor, T.D., *Computational Methods For Fluid Flow*, Springer Verlag, New York, 1983.

Roe, P. L., Approximate-Riemann Solvers, Parameter Vectors And Difference Schemes. Jour. of Computational Physics. Vol 43, pp. 357- 372, 1982.

Rubin, S. G. and Khosla, P. K., Polynomial Interpolation Methods for Viscous Flow Calculations, Journal of Comp. Physics, Vol 24, pp. 217 - 246, 1977.

Salas, M. D. and Gumbert, C. R., Breakdown of the Conservative Potential Equation, AIAA paper number 85 -0367, 1985.

Salas, M. D., Jameson, A., and Melnik, R. E., A Comparitive Study of the Nonuniqueness Problem of the Potential Equation, paper presented at 6th AIAA CFD conference held at Danvers, Massachusetes, 1983.

Steger, J. L. and Warming, R. F., A Flux-Vector Splitting of the Inviscid Gas Dynamic Equations With Application to Finite Difference Methods, Journal of Comp. Physics, Vol 40, pp. 263-293, 1981.

Steinhoff, J. and Jameson, A., Multiple Solutions of the Transonic Potential Flow Equations, AIAA Paper no. 81 - 1019, AIAA CFD conference, Palo Alto, California, 1981.

Takanashi, S., Large Scale Numerical Aerodynamic Simulations for Complete Aircraft Configurations, National Aerospace Laboratory, Tokyo, Report number TR - 1073T, 1990.

Williams, M. H., Bland, S. R., and Edwards, J. W., Flow Instabilities in TSDT, AIAA Jour., Vol 23, pp. 1491 - 1496, 1985.

12
A Finite Volume Method

12.1 Introduction

During the early days of computational fluid dynamics (CFD) finite-difference methods were the most popular. They are algorithmically simple, efficient, and accurate. However, they are best used on uniform grids and hence on regular computational domains. With advances of CFD, and its application to industrial problems, there is a need for methods for computing flows in complex geometries. To adapt the finite difference method to such geometries, we can map the complex domain into simple domains, either globally or locally, and solve the equations there. However, such transformation makes the governing equations take quite complicated forms and may lead to a loss of computational efficiency and accuracy. Alternatively, one can use schemes based on the finite volume methods directly on the physical domain (i.e. without transformation).

Finite volume methods are essentially a generalisation of the finite-difference method but use the integral form of the governing equations of flow, rather than their differential form. This gives greater flexibility in handling complex domains, as the finite volumes need not be regular. This approach too has some problems, the main one being that the staggered grid with non-coinciding pressure and velocity grid points – traditionally used by finite difference solvers to avoid pressure-velocity decoupling – is difficult and computationally expensive to use in complex geometries. Thus, there has been considerable interest in the last two decades in developing algorithms for fluid flow computations which can be applied on non-staggered nonorthogonal grids, without transformation of the domain and governing equations.

In this chapter, a finite volume method on a non-staggered nonorthogonal grid, applicable to complex geometries, will be discussed. However, for the sake of clarity and simplicity, this general case will be introduced later in the chapter after discussing the method applied to orthogonal Cartesian grids (which can be used for suitable problems of simple geometry).

12.2 Governing Equations

The Navier-Stokes equations for laminar flow in an arbitrary domain of volume V bounded by a closed surface $\mathbf{S}$ can be expressed in the following general convection-diffusion-source integral form:

$$\frac{\partial}{\partial t}\int_V \rho dV + \int_{\mathbf{S}} \rho \mathbf{u} \cdot \mathbf{dS} = 0 \tag{12.1}$$

$$\frac{\partial}{\partial t}\int_V \rho\phi dV + \int_{\mathbf{S}} [\rho\mathbf{u}\phi - \Gamma_\phi \nabla\phi] \cdot \mathbf{dS} = \int_V S_\phi dV \tag{12.2}$$

where ρ is the fluid density, $\mathbf{u}$ is the fluid velocity with components u, v, w, in the x, y, z directions respectively, Γ_ϕ is the diffusion coefficient (e.g., viscosity in the momentum equation) and S_ϕ is the volumetric source term. In this formulation we work with Cartesian components of velocity. So ϕ can be the three *Cartesian* components of velocity u, v, w, as well as any scalar e.g., temperature, T, which needs to be determined.

For incompressible flow of a Newtonian fluid, the equations take the form:

$$\int_{\mathbf{S}} \mathbf{u} \cdot \mathbf{dS} = 0 \tag{12.3}$$

$$\frac{\partial}{\partial t}\int_V \phi dV + \int_{\mathbf{S}} \left[\mathbf{u}\phi - \frac{\Gamma_\phi}{\rho}\nabla\phi\right] \cdot \mathbf{dS} = \frac{1}{\rho}\int_V S_\phi dV \tag{12.4}$$

where Equation 12.4 represents atleast three separate equations for $\phi = u, v$ and w with

$$S_\phi = -\frac{\partial p}{\partial x}, -\frac{\partial p}{\partial y} \text{ and } -\frac{\partial p}{\partial z}$$

respectively, that constitute the Navier-Stokes equations as well as others concerning temperature (T) and species concentrations, if required.

12.3 Regular Finite Volumes

We will first consider a three-dimensional solution domain subdivided into a number of contiguous hexahedral (six-faced) control volumes (CVs) of regular shapes, as shown in Figure 12.1. While these are regular parallelepiped, the solution procedure that follows can be easily generalized to irregular hexahedral shapes, as will be done later in this chapter. The CVs are defined by their eight vertices which are connected by straight lines. A *Collocated grid* arrangement is employed, i.e. all the dependent variables (u, v, w, p, T, etc.) are defined at the centroid of the CV which is called the node or cell/CV center. Shown in Figure 12.1 is a representative CV of dimensions Δx, Δy and Δz, with the centroid node indicated by P. Its six neighbouring nodes are indicated by E, W, N, S, T and B (for the east, west, north, south, top and bottom neighbours

respectively), whose center-to-center distance from the node P is denoted by $(\delta x)_e$, $(\delta x)_w$, $(\delta y)_n$, $(\delta y)_s$, $(\delta z)_t$ and $(\delta z)_b$, respectively. The face center points e, w, n, s, t and b are located at the corresponding face centroids of the CV with their respective unit vectors normal to the surface represented by $\mathbf{n}_e$, $\mathbf{n}_w$, $\mathbf{n}_n$, $\mathbf{n}_s$, $\mathbf{n}_t$ and $\mathbf{n}_b$.

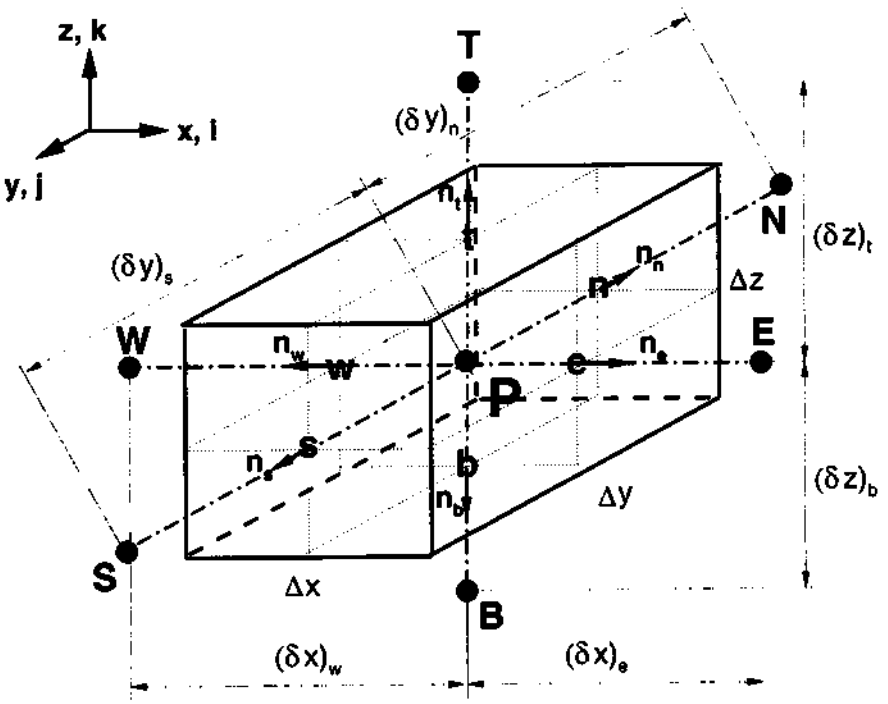


Figure 12.1: A representative control volume (CV) for simple geometries with nonuniform, orthogonal and Cartesian grid arrangement.

Before the time-stepping can proceed, all geometric parameters need to be computed. For each cell, these parameters include the surface vectors for each of the six faces and the distance between its node and each of the six neighbouring nodes. The volume of each cell, ΔV is also computed. For the regular parallelepiped in Figure 12.1, these computations are trivial: $\Delta\mathbf{S}_e = \Delta y \Delta z \hat{i}$, $\Delta\mathbf{S}_w = -\Delta y \Delta z \hat{i}$, $\Delta\mathbf{S}_s = \Delta x \Delta z \hat{j}$, $\Delta\mathbf{S}_n = -\Delta x \Delta z \hat{j}$, $\Delta\mathbf{S}_t = \Delta x \Delta y \hat{k}$, $\Delta\mathbf{S}_b = -\Delta x \Delta y \hat{k}$ and $\Delta V_P = \Delta x \Delta y \Delta z$, where $\hat{i}$, $\hat{j}$, $\hat{k}$ are the unit vectors in the x, y, z directions respectively. The corresponding computations for the irregular CVs will be shown later in the chapter.

12.4 Approximations in the Discretization Technique

Discretization is the process by which the integral equation, having infinite continuum values throughout some domain, is converted into set of linear algebraic equations for values at centroids of the CVs. The integral conservation equations 12.3 and 12.4 are applicable not just to the entire solution domain but to each CV as well. The finite volume method uses these equations for each CV:

$$\int_{\Delta \mathbf{S}} \rho \mathbf{u} \cdot d\mathbf{S} = 0$$

and

$$\frac{\partial}{\partial t} \int_{\Delta V_P} \rho \phi dV + \int_{\Delta \mathbf{S}} [\rho u \phi - \Gamma_\phi \nabla \phi] \cdot \mathbf{dS} = \int_{\Delta V_P} S_\phi dV \tag{12.5}$$

If we sum the equations for all CVs we obtain the global conservation equations 12.3 and 12.4, since surface integrals over inner CV faces cancel out. Thus global conservation is built into the method and this provides one of its principal advantages. As a result of the finite volume discretization procedure one obtains an algebraic equation for each CV, in which the centroid value ϕ_P and a number of neighbouring nodal values (ϕ_E, ϕ_W, etc.) appear. But this requires approximation at the two stages discussed separately in the following subsections.

12.4.1 Integration

The surface and volume integrals need to be approximated using quadrature formulae.

Approximation of Surface Integrals:- The net flux through the CV surface $\Delta \mathbf{S}$ is the sum of integrals over the four (in 2D) or six (in 3D) CV faces.

$$\int_{\Delta \mathbf{S}} \mathbf{F}_{cd} \cdot \mathbf{dS} = \sum_{f=e,w,n,s,t,b} \int_{\Delta \mathbf{S}_f} \mathbf{F}_{cd} \cdot \mathbf{dS} \tag{12.6}$$

where $\mathbf{F}_{cd}$ is the convective ($\rho \phi \mathbf{u}$) or diffusive ($\Gamma_\phi \nabla \phi$) flux vector at the CV face, f. To calculate the surface integrals in the Equation 12.6 exactly, one would need to know the integrand $\mathbf{F}_{cd}$ *everywhere* on the surface $\Delta \mathbf{S}_f$. This information is not available, as only the nodal values of ϕ are calculated. So an approximation must be introduced. The midpoint rule is the simplest one in which the mean value over the surface is approximated by the cell-face center value, $\mathbf{F}_{cd,f}$.

$$\int_{\Delta \mathbf{S}_f} \mathbf{F}_{cd} \cdot \mathbf{dS} \approx \mathbf{F}_{cd,f} \cdot \Delta \mathbf{S}_f \tag{12.7}$$

Here the symbol $\approx$ means that, in the chosen scheme, the quantity on the left is approximated by the quantity on the right. For a regular rectangular face,

this approximation of the integral is of second order accuracy (which can be preserved if the face-centroid value $\mathbf{F}_{cd,f}$ is computed with at least second order accuracy, as discussed in Subsection 12.4.2).

Approximation of Volume Integrals:- The transient and source terms in the transport equations require integration over the volume of the CV. The simplest second-order approximation is to replace for volume integral by the product of the mean value of the integrand and the volume, and approximate the former by the values at the CV center. That is

$$\int_{\Delta V_P} Q dV = \overline{Q} \Delta V_P \approx Q_P \Delta V_P \tag{12.8}$$

where Q_P is the value of Q at CV center. The above approximation becomes exact if Q is either constant or varies linearly within the CV; otherwise, it contains a second-order error.

12.4.2 Interpolation/Extrapolation and Differentiation

The face centered values of the mass flux, the variable ϕ, the pressure and the diffusion flux are needed in the approximation of surface integrals. These are determined by an interpolation method, except for the diffusion flux which contains a gradient term, and thus requires an approximation for differentiation for which the central difference scheme is generally used.

12.5 Discretization Procedure

We now show the discretization for the Equation 12.5 for the representative CV. The rate of change and source terms are integrated over the cell volume, whereas the convection and diffusion terms are represented in terms of fluxes through the CV faces.

12.5.1 Continuity Equation

The surface integral in Equation 12.5 is discretized in the following way:

$$\int_{\Delta \mathbf{S}} \rho \mathbf{u} \cdot \mathbf{dS} \approx \sum_{f=e,w,n,s,t,b} \rho \left(\mathbf{u} \cdot \Delta \mathbf{S}\right)_f = \sum_f \rho \mathbf{u}_f \cdot \Delta \mathbf{S}_f$$

where $\Delta \mathbf{S}_f$ is the surface vector representing the area of the fth cell face and $\mathbf{u}_f$ is the velocity defined at the face center f. We now define F_f, the outward mass flux through face f, by

$$F_f \equiv \rho \mathbf{u}_f \cdot \Delta \mathbf{S}_f \tag{12.9}$$

In discretized form the continuity equation 12.5 becomes

$$\sum_f F_f = F_e + F_w + F_n + F_s + F_t + F_b = 0 \tag{12.10}$$

12.5.2 General Equation

Rate of Change

Applying the approximation of volume integrals discussed in Subsection 12.4.1, we get

$$\frac{\partial}{\partial t}\int_{\Delta V_P} \rho\phi dV \approx \frac{(\rho\phi\Delta V)_P^{n+1} - (\rho\phi\Delta V)_P^n}{\Delta t} = \rho\Delta V_P \frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} \tag{12.11}$$

Convection Fluxes

The surface integral over the convection flux of variable ϕ can be approximated in the following form:

$$\begin{aligned} \int_{\Delta \mathbf{S}} \rho \mathbf{u}\phi \cdot \mathbf{dS} &\approx \sum_f \rho\phi_f(\mathbf{u}\cdot\Delta\mathbf{S})_f \\ &= \sum_{f=e,w,n,s,t,b} F_f\phi_f \\ &\equiv F_P^c \end{aligned} \tag{12.12}$$

where the quantity F_P^c represents the sum of convection fluxes ($\equiv F_f\phi_f$) over all the faces of the CV.

A Convection scheme is the method used to determine the value of the convected variable(ϕ_f) at the center of each CV face. As the ϕ values are known only at the cell-centers, finding ϕ_f requires interpolating ϕ from one or more of the neighbouring cell centers of the face. This interpolation is dictated by the direction of mass flux at the interface, F_f, and is expressed in terms of downstream, upstream and further upstream cell center values, represented by ϕ_D, ϕ_U, ϕ_{UU}. For example, if the mass flux is positive on the east face of the shaded CV shown in Figure 12.2 the cells with node E, P and W are the downstream, upstream and far-upstream neighbours, respectively. Otherwise, if $F_f < 0$ on the east face of the CV, the respective neighbours are the cells with node P, E and EE (the east cell of the east cell). The neighbours can be similarly classified for the other faces of CV. Numerous convection schemes are available that use ϕ_D, ϕ_U and ϕ_{UU} to find the face value ϕ_f. We mention a few that are most commonly used:

- **FOU** (*First Order Upwind*):- Constant upwind extrapolation is used involving one upstream neighbour.

$$\phi_f = \phi_U \tag{12.13}$$

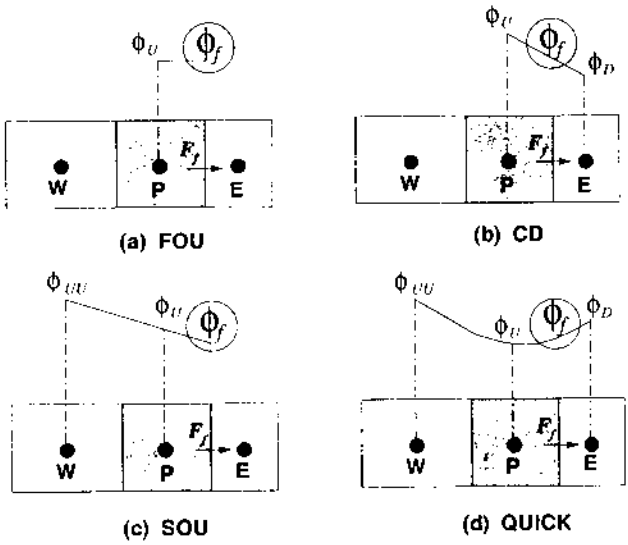


Figure 12.2: Schematic of the geometric interpolation/extrapolation employed by the various convection schemes for ϕ_f, shown for the face f(taken to be the east face in the figure) of shaded CV when the mass flux is in the positive direction indicated by the arrows on face f.

- **CD** (*Central Difference*):- Volume weighted linear interpolation is used involving one downstream and one upstream neighbour.

$$\phi_f = \frac{\Delta V_U}{\Delta V_D + \Delta V_U}\phi_D + \frac{\Delta V_D}{\Delta V_D + \Delta V_U}\phi_U \tag{12.14}$$

- **SOU** (*Second Order Upwind*):- Volume weighted linear upwind extrapolation is used involving two upstream neighbours.

$$\phi_f = \frac{2\Delta V_U + \Delta V_{UU}}{\Delta V_U + \Delta V_{UU}}\phi_U - \frac{\Delta V_U}{\Delta V_U + \Delta V_{UU}}\phi_{UU} \tag{12.15}$$

- **QUICK** (*Quadratic Upwind Interpolation for Convective Kinematics*):- Quadratic upwind interpolation is used involving one downstream and two upstream neighbours.

$$\begin{aligned}\phi_f = &\frac{\Delta V_U\,(2\Delta V_U + \Delta V_{UU})}{(\Delta V_D + \Delta V_U)\,(\Delta V_D + 2\Delta V_U + \Delta V_{UU})}\phi_D \\ &+ \frac{\Delta V_D\,(2\Delta V_U + \Delta V_{UU})}{(\Delta V_D + \Delta V_U)\,(\Delta V_U + \Delta V_{UU})}\phi_U \\ &- \frac{\Delta V_D \Delta V_U}{(\Delta V_U + \Delta V_{UU})\,(\Delta V_D + 2\Delta V_U + \Delta V_{UU})}\phi_{UU}\end{aligned} \tag{12.16}$$

where ΔV_D, ΔV_U and ΔV_{UU} are the volume of the downstream, upstream and far-upstream CVs respectively. The geometric shape of the respective interpolation or extrapolation profiles as shown in Figure 12.2, dictates the type and local order of accuracy of a convection scheme. The order of each of the convection scheme corresponds to the order of the leading error term in the Taylor series expansion; therefore, the QUICK, SOU and FOU are third, second and first order accurate, respectively, on orthogonal (but not necessarily uniform) grids. It is worth noting that the expressions for the convection schemes shown here are also applicable for non-orthogonal grid arrangements, discussed in Section 12.10, although the same order of accuracy is then not guaranteed. The Equations 12.14-12.16 can also be simplified further for the uniform grid arrangement where $\Delta V_D = \Delta V_U = \Delta V_{UU}$.

The numerical solution relies heavily on the numerical accuracy and computational stability of the convection scheme. The central difference scheme works very well for the diffusion term in the Navier-Stokes equation but when it is used for the convection term, spurious oscillations are known to result at high cell Peclet numbers ($=\rho u \Delta x/\Gamma$). To overcome this problem, the FOU scheme has been widely used, as it unconditionally satisfies the boundedness criteria and thus will never yield oscillatory solutions. However, it introduces excessive numerical diffusion, which increases with increasing cell Peclet number. A convection scheme should ideally take the advantage of upwinding for stability without losing accuracy. With this idea, the higher order convection schemes such as SOU and QUICK have been introduced. The numerical accuracy and the relative performance of the FOU, CD, SOU and QUICK schemes have been tested for this solution method, on the lid driven cavity flow problem, in the previous work of the present authors (2002). It was shown that QUICK converges to a grid independent solution with substantially fewer grid points and requires the least computational time for a given level of accuracy. With an increase in Reynolds number, it was found that the accuracy of the result obtained from the QUICK scheme remains almost constant- which is a property highly desired in a convection scheme whereas it decreases for the other convection schemes.

Diffusion Fluxes

The diffusion flux of the variable ϕ through the faces of the CV can be evaluated using the central difference scheme for the gradient of ϕ as follows:

$$\begin{aligned}
\int_{\Delta S} \Gamma_\phi \nabla\phi \cdot \mathbf{dS} &\approx \sum_{f=e,w,n,s,t,b} (\Gamma_\phi \nabla\phi \cdot \Delta\mathbf{S})_f \\
&= \Gamma_{\phi e}\left(\frac{\partial\phi}{\partial x}\right)_e \Delta S_e + \Gamma_{\phi w}\left(\frac{\partial\phi}{\partial x}\right)_w \Delta S_w + \Gamma_{\phi n}\left(\frac{\partial\phi}{\partial y}\right)_n \Delta S_n \\
&\quad + \Gamma_{\phi s}\left(\frac{\partial\phi}{\partial y}\right)_s \Delta S_s + \Gamma_{\phi t}\left(\frac{\partial\phi}{\partial z}\right)_t \Delta S_t + \Gamma_{\phi b}\left(\frac{\partial\phi}{\partial z}\right)_b \Delta S_b
\end{aligned}$$

$$
\begin{aligned}
&= \Gamma_{\phi e}\frac{\phi_E-\phi_P}{(\delta x)_e}\Delta S_e+\Gamma_{\phi w}\frac{\phi_P-\phi_W}{(\delta x)_w}\Delta S_w+\Gamma_{\phi n}\frac{\phi_P-\phi_N}{(\delta y)_n}\Delta S_n \\
&\quad +\Gamma_{\phi s}\frac{\phi_S-\phi_P}{(\delta y)_s}\Delta S_s+\Gamma_{\phi t}\frac{\phi_T-\phi_P}{(\delta z)_t}\Delta S_t+\Gamma_{\phi b}\frac{\phi_P-\phi_B}{(\delta z)_b}\Delta S_b \\
&\equiv F_P^d \qquad (12.17)
\end{aligned}
$$

where F_P^d represent the net diffusion flux across all faces of the CV with node P where ΔS_e etc. are the surface areas. In the equations above, in approximating the dot product $\nabla\phi\cdot\Delta\mathbf{S}_f$, use is made of the fact that the normal of the cell faces are in the x, y, or z coordinate directions (This will no longer be true when we consider non-orthogonal grids).

Sources

The source term is to be integrated over the cell volume. By applying the volume integral approximation:

$$\int_{\Delta V_P} S_\phi dV \approx (S_\phi)_P\Delta V_P$$

The pressure term in the momentum equation is also included as a source term. Its discretization is analogous to that of the ordinary diffusion flux, i.e., for the momentum equation for the velocity component u_i ($= u, v, w$, respectively), the pressure term is

$$-\int_{\Delta V_P}\nabla p\cdot\mathbf{n}_i dV \approx -(\nabla p\cdot\mathbf{n}_i)_P\Delta V_P \qquad (12.18)$$

where $\mathbf{n}_i$ is the unit vector in the direction of the velocity component, u_i. However, the Gauss divergence theorem can be used to convert the volume integral to a surface integral which can be discretized as

$$-\int_{\Delta S} p\mathbf{n}_i\mathbf{dS} \approx -\sum_{f=e,w,n,s,t,b} p_f\Delta S_{if} \qquad (12.19)$$

p_f is the pressure at the fth face center and ΔS_{if} is the ith direction component of the surface vector for face f. Both the above surface and volume integral forms of the pressure term have been used in the method discussed in the following section.

12.6 Semi Explicit Method

Existing methods for the incompressible Navier-Stokes equations fall into two categories, namely, semi-explicit and semi-implicit schemes. In the first, the momentum equations are discretized in an explicit manner with the exception of the pressure gradient terms, which are treated implicitly; the continuity

equation is also enforced implicitly. As a consequence, the equation-coupling reduces to a Poisson equation for the pressure corrections. Such schemes, however, because of their reliance on explicit time-stepping, suffer from time-step restrictions. In the second category, the equations are discretized implicitly, with the coupling being more complicated. Some methods which fall into this group employ sequential iteration (such as the SIMPLE and SIMPLER methods), in which the equation for each variable is solved in succession.

Both explicit and implicit methods have been extensively used over the past 30 years. Explicit methods have the advantage of algorithm simplicity, while the implicit ones that of greater numerical stability. In non-uniform grid arrangements with very fine grid size (near a solid surface, say), the implicit methods allow the use of reasonably large time-steps, while explicit methods cannot. However, explicit methods are favored by parallel computers, which is now the preferred architecture of the fastest computers. Explicit computations can be made at thousands of grid points simultaneously on such computers.

In the Semi-Explicit method, the discretized form of the momentum equation for the ith velocity component u_i:

$$\rho \Delta V_P \frac{u_{iP}^{n+1} - u_{iP}^{n}}{\Delta t} + (F_P^c + F_P^d)^n = -\sum_f p_f^{n+1} \Delta S_{if} \tag{12.20}$$

has to be solved with

$$\sum_f F_f^{n+1} = 0 \tag{12.21}$$

for each finite volume cell.

Here the subscript P indicates the cell P, the convective (F_P^c) and diffusive (F_P^d) fluxes are summed over all faces of P (see Equations 12.12 and 12.17). The superscript n indicates the (known) current time values and $n+1$ the (unknown) next values. Note that the pressure (p_f in Equation 12.19) and mass flux (F_f in Equation 12.10) are at the $n+1$ level. This makes the method incompletely (or semi-) explicit.We adopt a two step process to determine the convected velocity/variable and pressure at the cell centers, and the mass fluxes on all the face-centers of the CVs.

12.6.1 Predictor Step

The ith velocity component at the cell center, u_{iP}^* is predicted by approximating pressure in the Equation 12.20 by the known value p_f^n instead of p_f^{n+1}.

$$\rho \Delta V_p \frac{u_{iP}^* - u_{iP}^n}{\Delta t} + (F_P^c + F_P^d)^n = -\sum_f p_f^n \Delta S_{if}$$

Therefore

$$u_{iP}^* = u_{iP}^n - \frac{\Delta t}{\rho \Delta V_P}(F_P^c + F_P^d)^n - \sum_f p_f^n \Delta S_{if} \tag{12.22}$$

We define a new "tilde-velocity" variable, $\tilde{u}_{iP}$, which is

$$\tilde{u}_{iP} \equiv u_{iP}^n - \frac{\Delta t}{\rho \Delta V_P}(F_P^c + F_P^d)^n \tag{12.23}$$

Therefore, from Equation 12.22, we get

$$u_{iP}^* = \tilde{u}_{iP} - \sum_f p_f^n \Delta S_{if} \tag{12.24}$$

where linear interpolation is used to determine the pressure p_f on the faces from the cell center values p_P, p_N, p_S and others. However, it can be shown that the cell pressure value p_P will cancel out from Equation 12.24 in the summation over the p_f values. If the mass flux F_f or face velocity is also determined by the linear interpolation, it can be shown that there will be breakage of communication between the velocity and pressure at node P which will lead to checkerboard velocity and pressure fields (Patankar, 1980). A numerical method that allows such absurd solutions is certainly undesirable.

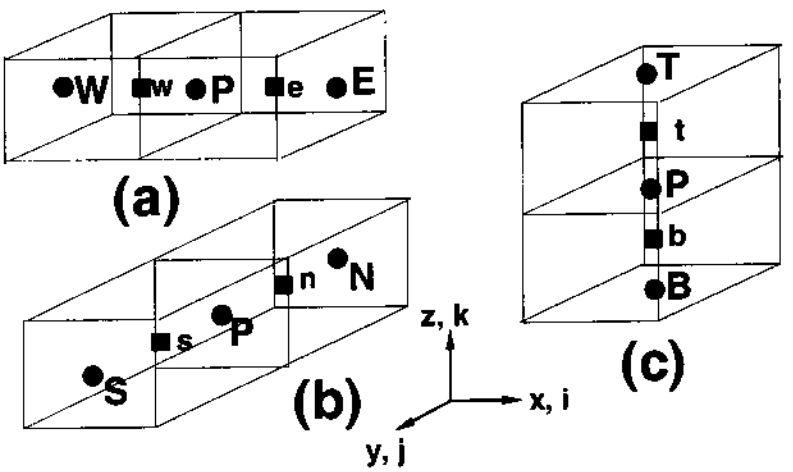


Figure 12.3: Imaginary control volumes staggered in (a) x, (b) y and (c) z directions.

The pressure-velocity decoupling is avoided by applying *momentum interpolation* to determine the velocity or mass flux on each face of the CV. Instead of averaging from cell-center values, equations similar to 12.22 can be obtained for the face velocities such as u_e and u_w, that are used to find the mass flux, F_f:

$$\rho \Delta V_e \frac{u_e^* - u_e^n}{\Delta t} + (F_e^c + F_e^d)^n = -\frac{p_E - p_P}{(\delta x)_e}\Delta V_e \tag{12.25}$$

where the discretization of pressure term is as shown in Equation 12.18 and ΔV_e is the volume of an imaginary staggered cell at the face center e, shown in Figure 12.3. Similar expressions can be obtained for the velocity on the other faces of the CV. Equation 12.25 can be further expressed as

$$\begin{aligned} u_e^* &= u_e^n - \frac{\Delta t}{\rho \Delta V_e}(F_e^c + F_e^d)^n - \frac{\Delta t}{\rho}\frac{p_E - p_P}{(\delta x)_e} \\ &= \widetilde{u}_e - \frac{\Delta t}{\rho}\frac{p_E - p_P}{(\delta x)_e} \end{aligned} \tag{12.26}$$

Now, we approximate $\widetilde{u}_e$ to get

$$u_e^* = \overline{\widetilde{u}_P, \widetilde{u}_E} - \frac{\Delta t}{\rho}\frac{p_E - p_P}{(\delta x)_e} \tag{12.27}$$

where over bar indicates a linear interpolation, i.e.

$$\overline{\widetilde{u}_P, \widetilde{u}_E} = \frac{\Delta V_E}{\Delta V_P + \Delta V_E}\tilde{u}_P + \frac{\Delta V_P}{\Delta V_P + \Delta V_E}\tilde{u}_E$$

Note that ΔV_e no longer appears in this final equation. Equation 12.27 gives the momentum interpolated face velocity value, which determines the mass flux. Note that the pressure difference between two *adjacent* nodes now becomes the driving force for the mass flux between these nodes. This is exactly as it is in the staggered grid, so by this we avoid the pressure-velocity decoupling.

Of course, u_e^* and other face velocities are only used to compute the mass flux $F_f^* \equiv \rho \mathbf{u}_f^* \cdot \Delta \mathbf{S}_f$. In fact, equations like 12.27 are nor actually used in the complex geometry case considered later. Rather, the predicted mass flux F_f^* is directly computed from the cell center values by using the following general expression

$$F_f^* = \rho \tilde{\mathbf{u}}_f \cdot \Delta \mathbf{S}_f - \Delta t \nabla p_f \cdot \Delta \mathbf{S}_f \tag{12.28}$$

where $\tilde{\mathbf{u}}_f$ is obtained by averaging cell-center $\tilde{\mathbf{u}}$ values, and ∇p_f is estimated from the cell-center pressure values.

12.6.2 Corrector Step

In the corrector step, we improve the pressure (approximated by p^n in the previous step) such that the mass flux satisfies the continuity equation 12.21 while the corrected velocity satisfies momentum equation 12.20. Note that in the predictor step no attempt was made to satisfy the continuity equation which is an important constraint on the solution.

Pressure, Velocity and Mass flux Correction

We assume that the correct pressure p^{n+1} is obtained by

$$p^{n+1} = p^n + p' \tag{12.29}$$

where p' is called the pressure correction. Next, we need to know how the velocity components and mass flux respond to this change in pressure. The corresponding velocity corrections u', v', w' and mass flux correction F' can be introduced in a similar manner:

$$u^{n+1} = u^* + u', \quad v^{n+1} = v^* + v', \quad w^{n+1} = w^* + w'; \quad F^{n+1} = F^* + F' \tag{12.30}$$

If we subtract Equation 12.22 from Equation 12.20, and substitute 12.30, we get the *velocity correction equation* as

$$\frac{\rho \Delta V_p}{\Delta t} u'_{iP} = -\sum_f p'_f \Delta S_{if}$$

or

$$u'_{iP} = -\frac{\Delta t}{\rho V_P} \sum_f p'_f \Delta S_{if} \tag{12.31}$$

The above equation is a discretization of the integral equation

$$\frac{\rho}{\Delta t} \int_{\Delta V_p} u'_i dV = -\int_{\Delta S} p' \mathbf{n}_i \mathbf{dS}$$

The right side of the equation can be transformed by the Gauss-divergence theorem to yield

$$\frac{\rho}{\Delta t} \int_{\Delta V_p} u'_i dV = -\int_{\Delta V_P} \nabla p' \cdot \mathbf{n}_i dV$$

where $\mathbf{n}_i$ is a unit vector in the direction of the velocity component u_i. As the volume of integration ΔV_P is arbitrary, the integrands must be equal. Therefore

$$u'_i = -\frac{\Delta t}{\rho} \nabla p' \cdot \mathbf{n}_i \Longrightarrow \mathbf{u}' = -\frac{\Delta t}{\rho} \nabla p' \tag{12.32}$$

Note that u'_i is a vector component whereas $\mathbf{u}'$ is a vector quantity. This equation is applicable to each CV as well as to the solution domain. By applying it for the face f of the CV, the outward mass flux correction through the face is obtained as

$$F'_f \equiv \rho \mathbf{u}'_f \cdot \Delta \mathbf{S}_f = -\Delta t \nabla p'_f \cdot \Delta \mathbf{S}_f \tag{12.33}$$

Thus the face center mass flux corrections are computed from the cell centered values of the pressure corrections that will appear in the gradient $\nabla p'_f$ (as in Equation 12.17).

Pressure Correction Equation

There is no explicit equation for pressure but the continuity equation can be converted into an equation for the pressure corrections. Substituting the mass flux from Equation 12.30 into Equation 12.21, we get

$$-\sum_f F_f^{'} = \sum_f F_f^*$$

which from Equation 12.33 gives

$$\sum_f \Delta t \nabla p_f^{'} \cdot \Delta \mathbf{S}_f = \sum_f F_f^* \tag{12.34}$$

Discretization of the pressure gradient on the LHS of the above equation 12.34 is similar to the diffusion flux (Equation 12.17). It leads to

$$\Delta t \left(\frac{p_E^{'} - p_P^{'}}{(\delta x)_e} \Delta S_e + \frac{p_P^{'} - p_W^{'}}{(\delta x)_w} \Delta S_w + \frac{p_P^{'} - p_N^{'}}{(\delta y)_n} \Delta S_n \right.$$
$$\left. + \frac{p_S^{'} - p_P^{'}}{(\delta y)_s} \Delta S_s + \frac{p_T^{'} - p_P^{'}}{(\delta z)_t} \Delta S_t + \frac{p_P^{'} - p_B^{'}}{(\delta z)_b} \Delta S_b \right) = \sum_f F_f^* \tag{12.35}$$

The R.H.S of this equation is known from the predictor step which computes F^*. We can now iterate the pressure corrections to satisfy the above equation 12.35 for all interior CVs in the domain. One convenient way is to use Gauss-Seidel iterative method where the new iterative values of $p_P^{'}$ are computed using previous iterative values of its neighbours $p_N^{'}$, $p_S^{'}$ etc. Thus we write Equation 12.35 as

$$\Delta t \left(\frac{p_E^{'m} - p_P^{'m+1}}{(\delta x)_e} \Delta S_e + \frac{p_P^{'m+1} - p_W^{'m}}{(\delta x)_w} \Delta S_w + \frac{p_P^{'m+1} - p_N^{'m}}{(\delta y)_n} \Delta S_n \right.$$
$$\left. + \frac{p_S^{'m} - p_P^{'m+1}}{(\delta y)_s} \Delta S_s + \frac{p_T^{'m} - p_P^{'m+1}}{(\delta z)_t} \Delta S_t + \frac{p_P^{'m+1} - p_B^{'m}}{(\delta z)_b} \Delta S_b \right) = \sum_f F_f^*$$

where superscript m represent the previous iteration number of Gauss-Seidel iterative method and $m+1$ to the new values. This can be further expressed as

$$a_P \left(p_P^{'m+1} - p_P^{'m} \right) = - \left(\sum_f F_f^* + \sum_f F_f^{'m} \right) \tag{12.36}$$

where

$$a_P = \Delta t \left(\frac{\Delta S_e}{\delta x_e} + \frac{\Delta S_s}{\delta y_s} + \frac{\Delta S_t}{\delta z_t} - \frac{\Delta S_w}{\delta x_w} - \frac{\Delta S_n}{\delta y_n} - \frac{\Delta S_b}{\delta z_b} \right)$$

$$\sum_f F_f^{'m} = -\Delta t \left(\frac{p_E^{'m} - p_P^{'m}}{(\delta x)_e} \Delta S_e + \frac{p_P^{'m} - p_W^{'m}}{(\delta x)_w} \Delta S_w + \frac{p_P^{'m} - p_N^{'m}}{(\delta y)_n} \Delta S_n \right.$$
$$\left. + \frac{p_S^{'m} - p_P^{'m}}{(\delta y)_s} \Delta S_s + \frac{p_T^{'m} - p_P^{'m}}{(\delta z)_t} \Delta S_t + \frac{p_P^{'m} - p_B^{'m}}{(\delta z)_b} \Delta S_b \right)$$

Equation 12.36 reduces to

$$p_P^{'m+1} = p_P^{'m} + \omega \frac{\Re}{a_P} \tag{12.37}$$

where the "residual" $\Re \equiv -\left(\sum_f F_f^* + \sum_f F_f^{'m}\right)$ and ω is a relaxation factor. The term $\Re$ represents a mass defect, which the pressure corrections – through their associated mass flux corrections – must reduce to zero. The mass flux correction is updated using the pressure correction obtained after each Gauss-Seidel iteration. These iterations and updating are continued till the residual for each CV becomes very close to zero. Thus while the predictor step is completed in a single sweep of all points, the corrector step requires repeated sweeps of many iterations to obtain the pressure corrections.

After obtaining the pressure corrections, the velocity correction at the node and the mass flux correction on the six faces of each cell is computed from Equations 12.31 and 12.33, respectively. The corrections are then added to the predicted values to obtain the pressure, velocity and mass flux for the next time level: $\mathbf{u}^{n+1} = \mathbf{u}^* + \mathbf{u}'$, $p^{n+1} = p^* + p'$ and $F^{n+1} = F^* + F'$. All this results in the completion of a single time step. The process of prediction and correction is done repeatedly to obtain the velocity fields time-step by time-step till some stopping criteria is finally satisfied.

In this section we have illustrated how the equations are handled for interior cells. However at the boundary cells, we also have to simultaneously impose boundary conditions at each step in both the predictor and corrector stages. How this is done is discussed in the next section.

12.7 Initial and Boundary Conditions

12.7.1 Initial conditions

The method discussed here is time accurate, i.e. it can be used to solve unsteady flow problems. To do this, we need to give the initial condition of velocity and pressure in accordance to the physical situation. For such a problem, we may know the initial conditions of velocity but yet not know that of pressure. However, compatible initial conditions for the pressure may be calculated from the initial velocity field by solving the pressure Poisson equation $\nabla^2 p = -\rho \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u})$ (obtained by using the divergence operation on the momentum equations and applying the continuity equation) with suitable boundary conditions. In

cases where the velocity field starts from initially quiescent (zero) conditions, the pressure may be assumed to be constant, often atmospheric.

Time-accurate methods are often used to solve steady-state problems using the false-transient approach, i.e. starting from an arbitrary field and time-marching to the steady-state solutions. The initial velocity and pressure field can be specified arbitrarily for such problems. As the time-stepping proceeds, the velocity and pressure fields are both corrected and finally both the velocity and pressure steady state fields are obtained. In such cases, if the initial velocity and pressure fields were not compatible, the pressure field so obtained is not the actual pressure but a *pseudo* pressure field. This will however not affect the correctness of the velocity field. Indeed, even otherwise the actual pressure boundary conditions are complicated, so it is best to use a pseudo pressure field to compute the velocity field, and then solve the pressure Poisson equation to get the actual pressure if, and when, necessary. If one is not interested in the pressure field, the pressure can even be initialized to zero at every time step without affecting the velocity field.

12.7.2 Boundary Conditions for Velocity, Pressure and Pressure Correction

One of the difficulties encountered with the free-stream fluid flow problems is that of specifying correct boundary conditions, especially at cross-stream and outflow boundaries. This is because the computational domain is bounded whereas the physical domain is unbounded. Thus, the computational domain has to be truncated from the real domain by using artificial *open* boundary conditions - such as free slip boundary conditions at the cross-stream boundaries and *Neumann* (NBC) or *convective* (CBC) boundary conditions for the outlet boundaries. For high accuracy, the computational domain must sometimes be very large and this increases CPU times and the cost of the computation. Often, proper boundary conditions can reduce the size of the required computational domain and decrease the cost. The following boundary conditions (BCs) are generally encountered in fluid flow and heat transfer problems: "Border" CVs shown in the Figure 12.5, are defined as the CV whose one face coincides with the boundary of the physical domain.

• **No-slip BC** (at solid walls)

$$\mathbf{u} = 0; \frac{\partial p}{\partial n} = \mu \frac{\partial^2 u_n}{\partial n^2}; \frac{\partial p'}{\partial n} = 0 \tag{12.38}$$

where n is the direction normal to the boundary. The pressure BC is obtained by using boundary layer approximation.

• **Inlet BC**

$$u_n = u_{in}, u_t = 0; \frac{\partial p}{\partial n} = -\left(\frac{\partial(\rho u_n)}{\partial t} + \nabla \cdot (\rho \mathbf{u} u_n) - \mu \nabla^2 u_n\right); \frac{\partial p'}{\partial n} = 0 \tag{12.39}$$

where u_n and u_t are the normal and tangential velocities, respectively and u_{in} is the velocity specified at the inlet which may be constant or function of position.

• **Free-slip BC**

$$u_n = 0, \frac{\partial u_t}{\partial n} = 0; \frac{\partial p}{\partial n} = -\left(\frac{\partial(\rho u_n)}{\partial t} + \nabla \cdot (\rho \mathbf{u} u_n) - \mu \nabla^2 u_n\right); \frac{\partial p'}{\partial n} = 0 \quad (12.40)$$

• **Outlet BC**:- In many numerical studies, especially those which involve vortex shedding, the outlet (open) boundary condition is a very important issue. A suitable outflow boundary (i) permits the flow to exit the domain with a smooth discharge of vortices, (ii) has a minimum effect on the flow in the domain near the outlet, and (iii) has a negligible effect on the near-body flow. Incorrect location of the outflow boundaries and unsuitable boundary conditions might seriously affect the whole flow structure, especially near the body. For finite difference and finite volume discretization, Neumann boundary condition(NBC) and convective boundary condition(CBC) are the two most popular boundary conditions. The pressure at the outlet boundary is taken to be constant (say, atmospheric) in most problems. As the pressure in an incompressible flow is known only in a relative, not absolute, sense we can then choose the outlet pressure to be zero. The NBC, CBC, pressure BC and pressure correction BC are respectively given by

$$\frac{\partial \mathbf{u}}{\partial n} = 0, \frac{\partial \mathbf{u}}{\partial t} + U_c \frac{\partial \mathbf{u}}{\partial n} = 0; p = 0; p' = 0 \quad (12.41)$$

at the outlet where U_c is the average stream wise velocity. Sohankar *et al.* (1998), have compared the CBC with the NBC at the outlet for the free stream flow across a cylinder of square cross-section and show that the CBC is better as it decreases the number of iterations per time step and the number of time steps to reach periodic flow. Furthermore, it allows a lower downstream computational domain length – between the rear surface of the cylinder and the outlet – to be used.

It is worth noting that the normal velocity is specified for all the boundary conditions discussed here, except the outlet boundary condition. Thus, the normal velocity correction will be zero on such boundaries and we get the pressure correction boundary condition from Equation 12.32 as $\partial p'/\partial n = 0$.

Also note that the actual pressure boundary conditions are very much more complicated than the pressure correction conditions. However, these boundary conditions are not usually applied. Instead simple homogeneous Neumann ($\partial p'/\partial n = 0$ at all boundaries except outlet) or homogeneous Dirichlet ($p = 0$ at outlet) conditions are applied. This means that the pressure field being calculated is actually a pseudo-pressure field. This is sufficient to obtain the velocity field. If the actual pressure is, at any time, needed we can solve the pressure Poisson equation with the actual pressure BC to obtain it. However, at any time instant, the pseudo pressure field differs from the true pressure field by an arbitrary constant. Thus, the pseudo pressure can be used to determine those

engineering parameters (such as coefficient of lift and drag) which requires the relative pressure differences, but not its absolute values.

12.8 Implementation Issues

12.8.1 Implementation of Boundary conditions

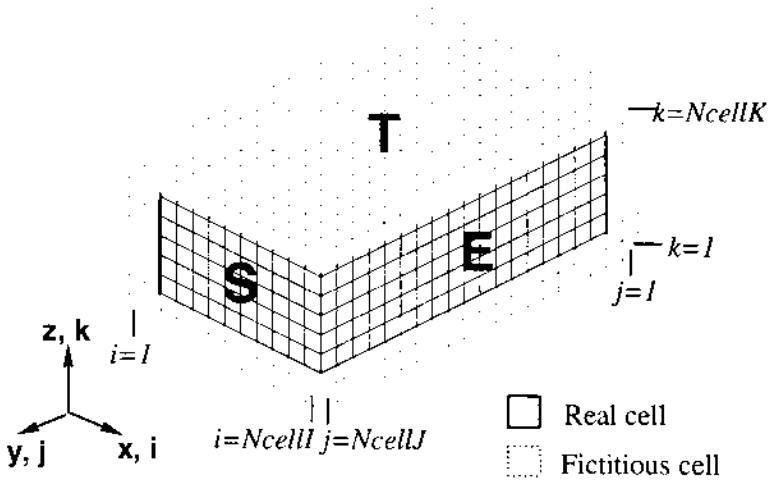


Figure 12.4: Sectional view of fictitious boundary cells surrounding the real cells (in the computational domain).

The boundary conditions are imposed by setting appropriate velocity, pressure, and pressure correction values in the fictitious cells surrounding the computational domain, shown in Figure 12.4. The boundaries of the physical domain lie on the cell faces between these boundary cells and their immediate interior cell neighbours. Therefore, the inner face of the fictitious cells and the outer face of the interior cells are the physical domain boundaries. The advantage of this fictitious cell approach to boundary conditions is that all interior cell values can be computed using the governing partial differential equations, and then the boundary (fictitious) cell values can be simply specified using the interior cell values. The BCs are thus quite simple to implement.

