FINITE ELEMENT METHOD AND APPLICATIONS

YONG HONG WU
Curtin University of Technology · Australia

BENCHAWAN WIWATANAPATAPHEE
Mahidol University · Thailand

Misterkopy Publishing Company
Bangkok

Published by
Misterkopy Publishing Company
Bangkok, Thailand

National Library of Thailand Cataloging in Publication Data

Benchawan Wiwatanapataphee
Finite Element Method and Applications
1. Engineering mathematics
2. Finite element method
I. Yong Hong Wu
II. Title
004.0151
ISBN: 974-94652-8-8

Finite Element Method and Applications

Printed in Thailand by Misterkopy Publishing Company

ISBN : 974-94652-8-8

Preface

With the development of computer technology, the finite element method has become more and more important. At present, it has become one of the most effective methods for solving a variety of problems arising from engineering design, industrial process control and scientific research. The aim of this book is twofold, to systematically introduce the finite element method for solving boundary value problems and to demonstrate its applications in engineering and medical science.

The first part of the book, including chapters 1-7, is an introduction to the finite element method. It focuses on the mathematical and computational aspects of the finite element method for solving elliptic and parabolic boundary value problems. The second part of the book, chapters 8-11, demonstrates the applications of the finite element method to the analysis of multi-phase heat transfer, electromagnetical stirring to fluid flows, and blood flows in stenotic arteries. Some of our recent research results in these application fields have also been included in the book.

The book was written based on our teaching and research experience in applied mathematical modeling using partial differential equations. Our experience has been built into the book through the design and organization of the contents and the writing of each chapter, section and paragraph. Our aim is to provide a text which is easy to learn and covers all elements essential to numerical simulation using the finite element method, and to provide a quick reference for those who use finite element method in their research and work. The book can be used as a text for a one-semester course at postgraduate level.

Yong Hong Wu
Benchawan Wiwatanapataphee

September, 2006

Contents

Preface **i**

1 Boundary Value Problems **3**
1.1 Classification of Differential Equations (2nd order) 4
1.2 Boundary and Initial Conditions 6
1.3 Methods of Solution 6

2 General Finite Element Formulation **11**
2.1 Classical Statements 13
2.2 Variational Statements 14
2.3 Weighted Residual Techniques 17
2.3.1 Point Collocation 19
2.3.2 Subdomain Collocation 19
2.3.3 The Galerkin Method 20
2.4 Finite Element Approximation 22
2.5 Classes of Admissible Functions 26
2.5.1 Spaces of continuous functions $C^m(\Omega)$ 27
2.5.2 Banach spaces 30
2.5.3 Hilbert spaces 31
2.5.4 Distributions 32
2.5.5 The p-integrable spaces $L^p(\Omega)$ 33
2.5.6 The Sobolev spaces $W^{k,p}(\Omega$ 36

3 Two-Point Boundary Value Problems **41**
3.1 Finite Element Formulation 41
3.2 Finite Element Approximation 44

4 Elliptic Boundary Value Problems **55**
4.1 Introduction . . . 55
4.2 Variational Statement . . . 56
4.3 The Galerkin Approximation . . . 58
4.4 The Finite Element Interpolation . . . 59
4.4.1 Triangular Elements . . . 60
4.4.2 Rectangular Elements . . . 63
4.4.3 Interpolation Error . . . 64
4.5 Finite Element Approximation . . . 65

5 Parabolic Boundary Value Problems **73**
5.1 Semi-discretization in space . . . 73
5.2 Time Differencing . . . 76

6 Element Calculations **83**
6.1 Element Transformation . . . 83
6.2 Finite Element Calculations . . . 89
6.3 Finite Element Program . . . 93

7 Solution of Linear Systems of Equations **101**
7.1 Direct Methods for Systems of Linear Equations . . . 102
7.1.1 Gaussian Elimination . . . 102
7.1.2 LU Factorization and Its Connection with Gaussian Elimination 107
7.1.3 Pivoting and Scaling . . . 108
7.1.4 Permuted LU Factorization . . . 111
7.1.5 LL^T and LDL^T Factorization Methods . . . 114
7.2 Solution of Sparse Systems of Linear Equations . . . 118
7.3 Iterative Methods for Systems of Linear Equations . . . 122
7.3.1 The Jacobi Iterative Method . . . 123
7.3.2 The Gauss-Seidel Iteration Method . . . 125
7.3.3 Convergence Conditions . . . 128
7.3.4 Error Bound and Speed of Convergence . . . 133
7.3.5 Relaxation Method . . . 136

8 Stokes Problem and Incompressible Flows **145**
8.1 Fundamental Equations for the Flow of Fluids . . . 145
8.2 Stokes Problem . . . 149
8.3 Flow of Incompressible Fluids . . . 151

9 Coupled Heat Transfer & Turbulent Flows **155**
9.1 The Continuous Casting Process 156
9.2 Heat Transfer-Turbulent Flow Model 157
9.3 Finite Element Solution 162
9.4 Numerical Investigation 165

10 Multi-Phase Flows under EM Force **171**
10.1 Steel Casting with Electromagnetic Stirring 172
10.2 Mathematical Model 173
10.3 Method of Solution 175
10.4 Numerical Investigation and Discussion 178

11 Blood Flows in Stenosed Arteries **183**
11.1 Stenosis and Cardiovascular Disease 183
11.2 Mathematical Model 187
11.3 Method of Solution 194
11.4 Numerical Results and Discussion 196

Bibliography **210**

APPENDICES **211**

List of Figures

2.1 Domain with 4 elements 5 nodes 23
2.2 Element shape functions $\phi_i(x)$. 24
2.3 Function $u(x)$ and its derivatives 28

4.1 Approximate function $u_h^e(i),\ i = 1, 2, 3$ 61
4.2 Pascal's triangle 62
4.3 A rectangular element with nine nodes 64
4.4 Computation domain Ω 68

6.1 Square elements with 4 nodes (linear element), 9 nodes (quadratic element) and 16 nodes (cubic element) 84
6.2 Element transformation T_e 84
6.3 Straight sides of $\bar{\Omega}$ map to curved sides of Ω_e 87
6.4 Mapping $\bar{\Omega}$ to Ω_1 and Ω_2 88
6.5 Finite element mesh for Example 6.2 93

9.1 The continuous steel casting process 157
9.2 Velocity and temperature profile (a) velocity vectors (m/s), (b) temperature contours (^oC) 167
9.3 Contour plot of (a) turbulent kinetic energy $K(m^2/s^2)$ and (b) dissipation rate $\varepsilon(m^2/s^3)$ 167
9.4 Temperature distribution in the computational region 168
9.5 Comparison of temperature profiles at the bottom of the mould obtained by models with turbulence effect (solid line) and with no turbulence effect (dotted line) 168
9.6 Comparison of temperature profiles in the first 3 metres below meniscus obtained by models with turbulence effect and with no turbulence effect 169

10.1 Computation domain (a = 0.1 m.) 175
10.2 Influence of external current density on (a) the magnetic flux density **B**; (b) the magnetic potential $\mathbf{A}_z$; (c) The electromagnetic force $\mathbf{F}_{em}$. . . . 180
10.3 Influence of external current density on the fluid flow and heat transfer (a) velocity field of molten steel; (b) Temperature profiles. 181
10.4 Influence of source current density on the magnitude of electromagnetic force at the horizontal section 0.055 m below the meniscus. 182
10.5 Influence of source current density on the temperature profile at the horizontal section 0.4 m below the meniscus. 182

11.1 The blood circulation in the heart 184
11.2 The cross section of an artery 188
11.3 The periodic blood pressure and flow rate waveforms of the right coronary artery oscillating within systolic and diastolic levels with cardiac period $T = 0.9s$. 192
11.4 The right coronary artery with stenosis. 197
11.5 The 3-D geometry of the 50% stenotic artery. 198
11.6 Velocity filed in the luminal channel of the 75% stenotic artery at the peak of diastole 199
11.7 Pressure distribution along a longitudinal line on the interface between the lumen region and the arterial wall during a cardiac cycle. 200
11.8 Pulsatile patterns of blood flow, pressure field and shear rate around the stenosis site in the lumen region (a) 50%-area severity (b) 75%-area severity. 201
11.9 Wall displacement during a cardiac cycle at the peak of diastole. 202
11.10 Wall shear stresses and wall shear rate along a longitudinal line on the interface between the lumen region and the arterial wall of stenotic arteries: (a-b) 50%-area severity; (c-d) 75%-area severity. 203

List of Tables

3.1 System Topology 45

7.1 Operation Count 106
7.2 Numerical Solutions by Jacobi and Gauss-Seidel Methods 126
7.3 The k^* value corresponding to different ρ 137

10.1 Parameters used in numerical simulation 178

11.1 Values of parameters used in computation 193

Chapter 1

Boundary Value Problems

Modelling of most real world problems in science and engineering usually leads to a boundary value problem (B.V.P.): a differential equation (or a set of differential equations) subject to certain initial and boundary conditions. For example, the transient temperature field in a bounded domain Ω with convection boundary $\partial\Omega$ can be modelled by

$$\rho c \frac{\partial T}{\partial t} = k\nabla^2 T + Q(x) \quad \forall \mathbf{x} \in \Omega, \;\; t \in [0, T] \tag{1.1}$$

subject to

$$k \frac{\partial T}{\partial n}(\mathbf{x}) = -h(T - T_\infty) \quad \forall \mathbf{x} \in \partial\Omega, \;\; t \in [0, T] \tag{1.2}$$

and initial condition

$$T(0, \mathbf{x}) = T_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \tag{1.3}$$

where $Q(\mathbf{x})$ is heat source and ρ, c, k and h are constants. In this chapter, we are concerned with the following topics

1. Classification of partial differential equations (P.D.Es).

2. Classification of boundary conditions (B.C.).

3. An overview of methods for solving boundary value problems (B.V.P.).

1.1 Classification of Differential Equations (2nd order)

Definition 1.1.1 A differential equation is said to be

- linear if it is a linear equation of the unknown function and its derivatives,

$$au_{xx} + bu_{yy} + cu_x + du = Q;$$

- quasi-linear if all the highest derivative terms are linear but some of the lower order derivatives are non-linear,

$$au_{xx} + bu_x^2 = f(x, y, u);$$

- non-linear if the equation is neither linear nor quasi-linear,

$$u_{xx} + 2u_{xy}^2 + bu = Q(x, y).$$

Most partial differential equations arising from real world problems are second order and thus we will focus only on second order equations. The general form of the second order quasi-linear partial differential equation is

$$au_{xx} + bu_{xy} + cu_{yy} + h(x, y, u, u_x, u_y) = 0,$$

which can be classified into three categories according to the value of $b^2 - 4ac$,

- elliptic : $b^2 - 4ac < 0$
- parabolic : $b^2 - 4ac = 0$

- hyperbolic : $b^2 - 4ac > 0$

Example 1.1

1) poisson equation

$$\nabla^2 u = \sigma$$

is elliptic $(a = c = 1, b = 0)$;

2) diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

is parabolic $(a = 1, b = c = 0)$;

3) wave equation

$$\frac{\partial^2 u}{\partial t^2} = \alpha^2 \frac{\partial^2 u}{\partial x^2}$$

is hyperbolic $(a = \alpha^2, b = 0, c = -1)$.

Remarks: If a,b and c are functions of x, y and u, the equation may change its type from one region to the other in the computation domain.

Example 1.2 The following partial differential equation

$$(1 - M^2(x, y))\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

may change its type from one sub-domain to the other. It can be classified as

1) elliptic equation if $M(x, y) < 1$;

2) parabolic equation if $M(x, y) = 1$;

3) hyperbolic equation if $M(x, y) > 1$.

1.2 Boundary and Initial Conditions

If we do not distinguish between time and space as independent variables, an initial condition can also be regarded as a boundary condition. For real world problems, usually, we know the value of the unknown function and/or its derivatives on part of the boundary $\partial\Omega$. As the solution must satisfy the boundary conditions, we have to solve the partial differential equation in Ω subject to the boundary conditions on $\partial\Omega$. Boundary conditions are usually of the following types:

- Dirichlet type (also called essential boundary condition in finite element method)
  eg. $u = \hat{u}$ on $\partial\Omega$
- Neumann type (natural boundary condition)
  eg. $\frac{\partial u}{\partial n} = \hat{\sigma}$ on $\partial\Omega$
- Robin type (mixed or general boundary condition)
  eg. $\alpha\frac{\partial u}{\partial n} + ku = f,\ \alpha \neq 0,\ k \neq 0$, on $\partial\Omega$

Boundary value problems are classified based on the type of partial differential equations and the type of boundary conditions. For example, a boundary value problem defined by an elliptic equation and a Neumann boundary condition is called a Neumann elliptic problem.

1.3 Methods of Solution

In general, a boundary value problem of an unknown function u can be written as

$$L(u) \ = \ f(\mathbf{x}) \quad \texttt{in } \Omega \tag{1.4}$$

$$B(u) = g(\mathbf{x}) \quad \texttt{on } \partial\Omega \tag{1.5}$$

where $f(\mathbf{x})$ and $g(\mathbf{x})$ are known functions, L denotes a linear or nonlinear differential operator and B is a boundary operator.

To solve a boundary value problem is to find the unknown function u that satisfies the differential equation in Ω and the boundary conditions on $\partial\Omega$. There are many alternative approaches available for solving linear and nonlinear boundary value problems, ranging from completely analytical to completely numerical. Of these, the following approaches deserve attention:

Direct Integration (yielding exact solutions)

- Separation of variables;
- similarity solutions;
- Fourier and Laplace transformations;

Approximate Solution Methods

- Perturbation, Power series, Probability schemes (Monte Carlo);
- The method of characteristics for hyperbolic equations;
- Finite difference technique;
- Ritz method;
- Boundary element method;
- Finite element method.

Remarks:

1) Only for very simple problems, it is possible to obtain an exact solution by direct integration of the differential equations.

2) The Power series method is powerful, but since the method requires generation of a coefficient for each term in the series, it is relatively tedious.

3) The perturbation method is applicable primarily when the nonlinear terms in the equation are small in relation to the linear terms.

4) The probability schemes (Monte Carlo Method) are used for obtaining a statistical estimate of a desired quantity by random sampling. These methods work best when the desired quantity is a statistical parameter and sampling is done from a selective population.

5) With the advent of high-speed computers, it appears that the three currently outstanding methods for obtaining approximate solutions of high accuracy are the finite difference method, the finite element method and the boundary element method. The finite difference method usually is only applicable to problems with simple geometry. The boundary element method is a more efficient and accurate method, which usually reduces the dimensionality of the problem by one. However, the application of the boundary element method requires a singular solution to the problem, which limits its application. The finite element method is a more general and versatile method. In principle, any problem, which can be solved by the finite difference method or the boundary element method, can also be solved by the finite element method.

In this book, we focus on the finite element methods for the solution of boundary value problems and their applications in fluid dynamic and heat transfer simulation.

Chapter 2

General Finite Element Formulation

The finite element method is a numerical technique for obtaining approximate solutions to boundary value problems. To find a solution u to a boundary value problem defined in domain Ω using the finite element method, we perform the following steps:

1. Discretize the computation domain Ω into a finite number of elements with N nodes, so that $\Omega = \cup_{e=1}^{E} \Omega_e$, and then take the values of u at these nodes as basic unknowns;

2. Transform the boundary value problem to a set of finite element equations;

3. Obtain coefficient matrices for each element;

4. Assemble each element matrix to form a global matrix;

5. Solve the global matrix equations which may be a system of algebraic equations (or ordinary differential equations) of u_i $(i = 1, N)$

The derivation of finite element equations is usually based on one of the following approaches:

- direct approach,
- variational approach,
- weighted residual approach,
- energy balance approach.

Remarks:

1) The direct approach can be used only for very simple problems (simple element shape).

2) The variational approach is a more general approach. It relies on the calculus of variations and we need to find a functional $J(u)$ corresponding to the boundary value problem such that the solution of the boundary value problem becomes : find the unknown function u such that the functional $J(u)$ is minimized. For problems in solid mechanics, the functional turns out to be the potential energy or some other physical quantities, and thus the method can be used. However, the approach is not applicable to those problems for which we do not have an associated functional (either has not been discovered or does not exist).

3) The weighted residual approach is a relative new and even more general and versatile method. It advantages the variational approach because it makes it possible to extend the finite element method to problems where no functional is available.

2.1 Classical Statements

The classical statement of a typical boundary value problem is:

Find a function $u(\mathbf{x})$ such that

$$\begin{aligned} L(u) - f(\mathbf{x}) &= 0 \quad \mathbf{x} \in \Omega, \\ B(u) - g(\mathbf{x}) &= 0 \quad \mathbf{x} \in \partial\Omega \end{aligned} \tag{2.1}$$

where Ω is the domain of the real-valued function $u(\mathbf{x})$
$\partial\Omega$ is the boundary of Ω
L and B are respectively differential and boundary operators.

Example 2.1 Two - point boundary value problem:

Find u such that

$$\begin{cases} u_{xx} = f(x) \quad x \in (a,b) \\ u(a) = A, \quad u(b) = B \end{cases}$$

Example 2.2 Two-dimension heat transfer problem:

Find T such that

$$\begin{aligned} \nabla \cdot (k \nabla T) &= f && \text{in } \Omega \\ T &= T_\infty && \text{on } \partial\Omega_1 \\ -k\frac{\partial T}{\partial \mathbf{n}} &= h(T - T_\infty) && \text{on } \partial\Omega_2. \end{aligned}$$

where $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ is the boundary of Ω.