We will now show how BCs are applied on the domain shown in Figure 12.4. We assume that the grid points are represented by indices i, j, k, which run from 1 to $NcellI$, 1 to $NcellJ$, and 1 to $NcellK$, respectively in the three

directions. The first and last cells are always fictitious, so the real interior cells run from 2 to $Ncell(I, J, K) - 1$.

Consider the bottom boundary of the domain. Let us assume that the boundary is a rigid no-slip wall. This condition can be imposed by setting the velocity at the fictitious node equal and opposite to the adjacent interior node so that the averaged velocity at the physical boundary (the cell face in-between) is zero. The pressure and pressure correction normal gradient is set to zero by equating the values at fictitious to those at the adjacent interior node. With reference to the Figure 12.4 and Equation 12.38, we have

$$\left.\begin{array}{l} u_{i,j,1} = -u_{i,j,2} \\ v_{i,j,1} = -v_{i,j,2} \\ w_{i,j,1} = -w_{i,j,2} \\ p_{i,j,1} = p_{i,j,2} \\ p'_{i,j,1} = p'_{i,j,2} \end{array}\right\} \begin{array}{lllll} \text{for} & i = 2 & \text{to} & NcellI - 1 \\ \text{and} & j = 2 & \text{to} & NcellJ - 1 \end{array} \tag{12.42}$$

Assume the south wall (S in the Figure 12.4) is a free-slip boundary. Then Equation 12.40 takes the form

$$\left.\begin{array}{l} u_{i,NcellJ,k} = u_{i,NcellJ-1,k} \\ v_{i,NcellJ,k} = -v_{i,NcellJ-1,k} \\ w_{i,NcellJ,k} = w_{i,NcellJ-1,k} \\ p_{i,NcellJ,k} = p_{i,NcellJ-1,k} \\ p'_{i,NcellJ,k} = p'_{i,NcellJ-1,k} \end{array}\right\} \begin{array}{lllll} \text{for} & i = 2 & \text{to} & NcellI - 1 \\ \text{and} & k = 2 & \text{to} & NcellK - 1 \end{array} \tag{12.43}$$

If the east wall (E in the Figure 12.4) is a outlet boundary. The CBC, pressure and pressure correction BC in Equation 12.41, are implemented as

$$\left.\begin{array}{l} \phi^{n+1}_{NcellI,j,k} = \phi^{n}_{NcellI,j,k} \\ \qquad -\Delta t U_c \frac{\left(\phi^n_{NcellI,j,k} - \phi^n_{NcellI-1,j,k}\right)}{(\delta x)_{wNcellI,j,k}} \\ p_{NcellI,j,k} = -p_{NcellI-1,j,k} \\ p'_{NcellI,j,k} = -p'_{NcellI-1,j,k} \end{array}\right\} \begin{array}{lllll} \text{for} & j = 2 & \text{to} & NcellJ - 1 \\ \text{and} & k = 2 & \text{to} & NcellK - 1 \end{array} \tag{12.44}$$

where $U_c(> 0)$ is the average streamwise velocity and ϕ is the convected variable which may be any component of velocity, as well as any scalar, e.g., temperature.

Let the west wall be an inlet boundary. The normal velocity component will be given, say, as a constant (u_0) or a parabolic profile ($u_{in} = f(j, k)$). Equation 12.39 is then represented as

$$\left.\begin{array}{l} u_{1,j,k} = (2u_0 - u_{2,j,k}) \text{ or } (2f(j,k) - u_{2,j,k}) \\ v_{1,j,k} = -v_{2,j,k} \\ w_{1,j,k} = -w_{2,j,k} \\ p_{1,j,k} = p_{2,j,k} \\ p'_{1,j,k} = p'_{2,j,k} \end{array}\right\} \begin{array}{lllll} \text{for} & j = 2 & \text{to} & NcellJ - 1 \\ \text{and} & k = 2 & \text{to} & NcellK - 1 \end{array} \tag{12.45}$$

where the first equation imposes the constant u_0 or parabolic $f(j,k)$ velocity at the east face of the fictitious ($i = 1$) cells.

Boundary condition on the top and bottom walls can be set similarly. Note that the implementation of homogeneous pressure BC ($\partial p'/\partial n = 0$) has been shown here for the BCs represented by Equations 12.38-12.40.

12.8.2 Conservative and Consistent Implementation of Convection Schemes

"Border" CVs shown in the Figure 12.5, are defined as the CV whose one face coincides with the boundary of the physical domain.

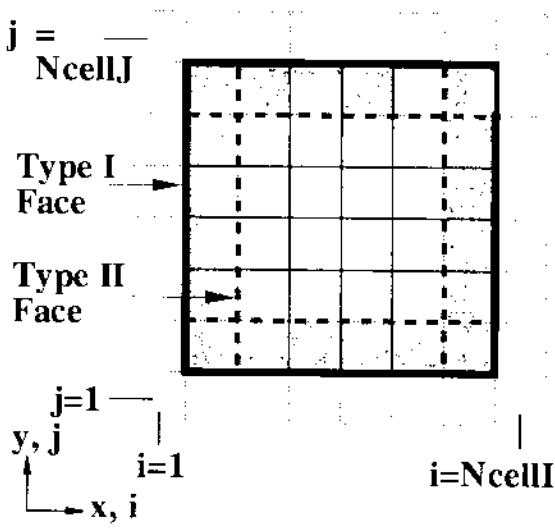


Figure 12.5: Types of control volume faces that are effected by the presence of boundaries where solid, dashed and dotted line represents the physical domain boundary, border CV faces parallel to it and fictitious cell faces, respectively.

As mentioned previously, we determine the value of the convected variable on each face of the interior cells in terms of the cell center values. The implementation of higher order convection schemes such as SOU and QUICK, requires two cell center neighbours upstream of the face (i.e. ϕ_U and ϕ_{UU} in Equations 12.15 and 12.16 and Figure 12.2), which may not be available for the face of the border CV coinciding with the domain boundary – when the mass flux is inward from the boundary. Thus, the implementation requires special treatment for the two types of CV faces shown in Figure 12.5 by thick solid and dashed lines: Type I coincides with, and Type II is parallel to, the physical

domain boundary. We now outline the treatment of these two types of faces in the algorithm.

For the CV faces of Type I, no convection scheme is needed if the BC is Dirichlet (value of ϕ is known at the boundary, say ϕ_b) as $\phi_f = \phi_b$. Otherwise, if the BC is Neumann (gradient of ϕ is known at the boundary), the fictitious cell center and the adjoining real cells are taken as ϕ_D, ϕ_U and ϕ_{UU}, respectively, irrespective of the sign of the mass flux. For the CV faces of Type II, when the mass flux is inward from the boundary and BC is Dirichlet, $\phi_{UU} = \phi_b$ (which is only half CV width away from the cell center of border CV (ϕ_U) instead of the usual full width) and $\Delta V_{UU} = 0$. The implementation for the other type of CV faces of real cells is straightforward as two upstream cell centers are available for them. The treatment for other types of BC is similar to that for the Neumann BC.

Thus, the higher order convection schemes (SOU and QUICK) are consistently implemented on all the faces of real cells. Furthermore, it is ensured that the convection fluxes are same for the two CVs at the interface – by using the same expression to determine the convection fluxes for both – and thus the conservative property is maintained.

12.8.3 Calculation of time step

Explicit methods are numerically unstable if the time-step exceeds some value prescribed, in principle, by a stability analysis of the governing equations. An exact stability analysis of the discretized Navier-Stokes equation does not exist. However, the *von Neumann* stability analysis applied to the advection and diffusion equations with an explicit FTCS (Forward in time and central in space) scheme can provide some reasonable guidance(Chapter 4). The stability criterion obtained by the analysis are as follows:

• **CFL (*Courant-Friedrichs-Lewy*) criterion**: It has to be satisfied for negligible diffusion.

$$\Delta t \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} \right) \leq 1 \tag{12.46}$$

This restriction for the time step means that a fluid particle cannot move more than one grid length in a single time step.

• **Grid Fourier Number criterion**: It has to be satisfied for negligible convection.

$$\nu \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right) \leq \frac{1}{2} \tag{12.47}$$

This means that each time the spatial mesh is halved, the time step has to be reduced by a factor of four.

When both convection and diffusion are present, the stability criterion is more complicated. Rather than dealing with this complexity, most often both criteria are individually applied and the smaller Δt is used. This may be little more restrictive than necessary, but it is safe.

12.8.4 Two-Dimensional Flows

Although the method discussed here is for three-dimensional flows, a comparable formulation with same algorithm can be easily developed for two-dimensional flows, by dropping all references to the third direction variables.

12.9 Solution Algorithm

The velocity/variable ϕ and pressure at the cell-center and velocity/mass flux at face-center are computed with the following algorithm:

1. Establish the problem by selecting the computational domain size, the governing parameters and by calculating the geometrical parameters.
2. Set the initial conditions and boundary conditions for velocity and pressure.
3. Compute the face-center mass flux for all real cells from Equation 12.27 or 12.28 using initial values.
4. Calculate the optimum time step from the CFL and grid Fourier number criterion using Equations 12.46 and 12.47. Use the smaller value as the time step.
5. Compute $\tilde{u}_{iP}$ and predict the cell center u^*_{iP} using Equations 12.23 and 12.24, respectively. Use linear interpolation (Equation 12.14) to obtain face center quantities $\bar{u}_f$ and p_f, and Equation 12.27 or 12.28 to obtain F^*. Compute $\sum_f F^*_f$ for each real cell.
6. Use BC for the predicted values at the fictitious cells.
7. Set pressure correction $p' = 0$ for all cells.
8. Impose pressure correction BC.
9. The face-center mass flux correction for each real cell is computed from Equation 12.33.
10. Compute residual for each real cell

$$\Re = -\left(\sum_f F^*_f + \sum_f F'_f\right)$$

11. Calculate the cell-center pressure correction from the Equation 12.37 for all real cells and find

$$\Re_{rms} = \sqrt{\frac{\sum_{i,j,k} \Re^2}{\text{Number of real cells}}}$$

12. Go to step 8 if $\Re > \epsilon$, where ϵ is the prescribed convergence criteria, say 10^{-6}, otherwise the final iterative values p'^{m} are now treated as the converged pressure correction, p'.

13. The cell-center velocity correction and face-center mass flux correction is computed from the converged pressure correction using Equations 12.31 and 12.33, respectively.

14. Use BC for the corrected values at the fictitious cells.

15. Go to step 4 and repeat the process until the stopping criterion (which may be steady state) is attained.

12.10 Treatment of Complex Geometry

In finite volume methods implemented directly on complex physical domains, the non-staggered (also called collocated) grid is the natural one to use. Such grids define all dependent variables at the same point – usually at the centroid of the finite volume. A representative hexahedral CV for complex geometries, shown in Figure 12.6, is defined by the coordinates of the vertices which are assumed to be connected by straight lines. The coordinates of these vertices are obtained from some grid generation procedure. Compared to the CVs for

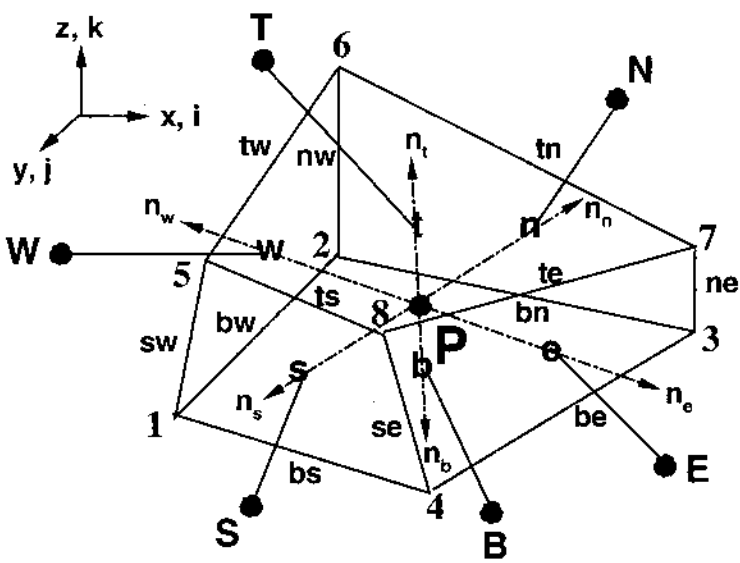


Figure 12.6: A representative CV for complex geometries with nonuniform, non-orthogonal and Cartesian grid arrangement.

simple geometries with orthogonal grid arrangement shown in Figure 12.1, the CV shown in Figure 12.6 has three major differences. First, the faces for the CV are not orthogonal to each other; second, the edges of the CV are not along the direction of the Cartesian coordinates and are also not perpendicular to each other, and third, the lines joining the cell center P and neighbouring cell centers do not pass through the corresponding face centers; for example, the line joining node P and E does not pass though face center e. Note, the edge-centers te, be, ne. se and others in Figure 12.6 are also needed in the computations.

The Semi-Explicit method discussed previously can be applied to complex geometries. However, this requires parameterization of the control volume surfaces and volumes and a diffusion flux formulation incorporating cross diffusion terms – which has no contribution for the orthogonal CVs discussed previously – in addition to the normal diffusion term. Each of these issues will be discussed separately in the following subsections. Finally, the terms for which the diffusion flux formulation is applied will be discussed in Subsection 12.10.3

12.10.1 Parameterization of Surface Areas and Volumes

The finite volume vertices are numbered 1 to 8 in the manner shown in the Figure 12.6. The *outward surface* normals and volume can be found in the following manner suggested by Kordulla and Vinokur (1983). Defining $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ where $\mathbf{r}_i$, $\mathbf{r}_j$ are the position vectors of points i and j, respectively, the surface areas are expressed as

$$\Delta\mathbf{S}_e = \frac{1}{2}(\mathbf{r}_{83} \times \mathbf{r}_{47}) \quad \Delta\mathbf{S}_s = \tfrac{1}{2}(\mathbf{r}_{54} \times \mathbf{r}_{18}) \quad \Delta\mathbf{S}_t = \frac{1}{2}(\mathbf{r}_{86} \times \mathbf{r}_{75})$$
$$\Delta\mathbf{S}_w = \frac{1}{2}(\mathbf{r}_{25} \times \mathbf{r}_{16}) \quad \Delta\mathbf{S}_n = \tfrac{1}{2}(\mathbf{r}_{72} \times \mathbf{r}_{36}) \quad \Delta\mathbf{S}_b = \frac{1}{2}(\mathbf{r}_{42} \times \mathbf{r}_{13}) \tag{12.48}$$

The volume of the CV is calculated from the cell corner coordinates assuming that the cell corners are joined by linear segments to form the six cell faces.

$$\Delta V = \frac{1}{3}\mathbf{r}_{71} \cdot (\Delta\mathbf{S}_s + \Delta\mathbf{S}_b + \Delta\mathbf{S}_w) \tag{12.49}$$

12.10.2 A Formulation of Diffusion Flux for the Complex Geometries

The diffusion flux of variable ϕ through the cell faces is evaluated as follows

$$\int_S \Gamma_\phi \nabla\phi \cdot \mathbf{dS} = \sum_{f=e,w,n,s,t,b} (\Gamma_\phi \nabla\phi \cdot \Delta\mathbf{S})_f$$

For any face, we can write

$$\Delta\mathbf{S}_f = \alpha_1\mathbf{n}_1 + \alpha_2\mathbf{n}_2 + \alpha_3\mathbf{n}_3 \tag{12.50}$$

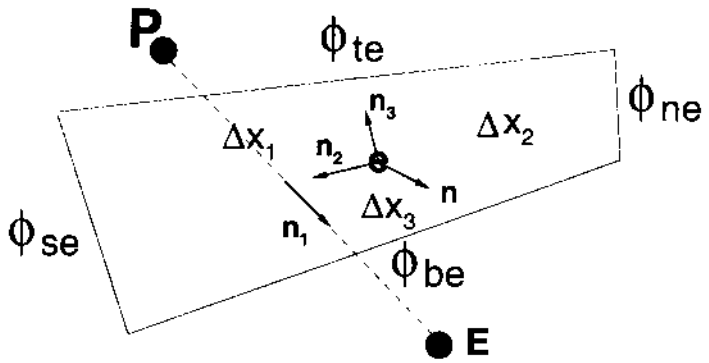


Figure 12.7: East face representation.

If $\mathbf{n}_1$, $\mathbf{n}_2$ and $\mathbf{n}_3$ are three linearly independent (not necessarily orthogonal, shown in Figure 12.7 for the east face) unit vectors. The unit vector $\mathbf{n}_1$ is in the direction of the line segment $\Delta\mathbf{x}_1$ joining the cell center P to its east neighbour E, $\mathbf{n}_2$ and $\mathbf{n}_3$ are unit vectors respectively parallel to the line segments $\Delta\mathbf{x}_2$ and $\Delta\mathbf{x}_3$ joining the opposite midpoints of the edges of the face. From Equation 12.50, we get

$$\begin{aligned}\nabla\phi\cdot\Delta\mathbf{S}_f &= \nabla\phi\cdot(\alpha_1\mathbf{n}_1+\alpha_2\mathbf{n}_2+\alpha_3\mathbf{n}_3)\\ &= \alpha_1\nabla\phi\cdot\mathbf{n}_1+\alpha_2\nabla\phi\cdot\mathbf{n}_2+\alpha_3\nabla\phi\cdot\mathbf{n}_3 \qquad (12.51)\end{aligned}$$

If $\Delta\phi_1$, $\Delta\phi_2$, $\Delta\phi_3$ are the differences in ϕ between the ends of the line segments $\Delta\mathbf{x}_1$, $\Delta\mathbf{x}_2$, $\Delta\mathbf{x}_3$, then we assume (with some approximation) that

$$\Delta\phi_1=\nabla\phi\cdot\Delta\mathbf{x}_1,\qquad \Delta\phi_2=\nabla\phi\cdot\Delta\mathbf{x}_2,\qquad \Delta\phi_3=\nabla\phi\cdot\Delta\mathbf{x}_3 \qquad (12.52)$$

Furthermore $\Delta\mathbf{x}_1$, $\Delta\mathbf{x}_2$, $\Delta\mathbf{x}_3$ are in the directions of $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ respectively, then it follows from Equation 12.52 that

$$\frac{\Delta\phi_1}{|\Delta\mathbf{x}_1|}=\nabla\phi\cdot\mathbf{n}_1,\qquad \frac{\Delta\phi_2}{|\Delta\mathbf{x}_2|}=\nabla\phi\cdot\mathbf{n}_2,\qquad \frac{\Delta\phi_3}{|\Delta\mathbf{x}_3|}=\nabla\phi\cdot\mathbf{n}_3 \qquad (12.53)$$

where $|\Delta\mathbf{x}_1|$, $|\Delta\mathbf{x}_2|$ and $|\Delta\mathbf{x}_3|$ are the magnitudes of $\Delta\mathbf{x}_1$, $\Delta\mathbf{x}_2$and $\Delta\mathbf{x}_3$, respectively. Consequently we can approximate Equation 12.51 using Equation 12.53 as

$$\nabla\phi.\Delta\mathbf{S}_f=\alpha_1\frac{\Delta\phi_1}{|\Delta\mathbf{x}_1|}+\alpha_2\frac{\Delta\phi_2}{|\Delta\mathbf{x}_2|}+\alpha_3\frac{\Delta\phi_3}{|\Delta\mathbf{x}_3|} \qquad (12.54)$$

This approximation is used to evaluate $\nabla\phi\cdot\Delta\mathbf{S}_f$ for the diffusion flux. We evaluate $|\Delta\mathbf{x}_1|$, $|\Delta\mathbf{x}_2|$, $|\Delta\mathbf{x}_3|$ and $\Delta\phi_1(\equiv\phi_E-\phi_P)$, $\Delta\phi_2(\equiv\phi_{se}-\phi_{ne})$, $\Delta\phi_3(\equiv\phi_{te}-\phi_{be})$, from the coordinates of the edge centers and their corresponding ϕ

values which are obtained by interpolation (see below). To get α_1, α_2 and α_3, we express

$$\mathbf{n}_1 = (n_{11} \;\; n_{12} \;\; n_{13}), \qquad \mathbf{n}_2 = (n_{21} \;\; n_{22} \;\; n_{23}), \qquad \mathbf{n}_3 = (n_{31} \;\; n_{32} \;\; n_{33})$$

where n_{11}, n_{12}, n_{13} are the Cartesian components of $\mathbf{n}_1$ and which can be easily determined by $\frac{\Delta x_{11}}{|\Delta \mathbf{x}_1|}, \frac{\Delta x_{12}}{|\Delta \mathbf{x}_2|}, \frac{\Delta x_{13}}{|\Delta \mathbf{x}_3|}$, where Δx_{11}, Δx_{12}, Δx_{13} are the three components of vector $\Delta \mathbf{x}_1$. The other values $n_{21...}n_{33}$ can be similarly determined from $\Delta \mathbf{x}_2$ and $\Delta \mathbf{x}_3$. Therefore, Equation 12.50 can be written as

$$\begin{bmatrix} n_{11} & n_{21} & n_{31} \\ n_{12} & n_{22} & n_{32} \\ n_{13} & n_{23} & n_{33} \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix} = \begin{Bmatrix} \Delta S_{1f} \\ \Delta S_{2f} \\ \Delta S_{3f} \end{Bmatrix} \tag{12.55}$$

where ΔS_{1f}, ΔS_{2f}, ΔS_{3f} are the Cartesian components of the surface vector $\Delta \mathbf{S}_f$ (which are known). Using Cramer's rule we can now find α_1, α_2, α_3. These α's (and $\Delta \mathbf{x}_1$,$\Delta \mathbf{x}_2$ and $\Delta \mathbf{x}_3$) need to be evaluated only once in the preprocessing stage, as they remain the same for all time steps.

The diffusion flux is made of two distinct parts: normal diffusion and cross diffusion. The second part arises from the nonorthogonality of the grid. The normal derivative diffusion flux of ϕ through any cell face involves the values of ϕ at cell centers whereas the cross-derivative diffusion flux takes into account the edge center values of ϕ. The example of the east face is taken to illustrate the diffusion model and is shown in Figure 12.7. We get the normal diffusion term $\frac{\phi_E - \phi_P}{|\Delta \mathbf{x}_1|}$ involving cell center values and the cross diffusion terms $\frac{\phi_{se} - \phi_{ne}}{|\Delta \mathbf{x}_2|}$ and $\frac{\phi_{te} - \phi_{be}}{|\Delta \mathbf{x}_3|}$ involving the edge center values ϕ_{te}, ϕ_{be}, ϕ_{se}, ϕ_{ne}. We compute the diffusion flux by

$$(\Gamma_\phi \nabla \phi \cdot \Delta S)_e = \Gamma_\phi \left(\alpha_1 \frac{\phi_E - \phi_P}{|\Delta \mathbf{x}_1|} + \alpha_2 \frac{\phi_{se} - \phi_{ne}}{|\Delta \mathbf{x}_2|} + \alpha_3 \frac{\phi_{te} - \phi_{be}}{|\Delta \mathbf{x}_3|} \right) \tag{12.56}$$

To calculate the edge center values appearing in cross-derivative diffusion flux, the following interpolation scheme is used.

$$\phi_{te} = \frac{\Delta V_{TE}\phi_P + \Delta V_P \phi_{TE} + \Delta V_T \phi_E + \Delta V_E \phi_T}{\Delta V_{TE} + \Delta V_P + \Delta V_T + \Delta V_E} \tag{12.57}$$

$$\phi_{te} = \frac{\Delta V_{BE}\phi_P + \Delta V_P \phi_{BE} + \Delta V_B \phi_E + \Delta V_E \phi_B}{\Delta V_{BE} + \Delta V_P + \Delta V_B + \Delta V_E} \tag{12.58}$$

$$\phi_{ne} = \frac{\Delta V_{NE}\phi_P + \Delta V_P \phi_{NE} + \Delta V_N \phi_E + \Delta V_E \phi_N}{\Delta V_{NE} + \Delta V_P + \Delta V_N + \Delta V_E} \tag{12.59}$$

$$\phi_{se} = \frac{\Delta V_{SE}\phi_P + \Delta V_P \phi_{SE} + \Delta V_S \phi_E + \Delta V_E \phi_S}{\Delta V_{SE} + \Delta V_P + \Delta V_S + \Delta V_E} \tag{12.60}$$

where the notation used is that ΔV_{TE} is the volume of the cell to the top-east of cell P and ϕ_{TE} is its cell center value. Other edge center values can be determined by similar interpolation at the other faces.

12.10.3 Application of the Diffusion Flux Formulation

The diffusion flux formulation is used not just for the diffusion term but also for the mass flux correction term represented by Equation 12.33. This is used in the solution of the pressure correction equation, 12.36, needed for the imposition of continuity. Note that the surfaces of the CV shown in Figure 12.6 have nonzero components in more than one direction, i.e. ΔS_e has non-zero components in the y and z directions as well as x direction, and the mass flux $F_e = \rho(u_e \Delta S_{ex} + v_e \Delta S_{ey} + w_e \Delta S_{ez})$. We evaluate the pressure gradient term in F_f^* (Equation 12.28) and $F_f^{'}$ (Equation 12.33) using Equation 12.56 (with $\Gamma_\phi = -\Delta t$ and ϕ is replaced by p^* and $p^{'}$, respectively for Equations 12.28 and 12.33).

The pressure correction equation 12.36 is then modified by additional terms for the mass flux correction corresponding to the cross diffusion of the pressure correction:

$$a_P \left(p_P^{'m+1} - p_P^{'m} \right) = - \left(\sum_f F_f^* + \sum_f F_f^{'m} \right)$$

where

$$a_P = \Delta t \left(\frac{\alpha_{1e}}{\Delta x_{1e}} + \frac{\alpha_{2s}}{\Delta x_{2s}} + \frac{\alpha_{3t}}{\Delta x_{3t}} - \frac{\alpha_{1w}}{\Delta x_{1w}} - \frac{\alpha_{2n}}{\Delta x_{2n}} - \frac{\alpha_{3b}}{\delta x_{3b}} \right)$$

$$\begin{aligned}
\sum_f F_f^{'m} \quad = \quad & -\Delta t \Bigg[\Bigg(\frac{p_E^{'m} - p_P^{'m}}{\Delta x_{1e}} \alpha_{1e} + \frac{p_P^{'m} - p_W^{'m}}{\Delta x_{1w}} \alpha_{1w} + \frac{p_P^{'m} - p_N^{'m}}{\Delta x_{2n}} \alpha_{2n} \\
& + \frac{p_S^{'m} - p_P^{'m}}{\Delta x_{2s}} \alpha_{2s} + \frac{p_T^{'m} - p_P^{'m}}{\Delta x_{3t}} \alpha_{3t} + \frac{p_P^{'m} - p_B^{'m}}{\delta x_{3b}} \alpha_{3b} \Bigg) \\
& + \Bigg(\frac{p_{se}^{'m} - p_{ne}^{'m}}{\Delta x_{2e}} \alpha_{2e} + \frac{p_{te}^{'m} - p_{be}^{'m}}{\Delta x_{3e}} \alpha_{3e} + \frac{p_{sw}^{'m} - p_{nw}^{'m}}{\Delta x_{2w}} \alpha_{2w} \\
& + \frac{p_{tw}^{'m} - p_{bw}^{'m}}{\Delta x_{3w}} \alpha_{3w} + \frac{p_{ne}^{'m} - p_{nw}^{'m}}{\Delta x_{1n}} \alpha_{1n} + \frac{p_{tn}^{'m} - p_{bn}^{'m}}{\Delta x_{3n}} \alpha_{3n} \\
& + \frac{p_{se}^{'m} - p_{sw}^{'m}}{\Delta x_{1s}} \alpha_{1s} + \frac{p_{ts}^{'m} - p_{bs}^{'m}}{\Delta x_{3s}} \alpha_{3s} + \frac{p_{te}^{'m} - p_{tw}^{'m}}{\Delta x_{1t}} \alpha_{1t} \\
& + \frac{p_{ts}^{'m} - p_{tn}^{'m}}{\Delta x_{2t}} \alpha_{2t} + \frac{p_{be}^{'m} - p_{bw}^{'m}}{\Delta x_{1b}} \alpha_{1b} + \frac{p_{bs}^{'m} - p_{bn}^{'m}}{\Delta x_{2b}} \alpha_{2b} \Bigg) \Bigg]
\end{aligned}$$

The residual on the RHS is evaluated and used to update $p^{'}$ in the same way as in Equation 12.37. With these modifications for the diffusion, mass flux and pressure correction terms, the solution algorithm for the complex geometry case is very similar to that presented in Section 12.9.

12.11 Application to Fluid Flow and Heat Transfer Problem

Applications of the finite volume method are shown below for two types of problems: first, with simple geometry and orthogonal grids (Figure 12.1) and second, with complex geometry and nonorthogonal grids (Figure 12.6). Free stream flow across a square cylinder and flow in a square duct with a 180^0 bend are the first and second problems discussed below.

12.11.1 Free-Stream Flow across a Square Cylinder

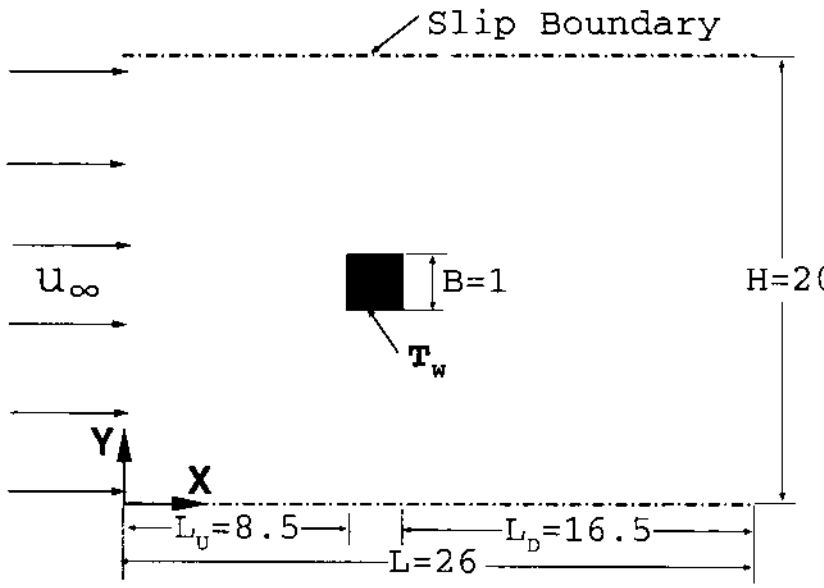


Figure 12.8: Computational domain for the flow around a square cylinder and the associated parameters. All distances are non dimensionalised with the width B of the square cylinder.

The flow configuration is shown in Figure 12.8. A fixed *two-dimensional* square cylinder with side B, which is also the non-dimensionalising length scale, is exposed to a constant free stream velocity represented by u_∞. In order to make the problem computationally feasible, artificial confining boundaries are placed around the flow where free slip boundary conditions are used. The vertical distance between the upper and lower walls, H=20B has been found to be sufficiently large to have little effect on the characteristics of the flow near the cylinder. The non-dimensional distance between the inlet plane and the front surface of the cylinder, L_U, is 8.5 and the non-dimensional distance

between the rear surface of the cylinder and the exit plane, L_D, is 16.5, with the total non-dimensional length of the computational domain, L=26.

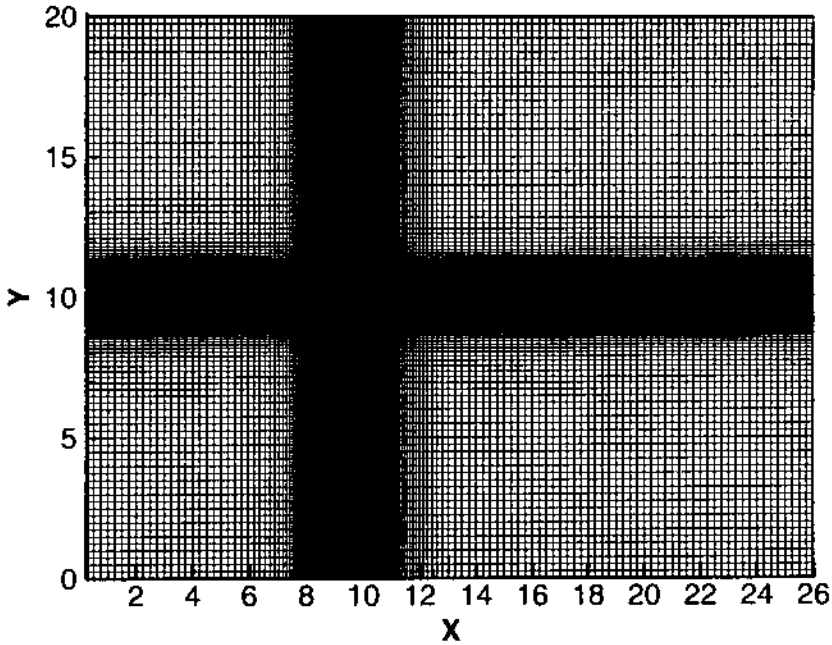


Figure 12.9: Non-uniform computational grid structure with 323 × 264 grids. Black patches are zones of very high grid density.

The grid structure employed is shown in Figure 12.9. The grid is divided into five separate zones in both directions, and uniform as well as nonuniform grid distributions are employed. The grid distribution is uniform with a constant cell size, $\Delta = 0.25$, outside a region around the cylinder that extends four units upstream, downstream, and sideways. Most of the action of fluid flow takes place near the cylinder. Therefore, a grid of much smaller size, $\delta = 0.01$, is clustered around the cylinder over a distance of 1.5 units, to adequately capture wake-wall interactions. The hyperbolic tangent function (Thompson *et al.*, 1985) is used for stretching the cell sizes between these limits (δ and Δ). A grid size of 323 × 264 has been used.

Figure 12.10 shows the computational results in the vicinity of the cylinder as the streamlines plots at Re=1, 2, 40, 50 (where Re $= u_\infty B/\nu$). It can be seen in Figure 12.10(a) that flow is fully attached with no separation at Re = 1. However, Figure 12.10(b) shows that the flow separates at the trailing

edge of the cylinder at Re=2. As a result, a closed steady recirculating region consisting of twin symmetric vortices (a separation bubble) forms behind the cylinder. This recirculating region increases in size with increasing Reynolds number and the flow remains steady at Re=40 as shown in Figure 12.10(c). However, by Re=50 (Figure 12.10(d)), the vortices in the separation bubble start to separate alternately from the trailing edge of the square cylinder and move downstream because of Bénard-von Kármán instability in the unsteady periodic flow phenomenon referred to as vortex shedding.

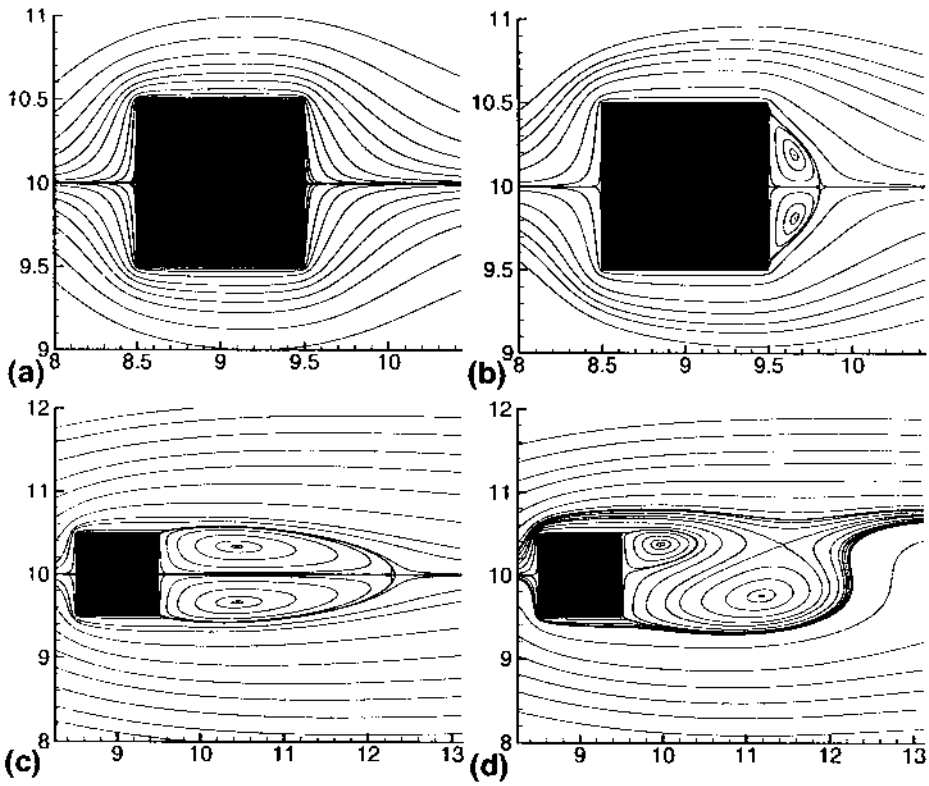


Figure 12.10: Streamlines around the square cylinder for different Reynolds number: (a) Re=1; (b) Re=2; (c) Re=40 and (d) Re=50.

The instantaneous streamlines in Figure 12.11(a)-(h) shows the detailed views of the laminar vortex shedding near the square cylinder at Re=100, for eight successive moments of time which span over one whole period of vortex shedding. Figure 12.11(d)-(g), shows a vortex shedding from the top of the cylinder while fluid from below the cylinder is drawn up into the recirculation

region. The vortex that is forming on the bottom of the rear face of the cylinder grows until it reaches the point where it breaks off, shown in Figure 12.11(h). The shedding process is repeated *ad infinitum*.

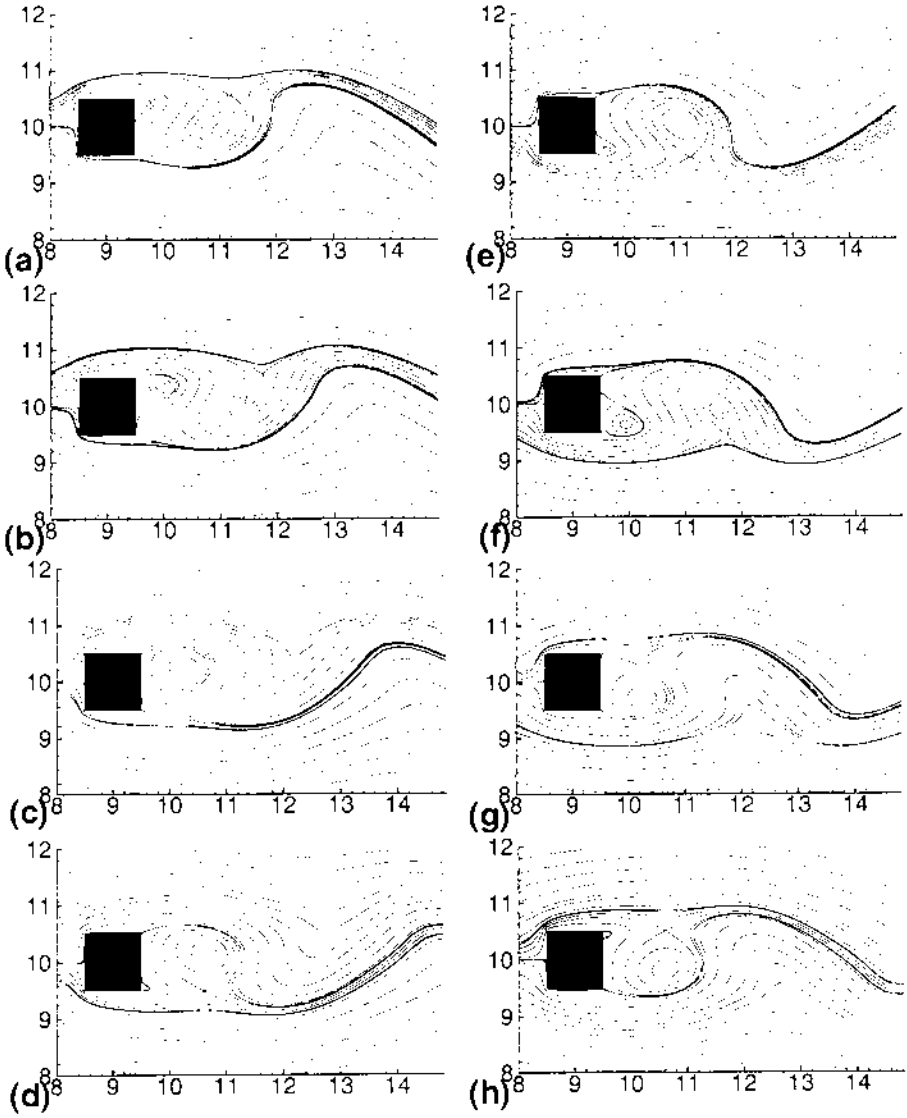


Figure 12.11: Instantaneous streamlines near the cylinder, separated by an interval of one-eighth of the time period of vortex shedding, at Re = 100.

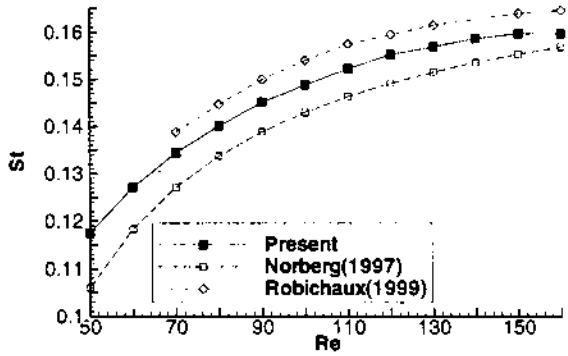


Figure 12.12: Variation of Strouhal number with Reynolds number.

An important quantity in unsteady periodic flow is the vortex shedding frequency f, normally expressed in non-dimensional terms as the Strouhal number ($\mathrm{St} = \frac{fB}{u_\infty}$). The time series for the lift coefficients can be used to determine the Strouhal number. Figure 12.12 shows the Strouhal versus Reynolds number variation obtained in these simulations along with the corresponding numerical results of Robichaux *et al.* (1999) and the experimental work of Norberg *et al.* (1997). All the results show a monotonic increase in Strouhal number with the Reynolds number.

12.11.2 Mixed Convective Free stream Flow across a Square Cylinder

In dealing with forced convection, we ignore the effects of free convection. Similarly, we assume forced convection is negligible in dealing with free convection. Situations may arise for which free and forced convection effects are comparable, in which case it is inappropriate to neglect either process. Hence the mixed (combined free and forced) convection is the case when an external flow is superposed on the buoyancy driven flow and there exists an imposed velocity. The corresponding non-dimensional governing equations (continuity, momentum and energy with Boussinesq approximation) for this problem are as follows:

$$
\begin{aligned}
\nabla \cdot \mathbf{U} &= 0 \\
\frac{\partial U}{\partial \tau} + \nabla \cdot (\mathbf{U} U) &= -\frac{\partial P}{\partial X} + \frac{1}{\mathrm{Re}} \nabla^2 U \\
\frac{\partial V}{\partial \tau} + \nabla \cdot (\mathbf{U} V) &= -\frac{\partial P}{\partial Y} + \frac{1}{\mathrm{Re}} \nabla^2 V + \mathrm{Ri}\,\theta \\
\frac{\partial \theta}{\partial \tau} + \nabla \cdot (\mathbf{U} \theta) &= \frac{1}{\mathrm{Re}\,\mathrm{Pr}} \nabla^2 \theta
\end{aligned}
$$

where all quantities are nondimensionlised and $\mathbf{U} = (U, V)$, where U is the horizontal and V is the vertical component of velocity, θ is the temperature and P is the pressure. The parameter that characterizes mixed convection is Richardson number, $\mathrm{Ri} = \mathrm{Gr}/\mathrm{Re}^2$ where $\mathrm{Gr} = g\beta(T_W - T_\infty)B^3/\nu^2$. If the flow is imposed in the upward direction, buoyancy acts upwards (along the flow direction) when Ri is positive, and downwards (opposite to the flow direction) when Ri is negative. Physically, the magnitude of Ri indicates the relative effect of natural to forced convection. The limiting values of $|\mathrm{Ri}| \to 0$ and $|\mathrm{Ri}| \to \infty$ corresponds to the forced and natural convection limits, respectively. Hence, the mixed convection regime is generally one for which $|\mathrm{Ri}| \approx 1$.

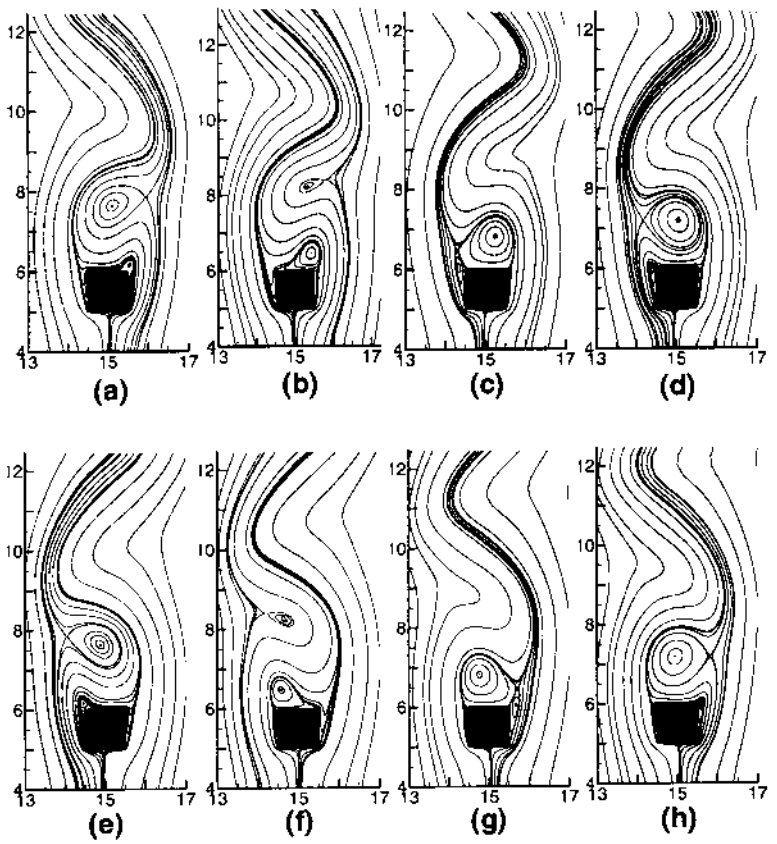


Figure 12.13: Instantaneous streamlines near the cylinder, separated by an interval of one-eighth of the time period of vortex shedding, at Re=100 and Ri=−0.25.

The effect of buoyancy on heat transfer in a forced flow is strongly influenced by the direction of the buoyancy force relative to that of the flow. Here, the upward forced motion of a fluid over a cooled cylinder has been considered. It corresponds to opposing flow (Ri < 0) case, where the buoyancy-induced and forced motion are in the the opposite directions. As the momentum and energy equation are coupled in this problem, they are solved simultaneously at each time step. A grid size of 304×376 has been used.

Figure 12.13 shows the detailed view of laminar vortex shedding at Re=100 and Ri=-0.25. As compared to the Figure 12.11 (Ri=0), it shows that the separation takes place from the leading edge of the cylinder at all times and the size of the wake is increased.

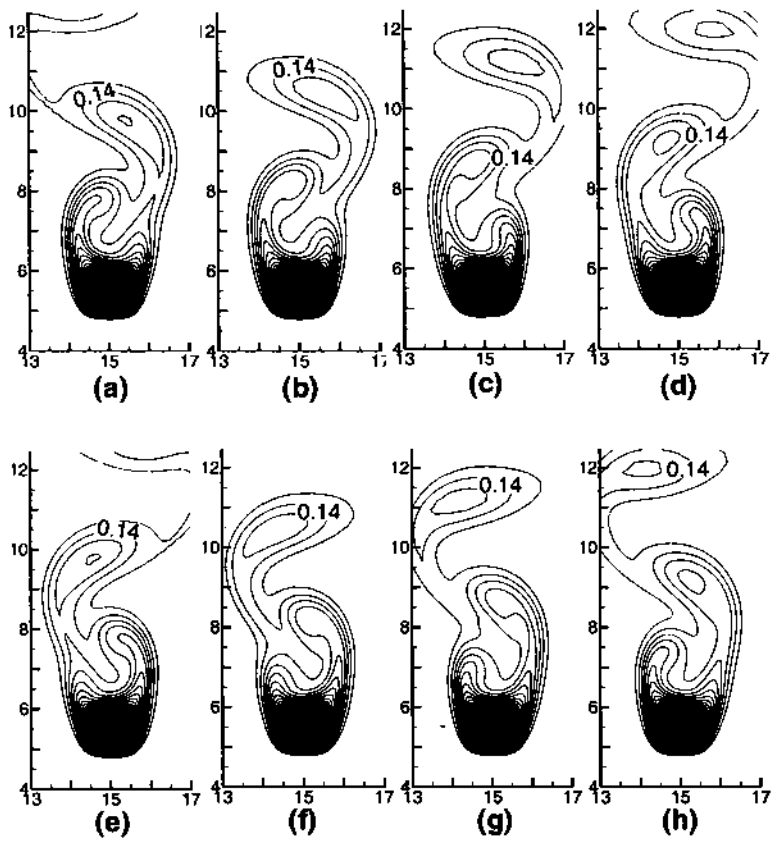


Figure 12.14: Instantaneous isotherms near the cylinder, separated by an interval of one-eighth of the time period of vortex shedding, at Re=100 and Ri=-0.25.

Figures 12.14(a)-(h) show the instantaneous isotherms near the square cylinder for Re= 100 and Ri=−0.25 at the same instants of time for which streamlines have been shown in Figure 12.13(a)-(h). The isotherms exhibit wavering motion as well as streamwise stretching.

12.11.3 Flow in a Square Duct with a 180^0 Bend

The problem considered here is the same in configuration and parameters as the experimental work of Phille *et al.* (1985) and is shown in isometric view in Figure 12.15. The model consists of a 180^0 degree bend of unit square cross section with a mean radius of curvature of R=6.45, and has straight extensions upstream and downstream that are 56 times the hydraulic diameter. The inlet section is sufficiently long to ensure fully developed flow.

The flow parameters are Reynolds number (Re=$\frac{V_c d}{\nu}$) of 574, where V_c is the bulk mean velocity and ν is the kinematic velocity of the fluid and a duct curvature ratio $\delta(\equiv$R/d)=6.45. This is a severe test case involving strong curvature effects. The present computations are done on a relatively coarse grid with 60 stations in the bend and a 21 × 21 cells in the cross plane.

The steady state solution is obtained by the false transient approach. The initial conditions are zero velocity conditions. The boundary conditions are no slip conditions at the walls , $u = 1$, $v = w = 0$ at inlet and $\frac{\partial u}{\partial x} = v = w = 0$ at outlet.

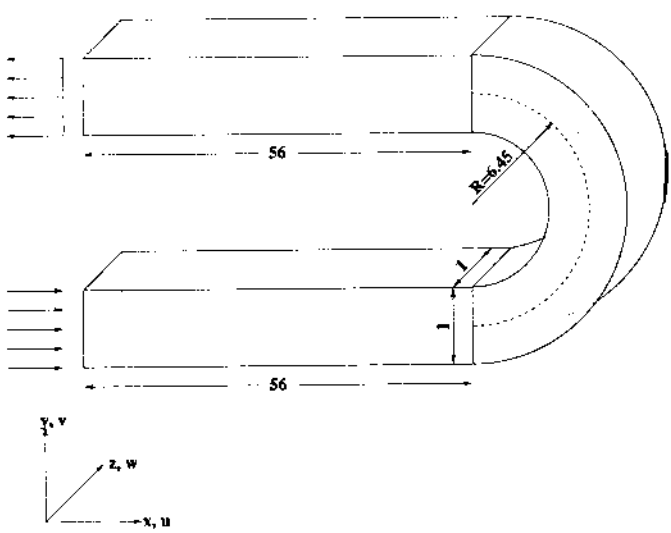


Figure 12.15: Schematic of 180^0 bend square duct.

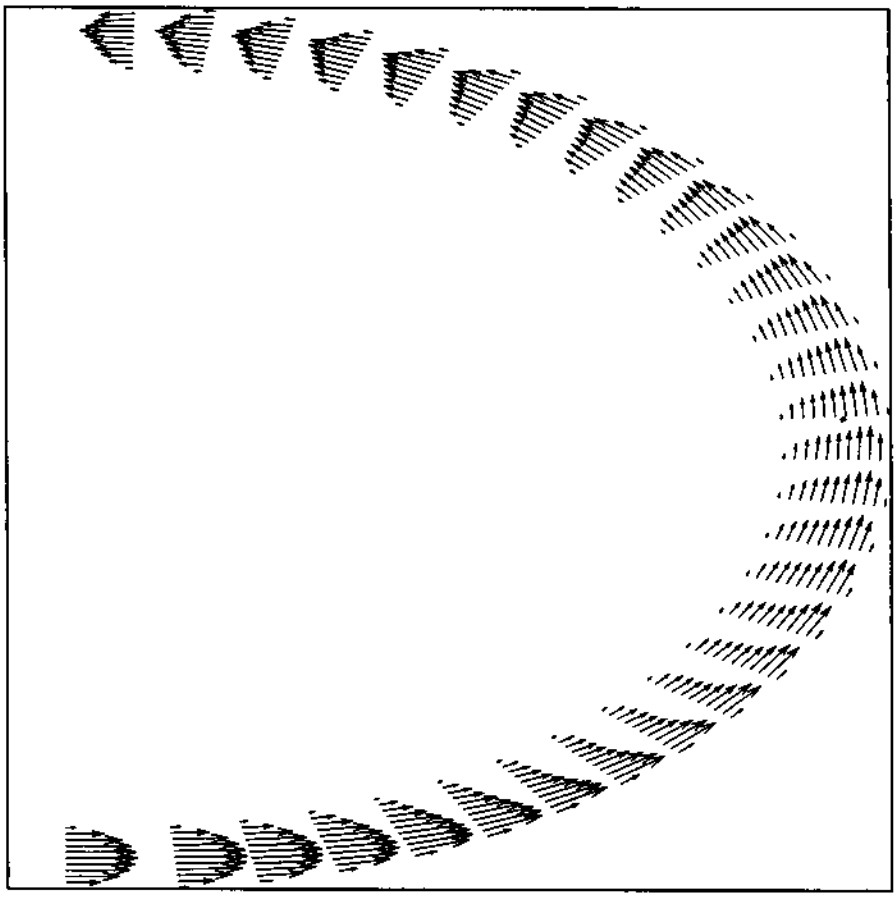


Figure 12.16: Streamwise velocity development in the bend shown as velocity vectors plotted in the x-y plane of symmetry.

The streamwise flow development of the steady state field is summarized in the Figure 12.16. Centrifugal forces arising due to the presence of longitudinal curvature causes the peak of the axial velocity profiles to shift towards the outer wall. The flow development in the bend is shown is Figure 12.17 which show the azimuthal velocity component U_ϕ measured as a function of radial coordinate in the mid plane of the cross section, $z = 0.5$, at intervals of 18^0. The figure show the comparison of the present results with the computed results of Kaushik and Rubin (1995), obtained on the same sized grid, and the experimental data (Phille *et al.*, 1985).

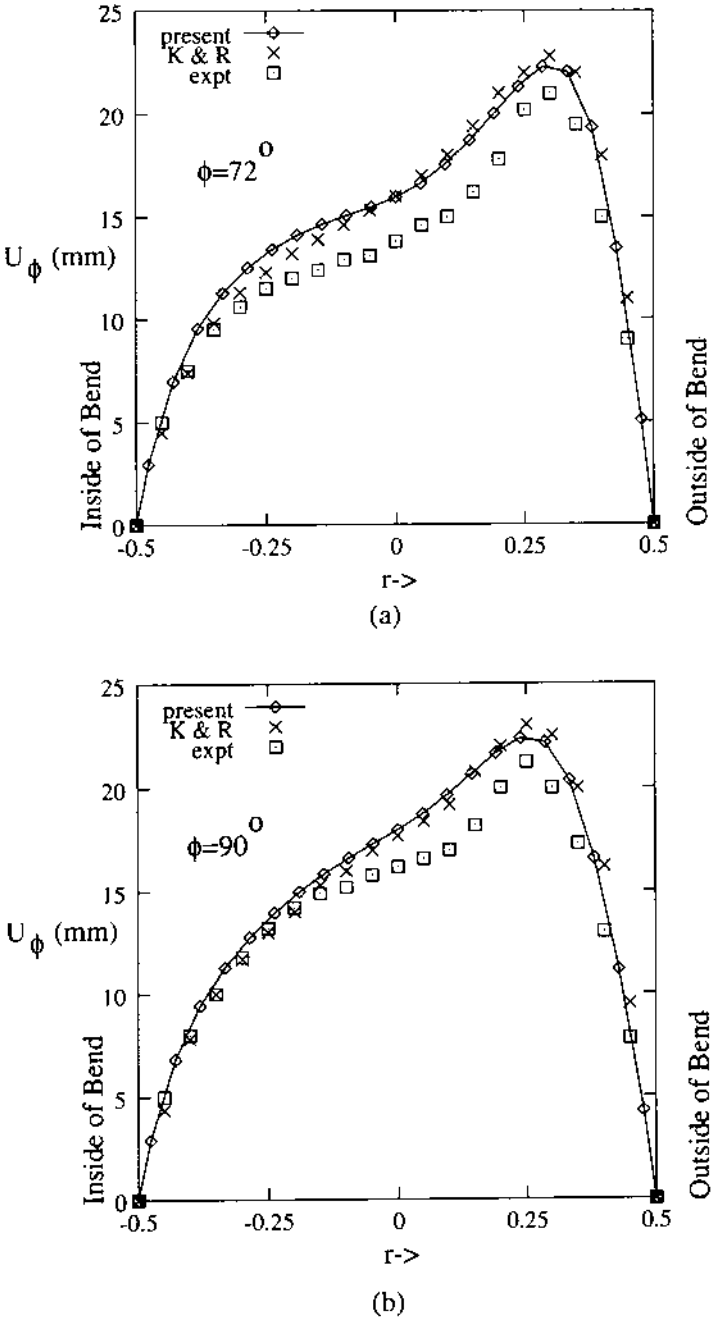


Figure 12.17: Mainstream velocity as a function of the radial location r at $z = 0.5$: (a) $\phi = 72^0$ and (b) $\phi = 90^0$.

References

[1] Kaushik, S. and Rubin S. G., Incompressible Navier Stokes solutions with a new primitive variable solver, Computer and Fluids, Vol. 24, pp. 27-40, 1995.

[2] Kordulla, W. and Vinokur, M., Efficient Computations of Volume in Flow Predictions, AIAA Journal, Vol.21, pp. 917-918, 1983.

[3] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, McGraw Hill, New York, 1980.

[4] Phille, P., Vehrenkamp, R., and Schulz-Dubois, E. O., The development and structure of primary and secondary flow in a curved duct., Journal of Fluid Mechanics, Vol. 151, pp. 219-241, 1985.

[5] Robichaux, J., Balachandar, S. and Vanka, S. P., Two-Dimensional Floquet Instability of the Wake of Square Cylinder, Phys. Fluids, Vol. 11, pp. 560-578, 1999.

[6] Sharma, A. and Eswaran, V., Conservative and Consistent Implementation and Systematic Comparison of SOU and QUICK Scheme for Recirculating Flow Computation on a Non-Staggered Grid, Proc. 2nd International and 29th National FMFP conference, Roorkee, India, pp. 351-358, 2002.

[7] Sohankar, A., Norberg, C. and Davidson, L., Numerical Simulation of Unsteady Low-Reynolds number flow around Rectangular Cylinders at Incidence, J. Wind Engg. Ind. Aerodyn., Vol. 69, pp. 189-201, 1997.

[8] Sohankar, A., Norberg, C. and Davidson, L., Low-Reynolds number flow around a Square Cylinder at Incidence: Study of Blockage, Onset of Vortex Shedding and Outlet Boundary Condition, Int. J. Num. Methods Fluids, Vol. 26, pp. 39-56, 1998.

[9] Thompson, J. F., Warsi, Z. U. A. and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, pp. 305-310, Elsevier Science, 1985.

13
Automatic Grid generation for Complex Geometry Problems

13.1 Introduction

The mesh employed for numerical simulation plays an important role towards determining the accuracy of the predicted solution. For instance, when the problem geometry is irregular, meshes which closely fit the boundary enable a better implementation of boundary conditions than rectangular meshes which envelope the domain. In the latter, the cells adjacent to the boundary will be incomplete. Over such cells tedious interpolation is required for satisfying the boundary conditions. Thus, the quality of solutions obtained on body-fitting grids (Figure 13.1) is expected to be better compared to the solutions on rectangular mesh envelopes.

A large number of problems require meshes with non-uniform grid spacing for obtaining accurate solutions without excessive computational effort. For

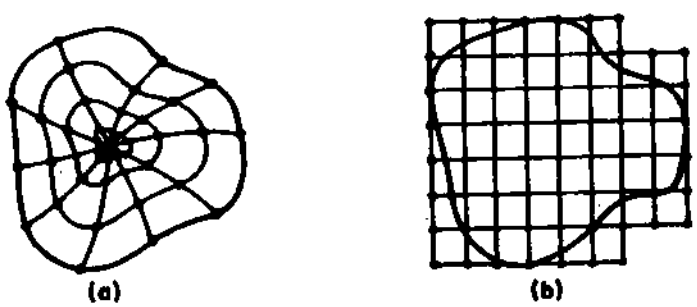


Figure 13.1: (a) Body-fitting and (b) Rectangular Envelope Grids

example, across boundary-layers or moving fronts such as shocks, flame sheets and phase-change boundaries, rapid changes in the velocity or temperature field occur. The grid spacing in these regions must be adequately small for accurate representation of the steep gradients. At this point, it is worth recalling from Chapters 4 and 5 that the discretization error has the form

$$\text{error} \approx \frac{\partial^m T}{\partial x^m} \Delta x^n$$

$(m > n)$ where n is the order of discretization. Clearly, it calls for small step sizes Δx in regions of large gradients. A large spacing may suffice in other zones where the variation in the solution is not as rapid. Thus an optimal grid point distribution within the solution domain greatly improves the accuracy and convergence of the numerical solution, at an affordable computational cost.

Grid generation could be a static process, i.e. the grid may be generated only once before the start of the numerical solution procedure. On the other hand, it could be adapted dynamically according to the evolution of computed solution with respect to time or iterations. In either case, it is essential that the grid generation process be automatic so that the drudgery of giving grid data manually is avoided.

Static grid generation is used when the requirement is to produce a body-fitting grid for a complex geometry. Adaptive grid generation is necessary when moving fronts are to be tracked accurately. In some of the adaptive procedures, a fixed number of grid points are distributed optimally, so as to accurately capture high gradient regions or to minimize some measure of error in the solution. A coordinated movement of grid points is required for this purpose so that no particular region has too many or too few grid points at any time.

The demands placed on the grid point distribution vary from problem to problem. While developing body-fitting grids, the chief concern is to ensure that the grid lines fit the boundary of the geometry well and they vary smoothly in the interior regions. It is also desirable to make the intersecting grid lines orthogonal to each other, if possible. In fact, if the angles between the grid lines become close to zero, the numerical solution will be highly inaccurate due to loss of independence between the respective coordinates. In adaptive grid generation, efficient grid spacing control is of primary interest. It is not possible to meet all of these demands simultaneously, as they are conflicting with each other in most situations. In any grid generation application, therefore, a particular grid property can be achieved satisfactorily, only with some compromise on the rest of the desired features.

13.2 Classification of Grids

Grids can be classified into two broad divisions, namely, (i) structured grids (ii) unstructured grids. In structured grid generation, the complex physical domain is transformed into a simple-shaped computational domain (usually rectangu-

lar), by the use of body-fitting coordinates. These coordinates may be curvilinear and non-orthogonal. Each segment of the physical domain boundary is described by a constant value of one of the curvilinear coordinates (Figure 13.2) and the entire mesh is formed by the intersection of the curvilinear coordinate lines. In the transformed domain, the grid is chosen to be rectangular and uniformly spaced.

Structured grids are primarily used for obtaining solutions based on finite-difference or finite-volume methods. This is in view of the fact that the number of nearest neighbours for each interior grid point is fixed in a structured mesh (i.e. 2,4,6 for one, two and three dimensions respectively). Further, since the grid lines are orthogonal in the transformed geometry, it is straight forward to convert all the governing partial differential equations into their equivalent difference forms.

In unstructured grid generation, usually no mapping is carried out. The physical domain as such is discretized algorithmically into small elements of desired shape. In most of the algorithms, no definite structure is maintained between any typical node and its neighbours. Therefore, unstructured meshes are best suited for finite element applications, where the solution procedure is performed in the physical domain itself. Although body-fitting coordinates are not employed, the overall mesh conforms to the body shape and the advantages of body- fitting grids (such as better representation of boundary conditions) are available.

13.3 Principles of Structured Mesh Generation

In structured mesh generation, a complex geometry is rendered simple through transformation, with the help of body-conforming coordinates. For instance, consider the domain $ABCD$ bounded by the lines $x = 0.5$; $y = 0.0$; $x = -0.5$ and the circular arc $y = (1 - x^2)^{0.5}$, (Figure 13.2(a)) and the transformation

$$\xi = \frac{y}{(1 - x^2)^{0.5}} \quad \text{and} \quad \eta = x + 0.5 \tag{13.1}$$

It is obvious that with the above transformation, $\xi = 0.0$ on line AB, $\eta = 1.0$ on BC, $\xi = 1.0$ on CD and $\eta = 0.0$ on AD. Thus, the transformation given by Equation 13.1 maps the domain $ABCD$ in the physical (x-y) plane onto a unit square in the transformed (ξ-η) plane. The coordinates ξ and η here are body-conforming. A grid formed by the intersection of the ξ = constant and η = constant lines in the physical domain shows the body-conforming nature of these coordinates quite clearly (Figure 13.2(b)). The corresponding grid in the transformed domain is evidently rectangular.

It should, however, be borne in mind that mapping of a given domain onto a rectangular region, is by no means unique. For instance, in the above example, one could select

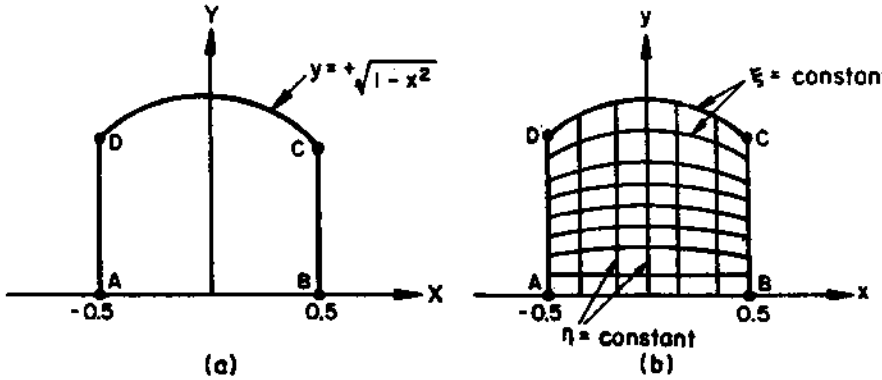


Figure 13.2: Structured Mesh Generation (a) Physical Region, (b) Body-Conforming Mesh

$$\xi = \frac{y^2}{(1-x^2)}; \quad \eta = x + 0.5 \tag{13.2}$$

In this case, if the ξ-constant lines are plotted with a uniform increment of $\Delta\xi$, the resulting grid will be more closely spaced near the circular arc boundary than for the transformation given by Equation 13.1. Apart from the variation due to the functional form of the body-fitting coordinates, the grid can differ depending on which part of the physical domain boundaries are mapped onto the different sides of the transformed domain. Indeed, this fact can be exploited advantageously to generate grid lines of desired shapes.

13.3.1 Mapping layouts

For multiply connected domains, three basic grid configurations are available. These are

(i) O- type grids

(ii) C-type grids

(iii) H- type grids.

For a given geometry, any one of these configurations can be obtained by suitable mapping.

Consider the multiply connected domain shown in Figure 13.3. If we introduce a branch-cut and identify points (A, B, C, D) on either side of the branch-cut as indicated in the figure, then O- type grids are obtained as follows. Map AB onto $A'B'$, BC onto $B'C'$, CD onto $C'D'$ and DA onto $D'A'$, where primes denote points in the transformed domain. Thus, the object boundary AB and the external boundary (CD) become opposite sides of the transformed domain.

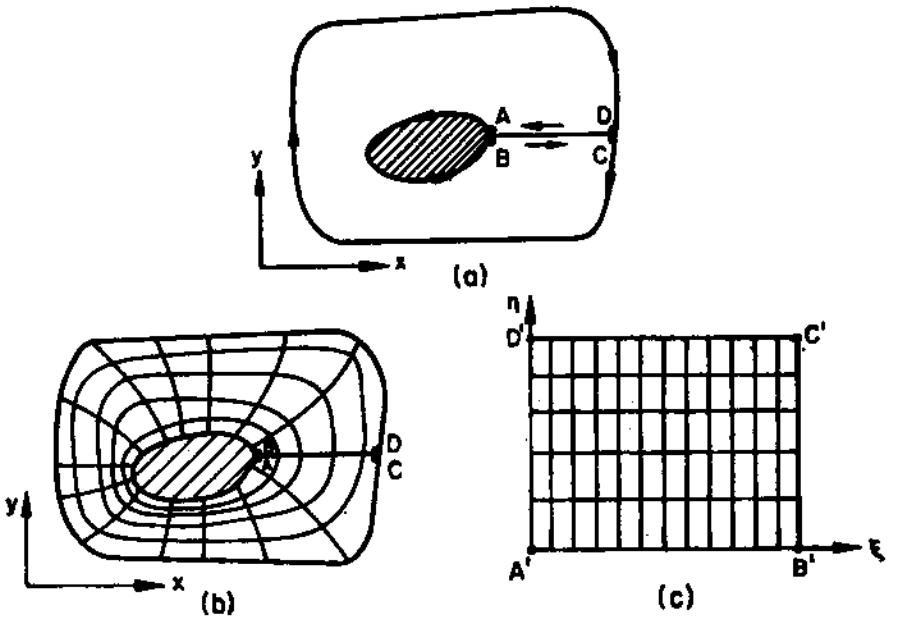


Figure 13.3: Generation of O-Grid

The two sides of the branch-cut (BC and AD) are also mapped onto two opposite sides of the rectangular domain. Now, a grid constructed by ξ- constant and η-constant lines in the physical domain will be O-type, as shown in Figure 13.3.

For C-type transformation of the same multiply connected domain, a branch-cut is introduced and two points A and B are identified (Figure 13.4) where the branch-cut meets the outer boundary. Points C and D are suitably selected on the external boundary and mapping is carried out with AB onto $A'B'$, BC onto $B'C'$, CD onto $C'D'$ and DA onto $D'A'$. Note that the forward sweep of the branch-cut (AP), the object surface (PQ) and the reverse sweep of the branch-cut (QB) comprise one side $A'B'$ of the transformed region. The object surface is mapped onto the patch $P'Q'$ on this side. It can be seen that in this transformation, the η-constant lines in the interior envelop the object and the branch-cut, thus forming a C-type configuration.

For H-type configuration, branch cuts are introduced on either side of the object and the upper and lower portions ($ABCD$ and $EFGH$) are separately mapped onto $A'B'C'D'$ and $E'F'G'H'$ in the proper sequence (Figure 13.5). Here, the object reduces to a line $P'Q'$ in the middle of the transformed domain.

It is evident from these examples that by choosing the mapping configuration, different types of grids can be generated for the same geometry. The appropriate choice depends on the nature of the problem to be solved. For complex domains with many objects, it may be necessary to map different regions separately, using local transformations. A variety of grid layouts such as the overlaid grids and embedded grids can be achieved through such procedures

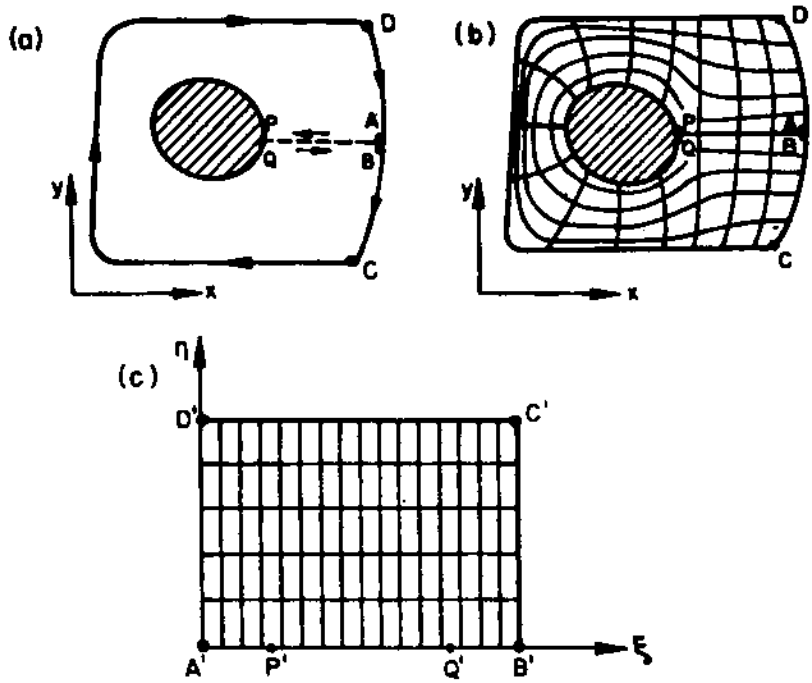


Figure 13.4: C-type Grid Generation

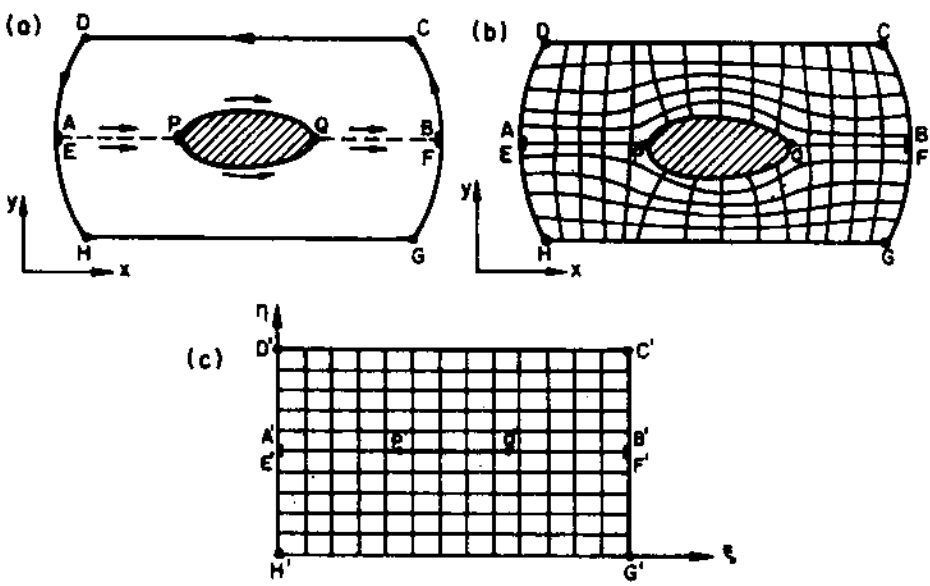


Figure 13.5: H-type Grid Generation

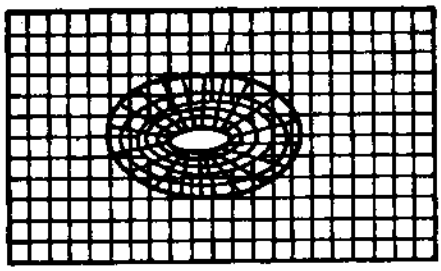


Figure 13.6: Overlaid grid (O-grid on Rectangular Grid)

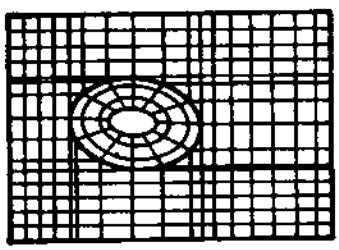


Figure 13.7: Embedded Grid (O-grid inside Rectangular Grid)

(Figure 13.6 and 13.7).

13.3.2 Transformation Relations

It was seen in the previous section that using body-fitting coordinates, a complex geometry can be transformed into a simpler shape. The price to be paid for this simplification is that the governing equations of the problem become complex, in the body- fitting coordinate system. However, since the discretization procedures for various linear/non-linear differential operators are well established in finite difference and finite volume methods, dealing with complex differential equations is not a matter of serious concern.

Let the body-fitting coordinate transformation be of the form $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$. From chain rule

$$\left\{ \begin{array}{c} d\xi \\ d\eta \end{array} \right\} = \left[\begin{array}{ccc} \xi_x dx & + & \xi_y dy \\ \eta_x dx & + & \eta_y dy \end{array} \right] = \left[\begin{array}{cc} \xi_x & \xi_y \\ \eta_x & \eta_y \end{array} \right] \left\{ \begin{array}{c} dx \\ dy \end{array} \right\} \tag{13.3}$$

where (ξ_x, ξ_y) and (η_x, η_y) are the partial derivatives of ξ and η with respect to x and y respectively. In a similar fashion, considering $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$, we obtain

$$\left\{ \begin{array}{c} dx \\ dy \end{array} \right\} = \left[\begin{array}{cc} x_\xi & x_\eta \\ y_\xi & y_\eta \end{array} \right] \left\{ \begin{array}{c} d\xi \\ d\eta \end{array} \right\} \tag{13.4}$$

Combining Equations 13.3 and 13.4

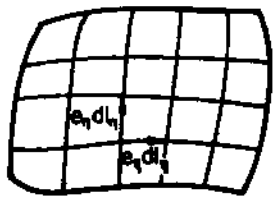


Figure 13.8: Transformation of Arc Lengths and Areas

$$\begin{Bmatrix} dx \\ dy \end{Bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{Bmatrix} dx \\ dy \end{Bmatrix} \tag{13.5}$$

or

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}^{-1} = \frac{1}{\mid J \mid} \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix} \tag{13.6}$$

where $\mid J \mid = x_\xi y_\eta - y_\xi x_\eta$ is the determinant of the Jacobian matrix

$$[J] = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \tag{13.7}$$

For one-to-one mapping to exist, the determinant $\mid J \mid$ should be finite and non-zero. From Equation 13.6, it is clear that the partial derivatives of ξ and η are related to those of x and y through the relations

$$\begin{aligned} \xi_x &= y_\eta / \mid J \mid; \qquad & \xi_y &= -x_\eta / \mid J \mid; \\ \eta_x &= -y_\xi / \mid J \mid; \qquad & \eta_y &= x_\xi / \mid J \mid . \end{aligned} \tag{13.8}$$

For three dimensional situations also, one can similarly derive the relationships between the derivatives of the cartesian coordinates (x, y, z) and curvilinear coordinates (ξ, η, ζ) in a matrix form as:

$$\begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}^{-1} \tag{13.9}$$

Now let us consider the transformation of an area element. Referring to Figure 13.8, the area vector $\hat{k}dA$ is

$$\hat{k}dA = \hat{e}_\xi \, d\ell_\xi \times \hat{e}_\eta \, d\ell_\eta \tag{13.10}$$

where $d\ell_\xi$ is the arc length in ξ-direction on a η-constant line and $d\ell_\eta$ is the arc length in η-direction on a ξ-constant line.

The unit vectors in the ξ- and η-directions are given by $\hat{e}_\xi$ and $\hat{e}_\eta$ respectively. On a η = constant line, it is possible to write

$$\hat{e}_\xi d\ell_\xi = \hat{i}dx + \hat{j}dy = \hat{i}x_\xi d\xi + \hat{j}y_\xi d\xi \tag{13.11}$$

where dx and dy are the projections of $d\ell_\xi$ in the x and y directions. Similarly

$$\hat{e}_\eta d\ell_\eta = \hat{i}dx + \hat{j}dy = \hat{i}x_\eta d\eta + \hat{j}y_\eta d\eta \tag{13.12}$$

on a ξ = constant curve. Using Equations 13.11 and 13.12 in Equation 13.10

$$\hat{k}dA = \hat{e}_\xi\, d\ell_\xi \times \hat{e}_\eta\, d\ell_\eta = \hat{k}\ (x_\xi\, y_\eta \quad y_\xi\, x_\eta)\, d\xi d\eta = \hat{k} \mid J \mid d\xi d\eta \tag{13.13}$$

The corresponding area in the transformed domain is $d\xi \cdot d\eta$, since the curvilinear cell made up of $d\ell_\xi$ and $d\ell_\eta$ in the physical domain will map into a rectangle with sides $d\xi$ and $d\eta$. Thus, it is clear that the determinant $\mid J \mid$ represents the local area scaling factor for two-dimensional domains. In 3-D, it can be shown that the volume of a curvilinear element dv is:

$$dv = (\hat{e}_\zeta d\ell_\zeta) \cdot (\hat{e}_\xi d\ell_\xi \times \hat{e}_\eta d\ell_\eta) = \mid J \mid d\xi d\eta d\zeta \tag{13.14}$$

where $\mid J \mid$ is the determinant of the matrix

$$[J] = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \tag{13.15}$$

Therefore, $\mid J \mid$ represents the local volume scaling factor. From the above discussions, it is evident that $\mid J \mid = 0$ will lead to the situation of an elemental area (or volume) in the transformed domain being mapped onto a point in the physical domain. Such a mapping is not one-to-one and therefore, is to be avoided. In fact, for preserving proper orientation of the area vector, $\mid J \mid$ should be non-zero.

13.3.3 Transformed Governing Equations

Let us first consider the transformation of derivatives of a scalar field variable $T(x, y)$. From chain rule

$$\frac{\partial T}{\partial x} = T_x = \xi_x \frac{\partial T}{\partial \xi} + \eta_x \frac{\partial T}{\partial \eta} \tag{13.16}$$

and

$$\frac{\partial T}{\partial y} = T_y = \xi_y \frac{\partial T}{\partial \xi} + \eta_y \frac{\partial T}{\partial \eta} \tag{13.17}$$

The second derivatives are

$$\begin{aligned}\frac{\partial^2 T}{\partial x^2} &= T_{xx} = \frac{\partial}{\partial x}\left\{\xi_x \frac{\partial T}{\partial \xi} + \eta_x \frac{\partial T}{\partial \eta}\right\} = \xi_{xx}\frac{\partial T}{\partial \xi} + \eta_{xx}\frac{\partial T}{\partial \eta} \\ &+ \xi_x^2 \frac{\partial^2 T}{\partial \xi^2} + \eta_x^2 \frac{\partial^2 T}{\partial \eta^2} + 2\xi_x\eta_x \frac{\partial^2 T}{\partial \xi \partial \eta}\end{aligned} \tag{13.18}$$

$$\begin{aligned}\frac{\partial^2 T}{\partial y^2} &= T_{yy} = \frac{\partial}{\partial y}\left\{\xi_y \frac{\partial T}{\partial \xi} + \eta_y \frac{\partial T}{\partial \eta}\right\} = \xi_{yy}\frac{\partial T}{\partial \xi} + \eta_{yy}\frac{\partial T}{\partial \eta} \\ &+ \xi_y^2 \frac{\partial^2 T}{\partial \xi^2} + \eta_y^2 \frac{\partial^2 T}{\partial \eta^2} + 2\xi_y\eta_y \frac{\partial^2 T}{\partial \xi \partial \eta}\end{aligned} \tag{13.19}$$

Hence

$$\begin{aligned}\nabla^2 T &= T_{xx} + T_{yy} = \left(\nabla^2 \xi\right)\frac{\partial T}{\partial \xi} + \left(\nabla^2 \eta\right)\frac{\partial T}{\partial \eta} + \left(\xi_x^2 + \xi_y^2\right)\frac{\partial^2 T}{\partial \xi^2} \\ &+ \left(\eta_x^2 + \eta_y^2\right)\frac{\partial^2 T}{\partial \eta^2} + 2\left(\xi_x\eta_x + \xi_y\eta_y\right)\frac{\partial^2 T}{\partial \xi \partial \eta}\end{aligned} \tag{13.20}$$

The above derivation shows that even the simple Laplace operator takes a complicated form when transformed into the (ξ, η) coordinates. This operator is of great interest as it arises in problems involving ideal fluid flows and diffusion processes. The importance of achieving some of the grid properties discussed earlier can be seen from the transformation of the Laplace operator. If it is desired to have the body-fitting coordinates ξ and η as smooth functions of (x, y), they can be chosen as harmonic functions, i.e. functions which satisfy

$$\nabla^2 \xi = 0$$

and

$$\nabla^2 \eta = 0$$

In this context, the first two terms in the right hand side of Equation 13.20 drop out and the expression for $\nabla^2 T$ simplifies. Additionally, the metric derivatives, for example ξ_x and ξ_y appearing in the coefficients of the transformed equations also vary smoothly over the domain, which is necessary for accurate prediction.

For the grid lines to be mutually orthogonal, they must satisfy the relation $\nabla\xi \cdot \nabla\eta = 0$. In expanded two dimensional form

$$\xi_x\eta_x + \xi_y\eta_y = 0 \tag{13.21}$$

Thus, for an orthogonal mesh, the cross derivative term drops out from the transformed Equation 13.20. In finite difference discretization, the cross derivative is cumbersome to handle and it is desired that this term be absent. Finally, if the mapping between (ξ, η) and (x, y) is conformal, the grid lines are orthogonal and smooth. Moreover, the coefficients $\left(\xi_x^2 + \xi_y^2\right)$ and $\left(\eta_x^2 + \eta_y^2\right)$ are equal

to each other. Thus, the Laplace equation remains invariant under conformal transformation, leading to

$$T_{\xi\xi} + T_{\eta\eta} = 0$$

Consider the transformation of the two- dimensional steady state heat conduction equation with a non- uniform heat source $Q(x, y)$. For a smooth grid

$$k\left\{\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right\} + Q(x,y) = k\left\{\left(\xi_x^2 + \xi_y^2\right)\frac{\partial^2 T}{\partial \xi^2} + \left(\eta_x^2 + \eta_y^2\right)\frac{\partial^2 T}{\partial \eta^2}\right.$$
$$\left. + \; 2\left(\xi_x\eta_x + \xi_y\eta_y\right)\frac{\partial^2 T}{\partial \xi \partial \eta}\right\} + \hat{Q}\left(\xi, \eta\right) = 0 \qquad (13.22)$$

In the above equation, $\hat{Q}\left(\xi, \eta\right)$ is the volumetric heat source term expressed in (ξ, η) coordinates. It is convenient to write the metric derivatives in Equation 13.22 in a form which can be evaluated on the computational domain (ξ, η). Substituting relations 13.8 into Equation 13.22

$$\frac{k}{\mid J \mid^2}\left\{\left(x_\eta^2 + y_\eta^2\right)\frac{\partial^2 T}{\partial \xi^2} + \left(x_\xi^2 + y_\xi^2\right)\frac{\partial^2 T}{\partial \eta^2} - 2\left(y_\eta y_\xi + x_\eta x_\xi\right)\frac{\partial^2 T}{\partial \xi \partial \eta}\right\}$$
$$+ \; \hat{Q}\left(\xi, \eta\right) = 0 \qquad (13.23)$$

The metric derivatives x_ξ, x_η, y_ξ and y_η in Equation 13.23 can be handled by finite differencing, in exactly the same manner as the derivatives of the solution variable, over the uniformly spaced computational grid.

For transient problems, the grid points may move in the physical domain, while they can be made stationary in the transformed domain, for convenience. This can be achieved by the transformation

$$\begin{aligned} & t = \tau; \quad x = x\left(\xi, \eta, \tau\right) \quad \text{and} \quad y = y\left(\xi, \eta, \tau\right) \\ \text{or} \quad & \tau = t; \quad \xi = \xi\left(x, y, t\right) \quad \text{and} \quad \eta = \eta\left(x, y, t\right) \end{aligned} \qquad (13.24)$$

The transient heat conduction equation is transformed in the body-fitting coordinate system as

$$\rho C_p \frac{\partial T}{\partial t} - k\left\{\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right\} - Q(x,y) = \rho C_p\left(\frac{\partial T}{\partial \tau} + \xi_t\frac{\partial T}{\partial \xi} + \eta_t\frac{\partial T}{\partial \eta}\right)$$
$$- \frac{k}{\mid J \mid^2}\left\{\left(x_\eta^2 + y_\eta^2\right)\frac{\partial^2 T}{\partial \xi^2} + \left(x_\xi^2 + y_\xi^2\right)\frac{\partial^2 T}{\partial \eta^2}\right.$$
$$\left. - \; 2\left(y_\eta y_\xi + x_\eta x_\xi\right)\frac{\partial^2 T}{\partial \xi \partial \eta}\right\} - \hat{Q}\left(\xi, \eta\right) = 0$$

The terms ξ_t and η_t are evaluated from the matrix relation

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ 1 \end{Bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \xi_t \\ \eta_t \\ 1 \end{Bmatrix} \tag{13.25}$$

after substituting $\dot{x} = dx/dt = 0$ and $\dot{y} = dy/dt = 0$. Rewriting the coefficients in the above equation in terms of the derivatives of the physical coordinates, one obtains

$$\begin{aligned} \rho C_p \left(\frac{\partial T}{\partial \tau} \right. &+ \left. \frac{(x_\eta y_\tau - y_\eta x_\tau)}{|J|} \frac{\partial T}{\partial \xi} + \frac{(y_\xi x_\tau - x_\xi y_\tau)}{|J|} \frac{\partial T}{\partial \eta} \right) \\ &- \frac{k}{|J|^2} \left\{ (x_\eta^2 + y_\eta^2) \frac{\partial^2 T}{\partial \xi^2} + (x_\xi^2 + y_\xi^2) \frac{\partial^2 T}{\partial \eta^2} \right. \\ &- \left. 2(y_\eta y_\xi + x_\eta x_\xi) \frac{\partial^2 T}{\partial \xi \partial \eta} \right\} - \hat{Q}(\xi, \eta) = 0 \end{aligned} \tag{13.26}$$

Here, x_τ and y_τ are the grid velocities in the respective directions. It is interesting to note that although the original problem in the physical domain has no convective contributions, in the transformed domain, convection-like terms arise due to grid point motion with respect to the computational time coordinate τ.

In the above examples, we illustrated the transformation of scalar equations. For vector equations such as the Navier-Stokes equations, three different approaches can be used. The vector velocity field can be described in terms of its covariant, contravariant or cartesian components and the transformed flow equations can be obtained for any of these three systems. The covariant velocity components have directions parallel to the body-fitting coordinates (ξ, η, ζ) at each point. The unit vectors $\hat{e}_\xi$, $\hat{e}_\eta$ and $\hat{e}_\zeta$ in these directions can be obtained from the expressions for the arc lengths along the body-fitting coordinates. The arc lengths are

$$\hat{e}_\xi d\ell_\xi = \hat{i} x_\xi d\xi + \hat{j} y_\xi d\xi + \hat{k} z_\xi d\xi \tag{13.27}$$

$$\hat{e}_\eta d\ell_\eta = \hat{i} x_\eta d\eta + \hat{j} y_\eta d\eta + \hat{k} z_\eta d\eta \tag{13.28}$$

$$\hat{e}_\zeta d\ell_\zeta = \hat{i} x_\zeta d\zeta + \hat{j} y_\zeta d\zeta + \hat{k} z_\zeta d\zeta \tag{13.29}$$

While deriving the above relations, we have used the fact that along the curvilinear coordinate lines, only one variable (ξ, η or ζ) changes. From the above equations, it is easy to show that

$$\hat{e}_\xi = \frac{\hat{e}_\xi d\ell_\xi}{d\ell_\xi} = \frac{\hat{i} x_\xi + \hat{j} y_\xi + \hat{k} z_\xi}{\sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2}} \tag{13.30}$$

$$\hat{e}_\eta = \frac{\hat{e}_\eta d\ell_\eta}{d\ell_\eta} = \frac{\hat{i} x_\eta + \hat{j} y_\eta + \hat{k} z_\eta}{\sqrt{x_\eta^2 + y_\eta^2 + z_\eta^2}} \tag{13.31}$$

$$\hat{e}_\zeta = \frac{\hat{e}_\zeta d\ell_\zeta}{d\ell_\zeta} = \frac{\hat{i}x_\zeta + \hat{j}y_\zeta + \hat{k}z_\zeta}{\sqrt{x_\zeta^2 + y_\zeta^2 + z_\zeta^2}} \tag{13.32}$$

The covariant velocity component along the ξ-coordinate line is

$$U_\xi = \mathbf{V} \cdot \hat{e}_\xi = \frac{ux_\xi + vy_\xi + wz_\xi}{\sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2}} \tag{13.33}$$

where u, v, w are the cartesian components of velocity vector $\mathbf{V}$. The other covariant velocity components U_η and U_ζ, can also be similarly obtained.

The contravariant unit vectors $\hat{e}^\xi$, $\hat{e}^\eta$ and $\hat{e}^\zeta$ are defined as the unit normals to the ξ-, η- and ζ-constant planes respectively. In order to distinguish between the covariant and contravariant components, we shall denote the former by subscripts and the latter by superscripts. The covariant and contravariant systems coincide with each other only in the case of orthogonal coordinates. The contravariant unit vectors are

$$\hat{e}^\xi = \frac{\nabla\xi}{|\nabla\xi|} = \frac{\hat{i}\xi_x + \hat{j}\xi_y + \hat{k}\xi_z}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}} \tag{13.34}$$

$$\hat{e}^\eta = \frac{\nabla\eta}{|\nabla\eta|} = \frac{\hat{i}\eta_x + \hat{j}\eta_y + \hat{k}\eta_z}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}} \tag{13.35}$$

$$\hat{e}^\zeta = \frac{\nabla\zeta}{|\nabla\zeta|} = \frac{\hat{i}\zeta_x + \hat{j}\zeta_y + \hat{k}\zeta_z}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}} \tag{13.36}$$

For deriving the above expressions, we have used the property that the gradient of a scalar field $\nabla\phi$ is perpendicular to the ϕ = constant surface, at any point. The contravariant velocity component in $\hat{e}^\xi$ direction is obtained as

$$U^\xi = \mathbf{V} \cdot \hat{e}^\xi = \frac{u\xi_x + v\xi_y + w\xi_z}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}} \tag{13.37}$$

The components in the other two directions can be obtained by expanding the dot products of $(\mathbf{V} \cdot \hat{e}^\eta)$ and $(\mathbf{V} \cdot \hat{e}^\zeta)$. Let us now consider the transformation of the continuity equation for a two-dimensional incompressible flow. In terms of the covariant components U_ξ and U_η, the equation becomes

$$\nabla \cdot \mathbf{V} = 0 = \frac{1}{|J|}\left\{\frac{\partial}{\partial\xi}(y_\eta U_\xi - x_\eta U_\eta) + \frac{\partial}{\partial\eta}(-y_\xi U_\xi + x_\xi U_\eta)\right\} \tag{13.38}$$

The corresponding form in terms of the contravariant components is

$$\nabla \cdot \mathbf{V} = 0 = \frac{1}{\mid J \mid} \left\{ \frac{\partial}{\partial \xi} (\mid J \mid U^{\xi}) + \frac{\partial}{\partial \eta} (\mid J \mid U^{\eta}) \right\} \tag{13.39}$$

If the cartesian velocity components (u, v) are themselves used, the transformed equation is obtained as

$$\nabla \cdot \mathbf{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \xi_x \frac{\partial u}{\partial \xi} + \eta_x \frac{\partial u}{\partial \eta} + \xi_y \frac{\partial v}{\partial \xi} + \eta_y \frac{\partial v}{\partial \eta} = 0 \tag{13.40}$$

The complete Navier-Stokes equations can be similarly transformed using the covariant components (U_ξ, U_η, U_ζ), contravariant components (U^ξ, U^η, U^ζ) or the cartesian components (u, v, w). General transformation rules for the covariant and contravariant systems are available and vector operators such as the gradient, divergence and curl, or scalar operators such as the Laplacian can be transformed directly with the help of these rules (Thompson *et al.*, 1985; Warsi, 1992). Thus, the use of the covariant or contravariant components provides a generalized approach for solving vectorial (coordinate-free) governing equations. However, interpretation of the computed results is a complicated procedure for the velocity components in the curvilinear systems.