2.2 Variational Statements

The variational statement of a boundary value problem is the integral representation of the problem. Consider the boundary value problem (2.1) with the approximate solution $u(\mathbf{x})$. We can define the local residual error r by

$$r(\mathbf{x}) \;=\; L(u(\mathbf{x})) - f(\mathbf{x}). \tag{2.2}$$

By multiplying the local residual error $r(\mathbf{x})$ by a weighting function $v(\mathbf{x})$ and integrating the weighted residual error over Ω, we obtain the total weighted residual error. Setting the total weighted residual error to zero, we obtain the following variational statement:

Find $u(\mathbf{x}) \in \tilde{H}$ such that

$$\int_\Omega v r(\mathbf{x})\, d\Omega \;=\; \int_\Omega v(L(u) - f(\mathbf{x}))\, d\Omega \;=\; 0 \quad \forall v \in H, \tag{2.3}$$

where v is known as weighting function or test function,

H is a linear space containing the set of all functions which are sufficiently well behaved that the above integral makes sense,

$\tilde{H}$ is a linear space containing the set of all trial (admissible) functions to which the solution u belongs.

The statement (2.3) is called a variational statement as the weighting function v is allowed to vary arbitrarily.

Remarks :

1) The variational statement (2.3) is equivalent to the classical statement (2.1).

2) The specification of the set $\tilde{H}$ of trial functions and the set H of test functions is an essential ingredient of an acceptable variational statement. The smoothness requirement demands that $\tilde{H}$ and H must be such chosen that $vL(u)$ is integrable over Ω for any functions u and v chosen from $\tilde{H}$ and H respectively.

3) Often by applying integration by part (or Green's formula) to the integral expression of equation (2.3), we can obtain expression containing lower-order derivatives of $u(x)$. That is, the order of derivatives of u can be reduced and (2.3) often becomes

$$\int_{\Omega} P(u,v)\, d\Omega + \int_{\partial\Omega} Q(u)v\, ds \;=\; 0.$$

The smoothness requirement on $u(x)$ is thus weakened and so the integral representation is also called the weak statement.

4) When integration by part is possible, it also offers a convenient way to introduce some of the boundary conditions.

Example 2.3 Find the variational statement of the following boundary value problem

$$u_{xx} - f \;=\; 0, \quad x \in (a,b)$$

$$u(a) = 0, \;\; u_x(b) = g.$$

Sol

From the given differential equation, the residual error function is determined by

$$r(x) \;=\; u_{xx} - f.$$

Thus, the total weighted residual error is

$$R \;=\; \int_a^b vr\, dx \;=\; \int_a^b v(u_{xx} - f)\, dx \;=\; -\int_a^b (u_x v_x + fv)\, dx + u_x v(x)|_a^b \tag{2.4}$$

Hence, the variational statement takes the following form:

Find u from a suitable class of admissible functions such that

$$\int_a^b (u_x v_x + fv)\, dx - u_x v|_a^b \;=\; 0 \qquad (2.5)$$

for all test functions in a suitable class of functions, H.

To identity the suitable class of functions, we consider the following two points:

Smoothness requirement on u and v

For the integral (2.5) to be well defined, it is sufficient to take u and v to be members of a class of functions whose derivatives of order 1 and less are square-integrable over Ω. Thus in this case,

$$\tilde{H} = H = \{v : v \text{ and } v_x \text{ are square integrable over } \Omega\}.$$

Note : a function g is said to be square -integrable if

$$\int_\Omega g^2 d\Omega < \infty.$$

Boundary restriction on u and v

As $u_x(a)$ is not given, we need to eliminate $u_x(a)v(a)$ from the variational statement. For this purpose, we choose $v(a) = 0$. In other words, we let v to be a member of the space H_0 defined by

$$H_0 \;=\; \{v : v \in H \text{ and } v(a) = 0\}.$$

Therefore, the variational statement is:

Find $u \in H_0$ such that

$$\int_a^b (u_x v_x + fv)dx - g(b)v(b) \;=\; 0 \quad \forall v \in H_0$$

where H and H_0 are as defined before.

2.3 Weighted Residual Techniques

Now we consider our general problem in variational form:

Find $u \in \tilde{H}$ such that

$$\int_\Omega P(u,v)d\Omega + \int_{\partial\Omega} Q(u)v\ ds \ = \ 0$$

$$\texttt{or} \ \int_\Omega (L(u)-f)v\ d\Omega \ = \ 0, \quad \forall v \in H. \tag{2.6}$$

In general, both u and v may belong to a large (infinite dimension) class of functions, i.e.

$$u(x) = \sum_{i=1}^{\infty} \alpha_i \phi_i(x), \quad v(x) = \sum_{i=1}^{\infty} \beta_i w_i(x) \tag{2.7}$$

where $\phi_i(x)$ and w_i denote the basis functions for $\tilde{H}$ and H.

The process of finding a solution from these broad classes without a constructive method is remote and we thus turn to find a numerical approximation. In numerical approximation, we pose the variational problem in the N-dimensional subspaces $\tilde{H}^h$ and H^h of $\tilde{H}$ and H, respectively. Thus, our problem becomes:

Find $u_N \in \tilde{H}^h$ such that

$$\int_\Omega P(u_N, v_N)\ d\Omega + \int_{\partial\Omega} Q(u_N)v_N\ ds \ = \ 0 \quad \forall v_N \in H^h. \tag{2.8}$$

Recall that the integral in (2.8) represents the total weighted residual error over Ω. We can state our method for solving the boundary value problem by

> Seek for an approximation solution u such that the total weighted residual error (with weighting function v_N) over Ω vanishes.

This method is called the weighted residual technique.

As $u_N \in \tilde{H}^h$ and $v_N \in H^h$, we have

$$u_N = \sum_{j=1}^{N} \alpha_j \phi_j(x), \quad v_N = \sum_{i=1}^{N} \beta_i w_i(x). \tag{2.9}$$

Thus, the integral equation (2.8) becomes

$$\int_\Omega P(u_N, \sum_{i=1}^{N} \beta_i w_i)\, d\Omega + \int_{\partial\Omega} Q(u_N) \sum_{i=1}^{N} \beta_i w_i\, ds \;=\; 0.$$

Assuming that $P(u, v)$ is a bilinear form of u and v, we have

$$\sum_{i=1}^{N} \left[\int_\Omega P(u_N, w_i)\, d\Omega + \int_{\partial\Omega} Q(u_N) w_i \right] \beta_i\, ds \;=\; 0. \tag{2.10}$$

Because β_i are arbitrary, (2.10) represents N equations to be satisfied by the α_i defining u_N rather than the single equation it may appear to be. To see this, consider the natural choice for the parameters β_i.

For $\beta_1 = 1,\ \beta_i = 0 \quad \forall i \neq 1$, (2.10) becomes

$$\int_\Omega P(u_N, w_1)\, d\Omega + \int_{\partial\Omega} Q(u_N) w_1\, ds \;=\; 0.$$

For $\beta_2 = 1,\ \beta_i = 0 \quad \forall i \neq 2$, (2.10) becomes

$$\int_\Omega P(u_N, w_2)\, d\Omega + \int_{\partial\Omega} Q(u_N) w_2\, ds \;=\; 0.$$

Continuing in this way, we can obtain N equations. Thus, the problem now becomes:

Find $\alpha = (\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_N) \in \mathcal{R}^N$ such that

$$\int_\Omega P(\sum_{j=1}^{N} \alpha_j \phi_j,\ w_i)\, d\Omega + \int_{\partial\Omega} Q(\sum_{j=1}^{N} \alpha_j \phi_j) w_i\, ds \;=\; 0 \quad (i = 1, 2, \ldots, N). \tag{2.11}$$

Remarks :

1) The form of error distribution over Ω depends on our choice for the weighting function $w_i(x)$. Once w_i are specified, equations (2.11) represent a system of

N equations (either algebraic equations or ordinary differential equations) which can then be solved to find α_i and then, the approximate representation of the unknown variable field $u(x)$ can be determined via (2.9).

2) We have a variety of weighted residual techniques because of the broad choice of weighting functions that we can use.

2.3.1 Point Collocation

In this technique, the weighting function is taken to be

$$w_i \;=\; \delta(\mathbf{x}-\mathbf{x}_i), \quad \mathbf{x},\mathbf{x}_i \in \Omega$$

where δ is the Direc delta function having the following properties

$$\delta(\mathbf{x}-\mathbf{x}_i) = \begin{cases} 0 & \mathbf{x} \neq \mathbf{x}_i \\ \infty & \mathbf{x} = \mathbf{x}_i \end{cases}$$

$$\int_\Omega G(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}_i)\, d\Omega \;=\; G(\mathbf{x}_i).$$

Thus, the total weighted residual error is

$$\int_\Omega r(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}_i)\, d\Omega \;=\; r(\mathbf{x}_i). \tag{2.12}$$

Hence, setting the total weighted residual error to zero for $w_i = \delta(\mathbf{x}-\mathbf{x}_i)$ is equivalent to making the local residual error to zero at the point $\mathbf{x}_i$. This means that in the point collocation technique the local residual error is forced to be zero at a number of chosen points $\mathbf{x}_i$ $(i = 1, N)$.

2.3.2 Subdomain Collocation

In this technique, the weighting function is chosen to be

$$w_i = \begin{cases} 1 & \mathbf{x} \in \triangle\Omega_i \\ 0 & \text{otherwise} \end{cases}$$

Thus, the total weighted residual error for corresponding to the above weighting function is

$$\int_\Omega r(\mathbf{x}) w_i \, d\Omega = \int_{\triangle\Omega_i} r(\mathbf{x}) \, d\Omega.$$

Hence, setting the total weighted residual error corresponding to the above weighting function is to make the integrated error over the element $\triangle\Omega_i$ to be zero. This means that in the subdomain collocation technique, the integrated error over each of the N subregions of the domain is forced to be zero.

2.3.3 The Galerkin Method

The error distribution principle most often used to derive finite element equations is known as the *Galerkin* criterion or Galerkin's method. According to the *Bubnov-Galerkin* (or Galerkin) method, the weighting functions are chosen to be the same as the basis functions used to represent u, that is

$$w_i(x) = \phi_i(x), \quad i = 1, 2, \ldots, N.$$

When $w_i \neq \phi_i$, the approach is called the *Petrov-Galerkin* method. Thus, to solve a boundary value problem using the Galerkin's method is to

find $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \in \mathcal{R}^N$ such that

$$\int_\Omega P\left(\sum_{j=1}^{B} \alpha_j \phi_j, \phi_i\right) d\Omega + \int_{\partial\Omega} Q\left(\sum_{j=1}^{N} \alpha_j \phi_j\right) \phi_i \, ds, \quad (i = 1, 2, \ldots, N), \tag{2.13}$$

where $\{\phi_i\}$ are the basis functions of H^h.

Example 2.4 Find the Galerkin approximation for the following variational problem.

Find $u \in H_0$ such that

$$\int_a^b (u_x v_x + fv)\, dx - g(b)v(b) \;=\; 0, \quad \forall v \in H_0,$$

where H_0 is as defined in example 2.3.

Sol

To find the Galerkin's approximation, we choose an N dimensional subspace of functions $H^h \subset H_0$ with basis functions $\{\phi_1, \phi_2, \ldots, \phi_n\}$. Then the problem is

Find $u_N \in H^h$ such that

$$\begin{aligned} & \int_a^b \left\{ \left[\textstyle\sum_1^N \alpha_j \phi_j(x)\right]' \phi_i' + f\phi_i \right\} dx - g(b)\phi_i(b) \;=\; 0 \\ \Rightarrow \quad & \textstyle\sum_{j=1}^N \left[\int_a^b \phi_i'\phi_j'\, dx\right] \alpha_j \;=\; -\int_a^b f\phi_i\, dx + g(b)\phi_i(b) \\ \Rightarrow \quad & \textstyle\sum_{j=1}^N K_{ij}\alpha_j \;=\; F_i \quad (i = 1, 2, \ldots, N), \end{aligned}$$

where $K_{ij} = \int_a^b \phi_i'\phi_j' dx,\ F_i = -\int_a^b f\phi_i dx + g(b)\phi_i(b)$.

Remarks :

1) The system of equations $KA = F$ has unique solution.

Proof : Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \neq \mathbf{0}$

$$\alpha K \alpha = \sum_{i,j=1}^N \alpha_i K_{ij} \alpha_j = \int_a^b \sum_{i=1}^N \alpha_i \phi_i' \sum_{j=1}^N \alpha_j \phi_j' dx = \int_a^b v'v' dx \geq 0$$

with equality only if $v' = 0$ (v = constant).

Now as $v(a) = 0$, $v' = 0$ if and only if $v = 0$ or $\alpha = \mathbf{0}$.

Therefore, K is positive definite and thus $KA = F$ has a unique solution.

Note: A symmetric matrix A is positive definite if

$$\eta \cdot A \cdot \eta = \sum_{i,j=1}^{N} \eta_i A_{ij} \eta_j > 0 \quad \forall \eta \in \mathcal{R}^N,\ \eta \neq \mathbf{0}$$

2) if $\phi_i(x)$ are determined, the determination of α_j is then just a computational matter.

2.4 Finite Element Approximation

While Galerkin's method provides an elegant strategy for constructing approximations of solutions to boundary value problems, it has some serious shortcoming:

1) In the method as we have described, there is no systematic way of constructing reasonable basis functions ϕ_i . Aside from being independent members of H^h, they are arbitrary. The analyst is left with a bewildering number of possibilities at his disposal and with the discomforting knowledge that the quality of his approximate solution will depend very strongly on the properties of the basis functions chosen. Moreover a poor choice of ϕ_i may produce an ill-conditioned system of equations.

2) For essential boundary condition $u(\mathbf{x}) = g(\mathbf{x})\ \ \forall \mathbf{x} \in \partial\Omega$, the test function ϕ_i must be designed to fit the boundary condition which is very difficult for domain with complex geometry.

For the above reasons, the classical Galerkin method is of rather limited use. These substantial difficulties can be resolved by using the finite element method. The finite

element method provides a general and systematic technique for constructing basis functions for the Galerkin approximation of boundary value problems. The main idea is that the basis functions can be defined piecewisely over subregions of the domain called finite elements and that over each element, ϕ_i can be chosen to be very simple functions such as polynomials of low degree.

Thus, to find a finite element solution, we firstly divide Ω into E subregions called elements. Then choose N points called *nodes* where the values of u are taken as basic unknowns.

Example 2.5 Let $\Omega = \{x : x \in [0,1]\}$ and

- divide Ω into 4 elements with 5 nodes as shown

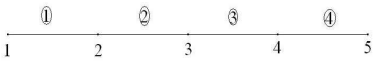


Figure 2.1: Domain with 4 elements 5 nodes

- use the values of u at nodes as basic unknowns, i.e. the unknowns now are $u_i = u(x_i)$ $(i = 1, 2, 3, 4, 5)$.

To identify the collection of elements and nodes (mesh), we need to do the following steps:

- number elements and nodes globally as shown in Figure 2.1;
- identify each individual element, i.e., record which nodes are contained in the element.

Suppose that there are M nodes in an element Ω_e, we denote these nodes

$$\{N_1^e, N_2^e, \ldots, N_M^e\}.$$

For example, in example 2.3, element 2 contains two nodes:

$$N_1^2 = 2, \quad N_2^2 = 3.$$

In the next step, we choose the global interpolating function $\phi_i(x)$ in such a way that

(1)

$$\phi_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2.14}$$

(2) $\phi_i(x) = 0$ on elements that do not contain node i

In example 2.5, the basis functions $\phi_1(x)$ and $\phi_2(x)$ are as shown below.

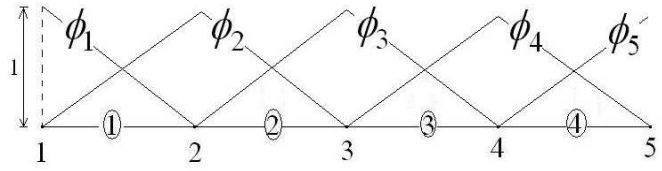


Figure 2.2: Element shape functions $\phi_i(x)$.

For convenience in discussion, we denote the part of $\phi_i(x)$ on element e by ϕ_i^e. Thus,

$$\phi_i = \cup \phi_i^e,$$

where e denotes any element around the node i. For the example 2.5, we have

$$\phi_2 = \phi_2^1 \cup \phi_2^2.$$

In the following, we give an interpretation of ϕ_i^e. With properties (1) and (2) above, the interpolating formula becomes

$$u_N(x) = \sum_{j=1}^{N} u_j \phi_j(x). \tag{2.15}$$

As $\phi_j(x)$ are defined piecewise, $u_N(x)$ is also a piecewise function. Now consider a typical element Ω^e with M nodes $\{N_1^e, N_2^e, \ldots, N_M^e\}$. Within Ω_e

$$u_N^e(x) = \sum_{j=1}^{N} u_j \phi_j(x).$$

From (2.14), only those ϕ_j with node j in Ω^e have contribution to $u_N(x)$.

Suppose the element Ω_e has M nodes $\{N_1^2, N_2^2, \ldots, N_M^e\}$. Then $u_N^e(x) = \sum_1^M u_{N_i^e} \phi_{N_i^e}^e$ which, for simplicity, can be written as

$$u_N^e(x) = \sum_{1}^{M} u_i^e \phi_i^e. \tag{2.16}$$

eg. for Ω_2, $\qquad u_N^2(x) = \sum_1^2 u_i^2 \phi_i^2 = u_2\phi_2^2 + u_3\phi_3^2.$

From (2.16), ϕ_i^e is the local interpolating function defined on Ω_e. As Ω_e is a small region, we can use simple function, such as low degree polynomial, for ϕ_i^e.

Now, it is clear that, the global $\phi_i(x)$ is the assembly of local interpolation functions $\phi_i^e(x)$. As $\phi_i(x)$ is defined piecewise over each Ω_e, system (2.13) can be rewritten as

$$\sum_{e=1}^{E} \int_{\Omega_e} P\left(\sum_{1}^{N} u_j \phi_j^e, \phi_i^e\right) d\Omega + \sum_{e=1}^{\partial E} \int_{\Omega_e} Q\left(\sum_{1}^{N} u_j \phi_j^e\right) \phi_i^e \, ds = 0 \quad (i = 1, 2, \ldots, N).$$

Remarks:

1) To ensure the regularity of the integrals in the variational statement, choosing proper classes of admissible functions for the solution u and the weighting function v is an important step in solving the problem using finite element method.

2) We often include boundary conditions in our definitions of Sobolev spaces. eg. We define

$$H_0^1 = \left\{ v \in H^1(0,1) \mid v = 0 \text{ at } x = 0 \right\}.$$

Thus, the global equations can be constructed by assembling the contributions from each of the elements. Hence, in summary, to solve a boundary value problem using the finite element method, we need to perform the following steps

1) Discretise Ω,

2) Choose local element shape function $\phi_i^e(x)$,

3) Construct the global equations by assembling contributions from each of the elements,

4) Impose the boundary conditions,

5) Solve the system of equations for u_i,

6) Determine the field variable via (2.16).

2.5 Classes of Admissible Functions

This section introduces some functional spaces which are important in solving boundary-value problems and are frequently referenced in finite element formulation. The unknown function u and test function v must be chosen from certain classes of functional spaces.

2.5.1 Spaces of continuous functions $C^m(\Omega)$

Suppose Ω is a bounded region in R^3, $u = u(x, y, z)$ is a real-valued function in Ω, then u is said to be of class C^m in Ω (or to belong to $C^m(\Omega)$, or to be a C^m-function) if u and all of its derivatives of order $\leq m$ are continuous at every point in Ω. Thus, $C^m(\Omega)$ is defined by

$$C^m(\Omega) = \Big\{ v = v(x,y,z) | v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial v}{\partial z}, \frac{\partial^2 v}{\partial x^2}, \ldots, \frac{\partial v^m}{\partial x^m}, \frac{\partial^m v}{\partial y \partial x^{m-1}}, \ldots, \frac{\partial^m v}{\partial z^m}$$
$$\text{are continuous in } \Omega \subset R^3 \Big\}$$

- The class $C^m(\Omega)$ is a linear space of functions. That is, if $u \in C^m(\Omega)$ and $v \in C^m(\Omega)$, then $\alpha u + \beta v \in C^m(\Omega)$ for any real scalars α and β.

- For a nonempty set Ω, the set $\mathcal{F}(\Omega)$ of all real functions defined in Ω is a linear space with the following properties

$$(u + v)(x) = u(x) + v(x) \qquad \forall x \in \Omega$$

$$(\alpha v)(x) = \alpha v(x) \qquad \forall x \in \Omega$$

- For an open subset Ω of $\Re^n$, the subset in $\mathcal{F}(\Omega)$ which are continuous is a subspace of $\mathcal{F}(\Omega)$ and is denoted by $C^0(\Omega)$.

Definition 2.5.1

$$C^0(\bar{\Omega}) = \{ v \in C^0(\Omega) : v \texttt{ is bounded and continuous in } \Omega \} \tag{2.17}$$

with the norm

$$\|v\|_{C^0(\bar{\Omega})} = sup_{x \in \Omega} |v(x)|. \tag{2.18}$$

Definition 2.5.2

$$C^k(\bar{\Omega}) = \{v \in C^k(\Omega) : D^\alpha v \in C^0(\bar{\Omega}) \texttt{ for } |\alpha| \leq k\} \tag{2.19}$$

with seminorm and norm:

$$|v|_{C^k(\bar{\Omega})} = \sum_{|\alpha|=k} \|D^\alpha v\|_{C^0(\bar{\Omega})} \tag{2.20}$$

$$\|v\|_{C^k(\bar{\Omega})} = \sum_{j=0}^{k} |v|_{C^j(\bar{\Omega})} = \sum_{|\alpha|\leq k} \|D^\alpha v\|_{C^0(\bar{\Omega})}. \tag{2.21}$$

Consider the function $u(x),\ v(x) = u'(x),\ w(x) = u''(x),\ x \in [0, a]$ as shown in Fig. 2.3. It is clear that w has a jump discontinuity at the point x_0 and v is continuous, which means $u \notin C^2([0, a]),\ u \in C^1([0, a])$ `and` $v \in C^0([0, a])$.

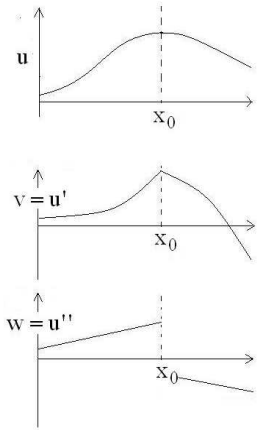


Figure 2.3: Function $u(x)$ and its derivatives

Definition 2.5.3 A function v on Ω is called Hölder-continuous of order λ for $0 < \lambda \leq 1$ iff there exists $M > 0$ such that $|v(x) - v(y)| \leq M|x - y|^\lambda,\ \forall x, y \in \Omega$.

- if $\lambda = 1$, the 1-Hölder-continuous functions are called Lipschitz-continuous functions.

- the λ-Hölder continuity implies uniform continuity [take $\delta = (\varepsilon/M)^{1/\lambda}$]. Hence bounded λ-Hölder continuous functions belong to $C^0(\bar{\Omega})$.

Definition 2.5.4 For $0 < \lambda \leq 1$, we have

$$C^{0,\lambda}(\bar{\Omega}) = \{v \in C^0(\bar{\Omega}) \ : \ v \ \texttt{is} \ \lambda - \texttt{Hölder continuous in} \ \ \Omega\} \tag{2.22}$$

with the following seminorm and norm:

$$|v|_{C^{0,\lambda}(\bar{\Omega})} = sup_{x,y\in\Omega, x\neq y}|v(x) - v(y)||x - y|^{-\lambda}, \tag{2.23}$$

$$\|v\|_{C^{0,\lambda}(\bar{\Omega})} = \|v\|_{C^0(\bar{\Omega})} + |v|_{C^{0,\lambda}(\bar{\Omega})}. \tag{2.24}$$

Definition 2.5.5 For $k \in \mathbf{N}$ and $0 \leq \lambda \leq 1$, we set

$$C^{k,\lambda}(\bar{\Omega}) = \{v \in C^k(\bar{\Omega}) : D^\alpha v \in C^{0,\lambda}(\bar{\Omega}) \ \texttt{for} \ |\alpha| \leq k\} \tag{2.25}$$

with the following seminorm and norm:

$$|v|_{C^{k,\lambda}(\bar{\Omega})} = \sum_{|\alpha|=k} |D^\alpha v|_{C^{0,\lambda}(\bar{\Omega})}, \tag{2.26}$$

$$\|v\|_{C^{k,\lambda}(\bar{\Omega})} = \sum_{|\alpha|\leq k} \|D^\alpha v\|_{C^{0,\lambda}(\bar{\Omega})} \tag{2.27}$$

Definition 2.5.6 For $k \in \mathbf{N}$ and $0 \leq \lambda \leq 1$, we set

$$C^{k,\lambda}(\Omega) = \{v \in C^k(\Omega) : v|_{\Omega_1} \in C^{k,\lambda}(\bar{\Omega}_1) \ \texttt{for every} \ \Omega_1 \subset\subset \Omega\} \tag{2.28}$$

Remarks:

- Notation $\Omega_1 \subset\subset \Omega$ means that Ω_1 and Ω are open, Ω_1 is bounded, and $\bar{\Omega}_1 \subset \Omega$;

- $C^{0,0}(\bar{\Omega}) = C^0(\bar{\Omega})$ and $C^{k,0}(\Omega) = C^k(\Omega)$.

Definition 2.5.7 We set

$$C^\infty(\Omega) = \{v \in C^0(\Omega) : v \in C^k(\Omega)\ \forall k\} \tag{2.29}$$

$$C^\infty(\bar{\Omega}) = \{v \in C^0(\bar{\Omega}) : v \in C^k(\bar{\Omega})\ \forall k\} \tag{2.30}$$

If V is any linear space and Ω is an open set in $\Re^n$, then a function $v : \Omega \to V$ is a **compact support function** iff there exists $\Omega_1 \subset\subset \Omega$ such that v vanishes outside Ω_1.

Definition 2.5.8 We set

$$C_0^\infty(\Omega) = \{v \in C^\infty(\Omega) : v \texttt{ is a compact support function}\}. \tag{2.31}$$

2.5.2 Banach spaces

Definition 2.5.9 A normed space V is called a *Banach space* iff every Cauchy sequence in V has a strong limit in V.

For example,

1) $\Re^n$ with the norm $\|\mathbf{x}\|_1 = \sum |x_i|$ is a Banach space.

2) $\Re^n$ with the Euclidean norm $\|(x_1, \ldots, x_n)\| = (\sum x_i^2)^{1/2}$ is a Banach space.

3) If $V_i (i = 1, \ldots, n)$ are Banach spaces with norms $\|\cdot\|_{V_i}$, then $V = V_1 \times \ldots \times V_n$ is a Banach space with norm

$$\|(V_1, \ldots, V_n)\| \ = \ (\sum \|v_i\|_{V_i}^2)^{1/2}.$$

Definition 2.5.10 A Banach space is called *separable* iff it contains a finite or countably infinite set A, such that `span` A is dense.

2.5.3 Hilbert spaces

Definition 2.5.11 A Banach space V is called a *Hilbert space* if the mapping

$$v \to \|v\|^2$$

is a quadratic form on V.

Definition 2.5.12 In a Hilbert space, there exists a unique symmetric bilinear form

$$u, v \to (u, v)_V \equiv \frac{1}{4}(\|u+v\|^2 - \|u-v\|^2) \tag{2.32}$$

such that $(v, v)_V = \|v\|^2$ for all v in V. The bilinear symmetric bilinear form (2.32) is called the *scalar product*.

For example,

1) $\Re^n$ with the Euclidean norm

$$\|(x_1, \ldots, x_n)\| = (\sum x_i^2)^{1/2}$$

is a Hilbert space.

2) If $V_i (i = 1, \ldots, n)$ are Hilbert spaces with norms $\|\cdot\|_{V_i}$, then $V = V_1 \times \ldots \times V_n$ is a Hilbert space with norm

$$\|(V_1, \ldots, V_n)\| \;=\; (\sum \|v_i\|_{V_i}^2)^{1/2}.$$

The corresponding scalar product is given by

$$(v, w)_V = \sum (v_i, w_i)_{V_i}.$$

Basic Properties of Hilbert Spaces

Let V be a Hilbert space, then

- $|(u,v)| \leq \|u\|\|v\| \quad \forall u, v \in V$ —- Schwarz inequality.
- for each $u \in V$, consider the mapping $Ju : V \to \Re$ defined by

$$Ju : v \to (u,v) \quad \forall v \in V.$$

  The map $u \to Ju$ is an isometric isomorphism from V to V'.

- V' is a Hilbert space.
- $\forall f \in V' \;\; \exists u (= J^{-1} f)$ such that

$${}_{V'}(f,v)_V = (u,v)_V \quad \forall v \in V$$

  implies $v_n \rightharpoonup v_0$ iff $lim_{n\to\infty}(v_n, v) = (v_0, v)$ for all $v \in V$

- Every Hilbert space is reflexive.

2.5.4 Distributions

Definition 2.5.13 Given $v_k,\ v \in C_0^\infty(\Omega)$, $\{v_k\}$ converges to $v \in \mathcal{D}(\Omega)$ iff there exists a bounded closed set $K \subset \Omega$ such that v_k vanishes outside K for any k, and for every α,

$$D^\alpha v_k \to D^\alpha v$$

uniformly in Ω.

Definition 2.5.14 A distribution on Ω is a linear functional L on $\mathcal{D}(\Omega)$ which is continuous in the sense that

$$v_k \to v \in \mathcal{D}(\Omega) \;\; \texttt{implies} \;\; L(v_k) \to L(v).$$

We define

$$_{\mathcal{D}'(\Omega)}< L, v >_{\mathcal{D}(\Omega)} = L(v), \tag{2.33}$$

for $L \in \mathcal{D}'(\Omega)$ and $v \in \mathcal{D}(\Omega)$. If $D_i = \partial/\partial x_i,\ u \in C^1(\Omega)$, and $v \in \mathcal{D}(\Omega)$, Green formula gives

$$\int_\Omega D_i u \cdot v dx = -\int_\Omega u \cdot D_i v dx$$

Definition 2.5.15 We define

$$_{\mathcal{D}'}< D_i u, v >_{\mathcal{D}} = -_{\mathcal{D}'}< u, D_i v >_{\mathcal{D}} \tag{2.34}$$

Definition 2.5.16 For higher-order derivative

$$_{\mathcal{D}'}< D^\alpha u, v >_{\mathcal{D}} = (-1)^{|\alpha|}\ _{\mathcal{D}'}< u, D^\alpha v >_{\mathcal{D}} \tag{2.35}$$

The sequence $\{u_k\}$ is converging to u in $\mathcal{D}'$ iff

$$_{\mathcal{D}'}< u_k, v >_{\mathcal{D}} \to\ _{\mathcal{D}'}< u, v >_{\mathcal{D}},\ \ \forall v \in \mathcal{D}(\Omega) \tag{2.36}$$

Note: The sequence $\{u_k\}$ is converging to u in $\mathcal{D}'$ ($u_k \to u \in \mathcal{D}'(\Omega)$) implies $D^\alpha u_k \to D^\alpha u \in \mathcal{D}'(\Omega)$ for all α.

2.5.5 The p-integrable spaces $L^p(\Omega)$

Definition 2.5.17 A function $v \in \mathcal{F}(\Omega)$ is called **measurable** iff a sequence of functions $v_n \in C^0(\Omega)$ exists such that $v_n \to v$ almost everywhere in Ω.

Let $\mathcal{M}(\Omega)$ denote the set of measurable functions which is a linear space, for $u,\ v \in \mathcal{M}(\Omega)$ and $f \in C^0(\Re)$, we have

- The composition $f \circ u \in \mathcal{M}(\Omega)$;

- $u^+,\ u^-,\ |u|^p \in \mathcal{M}(\Omega)$ for $p > 0$;
- If $u = v$ almost everywhere then $f \circ u = f \circ v$ almost everywhere;
- If the Lebesgue integral $\int_\Omega |v|^p = 0$ then $v = 0$ almost everywhere.

By definition we can conclude that every continuous function is measurable: $C^0(\Omega) \subset \mathcal{M}(\Omega)$.

Note Any set where $v_n(x)$ does not converge to $v(x)$ as $n \to \infty$ has zero Lebesgue measure.

Definition 2.5.18 $L^p(\Omega)$ is a subspace of measurable space defined by:

$$L^p(\Omega) = \left\{ u \in \mathcal{M}(\Omega) \ : \ \int_\Omega |u(\mathbf{x})|^p \, dx \ < \ \infty \right\} \tag{2.37}$$

with the norm

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega |u|^p \right)^{1/2}. \tag{2.38}$$

Note: For $u,\ v \in \mathcal{M}(\Omega)$, $u = v$ almost everwhere does not imply that $\texttt{sup}\ u = \texttt{sup}\ v$ and $\texttt{inf}\ u = \texttt{inf}\ v$.

Definition 2.5.19 For $v \in L^\infty(\Omega)$, we set

$$L^\infty(\Omega) = \{ v \in \mathcal{M}(\Omega) \ : \ v \ \texttt{is bounded} \} \tag{2.39}$$

with the norm

$$\|v\|_{L^\infty(\Omega)} = \texttt{ess sup}_\Omega |v|. \tag{2.40}$$

where

$$\begin{aligned} \texttt{ess sup}_\Omega\, v &= \texttt{ess sup}_{x \in \Omega}\, v(x) \\ &= \texttt{inf}\{ \mathcal{M} \in (-\infty, +\infty] \ : \ v(x) \leq \mathbf{M} \\ &\quad \texttt{almost everywhere in}\ \Omega \} \end{aligned} \tag{2.41}$$

and

$$\begin{aligned} \texttt{ess inf}_{\Omega}\, v &= \texttt{ess inf}_{x\in\Omega}\, v(x) \\ &= \texttt{sup}\{\mathcal{M}\in[-\infty,+\infty) \,:\, v(x)\geq\mathcal{M} \\ &\quad \texttt{almost everywhere in}\,\Omega\}. \end{aligned} \tag{2.42}$$

Space Properties

- For $1 \leq p \leq \infty$, $L^p(\Omega)$ is a Banach space.
- $L^p(\Omega)$ is separable iff $1 \leq p < \infty$
- $L^p(\Omega)$ is a Hilbert space iff $p = 2$. The scalar product is given by

$$(u, v)_{L^2(\Omega)} = \int_\Omega uv \, d\Omega. \tag{2.43}$$

- Let $p,\ p_1, \ldots,\ p_m \in [1, \infty]$ and $\sum p_i^{-1} = p^{-1}$ (with $\infty^{-1} = 0$). If $v_i \in L^{p_i}(\Omega)$ for $i = 1, \ldots, m$, then the function $\prod v_i$ belongs to $L^p(\Omega)$ and

$$\left\|\prod v_i\right\|_{L^p(\Omega)} \leq \prod \|v_i\|_{L^{p_i}(\Omega)} \quad (\texttt{Hölder's inequality}) \tag{2.44}$$

For example,

1) if $\sum p_i^{-1} = 1$, then

$$\left|\int_\Omega \prod v_i d\Omega\right| \leq \prod \|v_i\|_{L^{p_i}(\Omega)} \tag{2.45}$$

and with $m = 2$ we have

$$\left|\int_\Omega uv \, d\Omega\right| \leq \|u\|_{L^q(\Omega)} \|v\|_{L^{q'}(\Omega)} \ \ \forall u \in L^q,\ \forall v \in L^{q'}, \tag{2.46}$$

where the number q' is the conjugate of q and

$$\frac{1}{q} + \frac{1}{q'} = 1. \tag{2.47}$$

2) If $q = 2$, equation (2.46) is the **Schwarz inequality**.