13.4 Structured Grid Generation Techniques

The techniques available for generating structured grids can be divided into three major categories. These are

(i) algebraic methods

(ii) conformal mapping and

(iii) methods using partial differential equations.

Among these procedures, algebraic grid generation is efficient and it can be applied to 2-D or 3-D geometries. Some explicit control of grid spacing is also possible. However, slope discontinuities present at the boundaries tend to propagate to the interior and in such cases, the grid suffers from lack of smoothness. Conformal mapping provides a smooth orthogonal grid except at a few singular points, though its applicability to three dimensional problems is rather limited. Partial differential equation based methods generate grids which exhibit a good compromise between the various desired grid properties. They have a wide range of applicability. But the implementation of PDE-based grid generation is cumbersome and time consuming. Here each of the above-mentioned methods for the generation of structured grids is described.

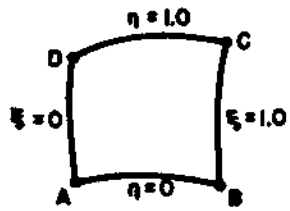


Figure 13.9: Algebraic Transformation

13.4.1 Algebraic Grid Generation

In algebraic methods, the transformation could be done analytically when the boundaries are regular. On the other hand, if the boundary is highly irregular and complex, the domain transformation is carried out numerically using suitable interpolation procedures. These ideas are illustrated in the following examples.

Consider the simply connected domain shown in Figure 13.9 whose sides AB, BC, CD and DA are given by the equations $f_1(x,y) = 0$, $f_2(x,y) = 0$, $f_3(x,y) = 0$ and $f_4(x,y) = 0$ respectively.

Without loss of generality, one can map the curves AB and CD onto the lines $\eta = 0$ and $\eta = 1.0$, using a transformation of the form

$$\eta = \frac{f_1(x,y)}{\{f_1(x,y) - f_3(x,y)\}} \tag{13.41}$$

Similarly, it can be assumed that

$$\xi = \frac{f_4(x,y)}{\{f_4(x,y) - f_2(x,y)\}} \tag{13.42}$$

Now let us consider some specific shapes of the boundaries and construct the corresponding grids. Let $f_1(x,y) = y$, $f_2(x,y) = 4 - y - 2x$, $f_3(x,y) = 2y - x - 2$ and $f_4(x,y) = y - 2x$. Using Equations 13.41 and 13.42

$$\eta = \frac{y}{(2 + x - y)} \quad \xi = \frac{(y - 2x)}{(2y - 4)} \tag{13.43}$$

The grid lines are obtained by constructing the ξ- and η-constant lines in the range $0 \leq \xi \leq 1$ and $0 \leq \eta \leq 1$ (Figure 13.10). The inverse relationships $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$ are obtained as

$$x = \frac{\{4\xi(\eta + 1) - 2\eta(2\xi - 1)\}}{(2\xi\eta + \eta + 2)}$$

$$y = \frac{4\eta(1 + \xi)}{(2\xi\eta + \eta + 2)} \tag{13.44}$$

All the metric derivatives can be analytically evaluated from Equations 13.43 and 13.44. In the present example, the grid is smooth and the metric derivatives

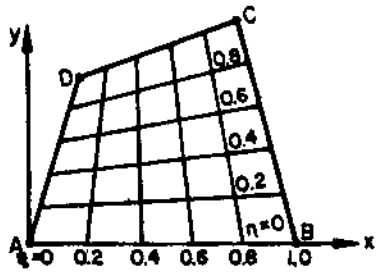


Figure 13.10: Transformation of a Quadrilateral

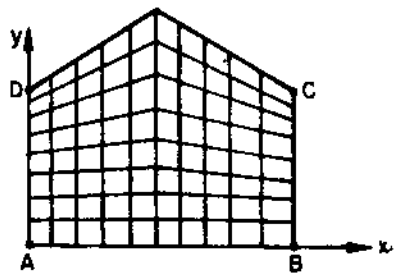


Figure 13.11: Non-smooth Grid in Algebraic Transformation

exist everywhere. Consider another simply connected geometry with boundaries given by

$$f_1(x,y) = y = 0 \quad f_2(x,y) = x - 2 = 0$$

$$f_3(x,y) = \begin{cases} 2y - x - 2 = 0 & \text{for} \quad 0 \le x \le 1 \\ 2y + x - 4 = 0 & \text{for} \quad 1 \le x \le 2 \end{cases} \quad ; \quad f_4(x,y) = x = 0$$

Here, the transformation functions are

$$\begin{aligned} \eta &= \frac{y}{(2 + x - y)} \quad \text{for} \quad 0 \le x \le 1 \\ \eta &= \frac{y}{(4 - x - y)} \quad \text{for} \quad 1 \le x \le 2 \\ \xi &= \frac{x}{2} \end{aligned} \tag{13.45}$$

The corresponding grid is shown in Figure 13.11. This grid is not smooth along the line $x = 1.0$, as the effect of the boundary discontinuity on the top side propagates to the interior. Although the metric derivatives do not exist on this line and they cannot be evaluated analytically, finite difference implementations tend to make the discontinuity smooth. Therefore, difference expressions for x_ξ and y_ξ do not exhibit any anomalies.

For a multiply connected geometry, for example, the annular region lying between a circle and an ellipse shown in Figure 13.12, O-type grid can be generated as:

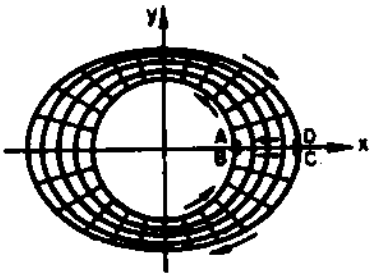


Figure 13.12: O-Type Grid Generation by Algebraic Transformation

Let AB be the circle $x^2 + y^2 = R^2$, BC (or AD) be the branch-cut taken along the x-axis and CD be the ellipse $x^2/a^2 + y^2/b^2 = 1$. Here, we have $f_1(x,y) = (x^2+y^2-R^2)/R^2$, $f_2(x,y) = (\tan^{-1}(y/x)) - 2\pi$, $f_3(x,y) = (x^2/a^2) + (y^2/b^2) - 1$ and $f_4(x,y) = \tan^{-1}(y/x)$. Using Equations 13.41 and 13.42, the body-fitting coordinates for this problem can be found as

$$\xi = \frac{1}{2\pi}\tan^{-1}\frac{y}{x}$$

$$\eta = \frac{(x^2/R^2 + y^2/R^2 - 1)}{(x^2/R^2 - x^2/a^2 + y^2/R^2 - y^2/b^2)}$$

Note that the branch-cut has been taken here as the straight line corresponding to $\theta = 0$ or 2π, in order to conveniently obtain the x- constant lines as radial lines passing through the origin.

In the examples cited above, the generation of grid was done analytically through suitable algebraic transformation. For irregular geometries, obtaining transformation functions is difficult; also, there is no guarantee that the particular transformation employed may produce an acceptable grid. A more general procedure for the algebraic generation of grids is based on interpolation. Here, the coordinates of the interior nodes are obtained by interpolation between the prescribed boundary data. A very useful technique for this purpose is *Transfinite Interpolation,* which is illustrated in the following example.

Consider the four-sided geometry ($ABCD$) (Figure 13.13(a)). It is desired to generate ξ-constant and η-constant lines within $ABCD$ which upon transformation would become equi-spaced orthogonal grid lines inside a rectangular domain of size 1×1. The first task to be completed is the placement of grid points on the boundary (see Figure 13.13(b)); here, in order to get a rectangular grid, the number of points on opposite sides should be equal. Also, if some idea is available regarding the nature of gradients in the problem, the boundary points can be located so as to resolve the high gradient regions. Now, let us apply linear interpolation in the ξ-direction between two grid points which lie on the same η =constant line. Since the total range of ξ variation is from 0 to 1, the linear interpolation formulae for the coordinates of an interior point P are written as

$$x'_P = (1-\xi)\,x_E + \xi x_F; \quad y'_P = (1-\xi)\,y_E + \xi y_F \tag{13.46}$$

where E and F are the two boundary grid points. Carrying out the same operation for all the η-constant lines including the boundaries ($\eta = 0$ and $\eta = 1$), the interpolated points (marked by $\times$) will appear as in Figure 13.13(c). It may be noted that on the boundaries AB and CD, the grid points obtained by unidirectional interpolation between the coordinates of the corner nodes (A, B) or (C, D), do not coincide with the actual grid points shown as dots $(\cdot)$. In order to remove this anomaly, the difference between the actual points and the interpolated points should be subtracted on the $\eta = 0$ and $\eta = 1$ boundary curves. Moreover, some corrections need to be applied to the coordinates of point P which have been obtained by unidirectional interpolation in the ξ-direction, along an η = constant line. Introducing such corrections brings in the influence of the boundary grid point data in the η-direction also, for determining the coordinates of point P. Considering two grid points G and H corresponding to the ξ=constant line on which P lies, the corrections for the coordinates of point P are

$$\Delta x_P = (1-\eta)\,\Delta x_G + \eta \Delta x_H; \quad \Delta y_P = (1-\eta)\,\Delta y_G + \eta \Delta y_H \tag{13.47}$$

where Δx_G, Δx_H, Δy_G and Δy_H, are the corrections for the boundary points given by

$$\begin{aligned} \Delta x_G &= x'_G - x_G; & \Delta y_G &= y'_G - y_G; \\ \Delta x_H &= x'_H - x_H; & \Delta y_H &= y'_H - y_H; \end{aligned} \tag{13.48}$$

In Equation 13.48, the coordinates indicated with prime are those obtained by unidirectional interpolation in ξ- direction and those without prime are the actual boundary grid point data. The final values of the coordinates of P (after interpolation in both ξ and η directions) are obtained as

$$x_P = x'_P - \Delta x_P \quad y_P = y'_P - \Delta y_P \tag{13.49}$$

Performing the above sequence of operations for every interior point gives the mesh in the physical domain as shown in Figure 13.13(d). The corresponding transformed mesh is shown in Figure 13.13(e).

It is important to note that unidirectional interpolation can be done first in the η direction along each ξ = constant curve. In that case, corrections will have to be applied for matching actual grid points and the interpolated points along the boundaries $AD(\xi = 0)$ and $BC(\xi = 1)$. The final grid obtained by both the above approaches will be exactly same. The general algorithm of transfinite interpolation, can therefore, be stated as:

(i) Place grid points on the boundaries of the domain as desired.

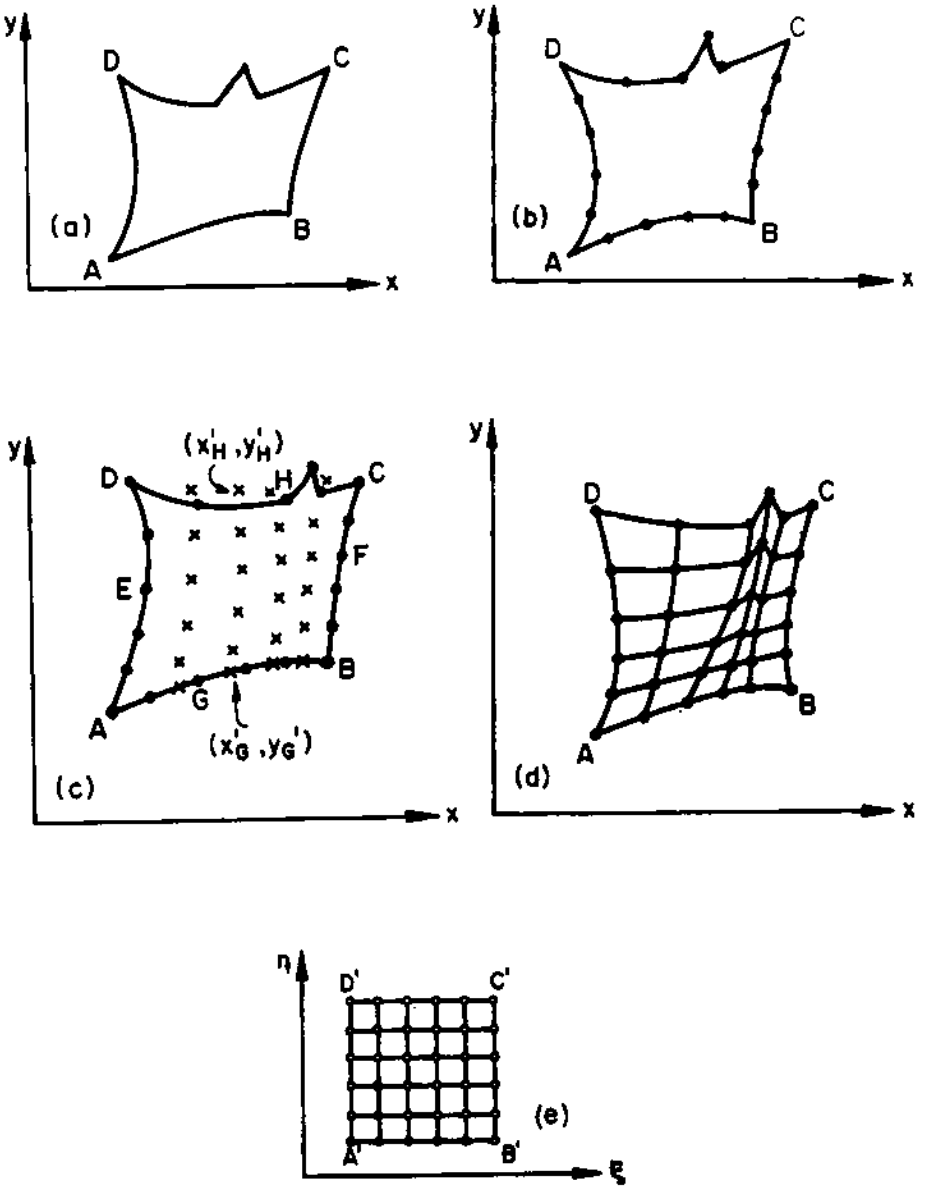


Figure 13.13: Transfinite Interpolation

(ii) Apply unidirectional interpolation in ξ-direction (or η-direction) between the boundary grid data given on the curves $\xi = 0$ and $\xi = 1$ (or $\eta = 0$ and $\eta = 1$) and obtain the coordinates x'_P, y'_P for every interior and boundary point.

(iii) Calculate the mismatch between the interpolated and the actual coordinates on the $\eta = 0$ and $\eta = 1$ (or $\xi = 0$ and $\xi = 1$) boundaries.

(iv) Linearly interpolate the difference in the boundary point coordinates and find the correction to be applied to the coordinates of every interior point.

(v) For each point, the final coordinate value is equal to difference between the uni-directionally interpolated value and the correction.

The transfinite interpolation described above is a very simple and powerful numerical procedure. It can be applied to 2-D or 3-D situations and for all grid configurations (O, C, and H-types). The main weakness of this procedure is that the slope discontinuities at the boundary propagate to the interior and spoil grid smoothness. However, since the metric derivatives are also evaluated numerically, the effect of a slope discontinuity is not severe, as in the case of analytical transformation.

13.4.2 Conformal Mapping

Conformal mapping is based on the theory of analytic functions in complex analysis. If a function $f(z)$ is analytic and $df/dz \neq 0$ (where $z = x + iy$), the mapping defined by

$$w = u + iv = f(z) \tag{13.50}$$

is conformal. In such a case, the angle between any two given lines in the z-plane will be equal to the angle between these lines in the w-plane and vice versa. In particular, a rectangular computational mesh (in w-plane) would correspond to an orthogonal mesh in the physical plane (z). Moreover, both the physical coordinates (x, y) and the transformed coordinates (u, v) vary smoothly over the domain. Thus, orthogonal smooth grids are possible by conformal mapping. Let us illustrate the features of conformal mapping through an example. Consider the transformation

$$w = u + iv = z^2 = (x^2 - y^2) + 2ixy \tag{13.51}$$

or

$$u = x^2 - y^2 \quad \text{and} \quad v = 2xy \tag{13.52}$$

Here, the $u =$ constant lines are hyperbolas with asymptotes $y/x = \pm 1$ and the $v =$ constant lines are rectangular hyperbolas with x and y axes as asymptotes. Let us consider a rectangular domain $A'B'C'D'$ bounded by the straight lines $u = u_1$, $v = v_1$, $u = u_2$ and $v = v_2$, where $(u_1, u_2, v_1, v_2) > 0$. It can be easily

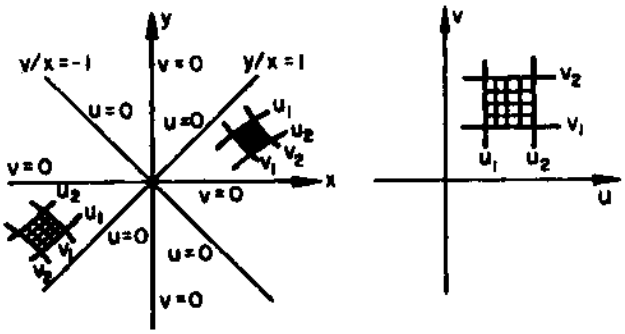


Figure 13.14: Conformal Mapping

proved that there are two pre-images of the domain $A'B'C'D'$ in the x-y plane as shown in the Figure 13.14. Either of the shapes in the physical domain can be mapped onto the rectangle in the first quadrant of the w-plane, using the conformal transformation given by the Equation 13.52. An orthogonal grid net formed by the hyperbolas x^2 - y^2 = constant and xy = constant, transforms into a rectangular grid in the w- plane. Considering the whole x-y plane, it can be said that each 45° sector (bounded between the lines $y/x = \pm 1$, $y = 0$ and $x = 0$) maps onto one of the quadrants of the w- plane. Angles between lines are preserved everywhere except at the origin where $f'(z) = 2z = 0$.

In many cases, it is easier to map the physical domain onto an annular region bounded by circles in the w-plane. Finite difference computations in annular regions are as simple as in rectangular domains and therefore, if conformal transformation onto a circular geometry is possible, it could be employed. Also, the mapping need not always have a form $w = f(z)$. It could be taken as $z = f(w)$ as well. For instance, consider the mapping

$$z = \frac{(w + 1/w)}{2} \tag{13.53}$$

Expressing w in polar coordinates as $\rho e^{i\phi}$, one gets

$$z = x + iy = \frac{1}{2}\left(\rho e^{i\phi} + \frac{e^{-i\phi}}{\rho}\right)$$

$$x = \frac{1}{2}(\rho + 1/\rho)\cos\phi; \quad y = \frac{1}{2}(\rho - 1/\rho)\sin\phi$$

Eliminating ρ

$$\frac{x^2}{\cos^2\phi} - \frac{y^2}{\sin^2\phi} = 1 \tag{13.54}$$

and eliminating ϕ

$$\frac{x^2}{(\rho + 1/\rho)^2} + \frac{y^2}{(\rho - 1/\rho)^2} = \frac{1}{4} \tag{13.55}$$

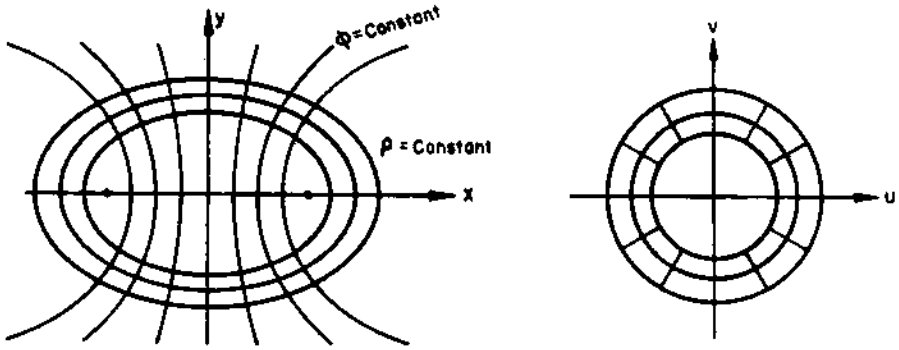


Figure 13.15: Conformal Mapping of an Elliptical Geometry

Thus, the circles (ρ = constant) in the w- plane map as confocal ellipses in the z- plane. Also, the radial lines (ϕ = constant) represent hyperbolas in the physical domain (Figure 13.15).

A very important mapping which is used widely is the Schwarz-Christoffel transformation (Currie, 1993). This transformation is of the form

$$z = C \int (w - u_1)^{k_1} (w - u_2)^{k_2} \ldots (w - u_n)^{k_n} \, dw + K \tag{13.56}$$

It maps an n-sided polygon onto the real axis of the w-plane (Figure 13.16). The vertices $z_1, z_2, z_3, \ldots z_n$ of the polygon become the points $u_1, u_2, u_3, \ldots u_n$ on the real axis. The indices $k_1, k_2, k_3, \ldots k_n$ are real and they are related to the interior angles $\alpha_1, \alpha_2, \alpha_3 \ldots \alpha_n$ of the polygon through the relationship

$$k_i = \alpha_i/\pi - 1 \tag{13.57}$$

It is easy to show that $\sum_{i=1}^{n} k_i = -2$, since $\sum_{i=1}^{n} \alpha_i = (n-2)\pi$. The constant factor C in Equation 13.56 represents an arbitrary magnification and rotation of the polygon. This is clear if C is expressed as $C = \alpha e^{i\beta}$, where α is the magnification factor and β is the angle of rotation. Also, the constant K corresponds to an arbitrary translation of the polygon. Thus, any polygonal figure in z-plane can be mapped onto the real axis of the w-plane. The points inside the polygon correspond to the upper-half (above the $u = 0$ axis), while the points outside map onto the lower half of the w-plane. The main task in transforming any given geometry, therefore, is to find the values of C, K and $u_1, u_2, u_3, \ldots u_n$. The Schwarz-Christoffel transformation can be applied to open regions as well. Some examples of conformal mapping by Schwarz-Christoffel transformation are given below.

Consider a forward-facing step as shown in Figure 13.17. It is intended to map the points $z_1 = 0$ and $z_2 = 1$, onto $u_1 = -1$ and $u_2 = 1$ respectively. If mapping is conformal, the region above the shaded line in the z-plane will map onto the upper half ($v \geq 0$) of the w-plane. The interior angles at z_1 and z_2

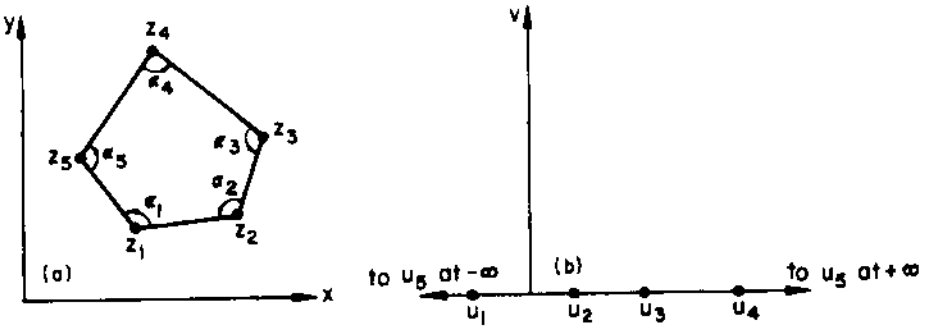


Figure 13.16: Conformal Mapping of a Polygon by Schwarz-Christoffel Transformation, a. z- and b. w-planes

are: $\alpha_1 = \pi/2$ and $\alpha_2 = 3\pi/2$. The corresponding indices are therefore given by $k_1 = -1/2$ and $k_2 = 1/2$. The mapping function for this problem is

$$z = C \int \left(\frac{w-1}{w+1}\right)^{1/2} dw + K \tag{13.58}$$

where C and K are determined by setting $z = 0$ for $w = -1$ and $z = i$ for $w = 1$.

For the triangle ABC shown in Figure 13.18, the mapping can be done as follows:

Points A and B are mapped onto the points $A'(0,0)$ and $B'(1,0)$ on the real axis of the w-plane. The point C' can be conveniently taken at infinity. The transformation is then obtained as

$$z = C \int (w-u_1)^{k_1} (w-u_2)^{k_2} dw + K \tag{13.59}$$

where $u_1 = 0$, $u_2 = 1$; $k_1 = (\alpha_1/\pi) - 1$, $k_2 = (\alpha_2/\pi) - 1$; and $\alpha_1 = \tan^{-1} 1 = \alpha_2$.

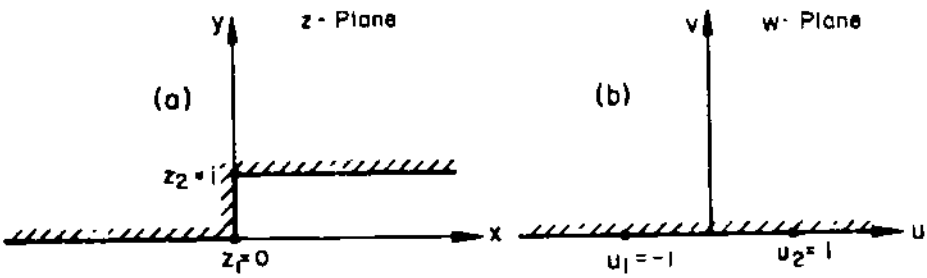


Figure 13.17: Transformation of a Forward Facing Step (a) Physical Domain, (b) Transformed Domain

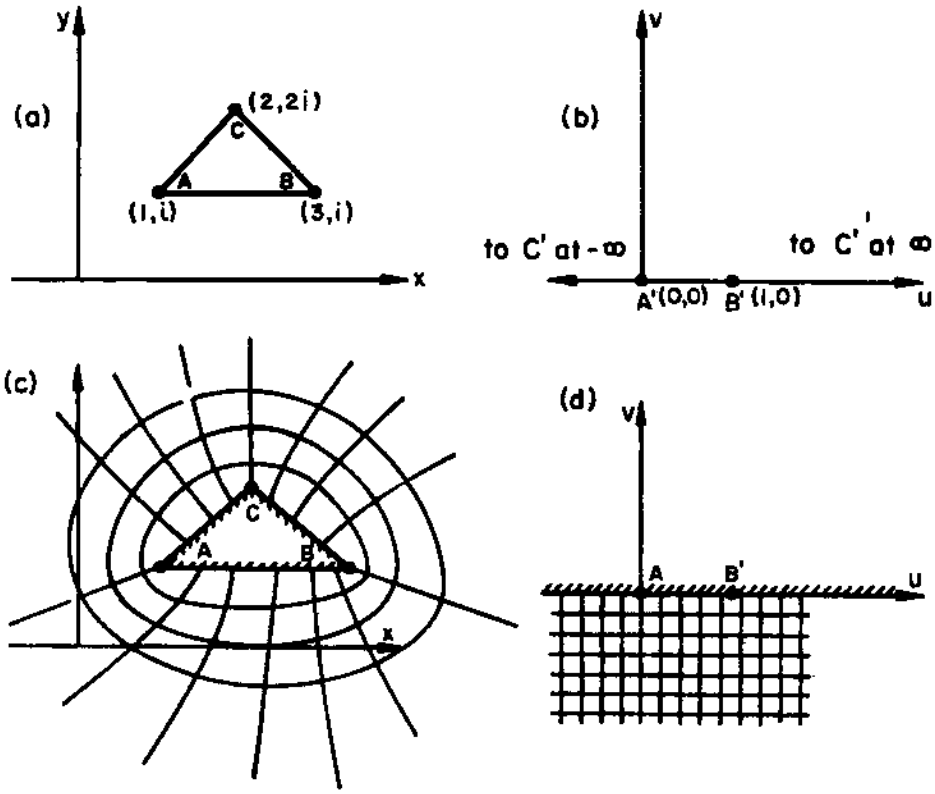


Figure 13.18: Conformal Transformation of a Triangle (a) Physical Domain, (b) Transformed Domain, (c) Grid in Physical Domain and (d) Grid in Transformed Domain

The constants C and K can be found by setting $z = (1, i)$ for $w = 0$ and $z = (3, i)$ for $w = 1$.

In the transformation described above, the interior region of the triangle ABC maps onto the upper half of the w-plane. This is because the interior lies to the left of the boundary lines AB, BC and CA in the physical plane and after conformal transformation also, it will lie to the left of the line segments $A'B'$, $B'C'$ and $C'A'$. Therefore, a rectangular grid in the upper half of the w-plane will correspond to a nearly orthogonal grid within the triangle. The orthogonality is disturbed only at the points A, B, C and the pre-image of $v \to \infty$. The region outside the triangle is mapped onto the lower half of the w-plane. The external grid obtained by conformal transformation is shown in Figures 13.18(c) and 13.18(d).

It is clear from the above examples that the boundary of a given geometry can be mapped onto the whole real axis ($v = 0$) of the transformed plane. If the

given geometry is open, then the infinity of the physical plane is transformed onto the infinity of the w-plane. For closed geometries, one of the vertices may be mapped as ∞ on the $v = 0$ axis. During such a transformation, the interior and exterior regions with respect to the boundaries of the given geometry conveniently become the upper and lower half planes in the transformed coordinates.

The Schwarz-Christoffel transformation is a powerful tool for mapping polygonal shapes onto a straight line. Complex curvilinear shapes can also approximated as polygons and nearly conformal grids can be generated using this procedure. Very often, the integral given in Equation 13.56 is evaluated numerically using suitable quadratures. In some situations, conformal mapping is done in a series of steps, using successive transformations. Fairly complex shapes can be transformed into simpler ones by this procedure. For instance, the transformation steps

$$Z'' = \zeta + \frac{b^2}{\zeta} \quad \text{and} \quad \zeta = Z''' + Me^{i\delta} \tag{13.60}$$

map a Joukowski airfoil in Z'' plane onto a circle on the Z''' plane (Figure 13.19).

For 3-D situations, the geometry can be sliced into several planes and conformal grids can be constructed for each plane. However, such stacked grids often lack orthogonality in the direction of stacking. A major disadvantage of the conformal mapping procedure is that it provides no control over grid spacing in the interior of the domain. Also, if uniform grid spacing is used in the transformed plane, the spacing in the physical grid may be unacceptably large or small. For the generation of orthogonal grids, however, conformal mapping is still one of the most popular techniques.

13.4.3 Grid Generation Based on Partial Differential Equations

PDE-based methods are commonly used for structured grid generation in a wide variety of applications. These techniques may be termed as elliptic, parabolic or hyperbolic, depending upon the characteristics of the grid generation equations.

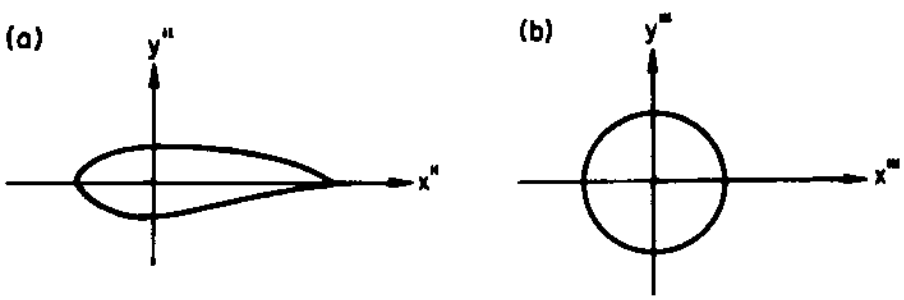


Figure 13.19: Successive Transformation of a Joukowski Airfoil

In most applications, the grid generation equations are also transformed onto the rectangular computational domain and solved along with the governing equations of the physical problem. The most attractive feature of the PDE based methods is that a compromise between the desired grid properties of smoothness, orthogonality and grid spacing control can be achieved. Also, a geometry independent, generalized approach can be successfully implemented for a large number of 2-D and 3-D problems with arbitrary shaped domains. However, these methods are computationally expensive, in general.

The partial differential equations for grid generation, in their original form, describe the variation of the body- fitting coordinates (ξ, η, ζ) in the interior of the physical domain, with prescribed values or slopes at the boundary. In the computational domain, however, the physical coordinates (x, y, z) are treated as the unknown variables on the grid formed by the ξ, η and ζ lines and they are determined by numerically solving the transformed grid generation equations. Elliptic grid generation methods offer good grid smoothness and spacing control. These are particularly useful for confined geometries. Hyperbolic equations can be employed for generating orthogonal grids in unbounded domains. Starting from the object boundary which may correspond to say, a ξ -constant surface, the grid lines can be constructed by marching in the ξ-direction up to the desired distance. Parabolic grid generation equations are not widely used and will not be covered here.

I. Elliptic Grid Generation

Poisson equations of the form

$$\begin{aligned} \xi_{xx} + \xi_{yy} + \xi_{zz} &= P(\xi, \eta, \zeta) \\ \eta_{xx} + \eta_{yy} + \eta_{zz} &= Q(\xi, \eta, \zeta) \\ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= R(\xi, \eta, \zeta) \end{aligned} \tag{13.61}$$

are normally employed for elliptic grid generation in 3-D geometries. A special case of elliptic grid generation is given by the Laplace system

$$\begin{aligned} \xi_{xx} + \xi_{yy} + \xi_{zz} &= 0 \\ \eta_{xx} + \eta_{yy} + \eta_{zz} &= 0 \\ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= 0 \end{aligned} \tag{13.62}$$

The elliptic operators satisfy certain extremum principles which are necessary to ensure that the mapping is one-to-one. For instance, considering the solution of the equation $\nabla^2 \phi = S$, the following properties can be proved:

(a) If $S > 0$, the maximum value of ϕ occurs only at the boundary;

(b) If $S < 0$, the minimum value of ϕ occurs only at the boundary;

(c) If $S = 0$, both the maximum and minimum occur only at the boundary.

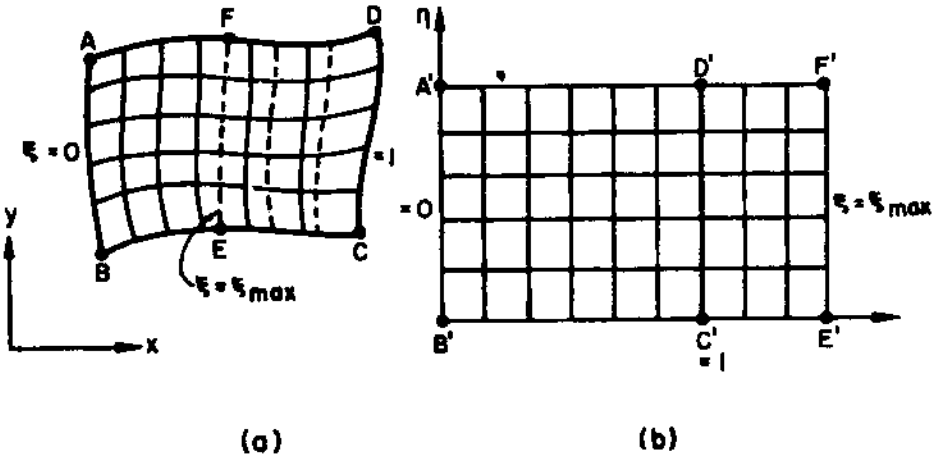


Figure 13.20: Violation of Maximum Principle

Property (c) is a highly desirable one in the context of grid generation. To illustrate its usefulness, let us assume that the maximum value of the curvilinear coordinate $\xi(= \xi_{\max})$ occurs in the interior of the solution domain (Figure 13.20(a)), while the boundary segments are mapped onto $\xi = 0$ and $\xi = 1$ in the computational domain as usual. Obviously, $\xi_{\max} > 1$. It is clear from the figure that the curvilinear lines for $1 < \xi < \xi_{\max}$, fold back into the interior of the physical domain and mapping is no more one-to-one. A similar folding back effect will take place if $\xi_{\min}, \eta_{\max}$ or $\eta_{\min}$ are assumed to occur in the interior of the physical domain.

The above example illustrates the importance of the extremum principles for achieving one-to-one mapping. The Laplacian grid generation system satisfies both the maximum and the minimum principles and is therefore a favourable candidate for use in the transformation procedure. Moreover, it can be proved that the solutions of the Laplacian operator are very smooth (harmonic) functions. Due to their excellent smoothness property, even discontinuities arising from the prescribed boundary data disappear at a short distance inside the domain. Therefore, the metric derivatives ξ_x, ξ_y, η_x and η_y exist everywhere within the solution domain and the variation of the metric derivatives is smooth, leading to an accurate solution of the transformed governing equations. Apart from the above mentioned advantages, grid orthogonality can also be achieved if the grid lines are chosen to intersect orthogonally at the boundaries.

Let us now consider the generation of 2-D body-fitting grids using the Laplacian system. The grid generation equations are

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

$$\nabla^2\eta = \frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2} = 0 \tag{13.63}$$

Rewriting $\nabla^2\xi$ as

$$\begin{aligned}\frac{\partial(\xi_x)}{\partial x} &+ \frac{\partial(\xi_y)}{\partial y} = \left\{\xi_x \frac{\partial}{\partial\xi}(\xi_x) + \eta_x \frac{\partial}{\partial\eta}(\xi_x)\right\} \\ &+ \left\{\xi_y \frac{\partial}{\partial\xi}(\xi_y) + \eta_y \frac{\partial}{\partial\eta}(\xi_y)\right\}\end{aligned}$$

and substituting for ξ_x, η_x, ξ_y and η_y from Equation 13.8

$$\nabla^2\xi = A(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}) + B(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}) = 0 \tag{13.64}$$

where

$$\begin{aligned}&A = -y_\eta / \mid J \mid^3,\ B = x_\eta / \mid J \mid^3,\ \alpha = x_\eta^2 + y_\eta^2, \\ &\beta = x_\xi x_\eta + y_\xi y_\eta,\ \gamma = x_\xi^2 + y_\xi^2 \quad \text{and} \quad \mid J \mid = (x_\xi y_\eta - x_\eta y_\xi).\end{aligned}$$

Similarly, it can be shown that

$$\nabla^2\eta = C(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}) + D(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}) = 0 \tag{13.65}$$

where $C = y_\xi / \mid J \mid^3$ and $D = -x_\xi / \mid J \mid^3$. If $AD - BC \neq 0$, Equations 13.64 and 13.65 simplify as

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \tag{13.66}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \tag{13.67}$$

Equations 13.66 and 13.67 constitute the elliptic grid generation system derived from the Laplace operator. They are valid if $\mid J \mid$ is finite and non-zero everywhere. These equations can be solved by finite difference method in the rectangular computational domain. A point to be noted, however, is that the coefficients α, β, γ are themselves functions of the unknown variables $x(\xi, \eta)$ and $y(\xi, \eta)$. Therefore, Equations 13.66 and 13.67 are a non-linear system which need to be iteratively solved. In practice, one starts with an initial mesh which is generated by an algebraic procedure (say, trans- finite interpolation) and improves the grid properties by applying the grid generation Equations 13.66 and 13.67.

The boundary conditions for the grid generation equations involve either prescribed grid spacing or grid line slope along the boundary curves. On a ξ - constant boundary (say $\xi = \xi_0$), prescribing the grid spacing leads to

$$\Delta s = (\Delta x^2 + \Delta y^2)^{1/2} = \left.(x_\eta^2 + y_\eta^2)^{1/2}\right|_{\xi=\xi_0} \Delta\eta \tag{13.68}$$

or

$$s_\eta = \frac{\partial s}{\partial \eta} = \left(x_\eta^2 + y_\eta^2\right)^{1/2} \tag{13.69}$$

The form given in Equation 13.68 can be easily implemented numerically. Considering a uniform spacing $\Delta\eta$ in the computational domain between $\eta = 0$ and $\eta = 1$, Equation 13.68 can be written as

$$\Delta s_i = \frac{1}{N}\left(x_\eta^2 + y_\eta^2\right)^{1/2}\Big|_{\xi=\xi_0} \tag{13.70}$$

where N is the number of divisions in the η-direction.

For prescribing the angle between the grid lines on the $\xi = \xi_0$ boundary, the following relations hold

$$\begin{aligned} \nabla\xi \cdot \nabla\eta &= \xi_x\eta_x + \xi_y\eta_y = \mid \nabla\xi \mid \mid \nabla\eta \mid \cos\theta \\ &= \left(\xi_x^2 + \xi_y^2\right)^{1/2}\left(\eta_x^2 + \eta_y^2\right)^{1/2}\cos\theta \end{aligned}$$

where θ is the angle between the normals of the ξ-constant and η-constant lines at the boundary. Using the transformation relations for the metric derivatives

$$\xi_x\eta_x + \xi_y\eta_y = \frac{-\left(y_\xi y_\eta + x_\xi x_\eta\right)}{\mid J \mid^2} = \frac{\left(\left(x_\eta^2 + y_\eta^2\right)^{1/2}\left(x_\xi^2 + y_\xi^2\right)^{1/2}\cos\theta\right)}{\mid J \mid^2}$$

or

$$-\left(x_\xi x_\eta + y_\xi y_\eta\right) = s_\eta\left(x_\xi^2 + y_\xi^2\right)^{1/2}\cos\theta \tag{13.71}$$

Similarly, the cross-product $\nabla\xi \times \nabla\eta$ gives rise to the equation

$$\left(x_\xi y_\eta - y_\xi x_\eta\right) = s_\eta\left(x_\xi^2 + y_\xi^2\right)^{1/2}\sin\theta \tag{13.72}$$

Eliminating y_η or x_η between these two relation leads to

$$x_\eta = s_\eta\frac{\left(-x_\xi\cos\theta - y_\xi\sin\theta\right)}{\left(x_\xi^2 + y_\xi^2\right)^{1/2}} \tag{13.73}$$

$$y_\eta = s_\eta\frac{\left(-y_\xi\cos\theta + x_\xi\sin\theta\right)}{\left(x_\xi^2 + y_\xi^2\right)^{1/2}} \tag{13.74}$$

For the special case of $\theta = 90^o$ (orthogonal grid lines), Equation 13.71 simplifies to

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{13.75}$$

It is seen that the boundary conditions for prescribed grid spacing (Equation 13.70) or prescribed grid line slope (Equations 13.73 and 13.74) are also non- linear. Therefore, these can be enforced only iteratively by successive improvement of the initial mesh.

The attractive features of the Laplace grid generation equations have been described at length earlier. One inconvenient aspect of using these equations arises from their convex property. This property causes grid lines to be attracted towards convex corners and repelled by concave corners (Figure 13.21).

Consider for example, a convex η = constant boundary as shown in Figure 13.22(a). With x and y directions as indicated in the figure, let us assume that η increases away from the surface $\eta = \eta_0$, i.e. $\eta_2 > \eta_1 > \eta_0$. Also let $\eta_2 - \eta_1 = \eta_1 - \eta_0 = \Delta\eta$. Due to the convex nature of the boundary, $\eta_{xx} > 0$ for points close to the boundary. This can be seen clearly from the finite difference expression

$$\eta_{xx}\,|_B = \frac{(\eta_A + \eta_C - 2\eta_B)}{\Delta x^2} > 0$$

since $\eta_A = \eta_C > \eta_B$ for the given configuration. If Laplace grid generation system is used, $\eta_{xx} > 0$ leads to $\eta_{yy} < 0$ (since $\nabla^2\eta = 0$). Consider the points 1, 2 and 3. From $\eta_{yy} < 0$, it can be shown that

$$\left(\frac{\partial\eta}{\partial y}\right)_{2-3} < \left(\frac{\partial\eta}{\partial y}\right)_{1-2}$$

since the gradient η_y decreases with y. Writing the gradients between the points (1,2) and (2,3) in terms of their finite difference expressions

$$\frac{\Delta\eta}{(y_3 - y_2)} < \frac{\Delta\eta}{(y_2 - y_1)}$$

or

$$(y_3 - y_2) > (y_2 - y_1).$$

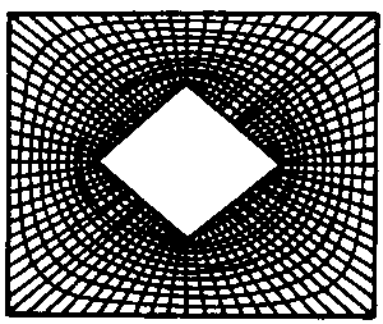


Figure 13.21: Boundary Grid Line Attraction and Repulsion for the Laplace System

This proves that the spacing is small near a convex boundary and it increases with distance towards the interior. In a similar fashion, it can be shown that the spacing is maximum adjacent to a concave boundary (Figure 13.22(b)) and it decreases towards the interior.

The assumptions made while deriving the above results are not restrictive in the sense that similar proofs can be given for any convex or concave boundary. The interior grid spacings produced by the Laplace grid generation equations tend to be more or less uniform. The loss of grid spacing control near the boundary could be severe when sharp corners or segments of high curvature are present.

In order to compensate for the effects of boundary curvature and to control the grid spacing as per desire, it is necessary to consider elliptic grid generation equations of the Poisson-type (Equation 13.61). After interchanging the roles of the dependent and independent variables, the Poisson grid generation system can be written as

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} \quad &+ \quad \mid J \mid^2 (P x_\xi + Q x_\eta) = 0 \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} \quad &+ \quad \mid J \mid^2 (P y_\xi + Q y_\eta) = 0 \end{aligned} \tag{13.76}$$

in a form similar to Equations 13.66 and 13.67. The control functions P and Q can be chosen appropriately to provide the desired grid spacing distribution. A slightly more convenient expression can be achieved for Equation 13.76 if P and Q are written as

$$P(\xi,\eta) = \phi(\xi,\eta)\frac{\alpha}{\mid J \mid^2} \quad \text{and} \quad Q(\xi,\eta) = \psi(\xi,\eta)\frac{\gamma}{\mid J \mid^2} \tag{13.77}$$

The grid generation equations are now given by

$$\begin{aligned} \alpha(x_{\xi\xi} + \phi x_\xi) - 2\beta x_{\xi\eta} \quad &+ \quad \gamma(x_{\eta\eta} + \psi x_\eta) = 0 \\ \alpha(y_{\xi\xi} + \phi y_\xi) - 2\beta y_{\xi\eta} \quad &+ \quad \gamma(y_{\eta\eta} + \psi y_\eta) = 0 \end{aligned} \tag{13.78}$$

The above form indicates that the variations of x and y in the ξ-direction (along an η-constant line) are governed by the function ϕ and their variations in η-direction are governed by the choice of ψ. One way of obtaining the functions ψ and ϕ is to set

$$x_{\eta\eta} + \psi x_\eta = 0 \quad \text{or} \quad y_{\eta\eta} + \psi y_\eta = 0 \tag{13.79}$$

on the $\xi =$ constant boundaries and

$$x_{\xi\xi} + \phi x_\xi = 0 \quad \text{or} \quad y_{\xi\xi} + \phi y_\xi = 0 \tag{13.80}$$

on the $\eta =$ constant boundaries. If the coordinates (x, y) of all the grid points are specified, then the functions ψ and ϕ can be determined on the boundaries by finite differencing the expressions (13.79) and (13.80). For the interior points,

ψ and ϕ can be obtained by interpolating between their boundary values, in the ξ and η directions respectively.

General remarks can be made regarding the effects of the control functions upon the grid spacing, based on the sign of P and Q at the concerned location. For instance, let us rework the example shown in Figure 13.22 for a Poisson system $\nabla^2\eta = Q$. If $Q < 0$, then near a convex boundary η_{yy} becomes more negative than in the case of Laplace grid generation equations. This leads to smaller grid spacing in the η-direction adjacent to the convex boundary. In general, if $\eta = \eta_0$ is a convex boundary segment, negative values of Q shift the η = constant grid lines towards the decreasing direction of η and $Q > 0$ results in a shift of η-lines towards the increasing direction of η (Figure 13.23). Similarly, negative or positive values of P move the ξ = constant lines in the decreasing or increasing direction of ξ respectively. If the grid spacing is specified on the $\eta = \eta_0$ boundary, $P > 0$ will alter the intersection angles of the ξ- lines so that they lean more in the direction of increasing ξ.