For $u \in L^{p'}(\Omega)$ and $v \in L^p(\Omega)$, by (2.46), the map $L(u) : v \to \int uv\, dx$ is continuous on $L^p(\Omega)$. Thus, $L(u) \in (L^p(\Omega))'$ and the map $u \to L(u)$ is one-to-one, linear, isometric from $L^{p'}(\Omega)$ into $(L^p(\Omega))'$. For $1 \le p < \infty$, $u \in L^{p'}(\Omega)$ and $v \in L^p(\Omega)$, we write

$$ {}_{(L^p)'} < u, v >_{L^p} = \int_\Omega uv\, dx \tag{2.48} $$

2.5.6 The Sobolev spaces $W^{k,p}(\Omega$

Definition 2.5.20 For a nonnegative integer k and p satisfying $1 \le p \le \infty$, the Sobolev space $W^{k,p}(\Omega)$ of order (k, p) is the linear space of functions in $L^p(\Omega)$ whose distribution derivatives $D^\alpha u$ of all order $|\alpha|$ such that $0 \le |\alpha| \le k$ are in $L^p(\Omega)$:

$$ W^{k,p}(\Omega) = \{u \mid D^\alpha u \in L^p(\Omega) \ \texttt{for}\ 0 \le |\alpha| \le k\} \tag{2.49} $$

The spaces $W^{k,p}(\Omega)$ are generally endowed with the norms

$$ \begin{aligned} \|u\|_{k,p,\Omega} &= \left[\int_\Omega \sum_{0\le|\alpha|\le k} |D^\alpha u(\mathbf{x})|^p\, dx\right]^{1/p} \\ &= \left[\sum_{0\le|\alpha|\le k} \|D^\alpha u\|^p_{L^p(\Omega)}\right]^{1/p}, \quad 1 \le p < \infty \\ \|u\|_{k,\infty,\Omega} &= max_{0\le|\alpha|\le k} \|D^\alpha u(\mathbf{x})\|_{L^\infty(\Omega)} \end{aligned} \tag{2.50} $$

and seminorm

$$ |u|_{k,p,\Omega} = \left[\int_\Omega \sum_{|\alpha|=k} |D^\alpha u(\mathbf{x})|^p\, dx\right]^{1/p}. \tag{2.51} $$

We also denote

$$ \begin{aligned} W_0^{k,p}(\Omega) &= \texttt{the closure of}\ C_0^\infty(\Omega)\ \texttt{in}\ W^{k,p}(\Omega) \\ H^k(\Omega) &= W^{k,2}(\Omega); \quad H_0^k(\Omega) = W_0^{k,2}(\Omega) \end{aligned} \tag{2.52} $$

and

$$W^{0,p}(\Omega) = W_0^{0,p}(\Omega) = L^p(\Omega). \tag{2.53}$$

The scalar product in $H^k(\Omega)$ is given by

$$(u,v)_{k,\Omega} = \sum_{|\alpha|\leq k} \int_\Omega D^\alpha u \cdot D^\alpha v d\Omega. \tag{2.54}$$

The sequence $\{v_m\}$ is converging to v in $W^{k,p}(\Omega)$ ($v_m \to v \in W^{k,p}(\Omega)$) iff $D^\alpha v_m \to D^\alpha v \in L^p(\Omega)$ for $|\alpha| \leq k$.

Note: The norm defined in (2.51) is a Hilbert norm iff $p = 2$.

A function v is in the class $H^m(\Omega)$ if v and all of its partial derivatives of order $\leq m$ are members of the class $L^2(\Omega)$. Compactly, we write for $\Omega \subset R^2$,

$$H^m(\Omega) = \left\{ v \mid v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \ldots, \frac{\partial^m v}{\partial z^m} \in L^2(\Omega) \right\}$$

The class $H^m(\Omega)$ is a linear space of functions : if $u \in H^m(\Omega)$ and $v \in H^m(\Omega)$, then $\alpha u + \beta v \in H^m(\Omega)$ for any real scalars α and β.

The Sobolev class $H^m(\Omega)$ is a natural generalization of the C^m-class for quantifying the smoothness or regularity of functions. In Fig 2.3, u is not in $C^2([0,a])$ since u'' is discontinuous at $x = x_0$. However, $w = u''$ is certainly square-integrable, thus

$$u \in H^2(0,a).$$

Similarly, $v' \in H^1(0,a)$, and $w = u'' \in H^0(0,a)$.

Major properties of $H^m(\Omega)$

- Linearity : $H^m(\Omega)$ is a linear space.

  This means that if u and v are in $H^m(\Omega)$, then their combinations $\alpha u + \beta v$ are also in $H^m(\Omega)$.

- Orthogonality : An inner product can be defined in $H^m(\Omega)$. For example, the scalar

$$(u,v)_m \equiv \int_0^1 \left(\frac{d^m u}{dx^m}\frac{d^m v}{dx^m} + \frac{d^{m-1}u}{dx^{m-1}}\frac{d^{m-1}v}{dx^{m-1}} + \ldots + \frac{du}{dx}\frac{dv}{dx} + uv \right) dx$$

defines an inner product on $H^m(0,1)$.

- Magnitude : $H^m(\Omega)$ is a norm space, and the norm of a function $u \in H^m(\Omega)$ being defined as the non-negative real number $\| u \|_m$ given by

$$\| u \|_m = \sqrt{(u,u)_m}$$

The norm is a measure of the magnitude of the function and has the properties that for any $u, v \in H^m(\Omega)$, and any real number α,

$\| u+v \|_m \leq \| u \|_m + \| v \|_m$,

$\| \alpha u \|_m = | \alpha | \cdot \| u \|_m$,

$\| u \|_m \geq 0$ and $\| u \|_m = 0$ if and only if $u = 0$,

$| (u,v)_m | \leq \| u \|_m \| v \|_m \quad \forall u, v \in H^m(\Omega)$—Schwarz inequality.

- Distance : The distance between functions in $H^m(\Omega)$ is defined by the H^m-norm of their difference. Thus the difference between u and v is $\| u - v \|_m$

EXERCISE 2

Question 1

Given a boundary value problem

$$-u''(x) = \delta(x - \frac{1}{2}) \quad x \in (0,1)$$

$$u(0) = 0, \quad u(1) = 0,$$

Construct a variational statement of the problem. (Ans : Find $u \in H_0^1$ such that $\int_0^1 u'v'dx = v(\frac{1}{2}) \quad \forall v \in H_0^1$, where $H_0^1 = \ldots.$)

Question 2

Show that one variational formulation of the boundary value problem

$$-xu'' - u' + u = sinx, \quad x \in (0,1)$$

$$u(0) = u(1) = 0$$

is as follows.

Find $u \in H_0^1$ such that

$$\int_0^1 (xu'v' + uv - vsinx)dx = 0 \quad \forall v \in H_0^1,$$

where $H_0^1 = \{v | v \in H^1(0,1) \text{ and } v(0) = v(1) = 0\}$.

Question 3

Consider

$$-u'' + u = x \quad x \in (0,1)$$

$$u(0) = 0, u(1) = 0$$

1) Find the variational statement of the boundary value problem

2) Find a Galerkin approximation using $N = 3$ and

$$\phi_i = sin(i\pi x) \quad (i = 1, 2, 3)$$

a) Calculate K_{ij} and F_i, then solve for α_j

b) Construct the approximate solution $u_N(x)$

Question 4

Develop a FE formulation for the B.V.P.

$$-\frac{d}{dx}\left(k(x)\frac{du}{dx}\right) + b(x)u(x) = f(x) \quad x \in (a,b)$$

$$u(a) = 0,\ u(b) = 0$$

$$(Ans : Ku = F \ \text{ with } \ K_{ij} = \ldots, F_i = \ldots)$$

Chapter 3

Two-Point Boundary Value Problems

The aim of this chapter is to enhance the understanding of the finite element method by working through all details of the finite element approximation via a simple 2-point boundary value problem defined by

$$\begin{cases} -\frac{d}{dx}(k(x)\frac{du}{dx}) + \gamma(x)u = f(x) & x \in (a,b) \\ u(a) = A,\ k(b)\frac{du}{dx}(b) = -p_b[u(b) - u_\infty] = \sigma(b). \end{cases} \tag{3.1}$$

3.1 Finite Element Formulation

Variational statements

The residual error function corresponding to the given differential equation is

$$r(x) = -\frac{d}{dx}(k\frac{du}{dx}) + \gamma u - f,$$

from which we obtain the overall weighted residual error

$$R = \int_a^b vr(x)dx = \int_a^b (ku'v' + \gamma uv - fv)dx - kv\left[\frac{du}{dx}\right]_a^b.$$

For the above integrals to be well defined, it is sufficient to choose

$$u \in H^1(a,b), \quad v \in H^1(a,b).$$

In addition, as $u'(a)$ is not given, choose v from $H_0^1 = \{v : v \in H^1 \text{ and } v(a) = 0\}$. Then the variational statement for the boundary value problem is

Find $u \in H^1$ such that $u(a) = A$ and

$$(ku', v') + (\gamma u, v) = (f, v) + v(b)\sigma(b) \quad \forall v \in H_0^1, \tag{3.2}$$

$$\text{where } (u_1, u_2) = \int_a^b u_1 u_2 dx. \tag{3.3}$$

Derivation of Finite Element Equations

To derive a set of finite element equations for the solution of the two-point boundary value problem, we divide $[a,b]$ into N_{ele} elements with N_{node} nodes. Let H_h^1 be the N dimensional subspace of H^1 ($H_h^1 \subset H^1$), with basis functions $\{\phi_i\}_{i=1}^N$ and $H_{0h}^1 \subset H_0^1$. Then, the finite element approximation is to

find $u_n \in H_h^1$ such that $u(a) = A$ and

$$(ku_N', v_N') + (\gamma u_N, v_N) = (f, v_N) + v(b)\sigma(b) \quad \forall v_N \in H_{0h}^1. \tag{3.4}$$

As

$$u_N = \sum_{j=1}^N u_j \phi_j, \quad v_N = \sum_{i=1}^N v_i \phi_i,$$

(3.4) becomes

$$\sum_{j=1}^N [(k\phi_i', \phi_j') + (\gamma\phi_i, \phi_j)]u_j \;=\; (f, \phi_i) + \sigma(b)\phi_i(b). \tag{3.5}$$

$$\Rightarrow \quad \mathbf{Ku} \;=\; \mathbf{F} + \mathbf{F_b}, \tag{3.6}$$

where $\mathbf{K} = (K_{ij})$ with $K_{ij} = (k\phi_i', \phi_j') + (b\phi_i, \phi_j)$,

$\mathbf{F} = (F_i)$ with $F_i = (f, \phi_i)$ and

$\mathbf{F}_b = (F_{bi})$ with $F_{bi} = \sigma(b)\phi_i(b)$.

As discussed in Chapter 2, ϕ_i are continuous functions defined piecewisely over each Ω_e. Thus, the global stiffness matrix $\mathbf{K}$ and the vector $\mathbf{F}$ can be obtained by assembling the corresponding contributions from each element, i.e

$$K_{ij} = \sum_{e=1}^{N_{ele}} K_{ij}^e, \quad F_i = \sum_{e=1}^{N_{ele}} F_i^e, \tag{3.7}$$

with element quantities

$$K_{ij}^e = \int_{\Omega_e} (k\dot{\phi}_i^e \dot{\phi}_j^e + \gamma \phi_i^e \phi_j^e) dx, \quad F_i^e = \int_{\Omega_e} f\phi_i^e \, dx. \tag{3.8}$$

Therefore, to solve a boundary value problem using the finite element method, we need to perform the following steps:

1) Discretise Ω;

2) Choose element shape function ϕ^e;

3) Calculate element matrix $\mathbf{K}^e$ and vector $\mathbf{F}^e$;

4) Construct global matrix $\mathbf{K}$ and vector $\mathbf{F}$ by assembling contributions from each element;

5) Impose boundary conditions;

6) Solve the system of equations.

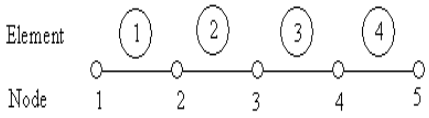
3.2 Finite Element Approximation

Consider solving

$$\begin{cases} -u_{xx} \;=\; 2 \quad x \in (a,b) \\ u(a) = \hat{u}_1,\ \frac{du}{dx}(b) = -p_b[u(b) - u_\infty] \;=\; \sigma(b). \end{cases} \tag{3.9}$$

Discretization and Topology of finite element mesh

Supose that we divide [a, b] into 4 equally spaced elements. Within each element, we choose 2 nodes (left and right ends). Then, we design a global numbering scheme for the elements and nodes.



Once the numbering scheme has been established for a finite element mesh, we must create the system's topology - the element definition. This topology tells how the elements are jointed together. On the element level, the topology is simply the ordered numbering of the nodes. Table 4.1 illustrates the system topology that has been established for our model. This information can be easily stored into a two dimensional matrix with each row storing the topology of one element.

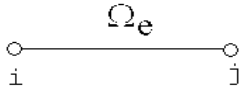
For example, the topology of element 2 can be recorded in the $2nd$ row of a matrix $Node(100,2)$: $Node(2,1)$=2 and $Node(2,2)$=3. For convenience, denote N_i^e as the ith node of element e, i.e., $N_i^e = Node(e,i)$.

Table 3.1: System Topology

element	Numbering scheme			
	Local		Global	
1	i	j	1	2
2	i	j	2	3
3	i	j	3	4
4	i	j	4	5

Selection of Element Shape Functions

For an arbitrarily chosen element Ω_e, to approximate $u(x)$ over Ω_e by polynomial of degree k, we need to choose $(k+1)$ points within Ω_e. Here we approximate $u(x)$ by a polynomial of degree 1. So we choose two nodes $i,\ j$ (left and right ends) for Ω_e as shown.



To standardize the calculation of element matrices, we firstly transform the element Ω_e into a standard element defined in $[-1, 1]$. This process is as follows:

Step 1. Introduce local coordinate ξ with $\begin{cases} \text{origin } \xi = 0 & \text{at the centre of element} \\ \xi = -1 & \text{at the left hand node} \\ \xi = 1 & \text{at the right hand node} \end{cases}$

This can be achieved by a linear transformation

$$\xi = \frac{2x - (x_i + x_{i+1})}{x_{i+1} - x_i}, \tag{3.10}$$

so that points $x \in [x_i,\ x_{i+1}]$ are transformed to points $\xi \in [-1, 1]$. As every element can be transformed into such element, we call this element as *master element* denoted by $\bar{\Omega}$ and then we perform our element calculation on this reference element.

Step 2. For shape functions of degree k, we need to identify $(k+1)$ nodes (including the end points).

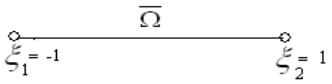
Let ξ_i denote the ξ-coordinate of the ith node,

u_i^e denote the value of u at node i

Then within Ω_e, $u(x)$ can be approximated by the Lagrange polynomial,

$$u^e(x) = \sum_{i=1}^{k+1} \bar{\phi}_i u_i^e \quad \text{with } \bar{\phi}_i = \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{(\xi - \xi_j)}{(\xi_i - \xi_j)}, \tag{3.11}$$

where $u^e(x)$ is the local approximation of $u(x)$ in Ω_e,
$\bar{\phi}_i(x)$ denote the local interpolating functions of the master element.



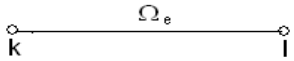
eg. For linear interpolation (two nodes in each element)

$$\begin{aligned} \bar{\phi}_1(\xi) &= \tfrac{(\xi-\xi_2)}{(\xi_1-\xi_2)} = \tfrac{\xi-1}{-1-1} = \tfrac{1}{2}(1-\xi) \\ \bar{\phi}_2(\xi) &= \tfrac{(\xi-\xi_1)}{(\xi_2-\xi_1)} = \tfrac{\xi+1}{1+1} = \tfrac{1}{2}(1+\xi) \end{aligned} \tag{3.12}$$

Remarks: As the transformation (3.10) is linear, a polynomial of degree k in the ξ-system will be transformed to a polynomial of the same degree k in the x-system.

Calculation of Element Contributions

Having selected an approximate set of shape functions, we now come to a crucial step in the analysis, i.e., the calculation of element matrices and vectors.



Consider $\Omega_e(x_k, x_l)$

$$k_{ij}^e = \int_{x_k}^{x_l} \bar{\phi}_i' \bar{\phi}_j' \, dx, \qquad f_i^e = \int_{x_k}^{x_l} 2\bar{\phi}_i \, dx.$$

Using the following coordinate transformation

$$\xi = \frac{2x - (x_k + x_l)}{x_l - x_k}, \quad d\xi = \frac{2}{x_l - x_k} dx = \frac{2}{h} dx$$

we have

$$k_{ij}^e = \frac{h}{2} \int_{-1}^{1} \bar{\phi}_i' \bar{\phi}_j' \, d\xi, \quad f_i^e = \frac{h}{2} \int_{-1}^{1} 2\bar{\phi}_i \, d\xi.$$

Note:

$$\bar{\phi}_k = \tfrac{1}{2}(1 - \xi), \;\; \bar{\phi}_l = \tfrac{1}{2}(1 + \xi)$$
$$\phi_k' = \tfrac{d\bar{\phi}_k}{dx} = \tfrac{d\bar{\phi}_k}{d\xi} \tfrac{d\xi}{dx} = -\tfrac{1}{2}\left(\tfrac{2}{h}\right) = -\tfrac{1}{h}$$
$$\phi_l' = \tfrac{d\bar{\phi}_l}{dx} = \tfrac{d\bar{\phi}_l}{d\xi} \tfrac{d\xi}{dx} = +\tfrac{1}{2}\left(\tfrac{2}{h}\right) = \tfrac{1}{h}$$

Therefore, $k^e = \begin{bmatrix} k_{kk}^e & k_{kl}^e \\ k_{lk}^e & k_{ll}^e \end{bmatrix}$, $f^e = \begin{bmatrix} f_k^e \\ f_l^e \end{bmatrix}$ with

$$k_{kk}^e = \tfrac{h}{2} \textstyle\int_{-1}^{1} \bar{\phi}_k' \bar{\phi}_k' \, d\xi = \tfrac{h}{2} \int_{-1}^{1} \tfrac{1}{h^2} \, d\xi = \tfrac{1}{h}$$
$$k_{lk}^e = k_{kl}^e = \tfrac{h}{2} \textstyle\int_{-1}^{1} \bar{\phi}_k' \bar{\phi}_l' d\xi = \tfrac{h}{2} \int_{-1}^{1} (\tfrac{1}{-h})(\tfrac{1}{h}) \, d\xi = -\tfrac{1}{h}$$
$$k_{ll}^e = \tfrac{h}{2} \textstyle\int_{-1}^{1} \bar{\phi}_l' \bar{\phi}_l' \, d\xi = \tfrac{1}{h}$$
$$f_k^e = h \textstyle\int_{1}^{1} \tfrac{1}{2}(1 - \xi) \, d\xi = h$$

i.e. for $e = 1, 2, 3, 4$

$$K^e = \tfrac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$
$$F^e = \begin{bmatrix} F_k^e \\ F_l^e \end{bmatrix} = h \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Thus, we can obtain all the element matrices.

For $\Omega_1(x_1, x_2)$,

$$K^1 = \begin{bmatrix} k_{11}^1 & k_{12}^1 \\ k_{21}^1 & k_{22}^1 \end{bmatrix} = K^e, \quad F^1 = \begin{bmatrix} F_1^1 \\ F_2^1 \end{bmatrix} = F^e.$$

For $\Omega_2(x_2, x_3)$,

$$K^2 = \begin{bmatrix} k_{22}^2 & k_{23}^2 \\ k_{32}^2 & k_{33}^2 \end{bmatrix} = K^e, \quad F^2 = \begin{bmatrix} F_2^2 \\ F_3^2 \end{bmatrix} = F^e.$$

For $\Omega_3(x_3, x_4)$,

$$K^3 = \begin{bmatrix} k_{33}^3 & k_{34}^3 \\ k_{43}^3 & k_{44}^3 \end{bmatrix} = K^e, \quad F^3 = \begin{bmatrix} F_3^3 \\ F_4^3 \end{bmatrix} = F^e.$$

For $\Omega_4(x_4, x_5)$,

$$K^4 = \begin{bmatrix} k_{44}^4 & k_{45}^4 \\ k_{54}^4 & k_{55}^4 \end{bmatrix} = K^e, \quad F^4 = \begin{bmatrix} F_4^4 \\ F_5^4 \end{bmatrix} = F^e.$$

Construction of global matrices

To construct the global **K** and **F**

i) Expand each element quantity to N dimension, i.e.

$$\text{For } \Omega_1, \quad K_1 = \begin{bmatrix} K_{11}^1 & K_{12}^1 & 0 & 0 & 0 \\ K_{21}^1 & K_{22}^1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} F_1^1 \\ F_2^1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{For } \Omega_2, \quad K_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & K_{22}^2 & K_{23}^2 & 0 & 0 \\ 0 & K_{32}^2 & K_{33}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 \\ F_2^2 \\ F_3^2 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{For } \Omega_3, \quad K_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33}^3 & K_{34}^3 & 0 \\ 0 & 0 & K_{43}^3 & K_{44}^3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 \\ 0 \\ F_3^3 \\ F_4^3 \\ 0 \end{bmatrix}$$

$$\text{For } \Omega_4, \quad K_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{44}^4 & K_{45}^4 \\ 0 & 0 & 0 & K_{54}^4 & K_{55}^4 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ F_4^4 \\ F_5^4 \end{bmatrix}$$

(ii) Add the expanded element quantities to form the global matrices.

$$\mathbf{K} = \sum_{e=1}^{E} K^e = \begin{bmatrix} K_{11}^1 & K_{12}^1 & 0 & 0 & 0 \\ K_{21}^1 & K_{22}^1 + K_{22}^2 & K_{23}^2 & 0 & 0 \\ 0 & K_{32}^2 & K_{33}^2 + K_{33}^3 & K_{34}^3 & 0 \\ 0 & 0 & K_{43}^3 & K_{44}^3 + K_{44}^4 & K_{45}^4 \\ 0 & 0 & 0 & K_{54}^4 & K_{55}^4 \end{bmatrix},$$

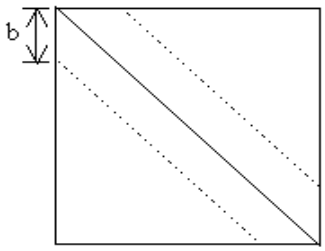
$$\mathbf{F} = \begin{bmatrix} F_1^1 \\ F_2^1 + F_2^2 \\ F_3^2 + F_3^3 \\ F_4^3 + F_4^4 \\ F_5^4 \end{bmatrix}.$$

Remarks:

1) Consider a typical entry K_{ij},

Contributions to this entry are only from those elements containing both nodes i and j.

2) The system matrix **K** has its nonzero terms clustered about its main diagonal while locations distant from the diagonal contain zero terms. The coefficient matrix is said to be banded as well as sparse. From point 1), the bandwidth b depends on the maximum difference of node number in each of the elements. If an efficient numbering scheme is used, the bandwidth can be minimized.



Boundary Conditions

Now the system of equations obtained so far is

$$\begin{bmatrix} K_{11} & K_{12} & & & \\ K_{21} & K_{22} & K_{23} & & \\ & K_{32} & K_{33} & K_{34} & \\ & & K_{43} & K_{44} & K_{45} \\ & & & K_{54} & K_{55} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 + \sigma(b)\phi(b) \end{bmatrix}. \tag{3.13}$$

Next, we need to impose the boundary conditions on the above system.

(i) Dirichlet boundary condition (also named essential boundary condition in the finite element method)

$$u(a) = u_1 = \hat{u}_1$$

– As u_1 is known, we move all known quantities $K_{i1}u_1$ in (3.13) to the right

hand side, thus

$$\begin{bmatrix} 0 & K_{12} & 0 & 0 & 0 \\ 0 & K_{22} & K_{23} & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} \\ 0 & 0 & 0 & K_{54} & K_{55} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} F_1 - K_{11}\hat{u}_1 \\ F_2 - K_{21}\hat{u}_1 \\ F_3 - K_{31}\hat{u}_1 \\ F_4 - K_{41}\hat{u}_1 \\ F_5 - K_{51}\hat{u}_1 + \sigma(b) \end{bmatrix}.$$

– In the variational statement, the test function $v(x)$ is required to satisfy $v(a) = 0$. However, the 1st equation of the system (3.13) is obtained by

$$(u_N, \phi_1) = (f, \phi_1) - \sigma\phi_1 \,|_a^b = \, 0$$

As $\phi_1(a) = 1 \neq 0$, $\phi_1(x)$ is not from the class of admissible test functions, $\phi_1(x) \notin H^1_{oh}$ and we should discard this equation and hence the system of equations becomes

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & K_{22} & K_{23} & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} \\ 0 & 0 & 0 & K_{54} & K_{55} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 \\ F_2 - K_{21}\hat{u}_1 \\ F_3 - K_{31}\hat{u}_1 \\ F_4 - K_{41}\hat{u}_1 \\ F_5 - K_{51}\hat{u}_1 + \sigma(b) \end{bmatrix}.$$

– Finally, we can either delete the 1st equation to yield an 4×4 system or add equation $u_1 = \hat{u}_1$ into the system to obtain

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & K_{22} & K_{23} & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} \\ 0 & 0 & 0 & K_{54} & K_{55} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} \hat{u}_1 \\ F_2 - K_{21}\hat{u}_1 \\ F_3 - K_{31}\hat{u}_1 \\ F_4 - K_{41}\hat{u}_1 \\ F_5 - K_{51}\hat{u}_1 + \sigma(b) \end{bmatrix}. \quad (3.14)$$

(ii) General natural boundary condition

$$k\frac{du(b)}{dx} = -p_b(u(b) - u_\infty) = \sigma(b).$$

The above natural boundary condition has been brought into the variational statement and consequently the 5th equation of (3.14) is

$$\begin{aligned} & K_{54}u_4 + K_{55}u_5 = F_5 - K_{51}\hat{u}_1 - p_b u_5 + p_b u_\infty \\ \Rightarrow \quad & K_{54}u_4 + (K_{55} + p_b)u_5 = F_5 - K_{51}\hat{u}_1 + p_b u_\infty \end{aligned}$$

Therefore system (3.14) becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & K_{22} & K_{23} & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} \\ 0 & 0 & 0 & K_{54} & (K_{55} + p_b) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} \hat{u}_1 \\ F_2 - K_{21}\hat{u}_1 \\ F_3 - K_{31}\hat{u}_1 \\ F_4 - K_{41}\hat{u}_1 \\ F_5 - K_{51}\hat{u}_1 + p_b u_\infty \end{bmatrix}.$$

which can then be solved to find u_2, u_3, u_4 and u_5.

Error Estimates

Suppose that the actual solution u of our boundary value problem has the property that its derivatives of order s are square-integrable on Ω, but those of order $s+1$ and higher are not, s being an integer greater than unity. Further, suppose that we use shape functions that contain complete polynomial of degree $\leq k$ and a uniform mesh of elements of equal length h. Then the approximate error, measured in H_1-norm, can be shown to satisfy the asymptotic error estimate

$$\| u - u_n \|_1 \leq ch^\mu \qquad (\text{note } \| v \|_1 = \left[\int_a^b (v'^2 + v^2) dx \right]^{1/2})$$

where c is a constant independent of h and $\mu = min(k, s)$.

Remarks: When the solution u is regular (i.e. $s > k$), then an improvement in the rate of convergence is obtained by increasing the degree k of the polynomial used in the approximate. However, for $s < k$, the rate of convergence is independent of k and no improvement is obtained by increasing k.

EXERCISE 3

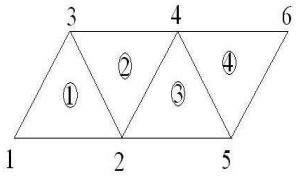
Question 1

Consider the FE mesh as shown.

(a) Describe the topology of the mesh.

(b) Write down the form of stiffness matrix for element 2

(c) Write down the form of the global stiffness matrix.



Question 2

Consider the boundary value problem defined by the differential equation

$$-u'' + b_0 u = 10\delta(x-1), \quad 0 < x < 2,$$

where b_0 is a constant, and the following sets of boundary conditions:

i) $u(0) = 1, \;\; u(2) = 3,$

ii) $u'(0) = 2, \;\; u'(2) = g_0 \;\; (g_0 \text{ is constant}),$

iii) $u'(0) + u(0) = 1, \;\; u(2) = 1.$

a) Using four elements of equal length and piecewise-linear basic functions, compute the global stiffness matrix and load vectors for this

general class of problems for the case is which $b_0 = 1$ and $b_0 = 0$. Give numerical values of all entries.

(b) Develop the reduced(non-singular) equations for problems (i) and (iii).

Chapter 4

Elliptic Boundary Value Problems

4.1 Introduction

The basic steps involved in solving a boundary value problem by finite element method are as follows:

1) Formulation of a variational statement with an appropriate space of admissible functions identified.

2) Construction of a finite element mesh and piecewise-polynomial basis functions defined on the mesh.

3) Construction of an approximation of the variational boundary value problem on a finite element subspace H^h. This generates a system of algebraic equations (or ordinary differential equations).

4) Solution of a system of equations.

In this chapter, we determine the finite element solution of boundary value problems of elliptic type,

$$\begin{array}{ll} -\nabla \cdot [k(x)\nabla u] + b(x)u \;=\; f(x) & x \in \Omega, \\ u(s) = \hat{u}(s) & s \in \partial\Omega_1, \\ -k(s)\frac{\partial u(s)}{\partial n} = p(s)[u(s) - \hat{u}(s)] = \hat{\sigma}(s) & s \in \partial\Omega_2, \end{array} \tag{4.1}$$

where ∇ is the gradient operator, $\nabla\cdot$ is the divergence operator and $\triangle = \nabla^2$ is the Laplace operator.

4.2 Variational Statement

To construct the variational statement of the boundary value problem (4.1), we define the residual function

$$r(x) = -\nabla \cdot [k\nabla u] + bu - f.$$

To test the residual over an arbitrary subregion, we multiply r by a sufficiently smooth test function v, integrate over Ω and set the resulting overall weighted residual to zero, we thus have

$$\int_\Omega [-\nabla \cdot (k\nabla u) + bu - f]v \; d\Omega \;=\; 0. \tag{4.2}$$

Then, as is typical in finite element work, we proceed to reduce the 2nd order terms to the 1st order by integration by parts. Using the product rule for differentiation

$$\begin{array}{ll} & \nabla \cdot (vk\nabla u) = k\nabla u \cdot \nabla v + v\nabla \cdot (k\nabla u) \\ \Rightarrow & v\nabla \cdot (k\nabla u) = \nabla \cdot (vk\nabla u) - k\nabla u \cdot \nabla v, \end{array} \tag{4.3}$$

we have from (4.2)

$$\int_\Omega [k\nabla u \cdot \nabla v - \nabla \cdot (vk\nabla u) + buv - fv] d\Omega = 0. \tag{4.4}$$

From the divergence theorem

$$\int_\Omega \nabla \cdot (vk\nabla u)\, d\Omega \;=\; \int_{\partial\Omega} vk\nabla u \cdot n\, ds \;=\; \int_{\partial\Omega} vk\frac{\partial u}{\partial n}\, ds, \tag{4.5}$$

equation (4.4) becomes

$$\int_\Omega [k\nabla u \cdot \nabla v + buv - fv]\, d\Omega - \int_{\partial\Omega} k\frac{\partial u}{\partial n} v\, ds \;=\; 0. \tag{4.6}$$

Choosing $v(x)$ such that $v(s) = 0$ on $\partial\Omega_1$ and using the boundary condition $(4.1)_3$, we obtain

$$\int_\Omega [k\nabla u \cdot \nabla v + buv - fv]\, d\Omega + \int_{\partial\Omega_2} puv\, ds - \int_{\partial\Omega_2} p\hat{u}v\, ds \;=\; 0. \tag{4.7}$$

To specify the appropriate class of admissible functions for problem (4.7), we examine the integrals in (4.7) and observe that the area integrals are well defined whenever u and v and their 1st order partial derivatives are smooth enough to be square-integrable over Ω. Thus, we need to choose u and v from $H^1(\Omega)$.

Hence, our variational boundary value problem can now be stated concisely in the following form:

Find $u \in H^1(\Omega)$ such that $u = \hat{u}$ on $\partial\Omega_1$ and

$$a(u, v) = L(v) \quad \forall v \in H^1(\Omega), \tag{4.8}$$

where $H_0^1 \;=\; \{v : v \in H^1 \text{ and } v = 0 \text{ on } \partial\Omega_1\}$,
$a(u, v) = \int_\Omega (k\nabla u \cdot \nabla v + buv)\, d\Omega + \int_{\partial\Omega_2} puv\, ds$ is a bilinear form of u and v,
$L(v) = \int_{\partial\Omega_2} p\hat{u}v\, ds + \int_\Omega fv\, d\Omega$ is a linear form of v.

Remarks: Natural boundary conditions enter implicitly in the variational statement, while the essential boundary conditions enter the problem in the definition of the class of admissible functions.

4.3 The Galerkin Approximation

A Galerkin approximation of (4.8) is obtained by posing the variational problem on a finite-dimensional subspace H^h of the space of admissible functions. Specifically, we

$$\begin{array}{ll} \text{seek} \quad u_h \in H_h^1 \quad \text{such that} \quad u_h(s) = \hat{u} \text{ on } \partial\Omega_1 \quad \text{and} \\ \qquad a(u_h, v_h) = L(v_h) \qquad\qquad \forall v_h \in H_{0h}^1 \end{array} \tag{4.9}$$

Let $\{\phi_i(x)\}_{i=1}^N$ be the basis functions of H_h^1, then

$$u_h(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x), \quad v_h(x) = \sum_{i=1}^{N} \beta_i \phi_i(x). \tag{4.10}$$

Substituting (4.10) into (4.9) yields

$$\begin{array}{ll} & \sum_{i=1}^N a(u_h, \phi_i)\beta_i = \sum_{i=1}^N L(\phi_i)\beta_i \qquad \forall \beta_i \\ \Rightarrow & a(u_h, \phi_i) = L(\phi_i), \quad (i = 1, 2, \ldots, N) \end{array} \tag{4.11}$$

Substituting (4.10) into (4.11) yields

$$\begin{array}{ll} & \sum_{i=1}^N a(\phi_i, \phi_j)\alpha_j = L(\phi_i), \quad (i = 1, 2, \ldots, N) \\ \Rightarrow & \mathbf{A}\alpha = \mathbf{F}, \end{array} \tag{4.12}$$

where $\mathbf{A} = (a_{ij})$ is an $N \times N$ matrix with $a_{ij} = a(\phi_i, \phi_j)$,

$\mathbf{F} = (F_i) \in \mathcal{R}^N$ with $F_i = L(\phi_i)$ and

$\alpha = (\alpha_i) \in \mathcal{R}^N$.