The above discussed ideas can be utilized for controlling the grid spacing in the interior of the domain also. In particular, attraction of grid lines towards a grid line or towards a given point can be achieved by suitable choices of P and Q. In order to have only a local effect, the control functions should be non-zero near the point or line of interest and they should rapidly decrease to zero far away. Exponential forms of control functions are useful for this purpose. Let us illustrate the choice of control functions through a specific example (Figure 13.24).

Suppose we require small grid spacing between the ξ = constant lines in the neighbourhood of $\xi = \xi_i$. This can be achieved by selecting

$$P(\xi, \eta) = -A \text{ sign}(\xi - \xi_i) \exp\{-B \mid \xi - \xi_i \mid\} \tag{13.81}$$

where the expression sign $(\xi - \xi_i)$ is $+1$ for $\xi > \xi_i$ and -1 for $\xi < \xi_i$. The constant A is equal to the maximum magnitude of P and it controls the extent by which the ξ-grid lines shift towards $\xi = \xi_i$. The constant B governs the decay

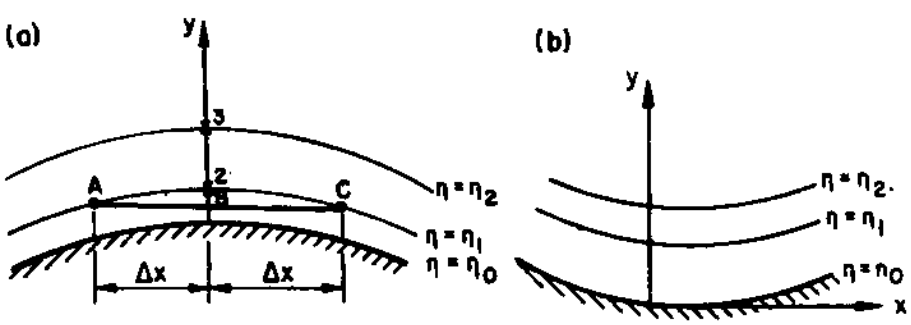


Figure 13.22: Effects of (a) Convex and (b) Concave Boundaries for Laplace Grid Generator

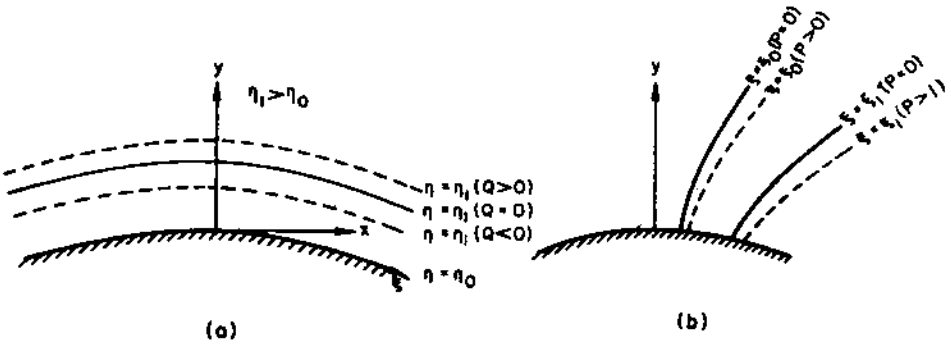


Figure 13.23: Effects of Control Functions on Boundary Grid Spacing

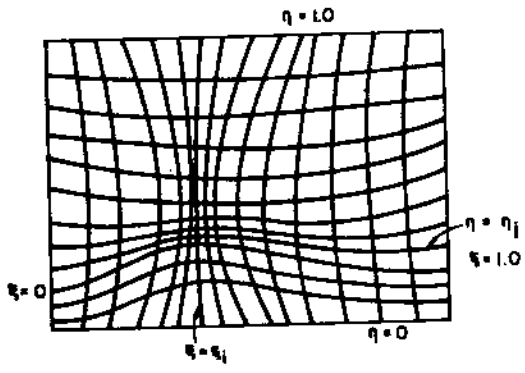


Figure 13.24: Attraction of Grid Lines towards an Interior Point or a Line

of P and so it is related to the extent of the region for which the attraction is effective. If instead of reducing the spacing adjacent to a grid line, we need to attract the ξ-constant line towards a given point (ξ_i, η_i), the control function P in this case can be selected as:

$$P(\xi,\eta) = -A \operatorname{sign}(\xi - \xi_i) \exp\left\{-B\left[(\xi-\xi_i)^2 + (\eta-\eta_i)^2\right]\right\} \tag{13.82}$$

For attracting the η-constant lines towards a given line or point, the functions $Q(\xi,\eta)$ can be selected analogously. A precaution to be applied while selecting the value of the constant A is that if it is too large, the extremum principles may be violated and the grid lines may cross each other. For adaptive grid generation, the control functions can be selected according to the error or gradients of the predicted solution.

It is evident from the previous discussions that the elliptic systems have a wide applicability towards the generation of structured grids. A major drawback of these systems, however, is that the grid generation equations have to be transformed into the computational (ξ,η) domain and such transformed equations are highly non-linear . The implementation of the boundary conditions is also not straight forward. One simple way to eliminate these difficulties is to solve the original elliptic equations in the physical (x,y) domain itself, by finite element method (Rao *et al.*, 1988). Applying the Galerkin's weighted residual procedure to the Poisson grid generation equations leads to

$$\left.\begin{aligned} \iint_\Omega W_i \left\{\nabla^2\xi - P\right\} dxdy = 0 \\ \iint_\Omega V_i \left\{\nabla^2\eta - Q\right\} dxdy = 0 \end{aligned}\right\} \quad i = 1,2\ldots k \tag{13.83}$$

where W_i, V_i are suitable weighting functions, Ω is the physical domain and k is the number of nodes in the finite element mesh (Figure 13.25). The unknown variables ξ and η are interpolated within each element using expressions of the form

$$\xi = \sum_{j=1}^{M} N_j(x,y)\xi_j, \quad \eta = \sum_{j=1}^{M} N_j(x,y)\eta_j, \tag{13.84}$$

where ξ_j, η_j are the nodal values of ξ and η, and M is the number of nodes per element. According to Galerkin's approach, the weighting functions W_i and V_i can be chosen equal to the interpolation functions N_i. Thus

$$W_i = V_i = N_i \tag{13.85}$$

for each element. Substituting Equations 13.84 and 13.85 into Equation13.83 and integrating the Laplace operator by parts leads to

$$\iint_\Omega \left\{\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right\} \xi_j dxdy \quad - \quad \oint_S \frac{\partial \xi}{\partial n} N_i \, d\ell = \iint_\Omega P N_i \, dxdy \quad .$$

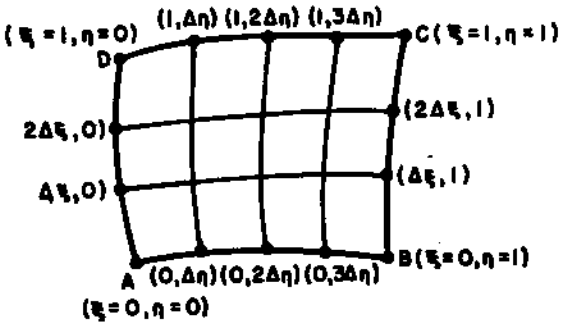


Figure 13.25: Generation of a Structured Grid by Finite Element Method

$$\iint_{\Omega} \left\{ \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right\} \eta_j dxdy \quad - \quad \oint_S \frac{\partial \eta}{\partial n} N_i \, d\ell = \iint_{\Omega} Q N_i \, dxdy \tag{13.86}$$

where S is the boundary of domain Ω and n is the local normal direction for the boundary. The integrals in the above equations are evaluated using numerical quadrature and the resulting algebraic equations are written in a compact matrix form

$$[M]\{\xi_j\} = \{F_1\} \tag{13.87}$$

$$[M]\{\eta_j\} = \{F_2\} \tag{13.88}$$

where

$$\begin{aligned} M_{ij} &= \iint_{\Omega} \left\{ \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right\} dxdy \\ F_{1i} &= \oint_S \frac{\partial \xi}{\partial n} N_i d\ell - \iint_{\Omega} P N_i dxdy \\ F_{2i} &= \oint_S \frac{\partial \eta}{\partial n} N_i d\ell - \iint_{\Omega} Q N_i dxdy \end{aligned}$$

The implementation of the boundary conditions is also very simple, for the above system of equations. For the 4-sided domain shown in Figure 13.25, let us assume that the grid spacing between all boundary grid points are given. Here, $\xi = 0$ on AB, $\xi = 1$ on CD and $\Delta\xi$ is equal between any two neighbouring points on BC and AD.

The ξ values at all boundary grid points are known and the relevant algebraic equations for the boundary nodes can be replaced by equations of the form ξ_j = given in the matrix system (13.87). Similar incorporation of the known

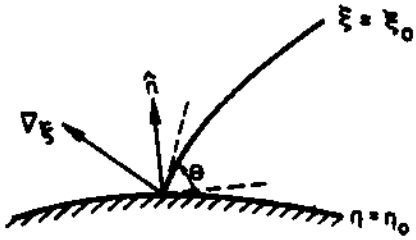


Figure 13.26: Implementation of Prescribed Slope Boundary Condition

values of η on the boundary can be made in the matrix Equation 13.88 also. When the grid point locations are not prescribed but the grid lines are required to intersect with a given angle at the boundary, we make use of the boundary integral terms of the right hand side vector. For instance, the term $\partial\xi/\partial n$ can be written as

$$\frac{\partial \xi}{\partial n} = \mathbf{n} \cdot \nabla \xi = \mid \nabla \xi \mid \cos\theta \tag{13.89}$$

where θ is the angle between the normal $\vec{n}$ to the boundary S and the normal to the ξ = constant line (Figure 13.26). It is evident from Figure 13.26 that θ is also the angle of intersection between the grid lines. As a special case, taking $\partial\xi/\partial n = 0$ on the η = constant boundaries will result in orthogonal grid lines at the boundary.

For generating body-fitting grids by the above procedure, one needs to start with an initial finite element (FE) mesh in the physical domain. However, this grid need not be smooth or structured; it can actually be generated by any algebraic technique or by unstructured grid generation methods to be discussed later. After obtaining the nodal values of ξ and η on the initial FE mesh by the solution of the matrix systems (Equations 13.87) and (13.88), the contours of ξ = constant and η = constant lines can be computed to obtain the structured mesh. Interior grid spacing control can also be achieved by proper choice of control functions P and Q. Examples of grids generated by the above procedure are shown in Figure 13.27.

II. Hyperbolic Grid Generation

In hyperbolic grid generation, the outer boundary need not be specified in advance. Starting from the object surface, (which forms the inner boundary) orthogonal grids can be constructed by marching in one curvilinear coordinate direction (say, η) with the help of the hyperbolic grid generation system. Steger and Sorenson (1980) have suggested the use of a hyperbolic system which provides control over the grid cell areas or volumes for the 2-D and 3-D applications respectively. For a two dimensional geometry, the area scaling factor is

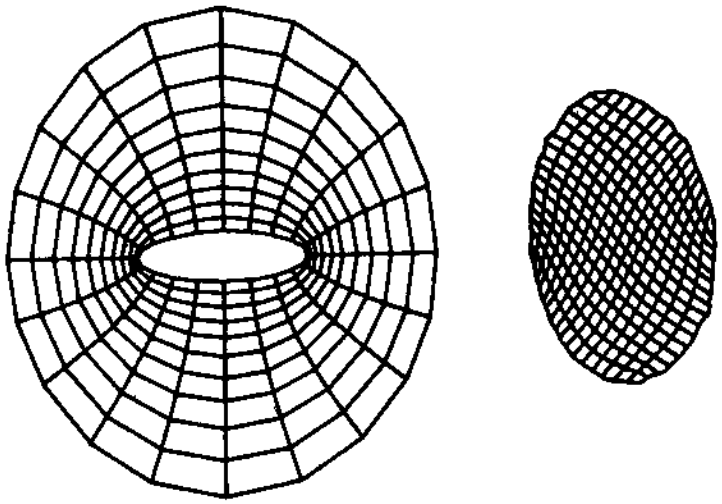


Figure 13.27: Structured Grids Generated by FEM

$$(x_\xi y_\eta - y_\xi x_\eta) = J \tag{13.90}$$

Further, if grid lines are orthogonal

$$(x_\xi y_\eta - y_\xi y_\eta) = 0 \tag{13.91}$$

Equations 13.90 and 13.91 can be linearized about a point $(\bar{x}, \bar{y})$ giving rise to

$$[A] \left\{ \begin{array}{c} x'_\xi \\ y'_\xi \end{array} \right\} + [B] \left\{ \begin{array}{c} x'_\eta \\ y'_\eta \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ J' \end{array} \right\} \tag{13.92}$$

where

$$[A] = \left[\begin{array}{cc} \bar{x}_\eta & \bar{y}_\eta \\ \bar{y}_\eta & -\bar{x}_\eta \end{array} \right] \quad \text{and} \quad [B] = \left[\begin{array}{cc} \bar{x}_\xi & \bar{y}_\xi \\ -\bar{y}_\xi & \bar{x}_\xi \end{array} \right]$$

and $x' = x - \bar{x}$, $y' = y - \bar{y}$. Equation 13.92 represents a hyperbolic system in the increments x' and y', which can be solved by marching in the η-direction starting from the grid point coordinates at the object surface. The function J can be selected suitably to control the cell area.

13.5 Solution of Flow Equations Using Body-fitting Coordinates

Here, the procedure for solving two-dimensional incompressible flows in a body-fitting coordinate system is discussed. The cartesian velocity components are retained as the flow variables for the transformed variables also. This scheme

has the advantage that the velocity field can be easily understood from the predicted results, without any further processing. Consider the following two-dimensional equations

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{13.93}$$

$$\begin{aligned} \rho \left\{ \frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} \right. &+ \left. \frac{\partial (uv)}{\partial y} \right\} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left\{ 2\mu \frac{\partial u}{\partial x} \right\} \\ &+ \frac{\partial}{\partial y} \left\{ \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right\} \end{aligned} \tag{13.94}$$

$$\begin{aligned} \rho \left\{ \frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} \right. &+ \left. \frac{\partial (v^2)}{\partial y} \right\} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial y} \left\{ 2\mu \frac{\partial v}{\partial y} \right\} \\ &+ \frac{\partial}{\partial x} \left\{ \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right\} \end{aligned} \tag{13.95}$$

Using the transformation relations (13.25), the conservative form of the above equations can be written in the curvilinear coordinate system (ξ, η) as

$$\left\{ (y_\eta u)_\xi - (y_\xi u)_\eta \right\} + \left\{ -(x_\eta v)_\xi + (x_\xi v)_\eta \right\} = 0 \tag{13.96}$$

$$\begin{aligned} & \rho \left[\frac{\partial u}{\partial \tau} - \dot{x} \left\{ (y_\eta u)_\xi - (y_\xi u)_\eta \right\} - \dot{y} \left\{ -(x_\eta u)_\xi + (x_\xi u)_\eta \right\} \right] \\ + \; & \rho \left[(y_\eta u^2)_\xi - (y_\xi u^2)_\eta - (x_\eta uv)_\xi + (x_\xi uv)_\eta \right] = -(y_\eta p)_\xi \\ + \; & (y_\xi p)_\eta + (y_\eta F_1)_\xi - (y_\xi F_1)_\eta - (x_\eta F_2)_\xi + (x_\xi F_2)_\eta \end{aligned} \tag{13.97}$$

and

$$\begin{aligned} & \rho \left[\frac{\partial v}{\partial \tau} - \dot{x} \left\{ (y_\eta v)_\xi - (y_\xi v)_\eta \right\} - \dot{y} \left\{ -(x_\eta v)_\xi + (x_\xi v)_\eta \right\} \right] \\ + \; & \rho \left[(y_\eta uv)_\xi - (y_\xi uv)_\eta - (x_\eta v^2)_\xi + (x_\xi v^2)_\eta \right] = +(x_\eta p)_\xi \\ - \; & (x_\xi p)_\eta + (y_\eta F_2)_\xi - (y_\xi F_2)_\eta - (x_\eta F_3)_\xi + (x_\xi F_3)_\eta \end{aligned} \tag{13.98}$$

where

$$F_1 = \frac{2\mu}{|J|} \left\{ (y_\eta u)_\xi - (y_\xi u)_\eta \right\}$$

$$F_2 = \frac{\mu}{|J|} \left\{ -(x_\eta u)_\xi + (x_\xi u)_\eta + (y_\eta v)_\xi - (y_\xi v)_\eta \right\}$$

$$F_3 = \frac{2\mu}{|J|}\left\{-(x_\eta v)_\xi + (x_\xi v)_\eta\right\}$$

In the above equations, $\dot{x}, \dot{y}$ are the grid speeds (x_τ and y_τ), which are equal to zero if the physical mesh is stationary.

Equations 13.96, 13.97 and 13.98 can be written in the general form

$$\frac{\partial(E)}{\partial\tau} + \frac{\partial(G)}{\partial\xi} + \frac{\partial(H)}{\partial\eta} = S \tag{13.99}$$

where E, G, H and S are vectors. These are given by

$$E = \begin{Bmatrix} 0 \\ \rho u \\ \rho v \end{Bmatrix}, G = \begin{bmatrix} (y_\eta u - x_\eta v) \\ \rho\left(y_\eta u^2 - x_\eta uv\right) + (y_\eta p) - y_\eta F_1 + x_\eta F_2 \\ \rho\left(y_\eta uv - x_\eta v^2\right) - (x_\eta p) - y_\eta F_2 + x_\eta F_3 \end{bmatrix}$$

$$H = \begin{bmatrix} (-y_\xi u + x_\xi v) \\ \rho\left(-y_\xi u^2 + x_\xi uv\right) - (y_\xi p) + y_\xi F_1 - x_\xi F_2 \\ \rho\left(-y_\xi uv + x_\xi v^2\right) + (x_\xi p) + y_\xi F_2 - x_\xi F_3 \end{bmatrix}$$

$$S = \begin{bmatrix} 0 \\ \rho\dot{x}\left\{(y_\eta u)_\xi - (y_\xi u)_\eta\right\} + \rho\dot{y}\left\{-(x_\eta u)_\xi + (x_\xi u)_\eta\right\} \\ \rho\dot{x}\left\{(y_\eta v)_\xi - (y_\xi v)_\eta\right\} + \rho\dot{y}\left\{-(x_\eta v)_\xi + (x_\xi v)_\eta\right\} \end{bmatrix}$$

The above vector equations can be solved by a variety of finite difference based time marching schemes, such as the implicit, explicit, Crank-Nicolson, ADI or Predictor-Corrector methods.

13.6 Unstructured Mesh Generation

The techniques of unstructured mesh generation are primarily used in conjunction with the finite element solution procedure of the governing equations. In unstructured grids, the number of computational cells to which a typical grid point belongs or the number of neighbouring nodes to which it is connected, may vary with location (Figure 13.28). The mesh is not generated through coordinate transformation as in the case of structured grids; instead, it is obtained by the algorithmic division of the geometry into small elements of desired shape.

The steps involved in unstructured mesh generation are

1. division of the geometry into elements of desired shape and placement of nodes on the boundary and interior of the elements;

2. numbering of the nodes and elements and definition of nodal connectivity for each element. A certain order (usually, anti-clockwise) is maintained while specifying the connectivity of elements;

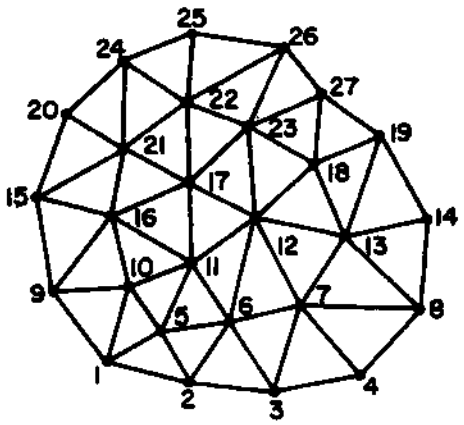


Figure 13.28: Unstructured Mesh

3. improvement of element aspect ratios by mesh smoothing.

Salient features of each of these steps are discussed in the following sections.

13.6.1 Generation of Nodal Connectivity for Elements

The numbering of elements and nodes and the definition of nodal connectivity can also be done algorithmically. For instance, consider the generation of these grid data in a quadrilateral geometry (Figure 13.29). Dividing the shape into $(NX \times NY)$ elements, a general formula for the connectivity of each element is found as (Figure 13.29)

$$\begin{aligned} n1 &= (iy-1)(NX+1)+ix \\ n2 &= n1+1D \\ n3 &= n2+NX+1 \\ n4 &= n3-1 \end{aligned}$$

For the complex shape shown in Figure 13.30, the mesh data for each quadrilateral zone can be generated by a similar procedure, with due care to avoid the already assigned numbers of elements and nodes. In general, numbering of nodes and elements in the shorter direction of the geometry is helpful for obtaining convenient matrix structure of the discretized equations. A mesh with 3-noded triangular elements could also be generated easily . For instance, each quadrilateral element (n) with nodal connection array (i, j, k, l) could be divided into two triangles with element numbers $(2n-1)$ and $(2n)$ and nodal connectivities (i, j, k) and (i, k, l) respectively. In most of the unstructured mesh generation procedures, the numbering of elements and nodes and the generation nodal connectivity data are performed simultaneously along with the division of the geometry into small elements.

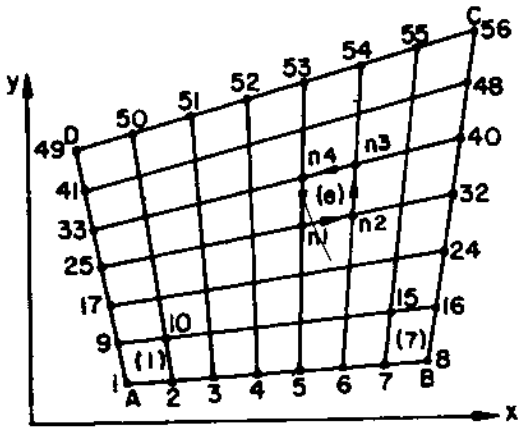


Figure 13.29: Generation of Unstructured Grid Data in a Quadrilateral

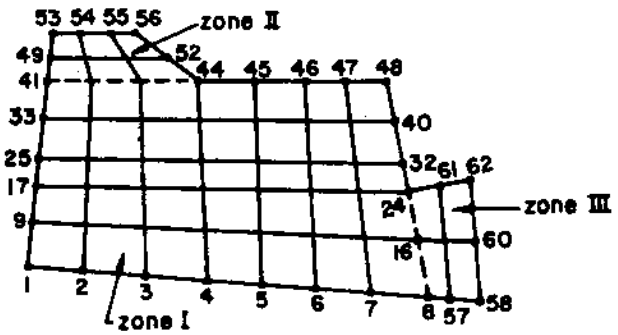


Figure 13.30: Unstructured Quadrilateral Mesh Generation in a Complex Domain

13.6.2 Mesh Construction Algorithms

Several techniques are available for dividing arbitrary geometries into elements of simple shapes. The commonly employed two-dimensional element shapes are triangles, rectangles, squares and quadrilaterals; for 3-D, tetrahedrons, cuboids and cubes are used. The mesh construction is based on techniques such as elemental subdivision, generation of Delaunay triangles/tetrahedrons and quadtree/octree division.

One of the earliest methods of generating an unstructured finite element mesh was proposed by Zienkiewicz and Phillips (1971). In this method, local isoparametric coordinates are employed for subdividing a curvilinear region (Figure 13.31). The geometry is mapped into a simpler shape, using relations

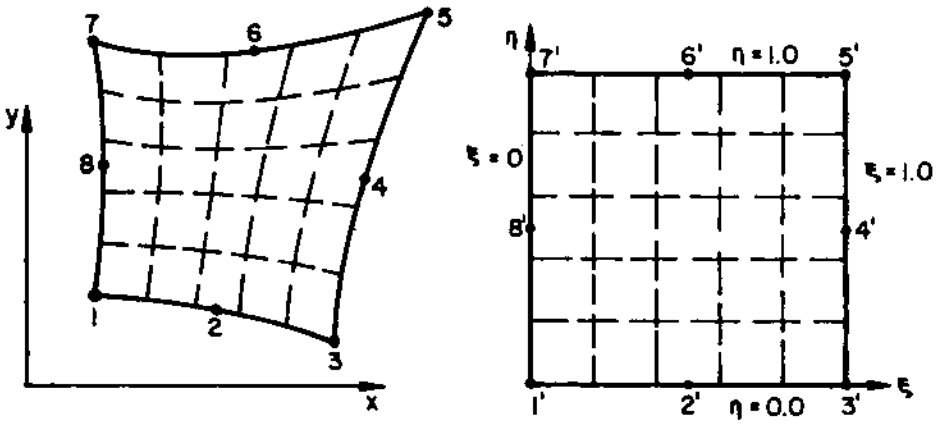


Figure 13.31: Element Subdivision by Isoparametric Transformation

of the form

$$x = \sum_{i=1}^{m} N_i\ (\xi, \eta)\ x_i \quad \text{and} \quad y = \sum_{i=1}^{m} N_i\ (\xi, \eta)\ y_i$$

where ξ and η are the isoparametric coordinates and N_i are the shape functions. The mesh is obtained from the above relations, after subdividing the $\xi - \eta$ domain into equal-sized elements.

Several interesting algorithms (known as *Triangulation* procedures) have been developed for generating triangular meshes within arbitrary shaped two-dimensional domains (Thacker, 1980). These are used widely because triangular elements can fill any complex shape with reasonable accuracy. Here, some of the commonly employed triangulation algorithms are described.

In Figures 13.32 (a-e), different ways of dividing a coarse triangular element are shown. For instance, three smaller triangles are formed by joining the vertices of a parent element to the centroid (Figure 13.32(b)). It is possible to subdivide into two or four elements, by connecting the mid-point of a long side to the opposite vertex (Figure 13.32(c)) or by joining the mid-side nodes with each other (Figure 13.32(d)). The new elements can be further subdivided, till sufficiently refined elements are obtained (Figure 13.32(e)). These subdivision procedures are quite useful for providing local refinement in a certain region, for a coarse triangular mesh.

Of the different options mentioned above for dividing a parent element, the methods shown in Figures 13.32(c) and (d) require subdivision of neighbouring elements also, while the method of Figure 13.32(b) can be applied locally even to a single element. However, the aspect ratios (height/base) for the later generation triangles are smaller, which is not a desirable consequence for the accuracy of numerical prediction. Some special techniques are available for improving the element aspect ratios, which are briefly described later.

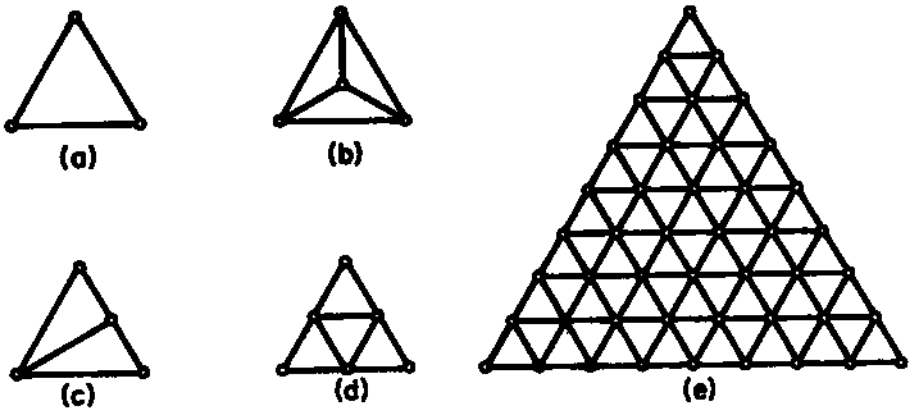


Figure 13.32: Division of Triangular Elements

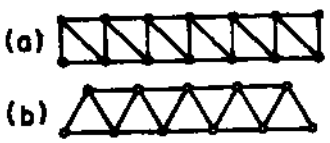


Figure 13.33: Triangulation of a Parallel Strip

In many triangulation algorithms, the given geometry is first divided into large elements of simple shape (say, triangles, rectangles, polygons or parallel strips) which are triangulated further. Triangulation of a thin parallel strip is shown in Figure 13.33(a). Each of the parallel sides is divided into m equal segments (with m being chosen as required) and triangles are formed as shown in the figure. It is also possible to select the number of divisions on the parallel sides as m and $(m+1)$, or m and $(m-1)$, depending on the difference in length between these sides (Figure 13.33(b)).

The algorithm of dividing a parallel strip can be used for triangulating any complex two- dimensional geometry. For instance, the geometry shown in Figure 13.34 is divided into equally spaced parallel strips starting from one side. The parallel strips are triangulated as described above. The two non-parallel strips at the ends are also handled in a similar fashion by dividing the curved side into n or $(n \pm 1)$ parts, where n is the number of divisions on the straight side. A point to be noted is that during such triangulation, the curvilinear boundaries are approximated by piecewise linear segments (shown as dotted lines in the figure). This approximation, however, can be reasonably good if the strip width is small enough.

A widely used technique which produces triangles of acceptable aspect ratios among N given points is *Delaunay triangulation*. The Delaunay triangles have the property that each of their vertices is surrounded by a polygon whose sides

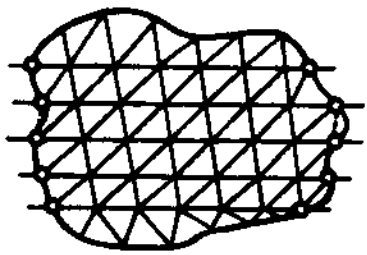


Figure 13.34: Triangulation by the Parallel Strip Method

are perpendicular bisectors of the sides of the triangles. The latter known as *Voronoi polygons* are non-intersecting and cover the whole domain consisting of the N given points (Figure 13.35). Thus, the generation of Delaunay triangles between N points has a dual problem of dividing the surrounding domain into N non-intersecting Voronoi polygons. Because of the equi-distant property of the Voronoi polygons, well-proportioned triangular elements are generated when the nodes enclosed by adjacent polygons are connected.

A simple algorithm quite effective for obtaining Delaunay triangles has been proposed by Watson (1981). In this method, nodes which can form Delaunay triangles are grouped as triplets. The radius of the circum circle (R_c) and the circum centre (O_c) are computed for each triplet (i, j, k). For a triplet to form Delaunay triangle, there should not be any other node within the circum disc (the region inside the circum circle). Therefore, of all possible triangles which might be formed, those with non-empty circum discs are rejected. When a new node n is introduced, it is verified whether n lies within any of the circum discs of the existing node triplets. This check is easily performed by comparing the distance (S) between the circum centre O_c and the point n, with the radius R_C; i.e., if $S < R_c$, the point is inside the circum disc. The triplets whose circum discs contain the newly introduced node are removed and the union of

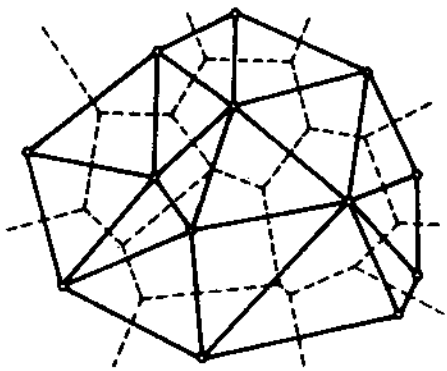


Figure 13.35: Delaunay Triangulation of a Given Domain

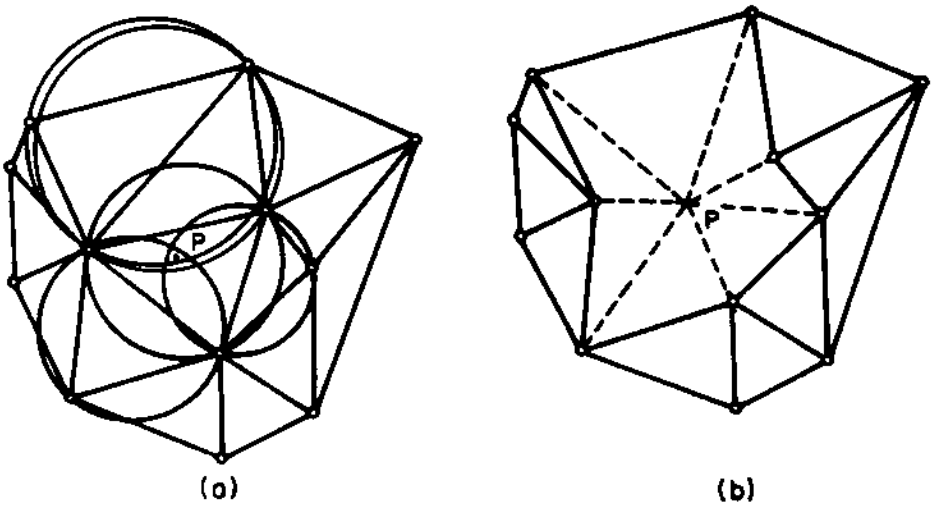


Figure 13.36: (a) Introduction of a New Node and (b) Division of the Insertion Polygon

all the rejected triangles forms what is known as the *Insertion Polygon*. Now each boundary node of the insertion polygon is connected to the node n to form new Delaunay triangles (Figures 13.36(a),(b)).

The process is started with three nodes which are farthest apart from each other and new nodes are introduced one at a time, until all the N nodes are exhausted. In actual applications, the N vertex points are also automatically generated. It is possible to generate Delaunay tetrahedra in 3-D problems by a similar procedure (Cavendish *et al.*, 1985).

Another important class of unstructured mesh generation is the Quadtree approach for 2-D geometries and the Octree method for 3-D situations. In quadtree procedure, a complex 2-D geometry is approximated by a sequence of square elements. The status of each square is assigned as: (i) completely inside, (ii) completely outside and, (iii) partially enclosing, as shown in Figure 13.37. The partially enclosing squares are further divided, while those completely outside or completely inside are left out from division.

The algorithm of checking whether a point lies inside or outside a given 2-D domain is repeatedly used for identifying the status of the newly formed squares. For this purpose, it is sufficient if only the intersection points are compared. For instance, compute the intersection points A and B between a vertical grid line and the boundary of the given geometry (Figure 13.37). The points P and Q which are formed by the intersection of the square grid lines can be identified as interior points since their coordinates lie between those of A and B. Repeating this process for all the horizontal and vertical grid lines, the points lying inside or outside can be identified. At later stages of the quadtree mesh structure, the check needs to be done only for the new nodes (and grid lines) which are

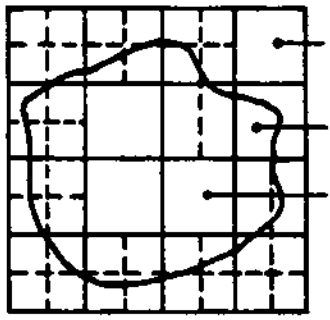


Figure 13.37: Quadtree Division

Figure 13.38: Smoothing of Badly Formed Grids

introduced by the division of the partially enclosing squares. After repetitive application of the quadtree procedure, the surface of the complex geometry is approximated by a stair-case like boundary.

In all the unstructured mesh generation techniques described above, some algorithmic division procedure is applied repeatedly. In order to terminate the division procedure, some limiting criterion is imposed on the total number of elements or the largest/ smallest element size or the number of divisional steps.

13.6.3 Mesh Smoothing Algorithms

In unstructured mesh generation, the final mesh often has a large variation of elemental aspect ratio over the physical domain. In such cases, smoothing techniques are applied to improve the quality of the grid. For instance, consider the badly formed triangles shown in Figure 13.38(a). By moving the point A to the centroid (A') of the polygon , some improvement is seen in the aspect ratios (Figure 13.38(b)). This procedure is applied iteratively to all the interior points of the mesh, until the element aspect ratios become uniform throughout the grid. For best results, the boundary points are also adjusted to obtain boundary segments of equal length. Occasionally the technique of *diagonal swapping* is performed between two triangles with a common side, to improve the aspect ratio of both the triangles (Figure 13.39). Such grid smoothing procedures are essential ingredients of most unstructured mesh generation algorithms.

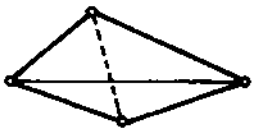


Figure 13.39: Diagonal Swapping

13.7 Adaptive Mesh Generation

Adaptive mesh generation implies the controlling of grid size depending on the magnitude of local gradients or some measure of local error in the solution. Thus, it is desired to reduce the grid size where the gradients (or error) is large, compared to the regions where variation in the solution is weak. This may be achieved by moving a given number of points within the solution domain to obtain an optimal grid structure or by increasing the number of grid points/order of interpolation until specified error criteria are met. These two approaches are referred as the Grid Distribution and Grid Refinement strategies respectively. For both the strategies, the grid evolves with respect to time or iterations depending on the computed solution. At every location within the grid, the grid spacing is selected according to the variation of a suitable weighting function which characterizes the error in the solution.

Historically, the grid distribution schemes have been used for the generation of structured meshes adaptively. This is because the transformed (computational) mesh always remains the same, while the grid points move in the physical domain according to the evolution of the solution. The grid refinement schemes, on the other hand, are suited for unstructured meshes since large elements may be subdivided locally to bring down the solution error. A variety of adaptive schemes have been developed based on the grid distribution and grid refinement approaches. A brief outline of some of the important methods in each category are discussed below.

13.7.1 Grid Distribution Methods

The task of distributing grid points optimally within the solution domain is akin to that of stationing observers to gather information from a region. The most active spots should have adequate number of observers (or grid points). Based on this concept, the distribution of grid points in adaptive grid generation problems is performed according to a weighting function W, which is calculated from the solution and its derivatives. The weighting function is selected such that wherever gradients are large, the grid spacing becomes small enough to accurately represent the sharp variations in the solution. In the most rudimentary form, for one dimensional grid adaption in x- direction, one could calculate the grid spacing by setting

$$\left|\frac{dT}{dx}\right| \cdot \Delta x = C \quad \text{(a constant)} \tag{13.100}$$

where T is the solution variable. Employing the above expression results in a grid with Δx proportional to $|dT/dx|$. Thus, $|dT/dx|^{-1}$ is the weighting function (W) for this case. The major difficulties in using the form given by Equation 13.100 are that the grid spacing Δx will be very large if the gradient approaches zero value (near an extremum point) and also $\Delta x \to 0$ if $T_x \to \infty$. A very small grid spacing can cause computational inaccuracy due to roundoff error. These difficulties can be overcome by choosing

$$W = \frac{1}{a + b\,|dT/dx|}$$

with suitable values of a and b to produce the required maximum and minimum grid spacings. More generalized expressions can also be attempted (Dwyer *et al.*, 1980) in the form

$$W = \frac{1}{a + b\,|dT/dx| + c\,|d^2T/dx^2| \ldots + k\,|d^nT/dx^n|} \tag{13.101}$$

where the constants $a, b, c \ldots$ can be chosen so as to get the desired grid distribution. Without any loss of generality, the grid spacing Δx which is varied in proportion to the value of W, can be taken in the form

$$\Delta x = W \Delta \xi \tag{13.102}$$

since the grid spacing $\Delta \xi$ in the computational domain is a constant. It is more convenient to express Equation 13.102 as follows

$$dx = W d\xi$$

or

$$\frac{\int_{x\,\min}^{x} dx}{\int_{x\,\min}^{x\,\max} dx} = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} = \frac{\int_0^{\xi} W d\xi}{\int_0^1 W d\xi} \tag{13.103}$$

It is also possible to rewrite Equation 13.102 as

$$\frac{1}{W} dx = d\xi \tag{13.104}$$

Defining $1/W = \bar{W}$

$$\xi = \frac{\int_{x\,\min}^{x} \bar{W} dx}{\int_{x\,\min}^{x\,\max} \bar{W} dx} \tag{13.105}$$

Thus, the transformation between ξ and x can be evaluated using either one of the expressions 13.103 or 13.105. Rewriting Equation 13.104 as $\Delta x/W =$

$\bar{W}\Delta x = \Delta\xi$ (a constant), it is clear that the grid is determined so as to equidistribute the quantity $(\bar{W}\Delta x)$ among all the grid points. For this reason, the grid distribution techniques are also called ***equidistribution*** methods.

For multi-dimensional grids, the above one-dimensional adaption technique can be extended by applying the equidistribution procedure along each of the grid lines. For instance, the curvilinear coordinate ξ can be evaluated through a modified form of Equation 13.105

$$\xi = \frac{\int_0^s \bar{W}(s)ds}{\int_0^{s\,\max} \bar{W}(s)ds} \tag{13.106}$$

where s is the arc length along the ξ-grid line (on which η and ζ are constants) and the weighting function is defined by

$$\bar{W}(s) = \frac{1}{W(s)} = a + b\left|\frac{dT}{ds}\right|_{\eta,\zeta} + c\left|\frac{d^2T}{ds^2}\right|_{\eta,\zeta} + \ldots k\left|\frac{d^nT}{ds^n}\right|_{\eta,\zeta} \tag{13.107}$$

The expressions for the other curvilinear coordinates (η and ζ) can also be derived in a similar fashion.

An interesting algorithm has been proposed by Nakahashi and Deiwert (1985), which considers the grid distribution problem as analogous to that of the stretching of an array of inter-connected springs . As shown in Figure 13.40, the nodes are assumed to be connected by tension and torsion springs, whose elastic constants are related to the gradients of the computed solution. In fact, the spring constant K of a tension spring between two neighbouring nodes is analogous to the weighting function W discussed earlier. The equidistribution principle applied in the present case is the force balance at each node, which leads to equations of the form $K_i\Delta s_i = K_{i-1}\Delta s_{i-1}$, when torsion springs are not used. The tension springs are used to control the spacing between grid points, while the torsion springs are needed to control the angles between grid lines.

For transient problems, it is possible to achieve the desired grid distribution by moving nodes at suitable velocities during each time step, instead of controlling the grid spacing or grid angles directly by static equidistribution. For instance, when a sharp moving front is present in the problem, a dense mesh surrounding the front can be moved with the same velocity as that of the front. Such a straight-forward displacement of mesh points is computationally inexpensive as compared to other adaptive schemes (Rao, 1993). Its success, however, depends on the accuracy of prediction for the front speed at any instant. For general applications also, grid speed schemes are available. One of the earliest grid speed methods is that due to Rai and Anderson (1981) in which the nodes are attracted to or repelled from each other depending on the magnitude of the local error. For one dimensional adaption, this author have proposed a grid speed equation of the form

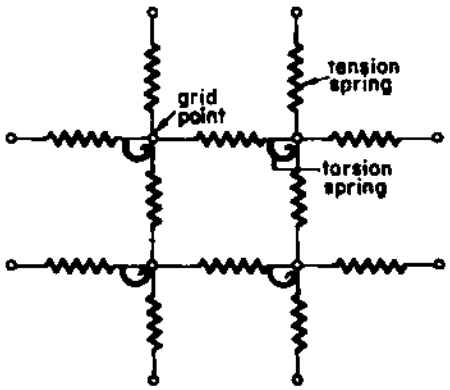


Figure 13.40: Spring Analogy Method

$$(\xi_t)_i = k\left(\sum_{j=i+1}^{n} \frac{|e|_j - |e|_{av}}{r_{ij}^n} - \sum_{j=1}^{i-1} \frac{|e|_j - |e|_{av}}{r_{ij}^n}\right) \tag{13.108}$$

$$(x_\tau)_i = -\frac{(\xi_t)_i}{(\xi_x)_i} \tag{13.109}$$

In the above equations, $(x_\tau)_i$ is the speed of the grid point i, e is the local solution error (i.e. the first truncated term of the Taylor series expansion at i), e_{av} is the average value of e over the whole domain and r_{ij} is the distance between two nodes i and j. Also, (x, t) are the physical coordinates, (ξ, τ) are the computational coordinates and n is a suitably selected constant. The above scheme provides higher grid speeds at locations of large local error, as compared to the regions of small error. Thus it tends to equidistribute the local error between grid points.

In the equidistribution and grid speed strategies discussed above, the emphasis is completely on controlling the grid spacings /angles and no effort is made to achieve the other desirable properties such as grid smoothness. An adaptive procedure which gives some weightage to all the required grid properties was proposed by Krishna and Sundararajan (1988). In this technique, the Poisson system of grid generation equations are solved, with the control functions P and Q selected as

$$P = \left(\frac{w\,(\partial T/\partial \xi)}{1 + (\partial T/\partial \xi)^2}\right)\left(\text{sign of } \frac{\partial^2 T}{\partial \xi^2}\right) \tag{13.110}$$

$$Q = \left(\frac{w\,(\partial T/\partial \eta)}{1 + (\partial T/\partial \eta)^2}\right)\left(\text{sign of } \frac{\partial^2 T}{\partial \eta^2}\right) \tag{13.111}$$

In the above equations, w is a suitable weight which is chosen so as to obtain the desired minimum spacing. The proposed forms for the control functions ensure that P and Q have the appropriate signs to attract points towards high gradient regions. Due to the inherent features of the Poisson solvers, the resulting grids approximately satisfy the extremum principles and exhibit a good deal of smoothness. However, the control on grid spacings/angles is only indirect. Also, this method requires more computational effort as compared to the grid speed and the equidistribution schemes.

For adaptive structured grid applications, the above discussed methods could serve as very useful tools. Depending on the requirements of each application problem, one method may be more attractive than the other.

13.7.2 Grid Refinement Methods

The grid distribution schemes, in spite of their versatility, suffer from some important drawbacks. For instance, in these schemes, there is no control on the magnitude of the global error. In other words, although the local error may be equidistributed well, the computed solution may not possess the desired accuracy due to insufficient number of nodes considered. There is no indication in the schemes to suggest that the number of nodes taken is too large or too small. Only by solving the problem repeatedly with different mesh sizes, it is possible to gauge the overall solution accuracy. Another difficulty often encountered with the grid distribution schemes is that grid clustering in high gradient regions occurs at the cost of extreme grid sparsity at other locations. This could lead to unacceptably coarse grids at some important locations such as domain boundaries. Therefore, it is necessary to consider schemes which have control over the global error and also, automatically arrive at the appropriate number of nodes for obtaining a specified level of solution accuracy.

The h-, p- and h-p refinement techniques available in the literature resolve the above- discussed difficulties quite effectively, for a large class of problems. Pioneering work has been done by Babuska and co-workers (1986) on the development of error estimates for the h- and p- adaptive strategies coupled with the finite element solution procedure. For transient problems, they have proposed an a posteriori error estimate (estimate of error derived from the computed solution) given by the expressions

$$\varepsilon(t) = \left\{ \sum_{n=1}^{N} E_n^2(t) \right\}^{1/2} \tag{13.112}$$

and

$$E_n(t) = \left\{ \frac{h_n^2}{12p^2} \int_{\Omega_n} r^2(x,t)dx \right\} \tag{13.113}$$

where $E_n(t)$ is the local error indicator, h_n is the size of the nth element (Ω_n), $r(x,t)$ is the finite element residue, p is the order of interpolation used in the

elements and $\varepsilon(t)$ is the a posteriori global error over the whole domain. For h-version refinement, $p = 1$ (linear degree approximation) for all the elements. In p-refinement, $p \geq 1$ for at least some elements. The objectives of the h- and p-refinements are (i) to find the element length (or grid spacing) h_n for each n so that the local error $E_n(t)$ is equidistributed (ii) to reduce the global error $\varepsilon(t)$ below a specified tolerance ε by proper choice of the number of elements N (in h- version) or by the choice of the order of interpolation (p) for each element (in p- version).