Therefore, the Galerkin approximation u_h of the solution u is of the form

$$u_h(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x), \tag{4.13}$$

where $\alpha \in \mathcal{R}^n$ is determined by (4.12) and

$\{\phi_j(x)\}_{j=1}^N$ are basis functions of H_h^1.

4.4 The Finite Element Interpolation

The finite element method provides a general and systematic technique for constructing the basis functions ϕ_i.

Consider an open bounded domain Ω in $\mathcal{R}^N$ with boundary $\partial\Omega$. Let $u \in C^m(\bar{\Omega})$ where $\bar{\Omega}$ is the closure of Ω, then the construction of a finite element interpolation of $u(\phi_i)$ can be accomplished by the following steps.

1) Partitioning of $\bar{\Omega}$

We replace $\bar{\Omega}$ by a collection $\bar{\Omega}_h$ of simple domain (element) $\bar{\Omega}_e$ such that

a) $\bar{\Omega}_h = \cup_{e=1}^{E} \bar{\Omega}_e$

b) $\bar{\Omega}_e \cap \bar{\Omega}_f = \phi$ for distinct $\bar{\Omega}_e$ and $\bar{\Omega}_f \in \bar{\Omega}_h$

c) every $\bar{\Omega}_e$ is closed and consists of a non-empty interior Ω_e and a boundary $\partial\Omega_e$.

2) Local Interpolation Over $\bar{\Omega}_e$-Local Basis ϕ_i^e

Over each $\bar{\Omega}_e$, we choose N_e nodes where the values of u and u_i^e are to be used as basic unknowns. Then we construct local interpolation function $\{\phi_i^e(\mathbf{x})\}_{i=1}^{N_e}$ such that the restriction of u_h to $\bar{\Omega}_e$ is

$$u_h^e(\mathbf{x}) = \sum_{i=1}^{N_e} u_i^e \phi_i^e.$$

The form of $\phi_i^e(\mathbf{x})$ for various type of elements will be studied in detail later.

3) Assembly of Global Basis Functions ϕ_i

Suppose there are N nodes in the finite element mesh, then there will be N global basis functions, each corresponding to one node (dominant node).

The global basis functions ϕ_i can be generated by patching together those local shape functions ϕ_i^e defined over $\bar{\Omega}_e$ which contain the node i. For example, suppose node i in the finite element mesh is shared by M elements. Then the local shape functions for point i corresponding to each of these elements are combined to form the global ϕ_i which satisfies

- the proper inter-element continuity
- $\phi_i(x_j) = \delta_{ij}$
- $\phi_i(\mathbf{x})$ is non-zero only over the particular patch of the M elements meeting at node i.

Thus, we can generate N linearly independent functions $\{\phi_i(x)\}_{i=1}^N$ which form a basis of an $N - \texttt{dimension}$ function space.

4.4.1 Triangular Elements

(1) Linear 3-point triangular elements

Approximate $u(x)$ over the element Ω_e by

$$u_h^e(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y, \quad \forall (x, y) \in \Omega_e \tag{4.14}$$

which determines a plane surface. Thus the use of linear interpolation on a triangular element will result in the approximation of a given smooth surface $v(x, y)$ by a plane as shown.

By evaluating (4.14) at each node, we have

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

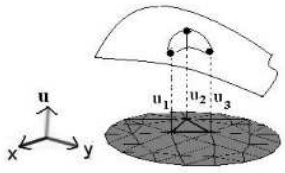


Figure 4.1: Approximate function $u_h^e(i),\ i = 1, 2, 3$

or $\quad u^e \ = \ p(x_i)\alpha$

$\Rightarrow \quad \alpha \ = \ p^{-1}(x_i)u^e$

Therefore $\quad u_h^e(x, y) = [1,\ x,\ y]p^{-1}(x_i)u^e$

which can be rearranged to yield

$$u_h^e(x, y) \ = \ u_1\phi_1^e \ + \ u_2\phi_2^e \ + \ u_3\phi_3^e \tag{4.15}$$

with element shape functions being

$$\begin{cases} \phi_1^e(x, y) = \frac{1}{2A_e}[(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y] \\ \phi_2^e(x, y) = \frac{1}{2A_e}[(x_3y_1 - x_1y_3) + (y_3 - y_1)x + (x_1 - x_3)y] \\ \phi_3^e(x, y) = \frac{1}{2A_e}[(x_1y_2 - x_2y_1) + (y_1 - y_2)x + (x_2 - x_1)y] \end{cases} \tag{4.16}$$

and $A_e = \frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$ is area of an element.

Remarks: The global piecewise basis function $\phi_i(x, y)$ constructed using the method described before is continuous across inter-element boundaries and, therefore, over Ω_h their 1st order partial derivatives are step functions and, hence, are square-integrable.

(2) Higher Order Triangular Element

Let us first display the terms appearing in polynomials of various degrees in two variables in the form as shown in Figure 4.2.

The above triangular array is called Pascal's triangle.

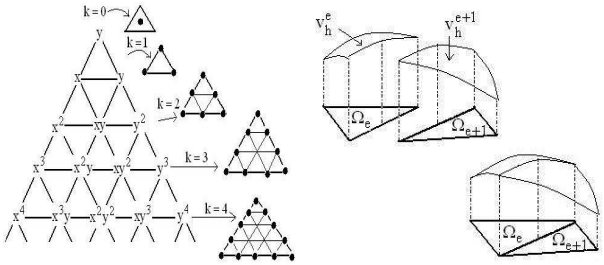


Figure 4.2: Pascal's triangle

Remarks:

1) A complete polynomial of degree k in x and y has $\frac{1}{2}(k+1)(k+2)$ terms. Thus, a polynomial of degree k in x and y can be uniquely determined by specifying its value at $\frac{1}{2}(k+1)(k+2)$ points in the plane.

2) The location of entries in Pascal's triangle can be used as the location of nodal points in triangular elements as shown in Figure 4.2.

3) The elements using the Pascal's triangle produce, for polynomial of degree > 0, basis functions that are continuous over the domain and, therefore, have square integrable 1st order partial derivatives.

Consider, for example, two adjacent six-node triangles Ω_e and Ω_{e+1} in the mesh. The local interpolation v_h^e and v_h^{e+1} are quadratic polynomials that must coincide at the 3 nodal points common to each element. However, the specification of 3 values of a quadratic in 1-D uniquely determines that quadratic. Hence, v_h^e and v_h^{e+1} will coincide everywhere on the common boundary of the 2 elements, and v_h will, therefore, be continuous across this boundary, as shown in Fig. 4.2.

4.4.2 Rectangular Elements

By taking the product of a set of polynomials in x with a set of polynomials in y, shape functions for a variety of rectangular elements can be obtained.

(1) Bilinear polynomials

The product of $(1, x)$ and $(1, y)$ produces a matrix

$$\begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 & y \end{bmatrix} = \begin{bmatrix} 1 & y \\ x & xy \end{bmatrix}. \tag{4.17}$$

A bilinear local interpolant can then be obtained by forming a linear combination of all the four terms in the matrix, i.e

$$v_h^e(x, y) = a_1 + a_2 x + a_2 y + a_4 xy. \tag{4.18}$$

Remarks:

1) if we choose four nodes in the rectangular element, one at each corner, the function v_h^e can then be uniquely determined by specifying its values at those nodes. We can rearrange above as

$$v_h^e(x, y) = \sum_1^4 u_i \phi_i^e(x, y),$$

where $\phi_i^e(x, y)$ denote the element basis functions.

2) Along the side x = constant or y = constant, v_h^e is linear in x or y. Thus, if two such elements Ω_e and Ω_{e+1} have a common side in the mesh, then $v_h^e(x, y)$ and $v_h^{e+1}(x, y)$ will coincide on the common side and therefore $v_h[\phi_i(x, y)]$ are continuous over Ω_h. Thus, the shape functions obtained using (4.18) will produce basis functions ϕ_i which have square-integrable 1st order derivatives over Ω_h.

(2) Higher Order Rectangular Elements

By considering tensor products of polynomials of higher degree, element shape functions can be constructed which contain polynomials of any desired degree and which lead to basis functions that are continuous throughout Ω_h.

eg. For a biquadratic local interpolation, we firstly find the matrix from

$$\begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} \begin{bmatrix} 1 & y & y^2 \end{bmatrix} = \begin{bmatrix} 1 & y & y^2 \\ x & xy & xy^2 \\ x^2 & x^2y & x^2y \end{bmatrix}$$

Then the biquadratic local interpolant v_h^e is obtained by forming a linear combination of all the nine terms in the matrix. To completely determine the interpolant, construct a rectangular element with nine nodes as shown.

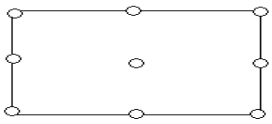


Figure 4.3: A rectangular element with nine nodes

4.4.3 Interpolation Error

Let g be a smooth function given,

g_n be the finite element representation which contains a complete polynomial of degree k.

If all partial derivatives of g of order $k+1$ are bounded in the domain Ω_h, then the interpolation error satisfies

$$\|g - g_h\|_{\infty,\Omega_e} = \max_{(x,y) \in \Omega_e} |g(x,y) - g_h(x,y)| \leq C h_e^{k+1}$$

where C is a positive constant and h_e is the *diameter* of Ω_e ; that is, h_e is the largest distance between any two points in Ω_e. Similarly

$$\left\| \frac{\partial g}{\partial x} - \frac{\partial g_h}{\partial x} \right\|_{\infty,\Omega_e} \leq C_1 h_e^k, \quad \left\| \frac{\partial g}{\partial y} - \frac{\partial g_h}{\partial y} \right\|_{\infty,\Omega_e} \leq C_2 h_e^k.$$

4.5 Finite Element Approximation

Assembly of Finite Element Equations

Return to the problem described in section 5.3, as we choose $\phi_i(x_j) = \delta_{ij}$, our finite element approximation of u is

$$u_h(\mathbf{x}) = \sum_{j=1}^{N} u_j \phi_j(\mathbf{x}).$$

Thus, our problem now is:

Find $\mathbf{u} \in \mathcal{R}^N$ such that $u_i = \hat{u}$ on $\partial\Omega_1$ and

$$\mathbf{Au} \;=\; \mathbf{F}$$

for all ϕ_i such that $\phi_i = 0$ on $\partial\Omega_1$ and

$\mathbf{A} = (a_{ij})$ with $a_{ij} = a(\phi_i, \phi_j) = \int_\Omega (k\nabla\phi_i \cdot \nabla\phi_j + b\phi_i\phi_j) + \int_{\partial\Omega_2} p\phi_i\phi_j \; ds$

$\mathbf{F} = (F_i)$ with $F_i = L(\phi_i) = \int_\Omega f\phi_i \; d\Omega + \int_{\partial\Omega_2} p\hat{u}\phi_i \; ds.$

As $\phi_i(\mathbf{x})$ are defined piecewisely over each element Ω_e, we have

$$a_{ij} = \sum_{e=1}^{E} \int_{\Omega_e} (k\nabla\phi_i \cdot \nabla\phi_j + b\phi_i\phi_j) \; d\Omega + \sum_{e=1}^{E} \int_{\partial\Omega_{2e}} p\phi_i\phi_j \; ds$$

$$F_i = \sum_{e=1}^{E} \left\{ \int_{\Omega_e} f\phi_i d\Omega + \int_{\partial\Omega_{2e}} p\hat{u}\phi_i \; ds \right\}.$$

To assemble $\mathbf{A}$, loop over all elements to calculate a^e and successively add in the contributions from each a^e as follows :

Set $\mathbf{A}(i,j) = 0, \quad b(i) = 0, \quad i,j = 1,2,...N$

For $e = 1,2...E$

calculate a^e

`Set` $\mathbf{A}_{g(e,a)g(e,\beta)} = \mathbf{A}_{g(e,a)g(e,\beta)} + a^e_{\alpha\beta}$

$\mathbf{F}_{g(e,a)} = \mathbf{F}_{g(e,a)} + F^e_\alpha \quad \alpha,\beta = 1,2,...,$`Number of Nodes in`$\Omega_e$.

where $g(e,k)$ denotes the global node number of the k^{th} node of element e.

Boundary Condition

The boundary conditions are of two types : natural and essential conditions. The natural boundary conditions are brought into the variational statement through the boundary integral, which modifies the coefficient matrix $\mathbf{A}$ and the vector $\mathbf{F}$. By contrast, essential boundary conditions are not enforced through the boundary integral. There are used to define the space H of admissible functions. However, in the construction of $\mathbf{A}$ and $\mathbf{F}$, the restriction on H due to the boundary conditions has not been taken into account. Thus, they must be enforced by overriding the main finite element equations at the boundary concerned.

Suppose at point $\ell, u_\ell = \hat{u}$ and the assembled system is

$$\begin{bmatrix} a_{11} & \cdots & a_{1\ell} & \cdots & a_{1N} \\ \vdots & & \vdots & & \vdots \\ a_{\ell 1} & \cdots & a_{\ell\ell} & \cdots & a_{\ell N} \\ \vdots & & \vdots & & \vdots \\ a_{N1} & \cdots & a_{N\ell} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_\ell \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_\ell \\ \vdots \\ f_N \end{bmatrix}$$

We impose the boundary condition $u_\ell = \hat{u}$ by performing the following steps:

1) Move the known values to the right hand side

$$\left[\begin{array}{ccc|c|ccc} a_1 1 & \cdots & & 0 & & \cdots & a_{1N} \\ \vdots & & & \vdots & & & \vdots \\ a_{\ell 1} & \cdots & & 0 & & \cdots & a_{\ell N} \\ \vdots & & & \vdots & & & \vdots \\ a_{N1} & \cdots & & 0 & & \cdots & a_{NN} \end{array}\right] \begin{bmatrix} u_1 \\ \vdots \\ u_\ell \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 & - & a_{1\ell}\hat{u} \\ & & \vdots \\ f_\ell & - & a_{\ell\ell}\hat{u} \\ & & \vdots \\ f_N & - & a_{N\ell}\hat{u} \end{bmatrix}$$

2) Impose the restriction $\phi_\ell = 0$ on the system.

Noting that $a_{\ell j} = a(\phi_\ell, \phi_j) = 0,\ f_\ell = L(\phi_\ell) = 0$, we have

$$\left[\begin{array}{ccccc} a_1 1 & \cdots & 0 & \cdots & a_{1N} \\ & & \vdots & & \\ \hline 0 & \cdots & \vdots & \cdots & 0 \\ \hline & & \vdots & & \\ a_{N1} & & 0 & & a_{NN} \end{array}\right] \left[\begin{array}{c} u_1 \\ \vdots \\ u_\ell \\ \vdots \\ u_N \end{array}\right] = \left[\begin{array}{ccc} f_1 & - & a_{1\ell}\hat{u} \\ f_2 & - & a_{2\ell}\hat{u} \\ & & \vdots \\ & & 0 \\ & & \vdots \\ f_N & - & a_{N\ell}\hat{u} \end{array}\right]$$

This set of equations is rank deficient and need to be modified by one of the following methods.

- Combine with the Dirichlet Condition $u_\ell = \hat{u}_\ell$ to yield

$$\left[\begin{array}{ccc|c|ccc} a_{11} & \cdots & & 0 & & \cdots & a_{1N} \\ \vdots & & & \vdots & & & \\ \hline 0 & \vdots & & 1 & & \vdots & 0 \\ \hline \vdots & & & \vdots & & & \\ a_{N1} & \cdots & & 0 & & \cdots & a_{NN} \end{array}\right] \left[\begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_\ell \\ \vdots \\ u_N \end{array}\right] = \left[\begin{array}{ccc} f_1 & - & a_{1\ell}\hat{u} \\ & & \vdots \\ & & \hat{u} \\ & & \vdots \\ f_N & - & a_{N\ell}\hat{u} \end{array}\right]$$

- Delete row ℓ and column ℓ to form an $(N-1) \times (N-1)$ system.

Example 4.1 Consider

$$\begin{cases} -\Delta(x,y) = f(x,y) & in\ \Omega \\ u = 0 & on\ \Gamma_{41} \\ \frac{\partial u}{\partial n} = 0 & on\ \Gamma_{12}, \Gamma_{25}, \Gamma_{67}, and \Gamma_{74} \\ \frac{\partial u}{\partial n} + \beta u = \gamma & on\ \Gamma_{56} \end{cases}$$

In this case $\partial\Omega_1 = \Gamma_{41}$

$$\partial\Omega_2 = \Gamma_{12} \cup \Gamma_{25} \cup \Gamma_{67} \cup \Gamma_{74} \cup \Gamma_{56}$$

Our analysis of this problem proceeds as follows:

- Partition Ω into six triangular elements.