A statistical equidistribution technique is often applied (Rao, 1993) during grid adaption by the h- and p-version techniques. In the statistical procedure, the local error indicator $E_n(t)$ is considered as a random distribution. Defining the mean value and the standard deviation of the error distribution as $\bar{E}$ and σ respectively, the number of sub-divisions L_n for the nth element in the h-version refinement is evaluated as

$$L_n = K \left| \frac{(E_n - \bar{E})}{\sigma} \right| \tag{13.114}$$

where K is a suitable proportionality factor. For p-version grid adaption, a similar formula can be derived for the order of interpolation, given by the expression

$$p_n = K_p \left| \frac{(E_n - \bar{E})}{\sigma} \right| \tag{13.115}$$

It is to be noted that both the number of sub-divisions L_n and the order of interpolation p_n are always integers; therefore, Equations 13.114 and 13.115 need not be applied exactly and for practical purposes, any rough proportionality will do.

In p-version adaption, the decrease in error with an increase in the order of interpolation is rapid only for small values of p. Typically, beyond cubic or fourth order interpolation the error does not decrease; indeed, the accuracy of the solution could deteriorate due to instabilities. Therefore, values of p above 4 or 5 are not used. In many situations, the global error tolerance ($\varepsilon(t) \leq \varepsilon$) may not be satisfied even when the maximum permissible value of p is reached. Under these circumstances, the best approach is to combine the p- and h-methods and reduce the element size for those elements with $p = p_{\max}$. This mixed scheme is called as the h-p refinement technique.

The h- refinement technique is simple to implement and is very effective in controlling the error. However, it is computationally more expensive than the p-version since more number of linear elements ($p = 1$) are needed to meet the same error criterion as compared to the higher order ($p \geq 1$) elements. Therefore, the storage memory required by p-version adaptive procedures will be comparatively less than the h-refinement schemes.

References

Babuska I., Feedback, Adaptivity and A Posteriori Estimates in Finite Elements, in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, edited by I. Babuska, John Wiley, 1986.

Brackbill, M.V. and Saltzmann, J.S., Adaptive Zoning for Singular Problems in Two Dimensions, J. Computational Physics, Vol. 46, pp 342-368, 1982.

Cavendish J.C, Field D.A., and Frey W.H., An Approach to Automatic Three Dimensional Finite Element Mesh Generation, Int. J. Num. Meth. Engng., Vol 21, pp 329-347, 1985.

Churchill, R. V. and Brown, J. W., *Complex Variables and Applications*, McGraw-Hill, New York, 1990.

Currie, I.G., *Fundamental Mechanics of Fluids*, McGraw-Hill, New York, 1993.

Dwyer H.A., Kee R.J., and Sanders B.R., Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer, AIAA Journal, Vol 18, No 10, 1980.

Krishna M.S.M. and Sundararajan T., A Novel Adaptive Finite Element Technique Based on Poisson Grid Generation Equations, Proceedings of the Second International Conference held at University of Miami (USA) during Dec. 5-10, 1988.

Nakahashi K. and Deiwert G.S., A Three Dimensional Adaptive Grid Method, AIAA paper 85-0486, 1985.

Rai M.M. and Anderson D.A., Application of Adaptive Grids to Fluid Flow Problems with Asymptotic Solutions, AIAA Journal, Vol 20, pp 496-502, 1981.

Rao V.C.V., A Finite Element Study of Body-fitting and Adaptive Grid Generation with Application to Flame Propagation Problems, Ph.D. Thesis, Dept. of Mathematics, IIT Kanpur, India, May 1993.

Rao V.C.V., Sundararajan T., and Das P.C., A New Approach to Grid Generation using Finite Element Technique, Proceedings of the Second International Conference on Numerical Grid generation held at University of Miami (USA) during Dec. 5-10, 1988.

Steger J.L. and Sorenson R.L., Use of Hyperbolic Partial Differential Equations to Generate Body-Fitted Coordinates, in *Numerical Grid Generation Techniques*, NASA Conference Publication 2166, pp 463-478, 1980.

Thacker W.C., A Brief Review of Techniques for Generating Irregular Computational grids, Int. J. Num. Meth. Engng., Vol 15, pp 1335- 1341, 1980.

Thompson J.F., Warsi Z.U.A., and Mastin C.W., *Numerical Grid generation-Foundation and Applications*, North Holland publishers, 1985.

Warsi, Z. U. A., *Fluid Dynamics: Theoretical and Computational Approaches*, CRC Press, Boca Raton, 1992.

Watson D.F., Computing the n-dimensional Delaunay Tessellation with Applications to Voronoi Polytopes, Computer Journal, Vol 24, 1981.

Zienkiewicz O.C. and Phillips D.V., An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Coordinates, Int. J. Num. Meth. Engng., Vol 3, pp 519-528, 1971.

Problems

1. Transform a quadrilateral with vertices $A(0,0)$, $B(1,1)$, $C(1,2)$ and $D(-1,2)$ into a unit square, using a suitable algebraic transformation. Find the transformed expression for the steady heat conduction equation, for the above domain.

2. Develop a Fortran program for implementing the trans-finite interpolation algorithm in a general 4-sided two dimensional domain, with the coordinates of the boundary points as input.

3. Develop a Fortran program for generating an unstructured mesh of quadrilateral elements in a complex domain as shown in Figure 13.31, using the trans-finite interpolation algorithm in each subdomain.

4. Write a Fortran program for generating an unstructured mesh with triangular elements for an arbitrary-shaped 2-D geometry using the quadtree approach. Also, implement the smoothing algorithm in the program.

5. Using a one-dimensional transient temperature profile of the form $T(x,t) = T_0 \tanh\{20(x+10t)\}$, develop a 100-element adaptive mesh for the interval [0,1]. The minimum element size should be 0.0001. Use an equidistribution procedure with a weighting function which depends on the first and the second spatial derivatives of the temperature field.

6. *Conformal transformation* is a powerful technique that is available for mapping a class of geometries to standard form in two dimensions (Section 13.4.2, also see Churchill and Brown, 1990). Develop a grid generation algorithm based on this approach. Show constant coordinate lines in the physical domain. Also show that the grid thus generated is intrinsically orthogonal in the interior as well as the boundaries of the domain. For geometries with sharp corners, the *Schwarz-Christoffel* transformation can be used (Currie, 1995). Apply the conformal transformation technique to geometries such as

   (a) a cylindrical annulus with the inner cylinder located eccentrically located with respect to the outer;

   (b) a channel with sudden expnsion;

   (c) an elliptical cylinder placed in an unbounded medium,

   (d) the family of Joukowsky aerofoils.

   In each example, identify the geometry of the physical domain after transformation arising from conformal mapping.

7. It has been discussed in the text (Section 13.4.3) that body conforming grids can be generated by solving elliptic partial differential equations for the nodal coordinate functions $x(\xi,\eta)$ and $y(\xi,\eta)$. These map the coordinates (x,y) in the physical domain to a uniform grid on a square in the (ξ,η) space. The flow and transport equations can be solved either in the transformed plane by a finite difference approach (as outlined in Chapter 7), or by the finite volume method of Chapter 12 in the physical domain. In either situation, the mapping can be used to get uniform and orthogonal coordinate lines throughout the physical domain. The grid generation equations for generating orthogonal grids in two

dimensions have been shown by Thompson *et al.* (1985) to be:

$$\frac{\partial}{\partial \xi}(f\frac{\partial x}{\partial \xi}) + \frac{\partial}{\partial \eta}(\frac{1}{f}\frac{\partial x}{\partial \eta}) = 0 \tag{13.116}$$

$$\frac{\partial}{\partial \xi}(f\frac{\partial y}{\partial \xi}) + \frac{\partial}{\partial \eta}(\frac{1}{f}\frac{\partial y}{\partial \eta}) = 0 \tag{13.117}$$

where f is the distortion function given by

$$f = \frac{\sqrt{x_\eta^2 + y_\eta^2}}{\sqrt{x_\xi^2 + y_\xi^2}} \tag{13.118}$$

and the subscripts indicate partial derivatives. These equations are solved with Dirichlet boundary conditions over a portion of the domain, as well as the Neumann condition

$$x_\xi x_\eta + y_\xi y_\eta = 0$$

that imposes orthogonality on the grid at the domain boundaries. Develop a computer program for implementing the above approach, for example in a wavy channel.

8. Develop the elliptic grid generation procedure to map a complex geometry in the physical plane to a unit circle in the transformed domain, rather than a unit square. This approach may be helpful when the original geometry is close to (but not exactly) circular. An example of such a geometry is an ellipse in the physical domain. Solve the steady heat conduction equation with sources in such a geometry, subjected to mixed Dirichlet-Neumann conditions.

9. Develop the elliptic grid generation equations that include orthogonality and grid-spacing control; start from the variational principle of Brackbill and Saltzmann (1982).

14
Operator-Splitting Algorithm

This chapter is in two parts. Sections 14.1-14.5 describe the operator-splitting algorithm in the context of model advection-diffusion problems. Sections 14.6-14.11 present the application of this algorithm to enhanced oil recovery from porous formations using thermal methods.

14.1 Introduction

The upwind scheme is commonly used for solving advection-diffusion problems arising in flow, heat and mass transfer applications. An alternative technique to upwinding called the operator-splitting algorithm is described here. This algorithm (to be called OS) properly identifies the mixed mathematical character of the governing differential equation and solves each homogeneous component of the equation as accurately as possible. A typical advection-diffusion equation arising in heat transfer problems is of the form

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} = \alpha \nabla^2 T \tag{14.1}$$

where T is temperature, t is time, $\mathbf{u}$ is the velocity vector, α is thermal diffusivity and ∇ is the gradient operator. In Cartesian coordinates $\nabla = (\partial/\partial x, \partial/\partial y)$. The classification of the equation is elliptic as $t \to \infty$, parabolic when $\mid \mathbf{u} \mid$ is zero and hyperbolic when $\mid \mathbf{u} \mid$ is large. In all other cases, the equation is of mixed character. These special cases are summarized as:

$$\begin{aligned}
\text{Steady state} \quad (t \to \infty)\ \mathbf{u} \cdot \nabla\, T &= \alpha \nabla^2 T \quad \text{(elliptic)} \\
\text{Conduction limit} \quad (\mid \mathbf{u} \mid = 0\ )\ \frac{\partial T}{\partial t} &= \alpha \nabla^2 T \quad \text{(parabolic)} \\
\text{Advection limit} \quad (\mid \mathbf{u} \mid \to \infty)\ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T &= 0 \quad \text{(hyperbolic)}
\end{aligned}$$

The OS algorithm solves Equation 14.1 in the following two steps:

$$\textit{Step } 1 \ \text{(Predictor)} \ : \quad \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = 0 \ \text{ over a time step } \Delta t$$

$$\textit{Step } 2 \ \text{(Corrector)} \ : \quad \frac{\partial T}{\partial t} = \alpha \nabla^2 T \ \text{ over the same time interval } \Delta t$$

At high values of the fluid velocity the correction required is quite small and at low velocities the predictor step acting alone is inadequate. However, the two steps combined will completely solve the full equation irrespective of the magnitude of $\mathbf{u}$. The following points are worth noting:

1. The OS approach is completly free of upwinding and hence free of the false diffusion error.

2. The only error arising in the OS algorithm is due to time discretization and it is of order Δt. If either the predictor or the corrector equation is solved numerically on a grid of size Δx, then additional discretization errors ($\sim \Delta x^n, n \geq 1$) are possible. Hence the overall error is $e \sim O(\Delta x^n, \Delta t), n \geq 1$.

3. Numerical techniques are well established for solving the diffusion step, $T_t = \alpha \nabla^2 T$. This equation does not have an analytical solution when the geometry involved is complex. Hence it is quite common to solve the diffusion equation numerically using a second order scheme. The discretization error of the OS algorithm then is $e \sim O(\Delta t, \Delta x^2)$.

4. The main source of error in the solution of heat transfer problems occurs while appproximating the advective terms $(\mathbf{u} \cdot \nabla T)$. In the OS algorithm these terms are isolated from the complete equation and solved as accurately as possible. It is desirable that this part of the problem be analytically solved globally or analytically on a local grid.

5. On non-dimensionalization, the energy equation 14.1 becomes

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \frac{1}{\mathrm{Pe}} \nabla^2 T$$

where $\mathrm{Pe} = UL/\alpha$ is the Peclet number. In the definition of the Peclet number U is a velocity scale and L is the length scale. Pe is a measure of the relative importance of convective transport with respect to diffusive transport. Most existing algorithms fail or become inaccurate when Pe is raised. The OS algorithm proposed here is uniformly valid for any value of Pe.

6. **Fronts**: As the Peclet number is raised, an initial discontinuity will propagate through the flow domain unchanged. This is called a *front*. Predicting the occurrence of a front and its location is a challenging numerical problem. In this chapter we demonstrate the applicabilty of the OS algorithm to front-dominated problems.

7. The OS algorithm is only conditionally stable since it involves explicit time marching. The stability criterion is the familiar Courant condition and is stated in words as: *the numerical movement of the front should not exceed one grid spacing in one time increment since the maximum front speed is the fluid*

velocity. In dimensional form,the above condition is, $u\,\Delta t/\Delta x < 1$ for stability. If u depends on position the time step is chosen on the basis of the maximum velocity in the flow field.

14.2 One-Dimensional Problems

14.2.1 Model Heat Transfer Equation

Consider the governing equation

$$T_t + T_x = \frac{1}{\mathrm{Pe}} T_{xx}$$

with the initial condition, $t = 0$, $T = 0$ and the boundary conditions, $x = 0, T = 1$ and $x \to \infty$, $T \to 0$. This represents a physical problem in which a hot fluid with unit velocity is introduced suddenly (at $t = 0$) at the inlet ($x = 0$) of a one dimensional flow domain. The fluid in the physical domain ($0 < x < \infty$) is initially at a constant cold temperature. If the Peclet number is large, a thermal front whose thickness is small will move through the flow region with unit velocity. As $\mathrm{Pe} \to 0$, the front will be diffuse and energy transport is primarily due to the molecular mechanism of thermal diffusivity. The closed-form solution of the above equation is

$$T(x,t) = \frac{2}{\sqrt{\pi}} \int_{x/\sqrt{4t/\mathrm{Pe}}}^{\infty} \left(\exp(-\eta^2) \exp(\mathrm{Pe}[\frac{x}{2} - \frac{x^2}{16\eta^2}\mathrm{Pe}] \right) d\eta \tag{14.2}$$

The above integral is well-behaved and can be evaluated using Simpson's rule for numerical integration (Chapter 2). Values of the upper limit between 8 and 10 reduce the integrand to an order 10^{-6} for the range of Peclet number and x studied here.

A comparison of the predictions of central differencing, upwind differencing, and operator-splitting algorithm against the analytical solution (Equation 14.2) is presented here. The governing differential equation discretized using central differencing for spatial derivatives and an implicit formulation for time yields

$$\frac{T^{p+1} - T_i^P}{\Delta t} + \frac{(T_{i+1} - T_{i-1})^{p+1}}{2\Delta x} = \frac{1}{\mathrm{Pe}} \frac{(T_{i+1} + T_{i-1} - 2T_i)^{p+1}}{\Delta x^2}$$

where Δt is a time step, Δx is the spacing between nodes on the x-axis, p is the current time level and $(p+1)$ is the new time level at which a solution is desired. The above equation determines the value of temperature at the ith node. It can be rewritten as

$$[AT_{i-1} + BT_i + CT_{i+1}]^{p+1} = DT_i^p \tag{14.3}$$

where

$$A = -\frac{1}{2\Delta x} - \frac{1}{\mathrm{Pe}\Delta x^2}, \quad B = \frac{1}{\Delta t} + \frac{2}{\mathrm{Pe}\Delta x^2},$$

$$C = \frac{1}{2\Delta x} - \frac{1}{\mathrm{Pe}\Delta x^2} \text{and} \quad D = \frac{1}{\Delta t}$$

The notation $[\ \ ]^{p+1}$ stands for the bracketed value evaluated at time level $p+1$. The matrix structure arising from this formulation is

$$\begin{bmatrix} B & C & \cdot & & & \\ A & B & C & & \mathbf{0} & \\ \cdot & A & & & & \\ & & & & & \\ & & & & & \\ & \mathbf{0} & & & & \\ & & & & & C \\ & & & & A & B \end{bmatrix} \begin{Bmatrix} T_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ T_{N-1} \end{Bmatrix}^{p+1} = \begin{Bmatrix} DT_1 - A \\ DT_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ DT_{N-1} \end{Bmatrix}^{p}$$

Here nodes are numbered as $i = 0, 1, ...N$ and node N is located far enough for the conditon $T_N = 0$ to hold. The above matrix can be inverted using the tridiagonal matrix algorithm (TDMA).

The first order upwind scheme used for the convection term T_x is given as

$$uT_x = \frac{u_e T_i - u_w T_{i-1}}{\Delta x}, u > 0 \text{ and } = \frac{u_e T_{i+1} - u_w T_i}{\Delta x}, u < 0.$$

Nodes e and w are located at the centre of the pairs of nodes $(i, i+1)$ and $(i, i-1)$ respectively. In the present problem $u = 1$ and the finite difference equation becomes

$$\frac{T_i^{p+1} - T_i^p}{\Delta t} + \frac{T_i^{p+1} - T_{i-1}^{p+1}}{\Delta x} = \frac{1}{\mathrm{Pe}} \frac{(T_{i+1} + T_{i-1} - 2T_i)^{p+1}}{\Delta x^2}$$

This can be written as

$$[AT_{i-1} + BT_i + CT_{i+1}]^{p+1} = DT_i^p \tag{14.4}$$

for each node i and the resulting matrix equation can be inverted using TDMA. Here

$$A = \frac{1}{\Delta x} - \frac{1}{\mathrm{Pe}\Delta x^2}, \quad B = \frac{1}{\Delta t} - \frac{1}{\Delta x} + \frac{2}{\mathrm{Pe}\Delta x^2}, \quad C = -\frac{1}{\mathrm{Pe}\Delta x^2}$$

and

$$D = \frac{1}{\Delta t}.$$

When the OS algorithm is used, the governing equation is split as

$$\text{Predictor:} \quad \frac{\partial T}{\partial t} + T_x = 0$$

$$\text{Corrector:} \quad \frac{\partial T}{\partial t} = \frac{1}{\mathrm{Pe}} T_{xx},$$

each equation being applied over the same interval Δt. The boundary conditions are relevant only over the corrector step. The convective operator $T_t + T_x = 0$ has an analytical solution, $T(t - x) =$ constant. This can be written as

$$T(t + \Delta t - x) = T(t - (x - \Delta t))$$

Choosing $\Delta t = \Delta x$, this equation becomes

$$T(t + \Delta t - x) = T(t - (x - \Delta x)). \tag{14.5}$$

Hence, the temperature at a point at the next time step is obtained by transferring the value prevailing at the current time at the rearward node. If Δt and Δx do not match, then the value of T in the interval $(i-1, i)$ at a distance Δt from node i is transferred to i.

The corrector step solved by finite differences using implicit time marching is

$$\frac{T_i^{\,p+1} - T_i^{\,p}}{\Delta t} = \frac{1}{\mathrm{Pe}} \frac{(T_{i+1} + T_{i-1} - 2T_i)^{p+1}}{\Delta x^2}$$

This can be cast in the form

$$[AT_{i+1} + BT_i + CT_{i-1}]^{p+1} = DT_i \tag{14.6}$$

where

$$A = -\frac{1}{\mathrm{Pe}\Delta x^2}, \quad B = \frac{1}{\Delta t} - \frac{2}{\mathrm{Pe}\Delta x^2}, \quad C = -\frac{1}{\mathrm{Pe}\Delta x^2}$$

and

$$D = \frac{1}{\Delta t}$$

Once again TDMA can be used to invert the matrix arising from the above equation. Succesive applications of the two steps completes the calculation for a single time increment Δt.

Figures 14.1a and 14.1b show the plot of temparature distribution as a function of distance for two time levels $t = 1$ and 5 and two Peclet numbers $\mathrm{Pe} = 0.1$ and 100. A typical value of $\Delta x = \Delta t$ used in these calculations is 0.1 at $\mathrm{Pe} = 0.1$ and 0.05 at $\mathrm{Pe} = 100$. All four approaches given above are compared in these figures. The agreement among the numerical predictions and the analytical result is very good at $\mathrm{Pe} = 0.1$. At $\mathrm{Pe} = 100$, only the operator splitting algorithm matches the analytical result closely. The upwind method fails due to false diffusion inherent in its formulation. The central difference method fails due to matrix ill-conditioning and loss of diagonal dominance at high Peclet numbers. The OS algorithm is uniformly accurate at low as well as high Peclet numbers.

Further testing of the OS algorithm for transport problems with decay, energy generation and non-trivial initial conditions is described below.

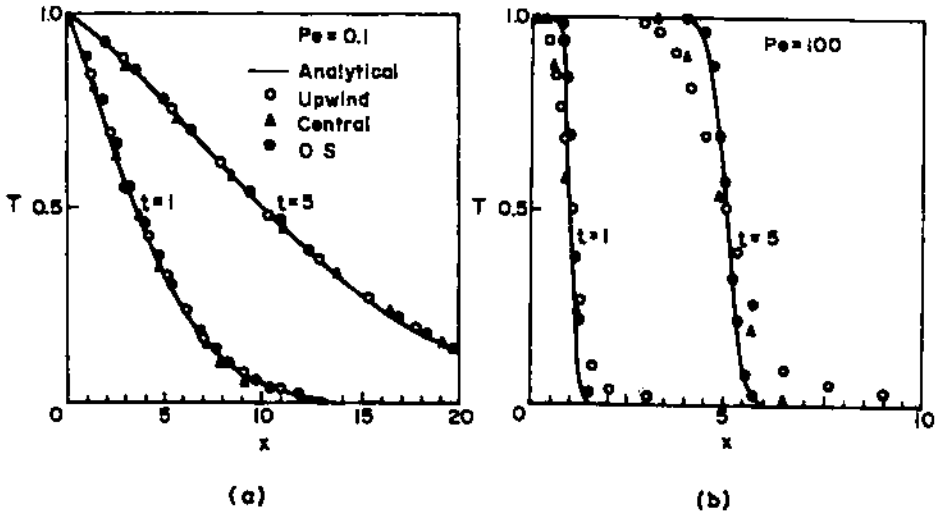


Figure 14.1: Evolution of Temperature with Distance and Time. (a) Pe = 0.1, (b) Pe = 100.

14.2.2 One Dimensional Transport with Decay

This problem is governed by the equation

$$\frac{\partial T}{\partial t} + T_x = \frac{1}{\mathrm{Pe}} T_{xx} - \lambda T \tag{14.7}$$

and the initial conditions, $t = 0$, $T = 0$ and the boundary conditions, $x = 0$, $T = 1$ and $x \to \infty$, $T \to 0$. It models transport of a chemical species whose concentration is T and has a finite life. λ, the dimensionless decay constant is $0.693\ L/(UT_{\frac{1}{2}})$. Here L is a length scale, U is a velocity scale and $T_{\frac{1}{2}}$ is the half-life of the element being transported in the flow. The OS algorithm splits this equation along the following steps:

$$\begin{aligned} \frac{\partial T}{\partial t} + T_x &= 0 && \text{(advection)} \\ \frac{\partial T}{\partial t} &= \frac{1}{\mathrm{Pe}} T_{xx} && \text{(diffusion)} \\ \frac{\partial T}{\partial t} &= -\lambda T && \text{(decay)} \end{aligned} \tag{14.8}$$

each being repeatedly applied over the same time step Δt. As described earlier the diffusion step is numerically solved by finite differences. The advection step has an analytical solution, $T(t - x) =$ constant while the decay step has

an analytical solution, $T \sim \exp(-\lambda t)$. The analytical solution of the decay problem is

$$T\ (x,t) = \frac{2}{\sqrt{\pi}} \int_{x/\sqrt{4t/\mathrm{Pe}}}^{\infty} (\exp(-\eta^2 + \mathrm{Pe}[\frac{x}{2} - \frac{x^2}{4\eta^2}(\frac{\mathrm{Pe}}{4} + \lambda)])d\eta \qquad (14.9)$$

The comparison among the upwind, central difference, operator-splitting and the analytical solutions are given in Figures 14.2a - 14.2d for a wide range of parameters. The presence of λ, the decay parameter prevents the formation of sharp fronts and all 4 methods are seen to be in good agreement with one another.

14.2.3 One Dimensional Transport with Sources

The governing equation for this problem is

$$\frac{\partial T}{\partial t} + T_x = \frac{1}{\mathrm{Pe}} T_{xx} + Q(x) \qquad (14.10)$$

subject to the initial and boundary conditions as in Equation 14.7. We choose energy source term to be $Q(x) = Q_0 e^{-\gamma x/\mathrm{Pe}}$. This form of Q models transport of a reactive mixture in which heat release occurs close to the inflow plane ($x = 0$). The mixture is initially either hot ($T = 1$ at $x = 0$) or cold ($T = 0$ at $x = 0$). The OS algorithm for this problem consists of the following steps:

$$\frac{\partial T}{\partial t} + T_x = 0 \qquad \text{(advection)}$$

$$\frac{\partial T}{\partial t} = \frac{1}{\mathrm{Pe}} T_{xx} + Q(x) \qquad \text{(diffusion + reaction)} \qquad (14.11)$$

The analytical solution of Equation 14.10 is

$$\begin{aligned} T(x,t) &= T_0 - \frac{2}{\pi} T_0 e^{x\frac{\mathrm{Pe}}{2}} \int_0^{\infty} \left[\frac{\beta \sin \beta x}{\beta^2 + (\frac{\mathrm{Pe}}{2})^2} \exp(-(\beta^2 + (\frac{\mathrm{Pe}}{2})^2)\frac{t}{\mathrm{Pe}} \right] d\beta \\ &+ \frac{Q_0 \mathrm{Pe}(1 - \exp(-\gamma \frac{x}{\mathrm{Pe}}))}{(\frac{\mathrm{Pe}}{2} + \frac{\gamma}{\mathrm{Pe}})^2 - (\frac{\mathrm{Pe}}{2})^2} \\ &-(\frac{2}{\pi} Q_0 \mathrm{Pe}) e^{x\frac{\mathrm{Pe}}{2}} \int_0^{\infty} \frac{\beta \sin \beta x \exp(-(\beta^2 + (\frac{\mathrm{Pe}}{2})^2)\frac{t}{\mathrm{Pe}})}{(\beta^2 + (\frac{\mathrm{Pe}}{2})^2)(\beta^2 + (\frac{\mathrm{Pe}}{2} + \frac{\gamma}{\mathrm{Pe}})^2)} d\beta \end{aligned} \qquad (14.12)$$

The indicated integrations have been performed in the present study with 2001 points using Simpson's rule and double precision arithmetic. Here $T_0 = 0$ for a cold flow at $x = 0$ and $T_0 = 1$ for hot flow at $x = 0$. For the results presented in this study we choose $Q_0 = 1$ and $\gamma = 1$. At low Peclet numbers a considerable amount of energy is released near $x = 0$ but the front is not sharp because of the dominant presence of molecular diffusion. Diffusional effects reduce in importance as Pe is raised. However at larger values of Pe the

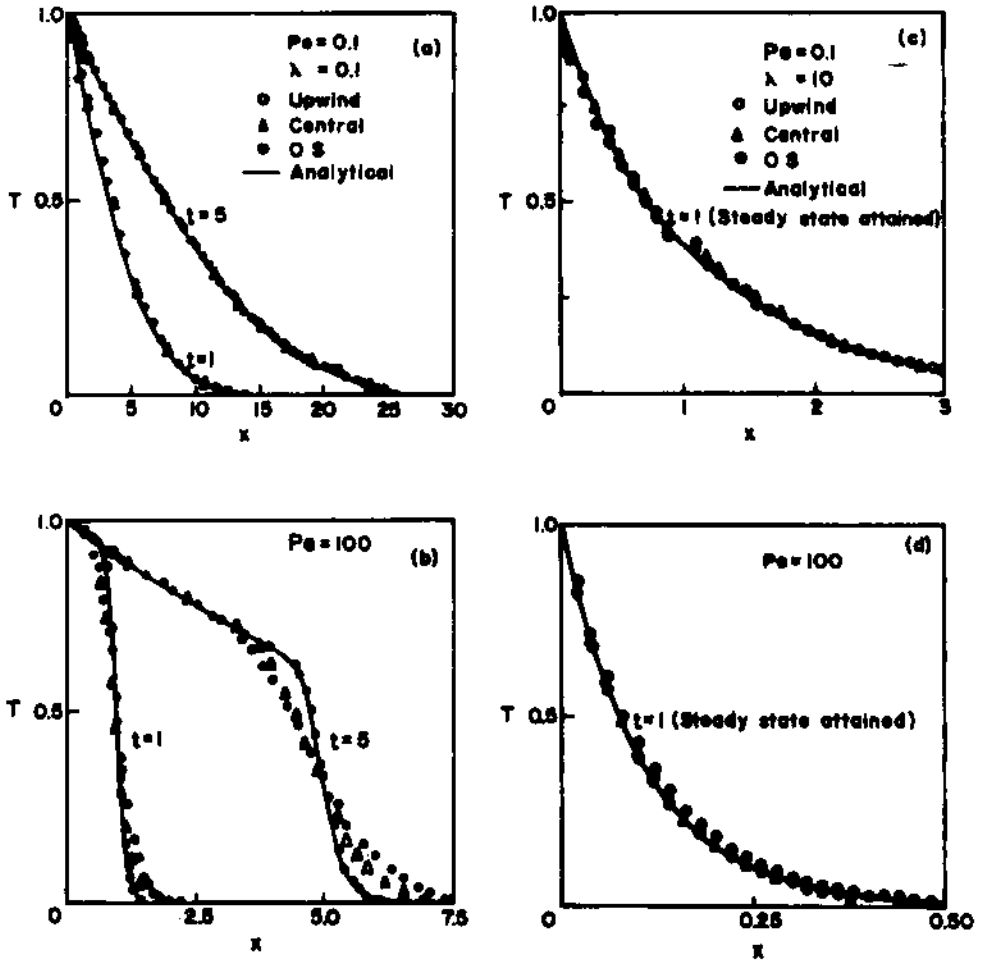


Figure 14.2: Evolution of Temperature with Distance and Time. Effect of Decay. (a) Pe = 0.1, $\lambda = 0.1$, (b) Pe = 100, $\lambda = 0.1$, (c) Pe = 0.1, $\lambda = 10$, (d) Pe = 100, $\lambda = 10$.

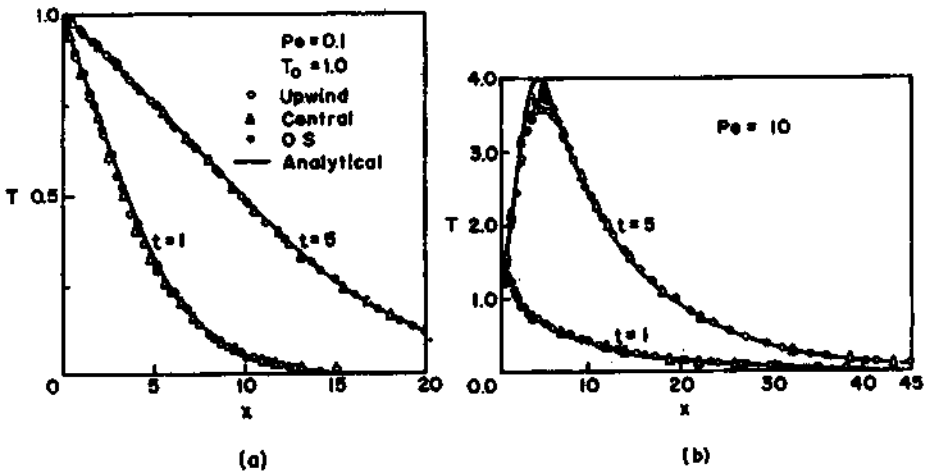


Figure 14.3: Evolution of Temperature with Distance and Time. Effect of Energy Generation. (a) Pe = 0.1, (b) Pe = 10.

reactants travel over a longer distance and energy is released over this length. Hence in the energy source problem, as in the decay problem formation of fronts is precluded. Figures 14.3a and 14.3b show a comparison of the upwind, central difference and the OS methods for the source problem. The agreement among the methods is seen to be good. The analytical solution is also shown in these figures for Pe < 10 and it matches the numerical solutions quite well. For Pe > 100 the analytical solution is difficult to evaluate because of the exponential term $\exp(x\mathrm{Pe}/2)$.

14.2.4 Gauss Test

The Gauss test consists of solving the following problem

$$\frac{\partial T}{\partial t} + T_x = \frac{1}{\mathrm{Pe}} T_{xx} \tag{14.13}$$

with $T(x,0) = \exp(-\alpha(x - x_0)^2)$, $T(0,t) = 0$ and $T(\infty,t) = 0$.

Note that the initial condition is inhomogeneous. It represents a Gaussian distribution in the variable T. At large values of Pe the expected solution is a uniform translation of this profile with unit velocity and no attenuation in amplitude. The analytical solution of this problem is

$$T(x,t) \quad = \quad \frac{1}{2\pi}\sqrt{\frac{\pi \mathrm{Pe}}{t}} \exp((\frac{x}{2} - \frac{t}{4})\mathrm{Pe}) U(x,t) \text{ where}$$

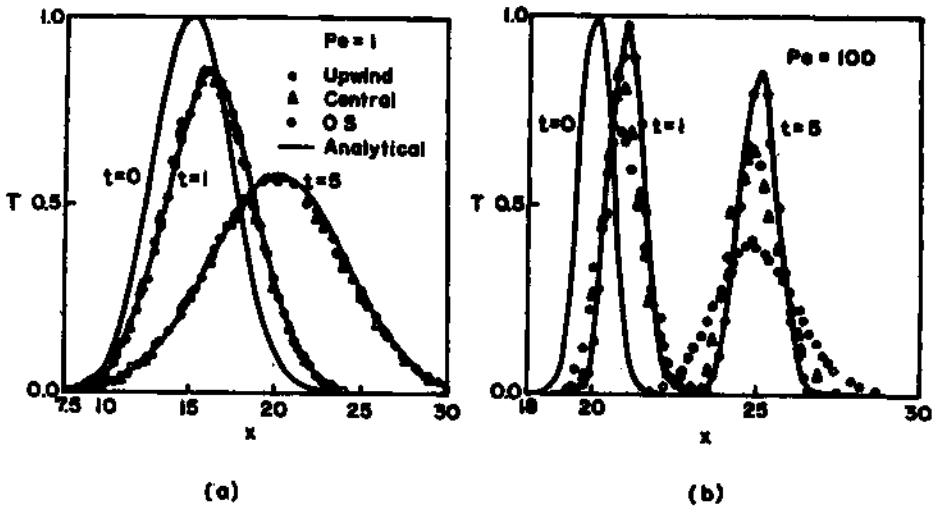


Figure 14.4: Spreading of a Gaussian Profile with Time. (a) Pe = 1, (b) Pe = 100.

$$U(x,t) = \int_0^\infty \exp(-\alpha(y-x_0)^2 - y\mathrm{Pe}/2)\,[e^{-(x-y)^2\mathrm{Pe}/4t} - e^{-(x+y)^2\mathrm{Pe}/4t}]dy \tag{14.14}$$

The integral $\int_0^\infty \exp\,(-\frac{x^2}{4\beta} - \gamma x)dx$ has a closed form value equal to $\sqrt{\pi\beta}$ exp $(\beta\gamma^2)(1-\phi(\gamma\sqrt{\beta}))$ where ϕ is the error function. This solution is used in evaluating U. Figures 14.4a and 14.4b show the transport of the initial Gaussian profile at two values of Pe. Once again the four methods (upwind, central differences, OS and analytical) are compared in these figures. The inadequacy of the upwind and central differencing procedures is clearly seen here at Pe = 100 since they show an unrealistic spreading of the initial profile as well as attenuation of the maximum temperature. The predictions of the OS algorithm match closely the analytical solution.

14.3 Two Dimensional Problems

Consider a model two dimensional convective heat transfer problem in which the flow is prescribed as uniform. The governing differential equation for this situation is

$$\frac{\partial T}{\partial t} + T_x = \frac{1}{\mathrm{Pe}}(T_{xx} + T_{yy}) \tag{14.15}$$

with the boundary conditions $T(x,0,t) = 0, T(x,Y,t) = 1$, $\partial T/\partial x(X,y,t) = 0$ and $T(0,y,t) = 0$ if $y < Y/2$ and $= 1$ if $y > Y/2$ and the initial condition $T(x,y,0) = 0$. These conditions represent two parallel streams of equal velocity ($u = 1, v = 0$) but unequal temperatures ($T = 0$ and $T = 1$) that come in contact suddenly at the inlet of a rectangular domain of size (X,Y). Simultaneously, the upper wall ($y = Y$) is maintained at a temperature $T = 1$, and the lower wall at $T = 0$. The fluid in the physical domain is initially at a constant cold temperature $T = 0$, possessing unit velocity in the x-direction. A thermal mixing layer forms at the interface of the two streams in which the temperature gradually changes from 0 to unity. The thickness of this layer grows in the downstream direction. At high Peclet numbers, the mixing layer is thin and the temperature distribution is almost a discontinuity. If Pe $\to 0$, the mixing layer spreads quickly to touch the boundaries.

In the present study the flow domain is taken to be square whose edge is 2 units. In the calculations a 26 × 26 grid is used with $\Delta x = \Delta y = 0.08$ and $\Delta t = 0.08$. The predictions of the OS algorithm are compared with those obtained using first order upwinding. The finite difference equation for a node (i,j) obtained using upwinding for the advective term is

$$[AT_{i-1,j} + BT_{i,j} + CT_{i+1,j} + DT_{i,j-1} + ET_{i,j+1}]^{p+1} = T_{i,j}^{p}$$

where

$$\begin{aligned} A &= -\frac{\Delta t}{\Delta x} - \frac{\Delta t}{\mathrm{Pe}(\Delta x)^2} \\ B &= 1 + \frac{\Delta t}{\Delta x} + \frac{2\Delta t}{\mathrm{Pe}(\Delta x)^2} + \frac{2\Delta t}{\mathrm{Pe}(\Delta y)^2} \\ C &= -\frac{\Delta t}{\mathrm{Pe}(\Delta x)^2} \\ D &= E = -\frac{\Delta t}{\mathrm{Pe}(\Delta y)^2} \end{aligned}$$

The TDMA algorithm is not applicable for this system of equations. However the criterion of diagonal dominance is satisfied, i.e. $|B| > |A| + |C| + |D| + |E|$ and the Gauss-Seidel method of matrix inversion can be used. When the OS algorithm is used the governing equation is split as

$$\begin{aligned} &I \quad \text{Predictor}: \quad \frac{\partial T}{\partial t} + T_x = 0 \\ &II \quad \text{Corrector}: \quad \frac{\partial T}{\partial t} = \frac{1}{\mathrm{Pe}}(T_{xx} + T_{yy}) \end{aligned}$$

each equation being applied over the same time interval Δt. The boundary conditions are relevant only over the second corrector step, as in the one dimensional problem. Solutions of the predictor and the corrector are calculated as follows. The advection steps (I) has an analytical solution, $T(t - x) =$ constant. This can be written as

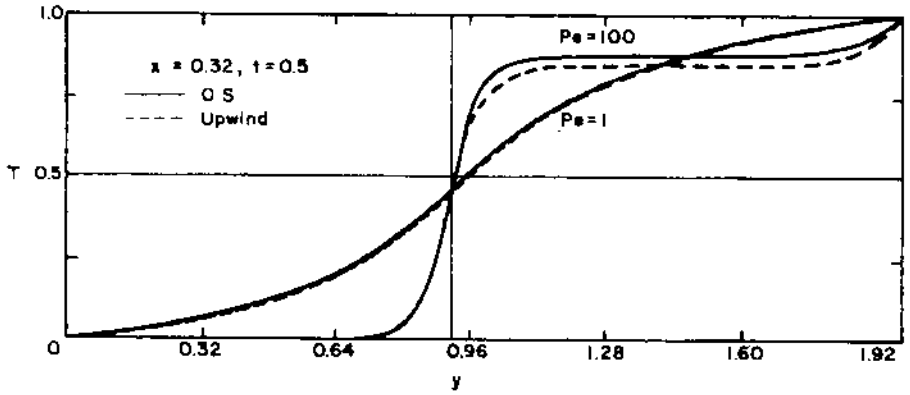


Figure 14.5: Temperature Profiles in a Two Dimensional Mixing Region

$$T^{p+1}(t + \Delta t) = T^{p}(t - (x - \Delta t))$$

As in the one dimensional problem, the temperature at a point at the new time step is obtained by transferring the value prevailing at the current time at the rearward node if $\Delta t = \Delta x$. If $\Delta t \neq \Delta x$, interpolation is required. The corrector step is the two dimensional heat equation and can be solved using the ADI technique.

Figure 14.5 shows temperature profiles at $x = 0.32$ and $t = 0.5$ computed using upwinding and the OS algorithm. At Pe = 1 the two profiles overlap. At Pe = 100 the upwind result is diffuse and shows a greater spreading rate of the mixing region.

14.4 Grid-Orientation Effect

In the one and two dimensional problems one can observe the effect of false diffusion in high Peclet number flows which are aligned with the Cartesian grid. In addition to the truncation errors in computations involving the field variable, there is also a problem of streamline-to-grid skewness in two and three dimensions. This is called the *grid-orientation effect*. An approximate expression for the false diffusion coefficient of the first order upwind scheme for a two dimensional transport problem is (Hughes, 1979; Roache, 1998):

$$\Gamma_{false} = \frac{\rho U \Delta x \Delta y \sin 2\Theta}{4(\Delta y \sin^2 \Theta + \Delta x \cos^2 \Theta)}$$

where U is the resultant velocity and Θ is the angle (between 0° and 90°) made by the velocity vector with the x direction. Note that Γ_{false} is a maximum when $\Theta = 45°$. The application ofthe OS algorithm to the problem of flow inclined to the coordinate lines is considered here. In the governing equation

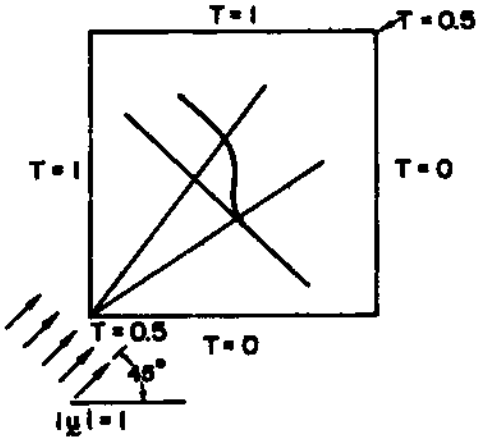


Figure 14.6: Flow Configuration in the Study of Grid Orientation effect

$$\frac{\partial T}{\partial t} + uT_x + vT_y = \frac{1}{\mathrm{Pe}}(T_{xx} + T_{yy}) \tag{14.16}$$

we specify $u = 1/\sqrt{2}, v = 1/\sqrt{2}$, so that the flow is at 45° to the grid. The initial condition $T(x, y, 0) = 0$ corresponds to an initially cold fluid. This problem is identical to the two dimensional problem in parallel flow except for the fact that the two parallel streams of equal unit velocity and of unequal temperature ($T = 0$ and $T = 1$), instead of being aligned with the grid, are inclined to it at 45° (Figure 14.6). Here again, a thermal mixing layer is formed. For definiteness, we study the problem at steady state. At steady state, the mixing layer thickness is inversely related to the magnitude of the Peclet number. This can be seen from the analytical solution of Equation 14.16, at large Peclet numbers namely,

$$T(\xi, \eta) = \frac{1}{2}(1 + \mathrm{erf}(\frac{\xi}{2}\sqrt{\frac{\mathrm{Pe}}{\eta}}))$$

where ξ and η are coordinates parallel and normal to the mean flow respectively.

Discretization using the first order upwind method for the advection terms results in the following equation valid at a node (i, j).

$$[AT_{i-1,j} + BT_{i,j} + CT_{i+1,j} + DT_{i,j-1} + ET_{i,j+1}]^{p+1} = T_{i,j}^{p}$$

where

$$\begin{aligned}
A &= -\frac{\Delta t}{\sqrt{2}(\Delta x)} - \frac{\Delta t}{\mathrm{Pe}(\Delta x)^2} \\
B &= 1 + \frac{\Delta t}{\sqrt{2}(\Delta x)} + \frac{\Delta t}{\sqrt{2}(\Delta y)} + \frac{2\Delta t}{\mathrm{Pe}(\Delta x)^2} + \frac{2\Delta t}{\mathrm{Pe}(\Delta y)^2} \\
C &= \frac{\Delta t}{\mathrm{Pe}(\Delta x)^2}
\end{aligned}$$

$$D = \frac{-\Delta t}{\sqrt{2}(\Delta y)} - \frac{\Delta t}{\mathrm{Pe}(\Delta y)^2}$$
$$E = -\frac{\Delta t}{\mathrm{Pe}(\Delta y)^2}$$

The system of equations can be solved for $T_{i,j}$ using Gauss-Seidel method. When OS formulation is used the governing equation is split as

$$I \quad \text{Predictor}: \quad \frac{\partial T}{\partial t} + \frac{1}{\sqrt{2}}T_x + \frac{1}{\sqrt{2}}T_y = 0$$
$$II \quad \text{Corrector}: \quad \frac{\partial T}{\partial t} = \frac{1}{\mathrm{Pe}}(T_{xx} + T_{yy})$$

The analytical solution of step I is obtained by using a new coordinate

$$\phi = \frac{1}{\sqrt{2}}(x + y)$$

This solution is $T(t - \phi) =$ constant and can be written as

$$T^{p+1}(t + \Delta t - \phi) = T^p(t - (\phi - \Delta t))$$

For $\Delta x = \Delta y$ and $\Delta t = \Delta\phi$, this equality means that the temperature at a point at the new time step is obtained by transferring the value prevailing at the current time at the diagonally opposite upstream node, i.e.

$$T^{p+1}_{i,j} = T^p_{i-1,j-1}$$

The corrector step II is solved as in Section 14.3 using ADI.

A comparison of temperature profiles obtained from the analytical solution, first order upwinding and OS algorithm is shown in Figure 14.7. The analytical solution cannot be used for comparison at low Peclet numbers such as Pe = 10 and Pe = 100 owing to the inapplicability of the boundary-layer approximation and the finite size of the region considered in numerical simulation. At Pe = 1000, the mixing layer is thin and temperature reaches values 1 and 0 well within the flow domain. At Pe = 1, both upwinding and operator splitting give identical results. The temperature profile becomes sharper at Pe = 100 and this is well reproduced by the OS solution. For Pe = 1000, we find the OS results come close to the analytical result. It was noted that at Pe = 100 and Pe = 1000, the upwind results were almost identical. Thus, it is to be concluded that for the grid used, false diffusion dominates over physical diffusion in the upwind solutions at Pe > 100.

For the fully evolved mixing layer, it can be shown that the mixing layer thickness δ, based on a temperature variation from 0.01 to 0.99 varies as $\delta\sqrt{\mathrm{Pe}}$ = constant. In Table 14.1, the $\delta\sqrt{\mathrm{Pe}}$ values as predicted by the two schemes are given for three Peclet numbers. We see that values from the upwind scheme

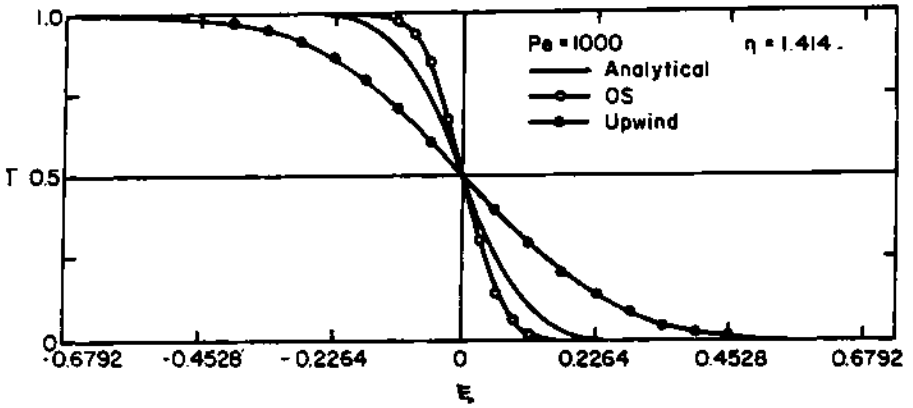


Figure 14.7: Temperature Distribution in Steady Two Dimensional Flow

come nowhere close to being a constant. OS scheme shows a much better response in relation to this analytical requirement. In Table 14.2, $\delta\sqrt{\text{Pe}}$ values predicted by OS method for different grid sizes is listed. Here M and N are number of nodes in x and y directions. It can be observed that despite a change in the magnitude of the constant $\delta\sqrt{\text{Pe}}$ remains nearly constant. This means that an increase in Pe reduces δ in the right proportion and produces a suitably sharp temperature profile.