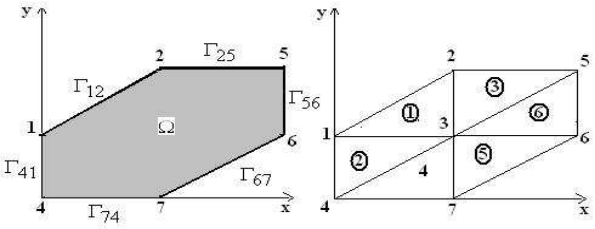


Figure 4.4: Computation domain Ω

- Compute the element matrices a^e and f^e $(e = 1, 2, ..., 6)$

$$a^e = \begin{bmatrix} a^e_{11} & a^e_{12} & a^e_{13} \\ a^e_{21} & a^e_{22} & a^e_{23} \\ a^e_{31} & a^e_{32} & a^e_{33} \end{bmatrix}, f^e = \begin{bmatrix} f^e_1 \\ f^e_2 \\ f^e_3 \end{bmatrix}$$

- Assemble the element matrices to form the global matrix using the following topology:

ele	node 1	2	3
1	1	2	3
2	1	3	4
3	2	5	3
4	3	4	7
5	3	6	7
6	3	5	6

Hence, we have

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & 0 & K_{25} & 0 & 0 \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} \\ K_{41} & 0 & K_{43} & K_{44} & 0 & 0 & K_{47} \\ 0 & K_{52} & K_{53} & 0 & K_{55}+K_b & K_{56} & 0 \\ 0 & 0 & K_{63} & 0 & K_{65} & K_{66}+K_b & K_{67} \\ 0 & 0 & K_{73} & K_{74} & 0 & K_{76} & K_{77} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5+F_b \\ F_6+F_b \\ F_7 \end{bmatrix}$$

$$
\begin{aligned}
K_{11} &= a^1_{11} + a^2_{11}, & K_{12} &= a^1_{12} \\
K_{13} &= a^1_{13} + a^2_{12}, & K_{14} &= a^2_{13} \\
K_{21} &= a^1_{21}, & K_{22} &= a^1_{22} + a^3_{11} \\
K_{23} &= a^1_{23} + a^3_{13}, & K_{25} &= a^3_{13} \\
K_{31} &= a^1_{31} + a^2_{21}, & K_{32} &= a^1_{32} + a^3_{31} \\
K_{33} &= \begin{matrix} a^1_{33} + a^2_{22} + a^3_{33} \\ + a^4_{11} + a^5_{11} + a^6_{11} \end{matrix}, & K_{34} &= a^2_{23} + a^4_{12} \\
K_{35} &= a^3_{32} + a^6_{12}, & K_{36} &= a^5_{12} + a^6_{13} \\
K_{37} &= a^4_{13} + a^5_{13}, & K_{41} &= a^2_{31} \\
K_{43} &= a^2_{32} + a^4_{21}, & K_{44} &= a^2_{33} + a^4_{22} \\
K_{47} &= a^4_{23}, & K_{52} &= a^3_{21} \\
K_{53} &= a^3_{23} + a^6_{21}, & K_{55} &= a^3_{22} + a^6_{22} \\
K_{63} &= a^5_{21} + a^6_{31}, & K_{66} &= a^5_{22} + a^6_{33} \\
K_{67} &= a^5_{23}, & K_{73} &= a^4_{31} + a^5_{31} \\
K_{74} &= a^4_{32}, & K_{76} &= a^5_{32} \\
K_{77} &= a^4_{33} + a^5_{33}
\end{aligned}
$$

- Impose the essential boundary condition.

EXERCISE 4

Question 1

Consider the boundary value problem

$$-\Delta u + \lambda u = f \text{ in } \Omega$$

$$u = 0 \text{ on } \partial\Omega.$$

Develop a variational statement of the problem using integration by part (or Green's formula).

Question 2

Consider a rectangular element with four nodes, one at each corner. In view of our criteria for acceptable finite element basis functions, why is the following choice of a local test function representation unacceptable ?

$$v_h^e(x, y) = a_1 + a_2 x + a_3 y + a_4 x^2$$

Here a_1, a_2, a_3 and a_4 are constants.

Question 3

Suppose that Ω_h is a square consisting of eight triangular elements of equal size. Describe by means of sketches, the global basis functions $\phi_i, i = 1, 2, ...9$ generated by piecewise-linear shape functions on each element.

Question 4

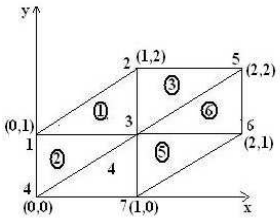
Furnish additional details for the example 5.1 as follows.

a) Suppose that all the elements in the mesh shown are equal isoscale triangles, the two equal sides being of length h. Derive the element stiffness matrix a^e and f^e for $f(x, y) = 1$.

b) Suppose the coordinates of the nodes in the mesh is as shown. Use the result in a) to calculate the element stiffness matrixes and load vectors for all six elements.

c) Construct the global matrices and load vector.

d) Impose the boundary condition to obtain the final system of equations.



Chapter 5

Parabolic Boundary Value Problems

In this chapter, we consider the solution of linear parabolic problems (diffusion problems) governed by the parabolic partial differential equation $(5.1)_1$ with boundary condition $(5.1)_2$ and initial condition $(5.1)_3$ as follows:

$$\begin{array}{lll} & & u_t - \nabla.(k\nabla u) + bu = f \quad \text{in } \Omega \times \mathrm{I} \\ \text{subj.} & \text{B.C.} & \frac{\partial u}{\partial n} + \alpha u = \gamma \quad \text{on } \partial\Omega \times \mathrm{I} \\ & \text{I.C.} & u(\mathbf{x}, 0) = \hat{u}(\mathbf{x}) \quad \text{in } \Omega \\ & \text{where I} & : [0, T] \end{array} \tag{5.1}$$

5.1 Semi-discretization in space

Variational statement

Multiplying (5.1), for a given t , by $v \in H^1$, then integrating over Ω and using Green's theorem, we get

$$\int_\Omega u_t v \; d\Omega + \int_\Omega (k\nabla u \cdot \nabla v + buv) \; d\Omega + \int_{\partial\Omega} k\alpha uv \; ds = \int_\Omega fv \; d\Omega + \int_{\partial\Omega} k\gamma v \; ds. \tag{5.2}$$

Thus, we are led to the following variational problem:

`Find` $u = u(\mathbf{x}, t) \in H^1(\Omega)$ `such that for every` $t \in I$

$$(u_t, v) + a(u, v) = L(v) \quad \forall\, v \in H^1(\Omega) \tag{5.3}$$

$$u(\mathbf{x}, 0) = \hat{u}(\mathbf{x}) \tag{5.4}$$

`where` $(\cdot, \cdot)$ = inner product

$a(u, v) = \int_\Omega (k\nabla u \cdot \nabla v + buv) d\Omega + \int_{\partial\Omega} k\alpha uv \; ds.$

$L(v) = \int_\Omega fv \; d\Omega + \int_{\partial\Omega} k\gamma v \; ds.$

Finite Element Approximation

Let H_h^1 be a finite dimensional subspace of H^1 with basis functions $\{\phi_1, \phi_2, ...\phi_n\}$. Then, the variational problem is approximated by :

Find $u_h(\mathbf{x}, t) \in H_h^1$ such that $u_h(\mathbf{x}, 0) = \hat{u}(\mathbf{x})$ and

$$\left(\frac{\partial u_h}{\partial t}, v_h\right) + a(u_h, v_h) = L(v_h) \quad \forall\, v_h \in H_h^1. \tag{5.5}$$

In the usual way, we introduce a discretization of Ω as a union of elements Ω_e, i.e. $\Omega \to \bigcup_{e=1}^{E} \Omega_e$ and approximate $u(\mathbf{x}, t)$ at t by.

$$u_h(\mathbf{x}, t) = \sum_{j=1}^{n} u_j\,(t)\varphi_j\,(\mathbf{x}) \tag{5.6}$$

From (5.5) and (5.6), by using the usual finite element formulation, we obtain

$$\begin{array}{c} \mathbf{M}\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} = \mathbf{F} \\ \mathbf{u}(0) = \hat{\mathbf{u}} \end{array} \tag{5.7}$$

where $\mathbf{M} = (m_{ij})$ with $m_{ij} = (\varphi_i, \varphi_j) = \sum_{e=1}^{E} \int_{\Omega_e} \varphi_i \varphi_j d\Omega$

$\mathbf{A} = (a_{ij})$ with $a_{ij} = a(\varphi_i, \varphi_j) =$

$\sum_{e=1}^{E} \int_{\Omega_e} (k\nabla\varphi_i \cdot \nabla\varphi_j + b\varphi_i\varphi_j)\; d\Omega + \sum_{e=1}^{\partial E} \int_{\partial\Omega_e} k\alpha\varphi_i\varphi_j \; ds$

$\mathbf{F} = (f_i)$ with $f_i = L(\varphi_i)$

Consistency and Stability

Definition: consistency

By **consistency** we mean that the numerical scheme converges to the correct governing equation as the mesh size and the time stepping independently go to zero.

Definition: stability

By **stability** we generally mean that a scheme is stable if the error measured in an appropriate norm does not become unbounded as time increases.

Error estimate theorem

Let u be the solution of (5.1) with $k = 1,\ b = f = 0,\ u = 0$ on $\partial\Omega$ and let u_n be the corresponding finite element solution using (5.7). Then $\exists$ a constant c such that

$$\max_{t\in I} \| u(t) - u_n(t) \| \leq c\left(1 + \left|\log\frac{T}{h^2}\right|\right) \max_{t\in I} h^2 \|u(t)\|_{H^2(\Omega)}. \tag{5.8}$$

Basic stability inequality (for $f = 0,\ u = 0$ on $\partial\Omega$).

Let $u_h(t)$ satisfy (5.7), then

$$\|u_h(t)\| \leq \|u_h(0)\| \leq \|\hat{u}\|,\ t \in I$$

Proof

For $u = 0$ on $\partial\Omega$, (5.5) becomes (on taking $v_h = u_h$)

$$\begin{aligned} &(\dot{u}_h, u_h) + a(u_h, u_h) = 0 \\ &\tfrac{1}{2}\ \tfrac{d}{dt}\|u_h\|^2 + a(u_h, u_h) = 0 \\ &\|u_h\|^2 + 2\int_0^t a(u_h(s), u_h(s))ds = \|u_h(0)\|^2 \end{aligned}$$

Therefore, $\|u_h\| \leq \|\hat{u}\|$

Note: we have used the notation $\|w\| = (w, w)^{1/2} = (\int w^2 d\Omega)^2$.

5.2 Time Differencing

We now consider the numerical technique to solve the following system of ordinary differential equations.

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} \;=\; \mathbf{F} \tag{5.9}$$

(i) Forward Difference Scheme

$$\texttt{Let} \quad \frac{d\mathbf{u}}{dt}(t) = \frac{\mathbf{u}(t+\Delta t_r) - \mathbf{u}(t)}{\Delta t} (or \frac{d\mathbf{u}_r}{dt} = \frac{\mathbf{u}_{r+1} - \mathbf{u}_r}{\Delta t_r}) \tag{5.10}$$

and use forward difference with O(Δt) accuracy, then (5.9) becomes

$$\mathbf{M}\ \mathbf{u}_{r+1} = (\mathbf{M} - \Delta t_r \mathbf{A})\mathbf{u}_r + \Delta t_r \mathbf{F}_r \tag{5.11}$$

where $\sum_{r=1}^{n} \Delta t_r = T$

Hence, starting with $\mathbf{u}_0$ at $r = 0$, we can generate a sequence of solutions $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n$ corresponding to $t_1, t_2, ..., T$.

Remarks:

1) If k, b and α depend on time, then A is a function of time, so that in the forward difference scheme, $\mathbf{A}$ is replaced by $\mathbf{A}(t)$.

2) Finite element code for the equilibrium problem ($\mathbf{u}_t = 0$) in Chapter 4 can be modified to solve this FE system at each time step.

Program Structure

Loop over time steps $r = 0, 1, 2, ...N_t$

Loop over elements $e = 1, 2, ...N_e$

For each Ω_e, calculate $a^e, m^e, f^e, \&\ b_r^e = (m^e - \Delta t_r k^e)u_r^e$

Assemble m^e to M & b_r^e to b_r

Modify M & b_r to satisfy essential B.C.'s

Solve $Mu_{r+1} = b_r$

Stability

To analyze the stability of the forward difference scheme, we consider the system (5.9) with the initial solution $\mathbf{u}(0) = \hat{\mathbf{u}}$.

Suppose $\mathbf{e}(t) :=$ `error in` $\mathbf{u}(t)$ due to a small change in $\hat{\mathbf{u}}$, then

$$\mathbf{M}(\dot{\mathbf{u}} + \dot{\mathbf{e}}) + \mathbf{A(u+e)=F}. \tag{5.12}$$

(5.12)-(5.9) $\Rightarrow$ $\quad \mathbf{M}\dot{\mathbf{e}} + \mathbf{Ae} = \mathbf{0}$

$\Rightarrow$ $\quad \frac{d\mathbf{e}}{dt} = -\mathbf{M}^{-1}\mathbf{Ae}.$

Thus, using forward difference scheme,

$$\mathbf{e}_{r+1} = (\mathbf{I} - \Delta t_r \mathbf{M}^{-1}\mathbf{A})\mathbf{e}_r = R_r\mathbf{e}_r = \left(\Pi_{i=0}^{r} R_i\right)\mathbf{e}_0.$$

If $\Delta t_r = \Delta t$ (constant), then

$$\mathbf{e}_{r+1} = R^{r+1}\mathbf{e}_0\ ,\ (r = 0, 1, ...\frac{T}{\Delta t}). \tag{5.13}$$

Let $\quad \lambda_i,\ \{\mathbf{w}_i\}_{i=1}^N$ be eigenvalues and eigenvectors of $\mathbf{M}^{-1}\mathbf{A}$.

Then $\quad \mathbf{M}^{-1}\mathbf{A}\mathbf{w}_i\ =\ \lambda_i\mathbf{w}_i$

$\rightarrow$ $\quad \Delta t\mathbf{M}^{-1}\mathbf{A}\mathbf{w}_i\ =\ -\Delta t\lambda_i\mathbf{w}_i$

$$\mathbf{I}\mathbf{w}_i - \Delta t\mathbf{M}^{-1}\mathbf{A}\mathbf{w}_i = (1 - \Delta t\lambda_i)\mathbf{w}_i$$

$$(\mathbf{I} - \Delta t\mathbf{M}^{-1}\mathbf{A})\mathbf{w}_i = (1 - \Delta t\lambda_i)\mathbf{w}_i$$

We can approximate the error at r=0 as $\mathbf{e}_0 = \sum_{i=1}^{N} \alpha_i \mathbf{w}_i$.

Hence, $R\mathbf{e}_0 = \sum_1^N \alpha_i (I - \Delta t\mathbf{M}^{-1}\mathbf{A})\mathbf{w}_i = \sum \alpha_i (1 - \lambda_i \Delta t)\mathbf{w}_i$

$$R^2\mathbf{e}_0 = \sum_1^N (1 - \lambda_i\Delta t)\alpha_i (I - \Delta t\mathbf{M}^{-1}\mathbf{A})\mathbf{w}_i = \sum_1^N (1 - \lambda_i\Delta t)^2 \alpha \mathbf{w}_i.$$

Therefore,

$$\mathbf{e}_{r+1} = R^{r+1}\mathbf{e}_0 = \sum_1^N (1 - \lambda_i\Delta t)^{r+1}\alpha_i\mathbf{w}_i \tag{5.14}$$

Remarks:

1) The error will not grow and the scheme is stable if

$$|1 - \lambda_i\Delta t| < 1, \ \texttt{i.e.}\ \Delta t < \frac{2}{\lambda_i} \ \ (i = 1, 2, ..., N), \tag{5.15}$$

2) The larger the value of λ_i , the greater the restriction on the time step.

3) The value of λ_i is related to the finite element mesh. For example, for linear element, from a study of the eigenvalue problem, the highest frequency for an operator of order $2m$ is $\lambda_m = \beta h^{-2m}$ for a constant β. In the diffusion problem considered, $m = 1$ and inequality (5.15) implies

$$\Delta t \leq \frac{2}{\beta}h^2 = ch^2 \tag{5.16}$$

(ii) Central and Backward Difference (Crank-Nicolson Method)

The forward difference extrapolation leads to the restriction on the time step size to ensure stability. Here, we derive a scheme with unconditional stability.

Crank-Nicolson Scheme

Let $\frac{d\mathbf{u}}{dt}(t+\frac{\Delta t}{2}) = \frac{\mathbf{u}(t+\Delta t)-\mathbf{u}(t)}{\Delta t}$

$\mathbf{u}(t+\frac{\Delta t}{2}) = \frac{1}{2}(\mathbf{u}(t)+\mathbf{u}(t+\Delta t))$

Then (5.9) becomes

$$(\mathbf{M}+\frac{\Delta t}{2}\mathbf{A})\mathbf{u}_{r+1} = (\mathbf{M}-\frac{\Delta t}{2}\mathbf{A})\mathbf{u}_r + \Delta t\mathbf{F}_{r+\frac{1}{2}} \tag{5.17}$$

Remarks: The only essential difference from the forward scheme lies in the actual form of the element matrix and vector contributions.

$$m^e + \frac{\Delta t}{2}a^e, \quad \texttt{and} \quad (m^e - \frac{\Delta t}{2}a^e)\mathbf{u}_r^e + \Delta t f^e_{r+\frac{1}{2}}.$$

Stability

We consider an initial error $\mathbf{e}_0$ and analyze the error growth in the recursion (5.17). Pre-multiplying (5.17) by $\mathbf{M}^{-1}$, we obtain

$$(I+\frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{A})\mathbf{e}_{r+1} = (I-\frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{A})\mathbf{e}_r, \tag{5.18}$$

$$\mathbf{e}_{r+1} = R_+^{-1}R_-\mathbf{e}_r = (R_+^{-1}R_-)^{r+1}\mathbf{e}_0, \tag{5.19}$$

where $R_\pm = \mathbf{I} \pm \frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{A}$.