Table 14.1: Comparison of $\delta\sqrt{\text{Pe}}$ Values on a 51 × 51 Grid.

Pe	Upwind	OS
10	6.86	6.66
100	19.32	7.81
1000	30.173	8.95

Table 14.2 : Effect of Grid-coarsening on $\delta\sqrt{\text{Pe}}$ Values.

Pe	$N(=M)=51$ $\Delta x(=\Delta y)=0.04$	$N(=M)=41$ $\Delta x(=\Delta y)=0.05$	$N(=M)=31$ $\Delta x(=\Delta y)=0.067$
10	6.66	6.64	6.614
100	7.81	7.954	8.014
1000	8.95	7.826	10.14

14.5 Interpolation Errors in Operator-Splitting Algorithm

When the hyperbolic part of the advection-diffusion equation is analytically solved interpolation is commonly required to transfer data along the characteristic direction. For example in one dimensional transport (Section 14.2) the

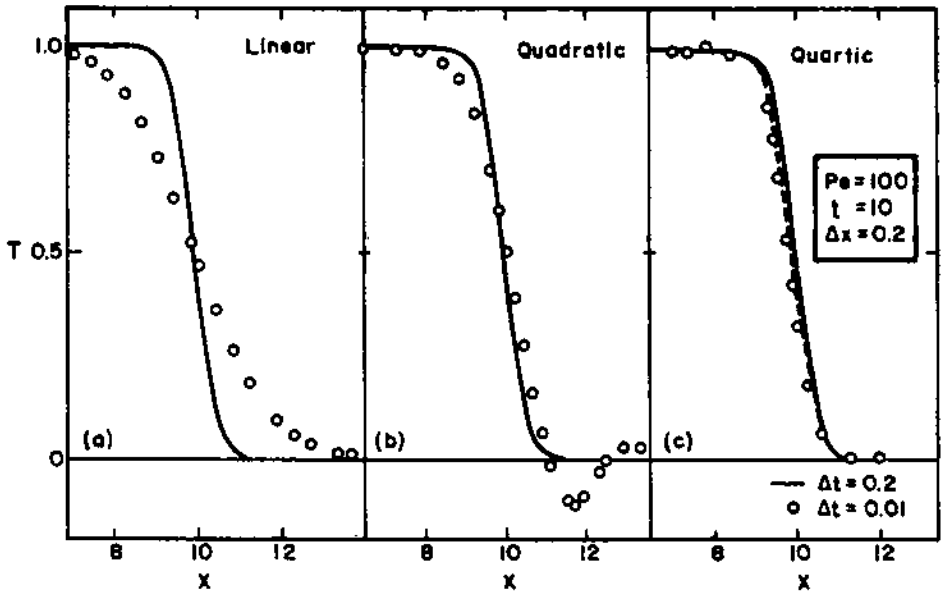


Figure 14.8: Additional Diffusion in OS Algorithm due to Interpolation

solution of the predictor step is $T(x, t+\Delta t) = T(x-\Delta t, t)$. If $\Delta t = \Delta x$ we obtain the new temperature T_i^{p+1} by equating it to T_{i-1}^{p}. Otherwise $\Delta t < \Delta x$ as per the stability requirement and T_i^{P+1} is obtained using interpolation among the temperature values $(T_i, T_{i-1}, T_{i-2} \ldots)^p$. Linear interpolation on a coarse mesh and for large time can result in diffusion-like errors and the advantage of having an analytical solution is lost. Figure 14.8 shows temperature profiles in a one dimensional transport problem with constant velocity at Pe = 100 and $t = 10$ units. The grid size is $\Delta x = 0.2$. Two values of Δt, namely 0.2 and 0.01 have been used. When $\Delta t = \Delta x$, the result overlaps the analytical solution. When $\Delta t < \Delta x$, Figure 14.8 shows temperature profiles obtained using (a) linear, (b) quadratic and (c) quartic interpolation respectively. Linear interpolation introduces diffusive errors and quadratic interpolation shows the formation of an undershoot below the minimum possible value of temperature. Fourth order quartic interpolation is seen to control the smearing of the front as well as oscillations and matches the analytical result quite well. Accordingly it is recommended as appropriate for OS calculations

In practice, the time and the space increments are independently chosen and velocity is not necessarily a constant. Hence the value of δx calculated as $u(x)\Delta t$ in the hyperbolic step will not equal the grid spacing Δx. Two interpolation schemes considered here for the variable velocity problem are 1. linear and 2. fourth order quartic with conditional limiters. The use of limiters controls oscillations in the temperature field. When a linear profile for temperature ia assumed we have $T_i^{n+1} = T_i^n(x_i - \delta x)$ where $T_i^n(x - \delta x) =$

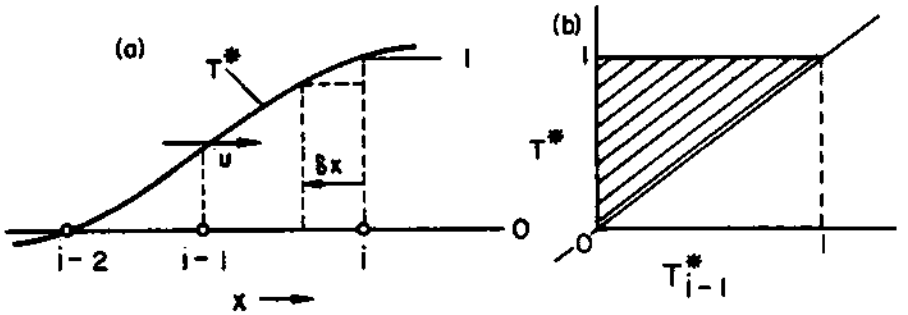


Figure 14.9: Nomenclature for Concept of Limiters

$T_i{}^n + (\delta x/\Delta x)(T_{i-1}^n - T_i{}^n)$. When a 4th order profile for temperature is assumed between two nodes i and $i-1$, we have $T(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$ where coefficients $a_0 \ldots\ldots a_4$ are functions of $T_{i-2}^n, T_{i-1}^n, T_i{}^n, T_{i+1}^n$ and T_{i+2}^n. $T(x_i - \delta x)$ is calculated using this formula subject to a constraint. This constraint is as follows. Define a dimensionless temperature (Figure 14.9a)

$$T^* = \frac{T - T_{i-2}}{T_i - T_{i-2}} \tag{14.17}$$

If the temperature at nodes $i-2, i-1$ and i respectively vary monotonically we require that $T^*(x_i - \delta x)$ be bounded by T_{i-1}^* and $T_i{}^*$. If T_{i-1} falls outside the bounds defined by T_i and T_{i-2}, then $T^*(x_i - \delta x)$, is equated to T_{i-1}^*. $T(x_i - \delta x)$ is subsequently reconstructed using Equation 14.17. The hyperbolic step then requires that the temperature at node i be upgraded as

$$T_i{}^{n+1} = T_i^n\ (x_i - \delta x)$$

This is equivalent to restricting the solution to the shaded region in Figure 14.9b.

The method of universal limiters described above has been developed extensively by Leonard (1988). To test the interpolation schemes referred above the following model one dimensional equation has been considered:

$$\frac{\partial T}{\partial t} + u(x) T_x = \frac{1}{\mathrm{Pe}} T_{xx} \tag{14.18}$$

Equation 14.18 is in dimensionless form. The initial and boundary conditions are, $T\ (x, 0) = 0, T\ (0, t) = 1$ and $T\ (\infty, t) = 0$. Front-like solutions form when $\mathrm{Pe} \gg 1$. The errors introduced by interpolation become significant under these conditions. A value of $\mathrm{Pe} = 100$ has been used to test the interpolation schemes. Four different velocity profiles ($u(x)$ in Equation 14.18) have been considered in the present study. These are

$$\begin{aligned} 1.\ u &= 1 \quad \text{[constant]} \\ 2.\ u &= 1 + 2(1 - e^{-x/2}) \quad \text{[acceleration]} \end{aligned}$$

$$
\begin{aligned}
3.\ u &= 2.73 - 2(1 - e^{-x/2}) \quad \text{[deceleration]} \\
4.\ u &= 1 + 2\exp[-5(x-1)^2] \quad \text{[Gaussian]}
\end{aligned} \tag{14.19}
$$

For the velocity profiles given by Equation 14.19 the Dirichlet condition at $x = \infty$ could be applied in the present study at $x = 4$ dimensionless units.

The OS algorithm has been applied to Equation 14.18 with the four velocity profiles (Equation 14.19) and two interpolation schemes. The results obtained here are compared with those from a first order upwind scheme. The hyperbolic part of the advection-diffusion equation in the OS algorithm has been solved by the method of limiters described by Leonard and Niknafs (1991). Solutions become grid independent when the domain $0 < x < 4$ is discretized with 81 nodes when the OS method is used. With the first order upwind scheme 201 nodes have been used, but the error arising from false diffusion is seen to be large on this refined mesh as well.

When the ratio $C = u\Delta t/\Delta x$, the Courant number is equal to unity, no interpolation is required. Interpolation errors increase as C departs from unity. The time step used for solving Equation 14.18 produces C in the range 0.1-0.3.

Figures 14.10-14.13 show temperature profiles at $t = 1$ and for a Peclet number of 100. The four velocity profiles given above have been used in these calculations and are also shown in these figures. When u is a constant an analytical solution is possible. For this case the predictions of the upwind scheme and the OS algorithm with two types of interpolation have been plotted along with the analytical solution. The data marked *ref* is the the solution obtained from the OS algorithm along with the method of Leonard and Niknafs (1991) for the hyperbolic step. Figures 14.10 (a-d) show the following general trends. For a constant velocity the OS method with quartic interpolation matches the analytical solution closely. The upwind scheme is consistently diffuse even on a grid with 201 points. The OS method with linear interpolation also shows diffusive errors. In contrast, the data marked *ref* produces a sharp front for temperature. This trend is seen for non-constant velocity profiles as well. As a rule, data for quartic interpolation with limiters is bounded by the *ref* solution above and linear interpolation below. Since for the constant velocity problem, quartic interpolation matches the analytical solution it may be considered as the most accurate among the interpolation schemes studied.

14.6 Application to Enhanced Oil Recovery

Liquid as well as gaseous hydrocarbons are found in the pores of sedimentary rocks. The host rock containing these fluids is often called a *reservoir*. The process of extraction of hydrocarbons from their natural surroundings is called *oil recovery*. Enhanced oil recovery (EOR) is an activity concerned with increasing the quantity of hydrocarbons from variously structured reservoirs. EOR is generally applicable to the secondary or the tertiary phase of recovery in the productive life of the reservoir. The primary recovery is by a natural

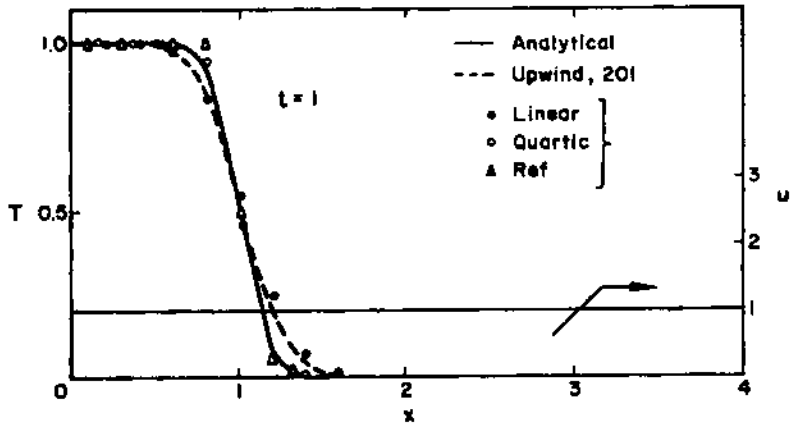


Figure 14.10: Temperature Profile in a Constant Velocity Field: Comparison of Five Solutions

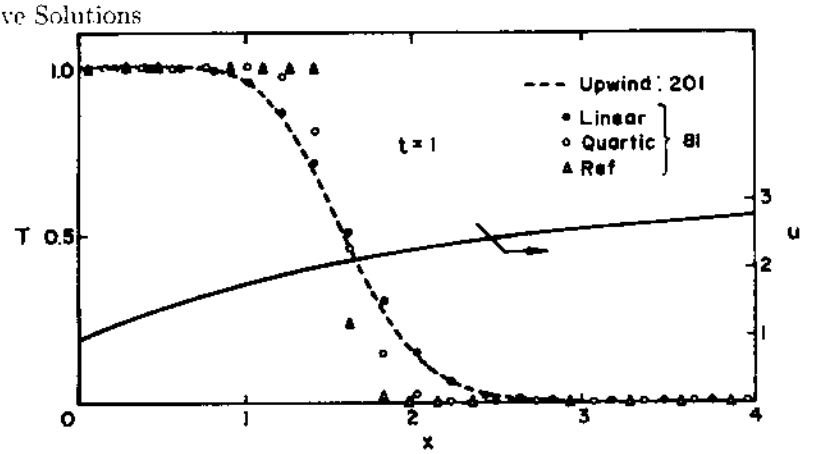


Figure 14.11: Temperature Profiles in an Accelerating Velocity Field

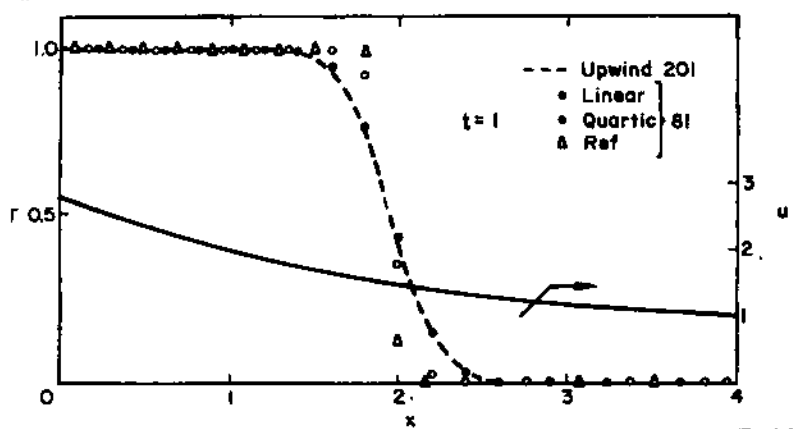


Figure 14.12: Temperature Profiles in a Decelerating Velocity Field

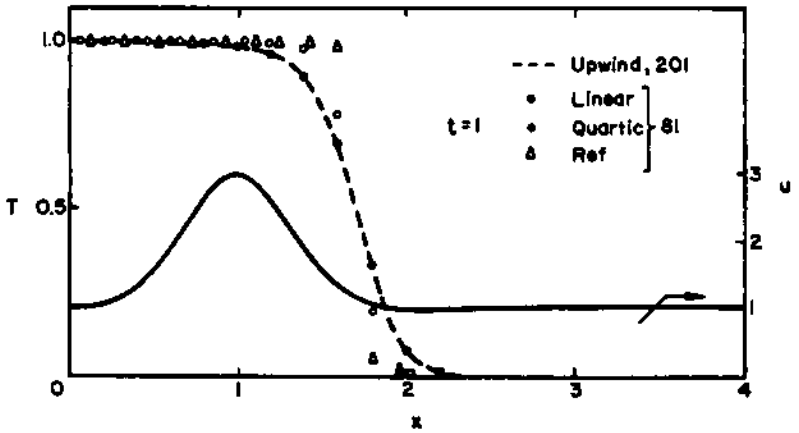


Figure 14.13: Temperature Profiles in a Gaussian Velocity Field

drive mechanism owing to the pressure of the resident fluid in the formation. The secondary recovery refers to techniques whose purpose is to maintain the reservoir pressure, by water injection for example. Here, the injected fluid displaces the resident fluid across a moving interface. The speed of the interface is controlled by a combination of various physico-chemical processes. The heavier, low specific gravity oils are too viscous to move to the production wells by this technique at rates sufficient to justify an economical operation. Thermal methods are therefore used at the tertiary phase to recover hydrocarbons by introducing heat to lower the oil viscosity. Hot water injection and *in situ* combustion are examples of such thermal methods.

Owing to heterogeneities in the porous formation or those in the pressure and temperature fields, the flow pattern in the reservoir can become quite complex. A pattern that is particularly disadvantageous is the appearance of single or multiple fingers in the physical domain. The fingering phenomenon refers to the bypassing of the resident fluid such as oil by an invading fluid, such as water. The finger in turn contains the bulk of the invading fluid. This definition encompasses the consequences of instability of the fluid interfaces caused by viscous forces, capillary forces and permeability distributions. Heterogeneities in the porous medium add to the complexity of the nonlinear fluid-fluid interactions in multiphase flow.

Enhanced oil recovery has all the characteristics of multiphase flow of oil and water in a porous medium. The flow behavior is dictated by gravitational, viscous and capillary forces. The gravity force causes phase migration in the direction of the gravitational field. The viscous and capillary forces influence the relative phase movement. Moreover, the capillary forces play a fundamental role in controlling the phase distribution in heterogenous porous media. Mathematical modelling is a useful tool for delineating the important flow mechanisms in a reservoir and thus predict its overall performance.

Extraction of oil from porous formations and fractured reservoirs using cold and hot water injection is numerically studied here. The flow is mathematically

modelled in terms of two pressure equations for the oil and water phases, and the thermal energy equation. The pressure equations are derived by combining the mass balance equation with modified Darcy's law that is applicable for flow in an unsaturated porous medium. The resulting system of equations is strongly coupled and non-linear. Further the energy equation is of the advection-diffusion type. The related Peclet numbers are quite high and temperature propagates as a front through the porous region. The energy equation has been solved in the present application by the operator-splitting method. The suitability of this method for oil recovery problems has been independently assessed. Results are presented for the amount of oil removed as a function of time. An increase in temperature is seen to improve oil recovery in general. However for large temperature difference between the oil and the water phases, oil production actually falls over the time period considered in the present work.

14.7 Nomenclature

Bi	Parameter related to heat loss to the host rock
c	Specific heat capacity, J/kg K
C	Courant Number
k_r	Relative permeability of porous medium
K	Absolute permeability of porous medium, Darcies
k_h	Thermal conductivity of porous medium, $W/m\ ^\circ$ K
p_{cow}	Capillary pressure in an oil-water system, Pa
Pe	Peclet number
S	Phase saturation
$t,\ \Delta\ t$	Time and time step, seconds
T	Temperature, °C
$\mathbf{u}$	Darcy velocity with components (u, v), m/s
$x,\ y$	Cartesian coordinates, m
$\Delta x, \Delta y$	Grid size along x and y axes respectively, m
X, Y	Domain dimensions along x and y axes respectively, m
Greek Symbols	
β	Expansivity $^\circ C^{-1}$
ε	Porosity
μ	Dynamic viscosity, Pa-s
ρ	Density, kg/m^3
$(\rho c)_R$	Heat capacity per unit volume of the porous medium, J/m^3 K
ξ	Compressibility, Pa^{-1}
Superscript	
n	Current time step
Subscript	
i, j	Node indices along x and y axes
$o,\ w$	Oil and water phases
R	Rock formation

14.8 Introduction to Oil Recovery

Enhanced oil recovery from a porous formation is numerically studied here. Resistance to oil movement arises from viscous as well as surface tension forces. The goal of enhanced oil recovery technique is to reduce this resistance and improve the mobility of oil (Boberg, 1988; Lake, 1994). Thermal methods are frequently used to lower oil viscosity while surfactants are used to lower surface tension. The present study concerns modelling of oil removal by hot and cold water injection.

Equations governing isothermal and nonisothermal two phase flow in a porous medium are obtained from the mass balance principle for the two phases, Darcy's law as applicable to an unsaturated porous region, and the energy equation in terms of a local volume-averaged temperature. These must be supplemented by constitutive relations that connect capillary pressure, relative permeability and water saturation. The governing equations can be reduced to a coupled system of non-linear partial differential equations in terms of three dependent variables. These variables can be chosen as oil pressure, water saturation and temperature. Alternatively they could be chosen as oil and water pressures and temperature. The first set of variables have been used by Ewing (1983) and Stevenson *et al.* (1991). In view of the similarity in the form of the governing equations for oil and water pressures, the second set of variables has been used in the present study. The pressure equations are diffusion-like while saturation and temperature are governed by advection-diffusion equations. When saturation is used as a dependent variable, the appropriate equation must be carefully solved. In the present study, pressure equations are solved using an implicit central difference scheme and saturation is determined from the difference between oil and water pressures. The energy equation has been solved using the operator splitting algorithm.

Results have been presented in this work for saturation and temperature profiles and oil recovery with hot and cold water injection. The data for the fluid and formation properties has been adapted from Boberg (1988). The linear dimensions and properties of the porous region are appropriate for a laboratory scale testing of the water injection technique.

14.9 Formulation of Phase Equations

The two dimensional physical domain that is considered for analysis and the coordinate system are shown in Figure 14.14. The domain is filled with a porous medium containing oil at a prescribed initial saturation and temperature. The medium is taken to be bounded by impermeable rocks at $y = 0$ and $y = Y$. Water at a pressure and temperature higher than that of oil is injected at the left face ($x = 0$) of the physical domain. Oil recovery is initiated by the physical displacement of oil trapped in the pores by water at a higher pressure.

The two problems studied here are as follows. 1. The injected water has the same temperature as the porous formation. This is referred to as isothermal

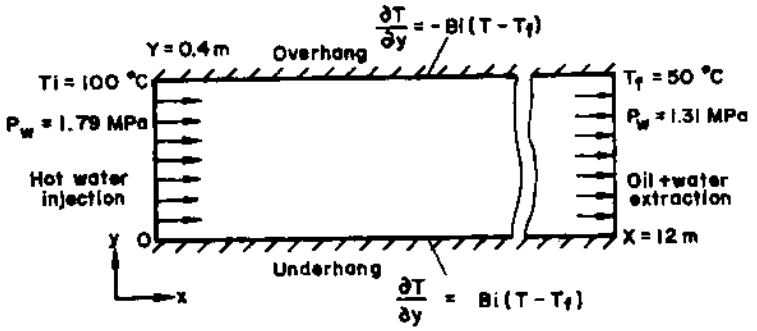


Figure 14.14: Description of Physical Domain, Coordinates and Boundary Conditions

injection. 2. The injected water is at a higher temperature than the formation. This is referred to as non-isothermal injection. It is assumed here that the two fluids oil and water fill the available pore space in the reservoir. The two phases are also assumed to co-exist, in the sense that the saturation of either of the phases does not reach an extreme value of zero or unity. Equations governing transient two phase flow in a porous medium can be developed under these set of approximations (Ewing, 1983; Sheorey *et al.*, 2001). They are given as follows:

Mass balance

$$\frac{\partial(\varepsilon S_i \rho_i)}{\partial t} + \nabla.\mathbf{u}_i \rho_i = 0 \quad (i = oil, water) \tag{14.20}$$

Darcy's law

$$\mathbf{u}_i = -\frac{K k_{ri}}{\mu_i} \nabla p_i \tag{14.21}$$

Energy Equation

$$\frac{\partial T}{\partial t} + UT_x + VT_y = \left(\frac{K_n}{\sigma_T}\right) \nabla^2 T \tag{14.22}$$

$$\begin{aligned} \text{where } U &= (u_o \rho_o c_o + u_w \rho_w c_w)/\sigma_T, \\ V &= (v_o \rho_o c_o + v_w \rho_w c_w)/\sigma_T \\ \text{and } \sigma_T &= \varepsilon[(1 - S_w)\rho_o c_o + S_w \rho_w c_w] + (1 + \varepsilon)(\rho c)_R \end{aligned} \tag{14.23}$$

In Equations 14.20 - 23, $p, \mathbf{u}, S$ and T refer to pressure, velocity, saturation and temperature respectively. Other symbols are defined in the nomenclature. The equations given above must be supplemented by constitutive relations of the type

$$
\begin{aligned}
k_{rw} &= k_{rw}(S_w) \\
k_{ro} &= k_{ro}(S_w) \quad [\text{ relative permeability }] \\
p_o - p_w &= p_{cow}(s_w) \quad [\text{ capillary pressure }] \\
\frac{\rho_i}{\rho_{i,ref}} &= 1 + \xi_i(p_i - p_{i,ref}) - \beta_i(T - T_{ref}) \\
S_o + S_w &= 1
\end{aligned} \tag{14.24}
$$

In Equation 14.20 mass sources are assumed to be absent. Gravity terms have been dropped in Equation 14.21 since the vertical extent of the physical domain is taken to be small. Data for Equations 14.20 - 14.22 is adapted from Boberg (1988) and is briefly reproduced in Appendix 14.1.

Equations 14.20 and 14.21 can be combined to generate two equations for oil and water respectively. These equations are

$$
\begin{aligned}
-S_o\beta_o\frac{\partial T}{\partial t} + S_o\xi_o\frac{\partial p_o}{\partial t} &- \frac{dS_w}{dp_{cow}}\left[\frac{\partial p_o}{\partial t} - \frac{\partial p_w}{\partial t}\right] \\
&= \frac{1}{\rho_o}\nabla.\left[\frac{Kk_{ro}\rho_o}{\mu_o\varepsilon}\right]\nabla p_o
\end{aligned} \tag{14.25}
$$

$$
\begin{aligned}
-S_w\beta_w\frac{\partial T}{\partial t} + S_w\xi_w\frac{\partial p_w}{\partial t} &- \frac{dS_w}{dp_{cow}}\left[\frac{\partial p_o}{\partial t} - \frac{\partial p_w}{\partial t}\right] \\
&= \frac{1}{\rho_w}\nabla.\left[\frac{Kk_{rw}\rho_w}{\mu_w\varepsilon}\right]\nabla p_w
\end{aligned} \tag{14.26}
$$

Equations 14.22, 14.25 and 14.26 are solved simultaneously for oil and water pressures and the local temperature. Saturation profiles are obtained from the constitutive relation 14.24. Oil recovery is calculated in terms of the percentage pore volume (ppv) defined as

$$
ppv = \frac{\text{Initial volume of oil} - \text{present volume of oil}}{\text{Pore volume}} \times 100
$$

The initial and boundary conditions correspond to a prescribed oil pressure and temperature at $t = 0$ and prescribed oil and water pressures on the inflow and outflow planes of the physical domain for all time. The side walls at $y = 0$ and Y are impermeable to flow and are treated as partly conducting to thermal energy. These conditions are

$$
\begin{aligned}
\text{Initial conditions } t &= 0,\ S_w, p_w,\ T \quad \text{prescribed.} \\
\text{Boundary conditions } x &= 0 \text{ and } X, s_w, p_w\ T \text{ prescribed} \\
y &= 0 \text{ and } Y,\ \frac{\partial}{\partial y}(p_o, p_w) = 0 \\
y &= 0 \text{ and } Y, \\
&\quad \frac{\partial T}{\partial Y} + \mathrm{Bi}(T - T_f) = 0
\end{aligned} \tag{14.27}
$$

The parameter Bi appearing in Equation 14.27 represents the extent of heat loss to the side walls. Bi = 0 refers to an insulating wall boundary condition. Typical values of the prescribed pressures and temperatures are shown in Figure 14.11. The magnitude of $X = 12$ m is chosen to be large enough for the Dirichlet conditions in Equation 14.27 to be valid at the outflow plane. The region over which *ppv* is calculated is chosen to be smaller than X.

14.10 Method of Solution

Equations 14.22, 14.25, and 14.26 along with the initial and boundary conditions (14.27) have been solved using a central difference scheme for marching in time. Average values of coefficients such as $Kk_r\rho/\mu\varepsilon$ in Equations 14.25 and 14.26 are evaluated using the harmonic average of values prevailing at the neighbouring nodes. The discretized form of the oil and water pressure equations on a Cartesian mesh of size Δx and Δy is as follows.

$$[A_o p_{o_{i+1,j}} + B_o p_{o_{i,j}} + C_o p_{w_{i,j}} + D_o p_{o_{i-1,j}} + E_o p_{o_{i,j+1}} + F_o p_{o_{i,j-1}}]^{n+1} = G_o^{n+1} \tag{14.28}$$

$$[A_w p_{w_{i+1,j}} + B_w p_{w_{i,j}} + C_w p_{o_{i,j}} + D_w p_{w_{i-1,j}} + E_w p_{w_{i,j+1}} + F_{w_{i,j-1}}]^{n+1} = G_w^{n+1} \tag{14.29}$$

Complete forms of the coefficients $A, B, ...G$ are given in Appendix 14.2. Equations 14.28 and 14.29 are coupled and produce a complex matrix structure. Alternately solving for oil and water pressures minimizes the bandwidth of the coefficient matrix to be inverted. This approach is not advantageous because the strong coupling between the phase pressures can delay the convergence of the iterations. The resulting equation arising jointly from equations 14.28-29 has been inverted using a generalized sparse matrix solver (Duff, 1980). These equations are nonlinear and hence must be repeatedly solved within a time step till convergence is attained. In the non-isothermal problem, iterations include successive solutions for the fluid pressures and temperature (Sheorey *et al.*, 2001).

A convergence criterion of 0.1 percent at each time level has been used in the present study. A grid consisting of 121 × 5 nodes for p_o, p_w and T has been used here. This level of refinement has been found to be adequate to obtain a grid-independent solution. The level of refinement in the y-direction is increased in problems where Bi, the heat loss parameter is large. However solutions for oil recovery with Bi > 0 lie within the band formed by the isothermal solution and the non-isothermal solution corresponding to Bi = 0. Hence results have not been presented here for non-zero values of Bi. Results have been presented in this study for oil recovery at the end of 3 hours. The time step used for this simulation is 0.01 hour.

The OS algorithm applied to the energy equation 14.22 splits it into two parts as

$$\begin{aligned} &\text{(Predictor)} \quad \frac{\partial T}{\partial t} + UT_x + VT_y = 0 \\ &\text{(Corrector)} \quad \frac{\partial T}{\partial t} = \left(\frac{K_h}{\sigma_T}\right) \nabla^2 T \end{aligned} \tag{14.30}$$

The predictor step is a hyperbolic equation and is solved using the method of characteristics (Section 14.2.1). The corrector step is a parabolic diffusion equation and is solved using an ADI scheme. The predictor equation is first simplified to the form

$$\frac{\partial T}{\partial t} + \bar{U} T_\psi = 0$$

Here $\bar{U} = \sqrt{U^2 + V^2}$ and ψ is the local streamline direction. This can be further simplified as

$$\frac{\partial T}{\partial t} + T_\phi = 0$$

with ϕ being defined as $\phi = \int d\,\psi/\bar{U}$. The temperature at the new time level is obtained from the solution of this reduced equation as

$$T^{n+1}(t + \Delta t - \phi) = T^n(t - (\phi - \Delta t)$$

Interpolation where required is accomplished using quartic interpolation with conditional limiters.

14.11 Results

Isothermal and non-isothermal water injection methods for oil recovery are considered here. For the non-isothermal problem results obtained using the OS algorithm and the three different methods for solving the hyperbolic step are compared against one another. The isothermal problem is studied first. Figure 14.15 shows the rise in saturation level of water with time in the flow domain along its centreline. The formation temperature as well as the injected water temperature is 50°C in this figure. The dashed line represents the smoothened initial condition used in the study. The movement of the saturation profile with time indicates the positive displacement of oil. The *ppv* calculated as a percentage clearly depends on the size of the physical domain used for its determination. In the present study, the overall length of the domain is 12 m

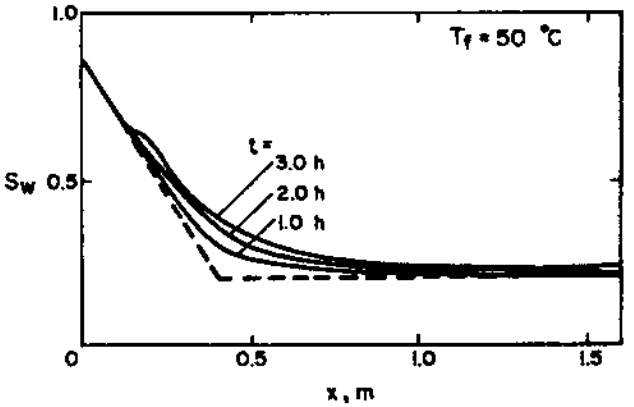


Figure 14.15: Progress of Saturation Front: Isothermal Injection

but *ppv* is calculated over a distance of 1.6 m. The large size of the physical domain guarantees the correctness of the Dirichlet condition at the exit plane of the porous region. For a 3 hr simulation, the movement of the saturation profile is within 1.6m. Figure 14.16 shows a plot of *ppv* versus time for three different formation temperatures. In all the three cases *ppv* increases with time. The increase is more marked at higher formation temperatures owing to a drop in the oil viscosity with an increase in temperature.

Non-isothermal problems are considered next. Figure 14.17 shows the temperature profile along the centreline of the porous region. The water injection temperature is 100°C. Heat loss to the rocks is taken to be zero in these calculations. The three methods used to solve the hyperbolic step of the OS algorithm are compared in this figure. As seen in the test cases with a prescribed variable velocity field, the data marked *ref* is the sharpest of the three temperature profiles. Linear interpolation produces the most diffuse solution, *ref* the sharpest and quartic interpolation lies between these two solutions.

Figure 14.18 shows the saturation profile for this problem. The movement of water in the non-isothermal case is front-like. This is clearly the effect of a local drop in oil viscosity upto the location of the temperature front. It must be noted that the Peclet number calculated in terms of the average water velocity and a composite oil-water diffusivity is in the range 1000 to 2000 for the time duration considered. Since this is a large value one may expect conventional upwinding for solving the energy equation to be in serious error.

Figure 14.19 shows a plot of *ppv* versus time for two pairs of values of injection and formation temperatures. The three methods used with the OS algorithm are again compared here. It can be seen that *ppv* is not too sensitive to the choice of the method that solves the hyperbolic step. However linear interpolation produces a slightly larger value of *ppv*. This is expected because diffusive errors inherent in this method (as well as in upwinding) move the

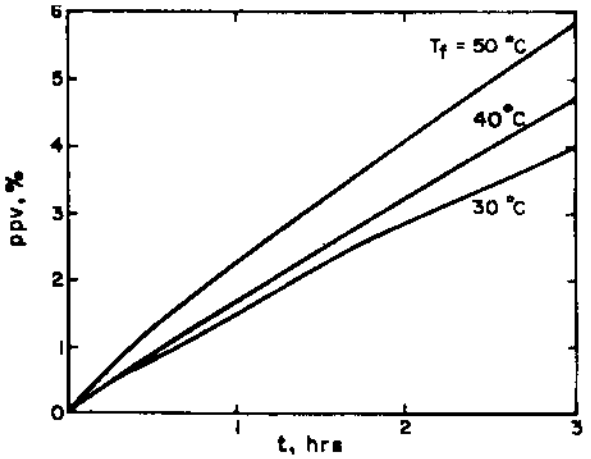


Figure 14.16: Effect of Formation Temperature on Oil Production : Isothermal Injection.

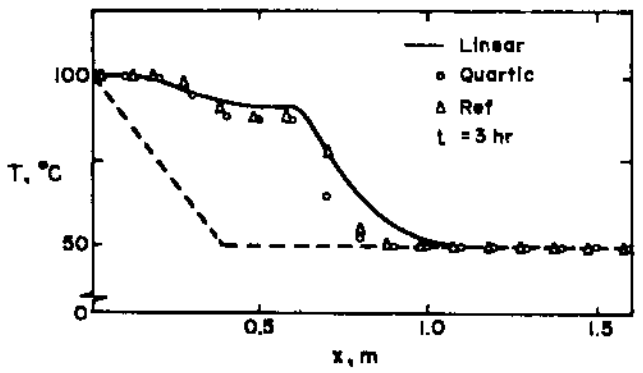


Figure 14.17: Temperature Profiles Predicted by three Interpolation Schemes: Nonisothermal Injection.

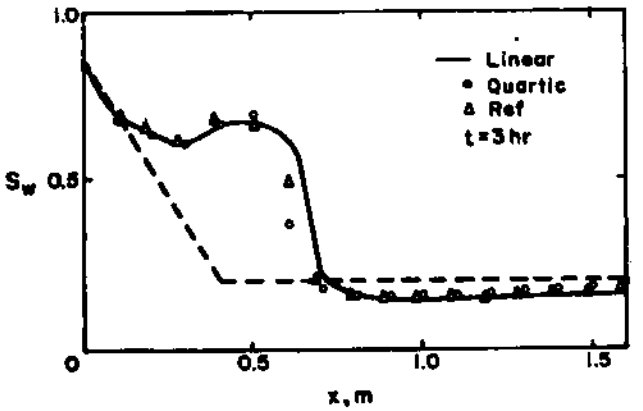


Figure 14.18: Saturation Profiles Predicted by Three OS Schemes: Nonisothermal Injection.

thermal front deeper into the oil-rich region. This results in an additional drop in oil viscosity and an overall increase in *ppv*.

Figure 14.20 compares the extent of oil recovery calculated with the isothermal and non-isothermal models for water injection. As stated earlier the effect of varying Bi, the heat loss parameter from 0 to large values is to shift respectively the curves from the non-isothermal limit to the isothermal limit. Results for intermediate values of Bi are not shown here. When T_f is 50°C isothermal water injection is seen to give a large value of *ppv* for $t < 2$ hours. Beyond this time the curves cross-over and non-isothermal injection is seen to be advantageous. When T_f is 30°C, *ppv* for isothermal injection is seen to be larger than for the non-isothermal problem for the full 3 hour simulation studied here.

These trends can be explained with reference to Figures 14.15 and 14.18 as follows. In the isothermal problem water saturation profiles uniformly move through the porous region resulting in a definite oil recovery. In the non-isothermal problem the saturation profiles are front-like and the increase in *ppv* with time is dependent on the front speed. Oil displaced by water gets accumulated ahead of the front in the space vacated by water. This is the cause of dip in the saturation profile ahead of the front as shown in Figure 14.18 and the sluggish response of *ppv*. This trend will continue till the saturation front gets close enough to the outflow plane of the problem domain and results in the cross-over of *ppv* curves.

In practice hot water injection will extract a larger volume of oil compared to the cold case, though this recovery is slower. The speed with which the front travels can be estimated using an idealized model. Consider a porous region containing primarily water (region 1) displacing an oil-saturated region 2 across a planar interface. The average resistance to front movement can be calculated from Darcy's law (Equation 14.21) to be the ratio the average pressure drop

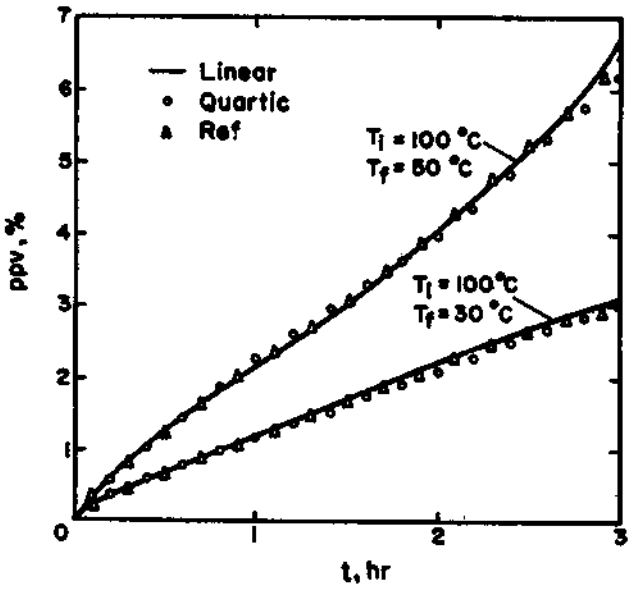


Figure 14.19: Prediction of *ppv* by OS Algorithm: Nonisothermal Injection.

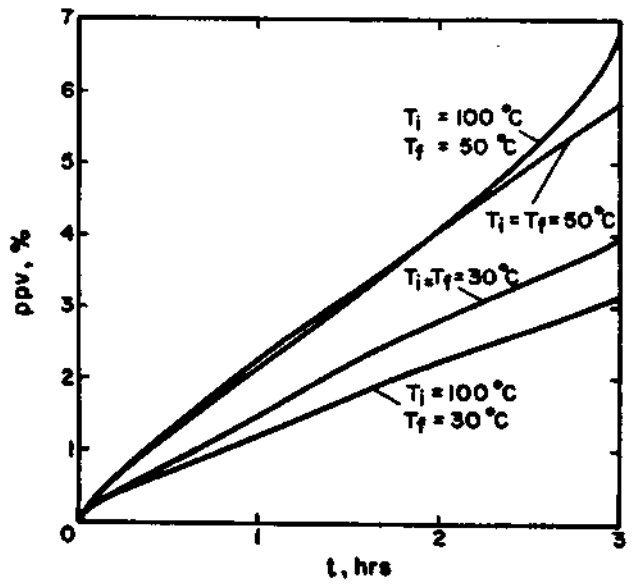


Figure 14.20: Increase in *ppv* in Isothermal and Nonisothermal Injection

and the flow rate as:

$$\text{Resistance} \sim \frac{\mu_1 x}{K k_{r1}(S_{w1})} + \frac{\mu_2 (L-x)}{K k_{r2}(S_{w2})}$$

Since $\mu_2 > \mu_1$ (Appendix 14.1) the two resistances will be comparable only when $(L-x)$ is small. Hence the initial front speed will be controlled by region 2 since its resistance is higher than that of region 1. At longer times x is large, $(L-x)$ is small, the front encounters a smaller overall resistance and as a result its speed is expected to increase. This will lead to a cross-over of the *ppv* curves as seen in Figure 14.20.

14.12 Closure

The OS algorithm falls in the family of numerical techniques that decompose a complex problem into a sequence of simpler problems. Splitting techniques have now been developed for solving the incompressible Navier-Stokes equations. The complexity of these equations is related to the non-linearity of the momentum equations and the absence of pressure in the mass-balance equation (Chapter 7). The suggested techniques include solving the Navier-Stokes equations as a sequence of Stokes problems (Briesteau *et al.*, 1985) and splitting the solution process into a series of decoupled operations on pressure and velocity (Issa, 1985). To handle complexity arising from geometry, an entire family of domain decomposition methods have now been developed (Glowinski *et al.*, 1983). The possibility of computing the set of simpler problems on a computer with a parallel architecture is a topic of current research (Topping, 1999).

References

Boberg, T.C., *Thermal Methods of Oil Recovery: An Exxon Monograph*, John Wiley and Sons, 1988.

Briesteau, M. O., Glowinski, R., Mantel, J., Periaux, J., and Perrier, P., Numerical Methods for Incompressible and Compressible Navier-Stokes Equations, in *Finite Elements in Fluids*, Vol 6, pp 1-40, John Wiley, New York.

Dawe, R.A. and Wilson, D.C., Editors, *Development in Petroleum Engineering - 1*, Elsevier, London and New York, 1985.

Duff, I.S., MA28 - A Set of Fortran Subroutines for Sparse Unsymmetric Linear Equations, AERE Harwell, UK, 1980.

Ewing, R.E., Editor, *Mathematics of Reservoir Simulation*, SIAM, Philadelphia, 1983.

Glowinski, R., Dinh, Q.V., and Periaux, J., Domain Decomposition Methods for Nonlinear Problems in Fluid Dynamics, Computer Methods in Applied Mechanics, Vol 40, pp. 27-109, 1983.

Hughes, T.J.R., Editor, *Finite Element Methods for Convection Dominated Flows*, ASME Publication, New York,1979.

Issa, R.I., Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting, J Computational Physics, Vol 62, pp. 40-65, 1985.

Lake, L.W., *Enhanced Oil Recovery*, Prentice-Hall, New Jersey, 1989.

Leonard, B.P., Universal Limiter for Transient Interpolation Modelling of the Advective Transport Equation : The ULTIMATE Conservative Difference Scheme, NASA-TM 100916, (ICOMP-88-11) NASA Lewis Research Centre, 1988.

Leonard, B.P. and Niknafs, H.S., Sharp Monotonic Resolution of Discontinuities without Clipping of Narrow Extrema, Computers and Fluids, Vol 19, No 1, pp. 141-154, 1991.

Muralidhar, K., Verghese, M., and Pillai, K.M., Application of an Operator Splitting Algorithm for Advection-Diffusion Problems, Numerical Heat Transfer, Vol 23, pp. 99-114, 1993.

Roache, P.J., *Fundamentals of Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico, 1998.

Stevenson, M.D., Kagan, M., and Pinczewski, W.V. Computational Methods in Petroleum Reservoir Simulation, Computers and Fluids, Vol 19, No 1, pp. 1-19, 1991.

Sheorey, T., Muralidhar, K. and Mukherjee, P.P., Numerical Experiments in the Simulation of Enhanced Oil Recovery, Int. J. Thermal Sciences, Vol. 40(11), pp. 981-997, 2001.

Topping, B.H.V., *Parallel and Distributed Processing for Computational Mechanics: Systems and Tools*, Saxe-Coburg Publications, Edinburgh, UK, 1999.

Yanenko, N.N. and Shokin, Y.I., *Numerical Methods in Fluid Dynamics*, Mir Publishers, Moscow, 1984.

Problems

1. Develop a computer code for the operator-splitting algorithm applied to a one dimensional transport equation and test the results presented in this chapter.
2. Apply the OS algorithm to the one dimensional Burgers' equation

$$\frac{\partial u}{\partial t} + uu_x = \frac{1}{\mathrm{Re}} u_{xx}$$

3. Develop a computer code for flow past a step in a two dimensional channel using stream function-vorticity equations. Apply the OS algorithm to the vorticity equation. Include quartic interpolation with limiters in two dimensions.
4. Suggest a method of applying the OS algorithm to incompressible flow equations in primitive variables.

Appendix 14.1

Fluid and Formation Properties

K = 132 Darcies (=1.32 × 10^{-8} m^2), k_h = 0.1661 W/mK, ε = 0.375, ξ_o = 0.03447 Pa^{-1}, ξ_w = 0.02137 Pa^{-1}

β_o = 2.28 x $10^{-4}/°C, C_o$ = 2092 W-s/kg K, C_w = 4184 W-s/kg K, $(\rho c)_R$ = 2413 kW-s/m^3K

Oil density = 90 percent of water density

Oil and Water Viscosities

T($°c$)	20	40	60	80	100
$\mu_0(Pa-s)$	4.12	2.37	0.61	0.10	0.06
$\mu_w(Pa-s)$	1.03×10^{-3}	6.6×10^{-4}	4.73×10^{-4}	3.67×10^{-4}	2.89×10^{-4}

Constitutive Relations

S_w	k_{rw}	k_{ro}	$p_{cow}(kPa)$
0.1(1)	0	1.0	28.34
0.2	0.0016	0.875	0.655
0.3	0.0081	0.735	0.496
0.4	0.0259	0.590	0.4205
0.5	0.0672	0.420	0.352
0.6	0.1000	0.210	0.283
0.7	0.1400	0.070	0.214
0.8	0.2000	0.016	0.145
0.86(2)	0.2500	0	0.0758

(1) Irreducible limit for water (2) Irreducible limit for oil

Appendix 14.2

Coefficients of Pressure Equation (14.28)

$$A_0 = -\left[\frac{\theta_e}{\rho_0(\Delta x)^2}\right]_{i,j}^{n+1}$$

$$B_0 = \left[\frac{\varepsilon_0\ S_0 - dS_w/dp_{cow}}{\Delta t} + \frac{1}{\rho_0}\left[\frac{\theta_e + \theta_w}{(\Delta x)^2} + \frac{\theta_n + \theta_S}{(\Delta y)^2}\right]\right]_{i,j}^{n+1}$$

$$C_0 = \left[\frac{dSw}{dp_{cow}}\right]_{i,j}^{n+1} \cdot \frac{1}{\Delta t}$$

$$D_0 = -\left[\frac{\theta_w}{\rho_0(\Delta x)^2}\right]_{i,j}^{n+1}$$

$$F_0 = -\left[\frac{\theta_S}{\rho_0(\Delta y)^2}\right]_{i,j}^{n+1}$$

$$G_0 = \left[\left[\frac{S_0\xi_0 - dS_w/dp_{cow}}{\Delta t}\right]^{n+1} p_0^n + \frac{1}{\Delta t}\left[\frac{dS_w}{dp_{cow}}\right]^{n+1} p_w^n\right]_{i,j} + \left[S_0^{n+1}\beta_0\frac{T^n - T^{n-1}}{\Delta t}\right]_{i,j}$$

$$\text{where } \theta = \frac{Kk_{ro}\rho_0}{\varepsilon\mu_0}$$

Backward differencing in time of the temperature term in G_0 is justified because the main effect of T on pressure distribution is through fluid properties. These do not change as rapidly with time as the primary variables p_0, p_w and S.

Coefficients of Pressure Equation (14.29)

$$A_w = -\left[\frac{\gamma_e}{\rho_w(\Delta x)^2}\right]_{i,j}^{n+1}$$

$$B_w = \left[\frac{S_w\xi_w - dS_w/dp_{cow}}{\Delta t} + \frac{1}{\rho_w}\left[\frac{\gamma_e + \gamma_w}{(\Delta x)^2} + \frac{\gamma_n + \gamma_s}{(\Delta y)^2}\right]\right]_{i,j}^{n+1}$$

$$C_w = \left[\frac{dSw}{dp_{cow}}\right]_{i,j}^{n+1} \cdot \frac{1}{\Delta t}$$

$$D_w = -\left[\frac{\gamma_w}{\rho_w(\Delta x)^2}\right]_{i,j}^{n+1}$$

$$E_w = -\left[\frac{\gamma_S}{\rho_w(\Delta y)^2}\right]_{i,j}^{n+1}$$

$$F_w = -\left[\frac{\gamma_S}{\rho_w(\Delta y)^2}\right]_{i,j}^{n+1}$$

$$G_w = \left[\left[\frac{S_w\varepsilon_w - dS_w/dp_{cow}}{\Delta t}\right]^{n+1} \cdot p_w^n + \frac{1}{\Delta t}\left[\frac{dS_w}{dp_{cow}}\right]^{n+1} p_0^n\right]_{i,j} + \left[S_w\beta_w\frac{T^n - T^{n-1}}{\Delta t}\right]_{i,j}$$

$$\text{where } \gamma = \frac{Kk_{rw}\rho_w}{\varepsilon\mu_w}$$

15
Computation of Transitional Flows

15.1 Introduction

Smooth and regular flows termed as laminar become chaotic and turbulent through a transition where the flow is neither laminar nor does it exhibit the feature of fully developed turbulence. In almost all the flow situations, this transition does not occur instantaneously over a sharp front. The spatial extent of transition is a strong function of other physical variables. In some special situations the extent of this region is quite large, for example in flows over turbine blades in the presence of high turbulence intensity and strong stabilizing pressure gradient, the flow is transitional over a large streamwise segment. It is also well known that the mass, momentum and energy transfer rates are qualitatively different in turbulent flows as compared to laminar flows. For the latter these quantities are an order of magnitude lower than the former. Hence it is necessary to locate and identify the extent of the transitional region for the accurate estimate of forces and heat transfer rates in practical problems. Moreover there is a distinct incentive to understand the physics of the transitional flow itself. In this chapter we present a few numerical methods for studying transitional flows. The entire subject area is vast; for an introductory treatment see Schlichting (1978) and Drazin and Reid (1981).

Here attention is focussed only on wall-bounded flows. The first experimental investigation related to transition phenomena is the celebrated dye experiment of Reynolds for pipe flows. If we define a Reynolds number based on centre-line velocity and the diameter, the Reynolds' experiment can be summarized by the following observations:

(i) The flow will not exhibit a sustained turbulent flow regime if the Reynolds number is less than 2000.

(ii) The Reynolds number at which the flow starts exhibiting irregular dye patterns is a strong function of turbulence intensity of the incoming flow.

It is now well-known that the so called transition Reynolds number for pipe flow could be increased to very high values by simply controlling the incoming disturbance level. These observations led Reynolds, and later Rayleigh to propose that the transition is related to the stability of the primary flow. In other words, the basic laminar flow remains laminar to all disturbances, if the Reynolds number is less than 2000 for the pipe flow. This then led to the study of stability of laminar flows by various investigators. Important early contributions were made by Tollmien (1931) and Schlichting (1932) for the boundary -layer problem. It was established that the onset of transition is associated with spatially growing waves, now named after Tollmien and Schlichting, since their results demonstrated the existence of such waves from stability calculations.

However a problem remained with the stability theory developed, namely that the predicted Tollmien-Schlichting waves were not observed experimentally for a long time. This was eventually demonstrated to exist by Schubauer and Skramstad (1947) in their classical vibrating ribbon experiment. In that experiment they reduced the general background noise level of the wind tunnel and the vibrating ribbon caused the Tollmien-Schlichting waves to be generated. Through secondary and three dimensional nonlinear growth, this caused the eventual transition to turbulence. In natural flows the actual transition process is caused by the background noise which has many characteristic frequencies acting as source of Tollmien-Schlichting waves of different wavelengths and growth rates. Therefore, free transition is difficult to simulate computationally, since it is not possible to characterise deterministically the noise occurring in engineering systems. This is one of the reasons that in the ribbon experiment the background noise was reduced while pure Tollmien-Schlichting waves were generated by externally exciting the flow deterministically.

In this chapter, attention is confined to computation of transitional flows which are triggered by an external deterministic disturbance source. Historically most of the investigations have been restricted to studying the stability aspect of flow starting with Rayleigh's work. The question that is addressed in this approach is the following: Can a given physical state when perturbed by disturbances return to its original state? If so, the state is called *stable*. If not, that particular state is *unstable*. In the stability approach, however no details of the nature of the source of disturbance is required and it is implicitly assumed to exist. Therefore the stability approach can only give a qualitative agreement with the experiments like the vibrating ribbon experiment, far away from the exciter and it cannot provide information about the phase of the wave that is generated.

It is now well accepted that the Navier-Stokes equations describe both the laminar and turbulent state of the flow. However not all the exact solutions, (if at all they are obtained), are observable in a physical experimental set-up. This is because these solutions may deviate from the nominal state due to the action of variously present disturbances. Although it is still not clearly established as to how the transition process is related to stability, the linear stability theory with all its assumptions are quite routinely used to empirically

predict transition.

Instead of following this stability analysis, a unified initial boundary value approach to both the linear and nonlinear model is presented in the following section. In particular, the simulation of the excitation problem for incompressible flows is considered. This is called the *receptivity* approach. The advantage of this approach is that it also provides all the information that one obtains through stability analysis.

15.2 Linearised Receptivity of Parallel Viscous Flows

The governing equations are those based on the conservation laws (Chapter 1) and are given by

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{15.1}$$

$$\rho \frac{du_i}{dt} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right\} \tag{15.2}$$

In the linearized analysis we are only interested in the response of the fluid dynamic system to small disturbances generally imposed in that part of the flow where the growth of boundary-layer is insignificant and hence can be neglected. This assumption is equivalent to a fully-developed (alternatively, *parallel*) flow approximation. The assumptions of small disturbances in parallel flow will enable us to simplify the governing partial differential equation to an ordinary differential equation, in the following manner. Under the small perturbation assumption, any physical variable will be written down as

$$q(\mathbf{r}, t) = Q(\mathbf{r}) + q'(\mathbf{r}, t) \tag{15.3}$$

Substituting Equation 15.3 in Equations 15.1 and 15.2 one obtains the equations for the mean flow quantities $Q(\mathbf{r})$ as well as the disturbance quantities $q'(\mathbf{r}, t)$. Here r is the position vector in the fluid region. Notice that the mean flow quantities are time independent and they are also given by the time independent Navier-Stokes equations. However for the attached wall-bounded flow further assumption of a boundary layer will enable us to obtain the mean flow from the boundary layer equations. The solution of the flow indicates that the normal velocity is small and furthermore for stations away from the leading edge of the body the streamwise component of the mean flow is more or less independent of the streamwise location, being a function of the normal coordinate only. This last assumption is a direct consequence of the parallel flow approximation. It is to be noted that excluding the entry section, Poisseulle flow and Couette flow satisfy such a requirement exactly, while for boundary layers this is quite a satisfactory assumption. Therefore we will write

$$\begin{aligned} U &= U(y) \\ V &= 0 \\ W &= W(y) \end{aligned} \tag{15.4}$$

where y is the predominant direction in which changes in velocity are observed. The equations for u', v', w' and p' are obtained as

$$\frac{\partial u'}{\partial t} + U\frac{\partial u'}{\partial x} + W\frac{\partial u'}{\partial z} + v'\frac{dU}{dy} = -\frac{1}{\rho}\frac{\partial p'}{\partial x} + \nu\nabla^2 u' \tag{15.5}$$

$$\frac{\partial v'}{\partial t} + U\frac{\partial v'}{\partial x} + W\frac{\partial v'}{\partial z} = -\frac{1}{\rho}\frac{\partial p'}{\partial y} + \nu\nabla^2 v' \tag{15.6}$$

$$\frac{\partial w'}{\partial t} + U\frac{\partial w'}{\partial x} + W\frac{\partial w'}{\partial z} + v'\frac{dW}{dy} = -\frac{1}{\rho}\frac{\partial p'}{\partial z} + \nu\nabla^2 w' \tag{15.7}$$

$$\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} + \frac{\partial w'}{\partial z} = 0 \tag{15.8}$$

The above equations are nondimensionalised further using the boundary layer thickness as the length scale, free stream velocity as the velocity scale and ρU_∞^2 as the pressure scale. We define a Reynolds number for the following discussion as

$$\mathrm{Re} = \frac{U_\infty\,\delta}{\nu}$$

We thus have four partial differential equation for the four disturbance quantities. To simplify matters further, we represent the disturbance quantities in terms of Fourier-Laplace transform as given by

$$(u', v', w', p') = \frac{1}{(2\pi)^3}\int\int\int (f, \phi, h, \pi)\, e^{i(\alpha x + \beta z - \omega t)} d\alpha d\beta d\omega \tag{15.9}$$

In Equation 15.9 the transforms are functions of y and the complex wave numbers (α, β) and circular frequency (ω) and the transforms are in general complex. The main advantage of this representation is that the disturbance equations are now ordinary differential equation with y as the independent variable. Secondly the ODEs can be solved independently of each other for each spectral variable combination. These differential equations will be solved subject to the boundary values:

$$y = 0;\ u',\ v',\ w' \quad \text{are prescribed functions of} \quad x, z \text{ and } t \tag{15.10}$$

$$\text{as} \quad y \to \infty, \quad u',\ v',\ w' \quad \text{decay to zero} \tag{15.11}$$

The boundary conditions Equation 15.10 will be given a precise form for illustration and definite solutions will be obtained later in this chapter. All the four

ODEs can be transformed into a single fourth order equation for the transform of the normal velocity component as

$$\begin{aligned}\phi^{iv} - 2\left(\alpha^2+\beta^2\right)\phi'' + \left(\alpha^2+\beta^2\right)^2\phi &= i\mathrm{Re}^{-1}\left\{(\alpha U + \beta W - \omega)\times\right.\\ \left[\phi'' - \left(\alpha^2+\beta^2\right)\phi\right] &- \left.(\alpha U'' + \beta W'')\,\phi\right\} \qquad (15.12)\end{aligned}$$

where i is the imaginary unit. Equation 15.12 is called the *Orr-Sommerfeld equation* (OSE). Next we discuss a computationally efficient method for solving Equation 15.12.

15.3 Numerical Solution of Orr-Sommerfeld Equation

Equation 15.12 is a variable coefficient, complex valued, ordinary differential equation. Though this differential equation is simple in appearance, it is one of the most difficult equation to solve numerically because of the possibility of the growth of the numerical error. It is related to the small coefficent Re^{-1} in Equation 15.12 (owing to a large Reynolds number), rendering the equation to be stiff (Chapter 3).

Of the few techniques that have proved to be successful, the compound matrix method of solving OSE will be discussed. For a survey of this method, refer Sengupta (1992) and Gaster and Sengupta (1993). For illustration, consider the Orr-Sommerfeld equation for two dimensional disturbances in a two dimensional mean flow. Then the governing differential equation is given by

$$\left[D^2 - \alpha^2 - i\alpha\mathrm{Re}\left(U - \frac{\omega}{\alpha}\right)\right]\left[D^2 - \alpha^2\right]\phi + i\alpha\mathrm{Re}U''\phi = 0 \qquad (15.13)$$

with D denoting the differential operator and U'' indicating the second derivative of streamwise mean velocity with respect to y. To solve Equation 15.13 the boundary conditions Equations 15.10 and 15.11 need to be expressed in the spectral plane. For example the free stream boundary conditions in this case will be

$$\phi \quad \text{and} \quad \phi' \quad \text{will decay as} \quad y \longrightarrow \infty. \qquad (15.14)$$

However as $y \longrightarrow \infty$, Equation 15.13 simplifies to a constant coefficient ODE and whose solution can be written in general as

$$\begin{aligned}\phi(y,\alpha,\omega) &= a_1\,e^{-\alpha y} + a_2\,e^{\alpha y} + a_3\,e^{-py} + a_4\,e^{py} \qquad (15.15)\\ \text{where} \quad p^2 &= \alpha^2 + i\,\alpha\mathrm{Re}\,(1-\omega/\alpha)\end{aligned}$$

The satisfaction of Equation 15.14 demands that a_2 and a_4 must be zero i.e.

$$\phi(y,\alpha,\omega) = a_1\,e^{-\alpha y} + a\,e^{-py}; \quad (\mathrm{Re}(\alpha) \quad \text{and} \quad \mathrm{Re}(p) \quad \text{are positive}) \qquad (15.16)$$

Note that of the four fundamental solutions only two are retained, to be called ϕ_1 and ϕ_3, and whose asymptotic behavior is given by Equation 15.16. Also to be noted is that ϕ_1 decays much more slowly than ϕ_3 as one moves away from the wall and any numerical error that may be present associated with numerical integration will tend to follow ϕ_1. Hence the linear independence of the solution vector in terms of ϕ_1 and ϕ_3 will be lost resulting in total loss of information. This is known in the literature on stability as the *parasitic error* growth problem.

The compound matrix method avoids this difficulty by working with a new set of dependent variables all of which have similar growth/decay rate with y. The essential details of the method is outlined below. To start the formulation, define a set of compound matrix variables

$$\begin{aligned} y_1 &= \phi_1\phi_3' - \phi_1'\phi_3 & y_4 &= \phi_1'\phi_3'' - \phi_1''\phi_3' \\ y_2 &= \phi_1\phi_3'' - \phi_1''\phi_3 & y_5 &= \phi_1'\phi_3'' - \phi_3'\phi_1'' \\ y_3 &= \phi_1\phi_3''' - \phi_1'''\phi_3 & y_6 &= \phi_1'\phi_3''' - \phi_3''\phi_1''' \end{aligned} \tag{15.17}$$

Using the above definitions and Equation 15.13, we can write down the governing differential equations as

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 + y_4 \\ y_3' &= y_5 + b_1 y_2 \\ y_4' &= y_5 \\ y5' &= b_2 y_1 + b_1 y_4 + y_6 \\ y_6' &= b_2 y_2 \end{aligned} \tag{15.18}$$

where

$$\begin{aligned} b_1 &= 2\alpha^2 + i\alpha \mathrm{Re}(U - \omega/\alpha) \\ b_2 &= \alpha^4 + i\alpha^3 \mathrm{Re}(U - \omega/\alpha) + i\alpha \mathrm{Re} U'' \end{aligned}$$

Equation 15.18 has six first order ordinary differential equation to be solved. As the nature of ϕ is known at the free stream, this information can be used to generate the initial condition for the y_i's for integrating from the free stream to the wall. These initial conditions are

$$\begin{aligned} y_1 &= 1 \\ y_2 &= -(\alpha + p) \\ y_3 &= \alpha^2 + \alpha p + p^2 \\ y_4 &= \alpha p \\ y_5 &= \alpha p\,(\alpha + p) \\ y_6 &= \alpha^2 p^2. \end{aligned} \tag{15.19}$$

Now that Equations 15.18 do not suffer from parasitic error growth, any good numerical technique for solving a set of first order ODEs can be used. The fourth order Runge-Kutta method is recommended (Chapter 3).

Having obtained the y_i's, one proceeds to obtain ϕ for the chosen value of α and ω. It is clear that ϕ is a linear combination of ϕ_1 and ϕ_3

$$\phi = p\phi_1 + q\phi_3 \tag{15.20}$$

Eliminating p and q between Equations 15.17 and 15.20 one obtains

$$y_1\phi'' - y_2\phi' + y_4\phi = 0 \tag{15.21}$$

$$y_1\phi''' - y_3\phi' + y_5\phi = 0 \tag{15.22}$$

$$y_2\phi''' - y_3\phi'' + y_6\phi = 0 \tag{15.23}$$

$$y_4\phi''' - y_5\phi'' + y_6\phi' = 0 \tag{15.24}$$

As described by Gaster and Sengupta (1993), the four equations are equivalent, but the last of the above is recommended for the numerical solution.

15.3.1 Simulation of Vibrating Ribbon Experiment

For this case the two dimensional disturbance field in a two dimensional mean flow has been computed when the boundary condition Equation 15.10 takes the specific form in terms of the disturbance stream function as:

$$\begin{aligned} u' &= 0 \quad \text{and} \\ \psi(x,0,t) &= H(t)\,\delta(x)\exp(-i\omega_0 t) \end{aligned} \tag{15.25}$$

We can represent the disturbance stream function in terms of its Fourier-Laplace transform as

$$\psi(x,y,t) = \frac{1}{(2\pi)^2}\int\int_{Br} \phi(y,\alpha,\omega)\, e^{i(\alpha x-\omega t)}\, d\alpha d\omega \tag{15.26}$$

Here Br is the Bromwich contour arising from inverse Laplace Transform. It is easy to show that the Fourier-Laplace transform satisfies the Orr-Sommerfeld equation as given by Equation 15.13.

In terms of the Fourier transform ϕ, the conditions given by Equation 15.25 can be written as

$$\phi'(0,\alpha,\omega) = 0 \tag{15.27}$$

$$\phi(0,\alpha,\omega) = BC_w \tag{15.28}$$

$$\tag{15.29}$$

where

$$BC_w = \frac{1}{\{i(\omega_0-\omega)\}} \tag{15.30}$$

The free stream conditions for ϕ are

$$\phi(\alpha, \omega, y \to \infty) \longrightarrow 0 \tag{15.31}$$

$$\phi(\alpha, \omega, y \to \infty) \longrightarrow 0 \tag{15.32}$$

Using Equation 15.29 and Equation 15.30 and indicating quantities evaluated at the wall by an additional subscript 0, one can write the disturbance stream function as

$$\psi(x, y, t) = \frac{-1}{(2\pi)^3} \int \int_{Br} \left[\frac{\phi_1(y, \alpha, \omega)\ \phi'_{30} - \phi_3(y, \alpha, \omega)\ \phi'_{10}}{(\phi_{10}\phi'_{30} - \phi_{30}\phi'_{10})\ (\omega - \omega_0)} \right] ie^{i(\alpha x - \omega t)} d\alpha d\omega \tag{15.33}$$

15.4 Results And Discussion

Computational results for the case where Re = 1000 and $\omega_0 = 0.1$ are presented first. For the evaluation of the Fourier-Laplace integral of Equation 15.33 we find out the integrand along the Bromwich contour by solving the governing Orr-Sommerfeld equation (15.13) using the compound matrix method. The Bromwich contour is the appropriate integration contour in both the α and ω planes which are parallel to or along the real axes in both the planes. The considerations for the choice of the contour are the *causality* principle of the disturbance in time and that the wave must propagate in the *proper* direction as dictated by the eigenvalues of the problem in the α plane. The reader is referred to Van der Pol and Bremmer (1959) for a discussion on the topic.

In the ω plane, a range of values from $(-2., 0.02)$ to $(2., 0.02)$ have been used with 256 equidistant points. In the α plane a range of values from $(-20., -.02)$ to $(20., -.02)$ with 4096 points have been used. It is necessary to use a window with the ϕ obtained since a finite range of values for α and ω is employed. For the chosen Reynolds number and ω, it is easy to show that the fundamental eigenvalue is at $\alpha^* = (0.279827, -0.00728721)$. The eigenvalues can also be obtained by the compound matrix method as discussed before. The only change that is necessary is for the wall boundary condition, which will be homogeneous. Thus one has to iterate on the value of α till the compound matrix variable y_1 is zero at the wall. This is the same expression as given in the denominator of Equation 15.33.

The Fourier-Laplace transform ϕ, as a function of ω and α shows a peaky behavior in the near vicinity of the eigenvalues of different modes and this is clearly shown in Figures 15.1 and 15.2, and Figures 15.3 and 15.4. Figures 15.1 and 15.2 show the variation of real and imaginary part of ϕ as one moves along the contour in the ω plane for a fixed α while Figures 15.3 and 15.4 show similar variations in the α-plane for a fixed ω value.

Since we have taken 1200 points in the y direction, the present results are equivalent to computing in a $(4096 \times 1200 \times 256)$ grid in the x-y-t plane. The

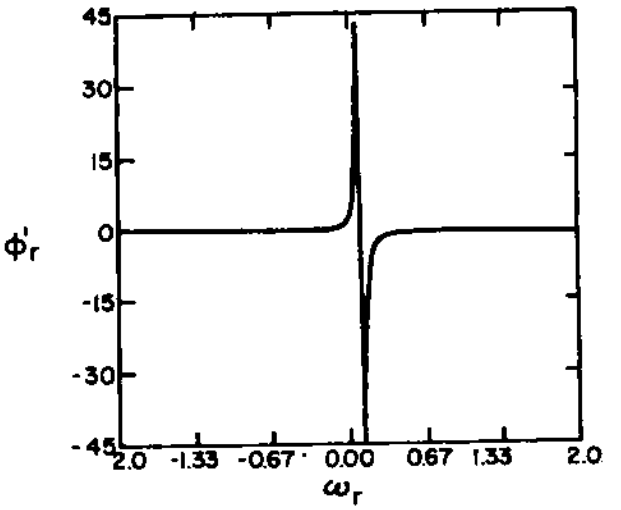


Figure 15.1: Variation of ϕ'_r along the path B for $\alpha = (0.289, -0.02)$

possibility of high resolution in space and time drections by the spectral approach is described in Chapter 12. The ϕ values as obtained by solving the Orr-Sommerfeld equation are reconstructed by performing the double inverse Fourier-Laplace transform using FFT for the relation Equation 15.33. The stream function value for a particular height over the plate is shown in Figure 15.5. A noteworthy feature of the figure is the behavior of the function in the immediate vicinity of the exciter for all times, which actually fixes the phase of the Tollmien- Schlichting waves that are seen downstream. The second advantage of the present procedure is that the transient is also obtained as a part of the solution. In a similar effort, Gaster and Sengupta (1993) have obtained the time periodic solution of the same problem. The time periodic problem refers to the Orr-Sommerfeld equation but with the altered boundary conditions. Instead of Equation 15.25 and Equation 15.29 in the physical and spectral plane respectively, the new conditions are given by

$$\psi\,(x, 0, t) = \delta(x) \exp\,(-i\,\omega_0\,t) \tag{15.34}$$

and

$$\phi\,(\alpha, \omega_0, 0) = 2\pi\,\delta\,(\omega - \omega_0) \tag{15.35}$$

The results of the full time dependent problem are compared with the time periodic solution in Figure 15.6. In this figure the streamwise component of the disturbance velocity is shown as a function of x for a given y coordinate. The figure for the time dependent simulation is shown for the maximum computation time. The figure shows good match of the amplitude, where they can be compared, but for obvious reasons there is a phase mismatch due to the effect of the transient.

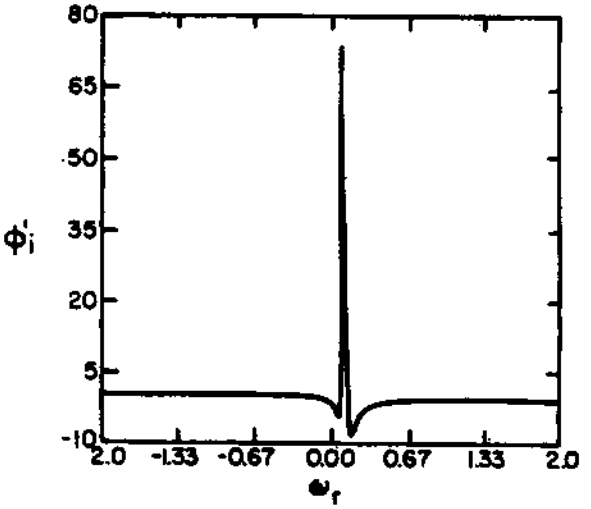


Figure 15.2: Variation of ϕ_i' along the path B for $\alpha = (0.289, -0.02)$

15.5 Nonlinear Nonparallel Receptivity problem

The results as obtained in Section 15.4, for the linearised problem are quite accurate. However they have severe limitations because they are associated with linearised analysis. A locally parallel flow approximation has been made and thereby the developed procedure cannot be used if the exciter is located in the growing part of the boundary layer. It is to be noted that till date the results of the Schubauer and Skramstad for small Reynolds numbers are not computed correctly not only by the linearised parallel flow model, but also by various nonparallel theories (see Sengupta *et al.*, 1994, 1997). The results from linear model would indicate continuous growth for the unstable cases in the downstream direction. However it is known that the actual transition process involves the linear growth mechanism followed by three dimensionality and nonlinearity. Therefore one would like to simulate the flow by a numerical technique which will take care of both the nonlinearity and nonparallelism of the flow. We restrict our discussion to two dimensional flows, noting that the three dimensional calculations would be preferred for predicting correctly the intricate details of transition.

For the present purpose, methods for solving directly the Navier-Stokes equation are indicated. These methods are classified as Direct Numerical Simulation (DNS). The earlier DNS simulations were limited to temporal formulation, in which a spatially periodic computational domain convects with the disturbances and the temporal evolution were computed and the temporal properties were converted to spatial evolution quantities by some approximate relations. Subsequently progress in spatial DNS has been made by, among others, Fasel and Konzelmann (1990), Fasel, Rist and Konzelmann (1990) and Spalart (1989). The reader is also referred to the review by Kleiser and Zang (1991).

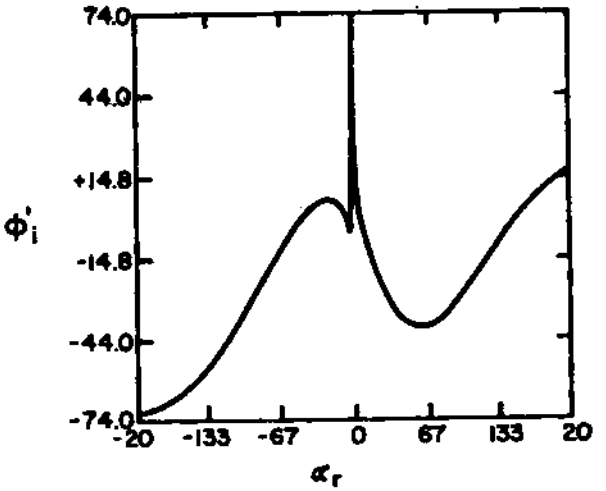


Figure 15.3: Variation of ϕ_i' along the path A for $\omega = (0.1019, 0.02)$

Although the spatial DNS simulation of transitional flows have yielded excellent qualitative comparison with linear and secondary stability theory, the author is not aware of any computation which has tried to simulate the vibrating ribbon experiment of Schubauer and Skramstad. The major problem areas of spatial DNS computations are realistic specification of inflow and outflow conditions and mainly the high demand on computational resources.

To circumvent the last problem, attention is limited to two dimensional flows, so that we can work with the stream function-vorticity formulation. This reduces the number of unknowns in two dimensions from three to two. The governing equations are then the stream function equation (SFE) and the vorticity transport equation (VTE) given by

$$\nabla^2 \psi = -\omega \tag{15.36}$$

$$\frac{d\omega}{dt} = \frac{1}{\mathrm{Re}} \nabla^2 \omega \tag{15.37}$$

The boundary conditions are given in Equation 15.25. In this formulation there is an additional need to compute the boundary vorticity at each time step by evaluating it using Equation 15.36 at $y = 0$ using the no-slip boundary condition and the conditions given by Equation 15.25. Fasel and Konzelmann (1990) have used time periodic blowing and suction conditions for specifying the excitation, except that the streamwise length of the excitation strip was one wavelength of the excited Tollmien-Schlichting waves. In terms of boundary layer thickness, this is about 5 to 6 units and thus the conditions prescribed are not identical to the Schubauer and Skramstad experiments. This was necessary for stable numerical calculations. An alternative form of the disturbance forcing is used in many other studies, where the disturbance is introduced by prescribing

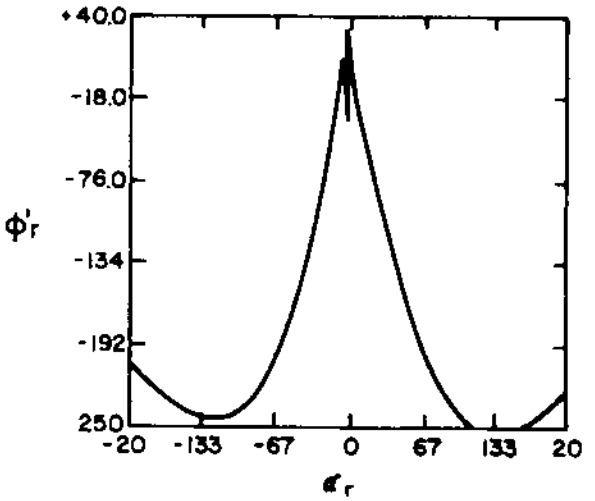


Figure 15.4: Variation of ϕ'_r along the path A for $\omega = (0.1019, 0.02)$

time-periodic function at the inflow or the free-stream boundary. These type of forcing also give qualitative understanding of the transition process.

15.6 Numerical Methods

The stream function equation is a Poisson equation and can be solved by an iterative technique like ADI method or multigrid method for solving the difference equations obtained by discretizing Equation 15.36. This author has used ADI for the difference equations obtained by using second order accurate central differencing, due to Peaceman and Rachford with acceleration parameters as given by Wachspress (Ames, 1978). However it is essential to treat the nonlinear terms of the vorticity transport equation by using higher order accurate schemes. Either fourth order central differencing or third order upwinding scheme is recommended. The third order upwinding scheme is given by

$$\left(f\frac{\partial u}{\partial x}\right)_i = \begin{cases} f_i\left(u_{i-2} - 2u_{i-1} + 9u_i - 10u_{i+1} + 2u_{i-2}\right)/6\Delta x & \text{for } f \geq 0 \\ f_i\left(-2u_{i-2} + 10u_{i+1} - 9u_i + 2u_{i-1} - u_{i-2}\right)/6\Delta x & \text{for } f < 0 \end{cases} \tag{15.38}$$

It is to be noted that if we treat function f as constant, then in the Fourier spectral representation both of these schemes have the same real part indicating identical resolution of the scales. However the third order upwind differencing has also an imaginary part which represents a small amount of numerical dissipation proportional to the fourth power of grid spacing. The third order upwinding is numerically stabilizing at high Reynolds number and is therefore recommended.

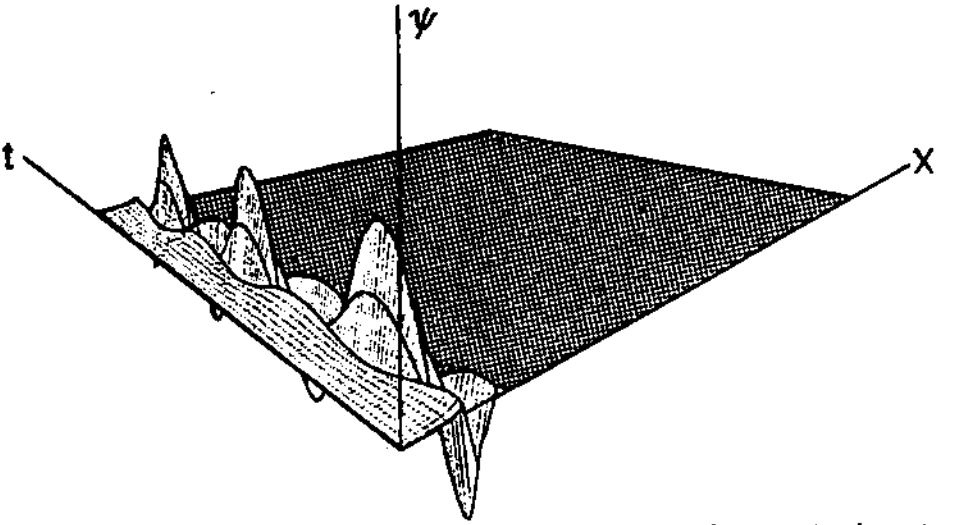


Figure 15.5: Three dimensional plot of disturbance streamfunction in the x-t plane for $y/\delta^* = 0.2789$

For this type of simulation where waves propagate downstream, it is extremely important to specify the outflow boundary conditions, such that the waves are not reflected from this boundary. This has been effectively implemented by Schwarz *et al.* (1992). In this procedure, an additional buffer zone is introduced after the outflow boundary and in this zone the oncoming disturbances are smoothly attenuated such that the disturbances reach a zero value at the downstream end of this buffer zone. This is now known as the artificial relaminarization method.

15.7 Closure

In this chapter the linearised and DNS method for the transitional flows, mainly boundary layer flows have been discussed. Undoubtedly the linearized model is fast and produces results which can be effectively interpreted for the early stages of transitional flow. Also it produces the local solution in the vicinity of the exciter, which are not only very accurate but may also prove very valuable for studying *by-pass transition* associated with transition process due to high level of turbulence intensity in the oncoming flow. Results produced by this method are of very high accuracy because of the inherent advantages of the spectral method that has been used. The equivalent number of points taken for this exercise is more than 1.25 billion points!

DNS becomes necessary for simulating flows to include the nonparallel and nonlinear effects in a direct fashion. Spatial DNS methods using finite differences have been discussed and its advantages are due to high efficiency. Although this method is restricted by the power of present-day computers, avail-

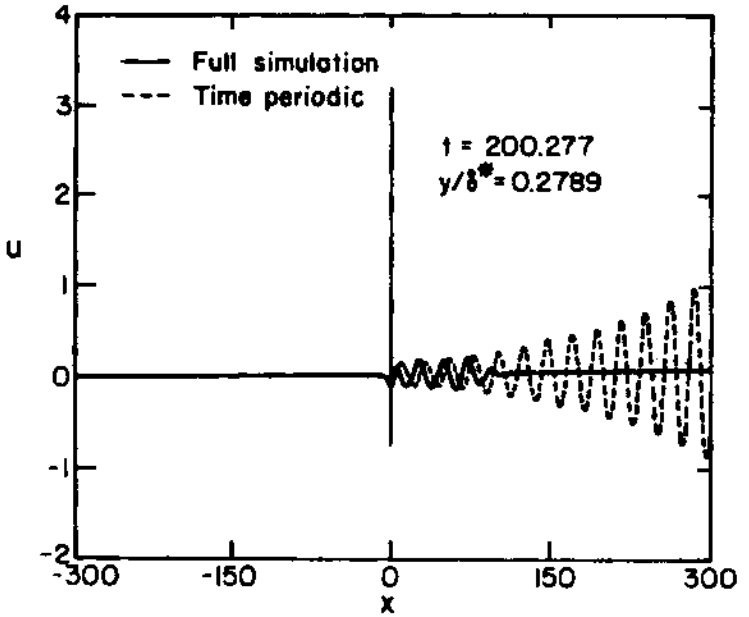


Figure 15.6: Comparison of solutions as function of streamwise distance obtained by full simulation and time-periodic assumption

able results give excellent qualitative agreement with other methods. Among the other methods, the Parabolised Stability Equation (PSE) method deserves mention (Bertolotti, 1991). However a word of caution is also mandatory here. It was seen that linearised receptivity analysis produces results which show a marginal upstream penetration of the excitation (Figures 15.5 and 15.6). Thus strictly speaking the PSE method will be in error in general and also where the local component of the solution will be dominant, as in case of by-pass transition.

Higher order finite difference methods have been used to study the three dimensional problem as well. In these methods, the information from experiments have been used to minimize the computational load. It is well known that the two dimensional wave evolution is followed by a three dimensional stage where the flow is distinctly periodic in the spanwise direction. Hence computations assume a spanwise periodicity and variables are expressed in terms of a Fourier series in the spanwise direction. These methods either use a pressure-velocity formulation or velocity-vorticity formulation and in either case they tend to be computationally intensive.

References

Ames, W. F., *Numerical Methods for Partial Differential Equations*, Academic Press, 1978.

Bertolotti, F. P., Linear and Nonlinear Stability of Boundary Layers With Stream-

wise Varying Properties, Ph.D. Thesis, Ohio State University, 1991.

Drazin, P. G. and Reid, W. H., *Hydrodynamic Stability*, Cambridge University Press, 1981.

Fasel, H. and Konzelmann, U., Non-Parallel Stability of a Flat Plate Boundary Layer Using the Complete Navier-Stokes Equations, J. Fluid Mech., Vol 221, pp 311-347, 1990.

Fasel, H., Rist, U. and Konzelmann, U., Numerical Investigation of the Three-Dimensional Development in Boundary Layer Transition, AIAA J., Vol. 28, pp 29-37, 1990.

Gaster, M. and Sengupta, T. K., The Generation of Disturbances in a Boundary Layer by Wall perturbations in "Instabilites and Turbulence in Engineering Flows", (Eds. D. Ashpis, T. B. Gatski and R. Hirsh), pp 31-49, Kluwer Academic Publishers, 1993.

Kleiser, L. and Zang, T.A., Numerical Simulation of Transition in Wall-Bounded Shear Flows, Annual Review of Fluid Mechanics, Vol 23, (Ed. J. L. Lumley, M. Van Dyke and H. L. Reed) Annual Reviews Inc., pp 495-537, 1991.

Schlichting, H., Uber die Entstehung der Turbulenz in Einem Rotierenden Zylinder, Math-Phys Kl., No 2, pp 160-198, 1932.

Schlichting, H., *Boundary Layer Theory*, Seventh edition, McGraw-Hill, New York, 1979; also see 8th revised and enlarged edition by Schlichting, H. and Gersten, K., Springer, Berlin, 2000..

Schubauer, G. B. and Skramstad, H. K., Laminar Boundary-Layer Oscillations and Stability of Laminar Flow, Journal of Aeronautical Sciences, Vol 14, No 2, pp 69-78, 1947.

Schwarz, V., Bestek, H., and Fasel, H., Numerical Investigation of Instability Waves in The Wake of An Axisymmetric Bluff Body, in the Proc. of IUTAM Symp. on Bluff-Body Wakes, Dynamics and Instabilities, (Eds. H. Eckelmann *et al.)*, Springer-Verlag, Goettingen, Germany, Sept 7-11, (1992).

Sengupta, T. K., Solution of Orr-Sommerfeld Equation for High Wave numbers, Comp. and Fluids, Vol 21(2), pp 301-303, 1992.

Sengupta, T.K., Ballav, M., and Nijhawan, S., Generation of Tollmien-Schlicting waves by Harmonic Excitation, Physics of Fluids, Vol. 6(3), pp 1213-1222, 1994.

Sengupta, T.K., Nair, M.T., and Rana, V., Boundary-layer Excited by Low Frequency Disturbances - Klebanoff Mode, J. Fluids and Structures, Vol. 11, pp 845-853, 1997.

Spalart, P., Direct Numerical Study of Leading-Edge Contamination, Fluid Dynamics of Three-Dimensional Turbulent Shear Flows and Transition, AGARD-CP-438, pp 5.1-5.13, 1989.

Tollmien, W., The Production Of Turbulence, NACA TM 609, 1931.

Van der Pol, B. and Bremmer, H., *Operational Calculus Based on Two-sided Laplace Integrals,* Cambridge University Press, 1959.

Index

Adaptive mesh generation, 529
ADI, 195, 550
Advection-diffusion equation, 12, 315, 539
Aerodynamic analysis, 367
Algebraic grid generation, 496
Artificial viscosity, 158, 372, 416
Automatic grid generation
    -see Grid generation

Bernoulli's equation, 18
Body-fitting coordinates, 489
Body-fitting grids, 485
Boundary conditions, 48
    Dirichlet conditions, 49
    Neumann conditions, 49
Boundary-layer theory, 18, 203
Boussinesq approximation, 21
Buoyancy-driven convection, 21
Burgers equation, 154

Cartesian coordinates, 4
Cell Peclet number, 252
CFL condition, 153, 240
Cholesky decomposition, 80
Classification of PDE, 47
Collocation method, 282
Compound matrix method, 578
Compressible flow, 23, 367, 403
Conduction of heat, 170
    Cartesian, 170
    cylindrical, 188
    spherical, 190
    steady state, 170, 286
    transient, 192, 295
Conjugate gradient method, 85
    preconditioning, 88, 93
    incomplete PCG, 93
Conservative property, 155
Conservative body forces, 15
Continuity equation, 4
Contravariant velocity components, 53, 495
Control volume, 186, 262
Convection equation, 169, 197, 427
Convection scheme, 450
Courant number, 150
Crank-Nicolson method, 189, 313
Crout's algorithm, 80
Curvilinear coordinates, 214
Cylindrical coordinates, 4, 188, 219

Dean number, 27
Determinant, 101
Diagonal dominance, 75
Differencing
    backward, 138
    central, 138
    forward, 138
Dirichlet conditions, 49
Dissipation, 42
Drag, 223, 349

Eigenvalues, 89, 423
Elliptic equation, 47
Energy equation, 11, 244, 404

Thermal energy equation, 11
Errors, 145
discretization, 146, 217
round-off, 146
truncation, 146
Euler equation, 15, 422
Explicit method, 113, 143, 305
Exponential scheme, 256

Falkner-Skan equation, 125
False diffusion,
-see artifical viscosity
Field grid, 375
Finite difference method, 135, 170
consistency, 142
stability analysis, 145
Finite element method, 275
assembly, 296
boundary conditions, 296, 298
Finite volume method, 438, 445
Fourier transforms, 64
discrete, 439
FFT, 66
Fourier-Laplace transform, 576
Fronts, 540
Full potential equation, 26, 369, 407
transonic, 408, 414

Galerkin method, 281
Gaussian elimination, 76
Gaussian quadrature, 65, 436
Gauss-Chebyshev, 65
Gauss-Laguerre, 66
Gauss-Jordon method, 78
Gauss-Legendre quadrature, 65
Gauss-Seidel method, 74, 186
Grid generation, 483
algebraic, 497
classification, 484
conformal mapping, 502
Delaunay triangulation, 525
elliptic, 508
Hyperbolic, 518
structured, 496, 507
C-type, 486
H-type, 486
O-type, 486
unstructured, 521
Generalized coordinates, 51, 491
Gram-Schmidt orthonormalization, 106
Green's function, 335
Green's third identity, 369
Grid refinement, 514

Harmonic average, 181, 563
Hyperbolic equation, 47
Hyperbolic grid generation, 518

Image nodes, 176
Implicit method, 114, 121, 143, 305
Incompressible flow, 4-5, 318
Influence coefficients, 342, 378
Initial conditions, 48
Initial value problem (IVP), 108
Integral equation, 369
Integral equation method, 372
Interpolation, 284
Hermitian, 285
Lagrangian, 284
Iterations, 67, 74, 87, 380

Jakobian, 53
Jameson's rotated difference scheme, 414

Karman-Pohlhausen method, 350
k-ϵ model, 41-42
Turbulent kinetic energy, 41
Kutta condition, 360

Lagrange polynomials, 285
Laminar flow, 13
Laplace equation, 139
Least squares, 281
Lift, 355
Logarithmic law, 37-38
LU decomposition, 79

MAC algorithm, 229, 231
Mach number, 24, 369, 408
Matrix inversion, 74
    iterative, 74
    direct, 76
Momentum interpolation, 455
Monotonicity, 428
Murman-Cole switching, 411, 412

Navier-Stokes equations, 8
    expanded form, 8, 9
Neumann conditions, 49
    von Neumann stability, 148
Newton-Raphson scheme, 67, 301
Numerical integration, 57, 293
    Gaussian qudrature, 60
    Simpson's rule, 58
    trapezoidal rule, 58

Oil recovery, 558
    phase equations, 560
    percentage pore volume, 562
    properties, 570
Operator-splitting, 539
    grid-orientation effect, 550
    interpolation errors, 553
Ordinary differrential equations, 107
    stiff, 121
Orr-Sommerfeld equations, 26, 31, 577

Panel method, 333
    source, 333, 337
    vortex, 357
    three dimensional, 367
Parabolic equation, 47
Parasitic error, 577
Partial differential equations, 47
Pascal triangle, 289
Peclet number, 14, 316
Pivoting, 79
Poisson equation for pressure, 220
Potential flow, 16
    theory, 16
Power-law scheme, 258
Prandtl number, 19
    turbulent, 38
Predictor-corrector method, 112
Pressure correction equation, 264
Pressure equation, 15

Quadtree division, 527
QUICK scheme, 198

Rankine-Hugoniot relation, 25, 430
Rarefaction waves, 431
Rate-of-strain tensor, 6
Rayleigh equation, 32
Rayleigh number, 23
Rayleigh-Ritz method, 278
Receptivity analysis, 575
Reynolds number, 14
    critical, 30
Reynolds transport theorem, 2
Reynolds stresses, 34, 325
    algebraic Reynolds stress model, 45
Reynolds transport theorem, 2
Riemann invariants, 431
Riemann problem, 429
Roots of a polynomial, 67
Runge-Kutta method, 110
    Adaptive stepping, 118
    BVP, 124

first order equation, 109
fourth order scheme, 110
higher order equation, 117

Scarborough condition, 75
Schwarz-Christoffel method, 504
Semi-explicit method, 453
Shocks, 407
shock relations, 430
Shooting method, 122
SIMPLE, 229, 248, 267
algorithm, 267
Source term, 183
linearization, 184
Spectral condition number, 89
Spectral radius, 89
Spherical coordinates, 4, 186, 221
Square cylinder, 472
Stability
analysis, 145
considerations, 233, 412
conditional, 113
Steger and Warming scheme, 440
Staggered grid, 230
Stokes equations, 15
Stream function, 16
-vorticity formulation, 16, 209
Stress tensor, 5
Structured mesh generation, 485, 496
Subsonic flow, 24
Supersonic flow, 24
Surface panels, 375

TDMA, 82, 265, 542
Transfinite interpolation, 499
Transformed coordinates, 405, 519
-see body-fitted coordinates
Transition, 27, 573
Transonic flow, 24, 385, 414
unsteady, 402
Transportive property, 158
TSP equation, 409

Turbulent flow, 13, 32
Turbulent heat flux, 34

Universal limiters, 555
Unstructured mesh generation, 521
Upwinding, 160, 194, 216, 234, 255
-also see false diffusion, artificial viscosity

Van Leer's scheme, 442
Variable thermal conductivity, 180, 312
Vibration ribbon experiment, 579
Vornoi polygons, 526
Vorticity, 16
transport equation, 16, 210

Wake, 362, 370
Wedge flow, 20
Weighted residual formulation, 276

አዳማ ዩኒቨርሲቲ
Adama University
ቤተመጻሕፍት
Library