Further, assume that $\mathbf{M}^{-1}\mathbf{A}$ has N linearly independent eigenvectors $\mathbf{w}_i$, then

$\mathbf{e}_0 = \sum_1^N \alpha_i\,\mathbf{w}_i$,

$R_\pm\mathbf{w}_i = (I \pm \frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{A})\mathbf{w}_i = (1 \pm \frac{\Delta t}{2}\lambda_i)\mathbf{w}_i$,

$R_+^{-1}\,\mathbf{w}_i = (1+\frac{\Delta t}{2}\lambda_i)^{-1}\mathbf{w}_i$.

Therefore,

$$\mathbf{e}_{r+1} = (R_+^{-1}R_-)^r\; R_+^{-1}(R_-\,\mathbf{e}_0) = (R_+^{-1}R_-)^r\;\sum R_+^{-1}(1-\frac{\Delta t}{2}\lambda_i)\alpha_i\mathbf{w}_i$$

$$= (R_+^{-1}R_-)^r \sum_{i=1}^{N} \frac{1-\frac{\Delta t}{2}\lambda_i}{1+\frac{\Delta t}{2}\lambda_i}\alpha_i \mathbf{w}_i \;=\; \sum_{i=1}^{N} \rho_i^{r+1}\alpha_i \mathbf{w}_i.$$

As the eigenvalues λ_i are all positive, $\rho_i = \frac{1-\frac{\Delta t}{2}\lambda_i}{1+\frac{\Delta t}{2}\lambda_i} \leq 1$. Consequently, the error will not grow and the scheme is stable.

Remarks:

1) If $\lambda_i < \frac{2}{\Delta t}$, then $\rho_i > 0$ and the error components decay monotonically;
   if $\lambda_i > \frac{2}{\Delta t}$, then $\rho_i < 0$ and the error components decay in an oscillatory manner from one step to the next. Therefore, we can define $\lambda^* = \frac{2}{\Delta t}$ as natural frequency.

2) The highest frequency depends inversely on the mesh size h with $\lambda_n = \beta h^{-2m}$ for a constant β. Accordingly, if the finite element mesh is repeatedly refined, inevitably when $h^{2m} < \beta\frac{\Delta t}{2}$, some of the higher order components enter and decaying oscillations appear. For m = 1 and linear element in our diffusion problem in one dimension, the oscillations in components occur when $\frac{\Delta t}{h^2} > \frac{2}{\beta}$, which is, incidentally, the stability limit of the previous forward scheme.

(iii) Backward difference scheme

Scheme : $(\mathbf{M} + \Delta t\mathbf{A})\,\mathbf{u}_{r+1} = \mathbf{M}\mathbf{u}_r + \Delta t\mathbf{F}_{r+1}$.

Using the procedure similar to that in (ii), it can be shown that the above scheme is

- O(Δt) accuracy,
- unconditionally stable,
- $\rho_i = (1 + \lambda_i \Delta t)^{-1}$.

EXERCISE 5

Question

Consider the convection-diffusion-problem

$$\frac{\partial u}{\partial t} - \mu \Delta u + \beta_1 \frac{\partial u}{\partial x_1} + \beta_2 \frac{\partial u}{\partial x_2} = f \quad \text{in } \Omega \times I$$

$$u \;=\; 0 \;\text{ on }\; \partial\Omega \times I$$

$$u(\mathbf{x}, 0) = u_0 \quad \text{on } \; \Omega$$

a) Find the variational statement of the problem.

b) Determine the finite element equation.

Chapter 6

Element Calculations

This chapter focuses on a general and systematic method for calculating element matrices and finite element programming.

6.1 Element Transformation

Calculation of element matrices in x, y coordinates is awkward as integration region is complex and limit of integration changes from element to element. If we can find a transformation

$$T_e \ : \ \begin{cases} x = x(\xi, \eta) \\ y = y(\xi, \eta) \end{cases}$$

which maps an arbitrarily chosen element e into a standard (master) element $\bar{\Omega}$, then the calculation of element matrices can be standardized using numerical quadrature.

(1) Master Element & Its Connection with Finite Element Mesh

The geometry of the master element is chosen as simple as possible, eg. the square as shown.

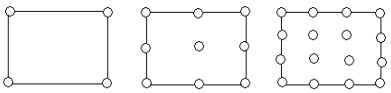


Figure 6.1: Square elements with 4 nodes (linear element), 9 nodes (quadratic element) and 16 nodes (cubic element)

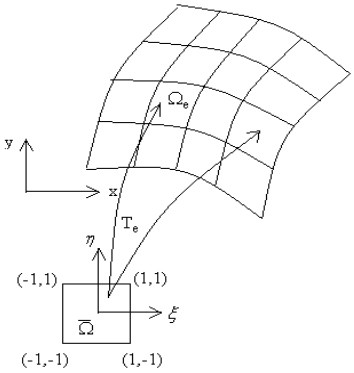


Figure 6.2: Element transformation T_e

- A point $P(\xi = \alpha, \eta = \beta)$ in the standard element $\bar{\Omega}$ is mapped into a point

$$P[x(\alpha, \beta), y(\alpha, \beta)]$$

  in local element Ω_e.

- A line $(\xi = \alpha)$ in $\bar{\Omega}$ is mapped into a curve

$$[x = x(\alpha, \eta), y = y(\alpha, \eta)]$$

  in the plane, which is called the curvilinear coordinate line $(\xi = \alpha)$.

- A finite element mesh can be viewed as a sequence of transformation $\{T_1,\ T_2, ...T_E\}$ of the fixed master element. Each element Ω_e is the image of the master element $\bar{\Omega}$ under a coordinate map T_e.

• All properties of a given type of elements (number and location of nodes, shape functions, stiffness and etc) can be prescribed for the fixed element $\bar{\Omega}$, and then carried to any Ω_e in the mesh by using the map T_e.

(2) Properties of Coordinate Transformation

Relations between dx, dy with $d\xi$ and $d\eta$

Suppose $x(\xi,\eta)$ and $y(\xi,\eta)$ are continuously differentiable, then

$$dx = \frac{\partial x}{\partial \xi}d\xi + \frac{\partial x}{\partial \eta}d\eta \text{ and } dy = \frac{\partial y}{\partial \xi}d\xi + \frac{\partial y}{\partial \eta}d\eta$$

$$\text{or} \quad \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = J \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}, \tag{6.1}$$

where J = Jacobian matrix of the transformation.

If at point (ξ,η) we have $|J| = det(J) \neq 0$

then an inverse map $T_e^{-1}(x, y \to \xi, \eta)$ exists at this point and thus

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = J^{-1} \begin{bmatrix} dx \\ xy \end{bmatrix} \tag{6.2}$$

and

$$T_e^{-1} \; : \; \begin{matrix} \xi = \xi(x,y) \\ \eta = \eta(x,y) \end{matrix} \tag{6.3}$$

defines a map $(x,y) \to (\xi,\eta)$. As in (6.1), we have

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}. \tag{6.4}$$

Hence, by equating terms in (6.4) and (6.2), we have the following relations

$$\frac{\partial \xi}{\partial x} = \frac{1}{|J|}\frac{\partial y}{\partial \eta}, \quad \frac{\partial \xi}{\partial y} = -\frac{1}{|J|}\frac{\partial x}{\partial \eta}, \quad \frac{\partial \eta}{\partial x} = -\frac{1}{|J|}\frac{\partial y}{\partial \xi}, \quad \frac{\partial \eta}{\partial y} = \frac{1}{|J|}\frac{\partial x}{\partial \xi} \tag{6.5}$$

(3) Construction of the Transformations T_e

Criteria for selection of T_e

(i) Within $\Omega_e, \xi(x,y)$ and $\eta(x,y)$ must be invertible and continuously differentiable.

(ii) $\{T_e\}_{e=1}^{E}$ must generate a mesh with no spurious gaps between elements and with no element overlapping another.

(iii) T_e should be easy to construct from the geometric data of the element.

Construction of T_e

In finite element method, the transformation T_e is constructed based on the element shape functions.

Let ψ_j be the shape function defined on $\bar{\Omega}$ for $j = 1, 2...N$, where N is the total number of nodes in $\bar{\Omega}$. Then, any function $g = g(\xi, \eta)$ in $\bar{\Omega}$ can be approximated by

$$\bar{g}(\xi, \eta) = \sum g_j \psi_j(\xi, \eta). \tag{6.6}$$

Let $g = x$ and $g = y$ respectively, from (6.6) we have

$$T_e \ : \quad \begin{array}{l} x = \sum_{j=1}^{N} x_j \psi_j(\xi, \eta), \\ y = \sum_{j=1}^{N} y_j \psi_j(\xi, \eta), \end{array} \tag{6.7}$$

which maps $\bar{\Omega}$ to Ω_e. To see this, consider a node i in $\bar{\Omega}$, the coordinates is (ξ_i, η_i). From (6.7), this point is mapped into point $x = x_i$, $y = y_i$ in the $x - y$ plane i.e, node i.

Remarks:

1) Criterion (iii) is easily verified. T_e is readily constructed from element data $(x_i,\ y_i,\ ...)$.

2) Criterion (ii) is usually not difficult to satisfy.

For example, the quadratic shape function on the master square as shown in Figure 6.3 maps the element to the corresponding elements Ω_e in the $x-y$ plane in such a way that straight sides of the $\bar{\Omega}$ are mapped to quadratic curved sides of Ω_e. On a given curved side between Ω_e and Ω_{e+1} , the maps T_e and T_{e+1} reduce to the same quadratic functions. Hence, the inter-element boundary is uniquely determined – no gaps between elements.

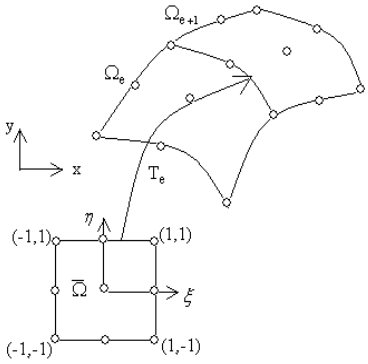


Figure 6.3: Straight sides of $\bar{\Omega}$ map to curved sides of Ω_e

3) For T_e to be invertible, we require $det(J) \neq 0$. In addition, from the integration theory,

$$dxdy = |J|\, d\xi d\eta.$$

Clearly, for the mapping defined by (6.3) to be acceptable, we must have positive values of $|J|$ at all points in $\bar{\Omega}$. The satisfaction of this condition is not assured in general for all maps of the form (6.7). Each set of shape functions must be examined to ensure that $|J| > 0$ throughout $\bar{\Omega}$.

Example 6.1

Figure 6.4 shows a 4-node master element $\bar{\Omega}$ and 2 elements Ω_1 and Ω_2 generated from it using the map (6.7). The shape function defined on $\bar{\Omega}$ are

$$\psi_i = \frac{1}{4}(1 - \xi\xi_i)(1 - \eta\eta_i), \quad (i = 1, ..., 4)$$

where (ξ_i, η_i) are coordinates of node i. In this example, straight lines ξ = constant or

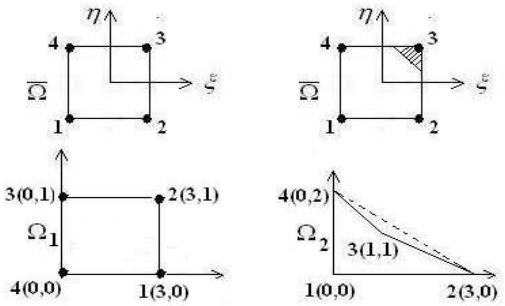


Figure 6.4: Mapping $\bar{\Omega}$ to Ω_1 and Ω_2

η = constant in $\bar{\Omega}$ map to corresponding straight lines in Ω_e .

For Ω_1

$$T_e : \begin{array}{rcl} x &=& 3\psi_1 + 3\psi_2 = \frac{2}{3}(1 - \eta) \\ y &=& \psi_2 + \psi_3 = \frac{1}{2}(1 + \xi). \end{array}$$

$$|J| = det \begin{bmatrix} 0 & -\frac{3}{2} \\ \frac{1}{2} & 0 \end{bmatrix} = \frac{3}{4} > 0$$

Therefore, the map is invertible.

For Ω_2

$$|J| = \frac{1}{8}(5 - 3\xi - 4\eta) \begin{cases} = 0 & \text{along } L : \xi = \frac{5}{3} - \frac{4}{3}\eta \\ > 0 & \text{below } L \\ < 0 & \text{above } L. \end{cases}$$

The region above L is mapped outside of Ω_2 by T_2. Clearly, Ω_2 is unacceptable. The trouble can be traced to the fact that the interior angle at node 3 is greater than π. It can be shown that for the 4-node element $\bar{\Omega}$ and the bilinear shape function used, T_e will be invertible if and only if all angles of the element are less than π.

6.2 Finite Element Calculations

The key to the finite element approximation is the calculation of the element matrices for each element in the mesh. For the approximation of the problem described in Chapter 4, we need to calculate the following integrals

$$\begin{aligned} k_{ij}^e &= \int_{\Omega_e} \left[k\left(\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y} \right) + b\phi_i\phi_j \right] dxdy, \\ f_i^e &= \int_{\Omega_e} f\phi_i \, dxdy, \\ p_i^e &= \int_{\partial\Omega_{2e}} p\phi_i\phi_j \, ds, \\ \gamma_i^e &= \int_{\partial\Omega_{2e}} P\hat{u}\phi_i \, ds. \end{aligned} \tag{6.8}$$

To calculate the above integrals, we begin by choosing the master element $\bar{\Omega}$ with geometry as simple as possible, such as square.

For a chosen Ω, we need to

- identify M nodes and shape function φ to define the coordinates map T_e,
- identify N nodes and shape function $\bar{\varphi}$ for local approximation of the unknown function.

Remarks: M and N need not to be the same.

- If $M > N_e$, then it is super-parametric map.
- If $M = N_e$, then it is iso-parametric map (iso-parametric element).
- If $M < N_e$, then it is sub-parametric map.

In the following, we will consider only the iso-parametric element.

Having selected $\bar{\Omega}$ and φ_j , we perform the following steps:

(1) Element map

$$T_e \ : \quad \begin{array}{l} x = \sum_{j=1}^{N} x_j \varphi_j(\xi, \eta) \\ y = \sum_{j=1}^{N} y_j \varphi_j(\xi, \eta) \end{array} \tag{6.9}$$

(2) Transformation of shape functions

As T_e is invertible, $\xi = \xi(x, y)$, $\eta = \eta(x, y)$ and the element shape functions are

$$\phi_j(x, y) = \varphi_j[\xi(x, y), \eta(x, y)] \tag{6.10}$$

Therefore,

$$\frac{\partial \phi_j}{\partial x} = \frac{\partial \varphi_j}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \varphi_j}{\partial \eta}\frac{\partial \eta}{\partial x}, \quad \frac{\partial \phi_j}{\partial y} = \frac{\partial \varphi_j}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \varphi_j}{\partial \eta}\frac{\partial \eta}{\partial y}.$$

According to (6.9)

$$\frac{\partial x}{\partial \xi} = \sum_{1}^{N_e} x_k \frac{\partial \varphi_k}{\partial \xi}(\xi, \eta), \quad \frac{\partial x}{\partial \eta} = \sum_{1}^{N_e} x_k \frac{\partial \varphi_k}{\partial \eta}(\xi, \eta),$$

$$\frac{\partial y}{\partial \xi} = \sum_{1}^{N_e} y_k \frac{\partial \varphi_k}{\partial \xi}(\xi, \eta), \quad \frac{\partial y}{\partial \eta} = \sum_{1}^{N_e} y_k \frac{\partial \varphi_k}{\partial \eta}(\xi, \eta),$$

Thus, using (6.5) and (6.9), equation (6.10) becomes

$$\frac{\partial \phi_j}{\partial x} = \frac{1}{|J|}\left\{ \frac{\partial \varphi_j}{\partial \xi} \sum_{k=1}^{N} y_k \frac{\partial \varphi_k}{\partial \eta}(\xi, \eta) - \frac{\partial \varphi_j}{\partial \eta} \sum_{k=1}^{N} y_k \frac{\partial \varphi_k}{\partial \xi}(\xi, \eta) \right\}$$

$$\frac{\partial \phi_j}{\partial y} = \frac{1}{|J|}\left\{ \frac{\partial \varphi_j}{\partial \xi} \sum_{k=1}^{N} x_k \frac{\partial \varphi_k}{\partial \eta}(\xi,\eta) - \frac{\partial \varphi_j}{\partial \eta} \sum_{k=1}^{N} x_k \frac{\partial \varphi_k}{\partial \xi}(\xi,\eta) \right\}$$

Remarks:

(a) The partial derivatives of ϕ_j w.r.t. x and y are completely determined by calculation defined only on $\bar{\Omega}$.

(b) From (6.8), for 4-node element, K^e is a 4*4 matrix which can be expressed as

$$K^e = \int_{\Omega_e} (k(D\phi)^T (D\phi) + b\phi^T \phi)\, d\Omega \tag{6.11}$$

where $\phi = (\phi_1, \phi_2, \phi_3, \phi_4)$ and

$$D\phi = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} & \frac{\partial \phi_2}{\partial x} & \frac{\partial \phi_3}{\partial x} & \frac{\partial \phi_4}{\partial x} \\ \frac{\partial \phi_1}{\partial y} & \frac{\partial \phi_2}{\partial y} & \frac{\partial \phi_3}{\partial y} & \frac{\partial \phi_4}{\partial y} \end{bmatrix}.$$

(3) Integration

Let $I = \int_{\Omega_e} g(x,y)\, dxdy$

then $I = \int_{\bar{\Omega}} G(\xi,\eta)\, d\xi d\eta$,

where

$$G(\xi,\eta) = g(\sum_{1}^{N} x_j \varphi_j(\xi,\eta), \sum_{1}^{N} y_j \varphi_j(\xi,\eta))|J(\xi,\eta)| \tag{6.12}$$

Numerical quadrature (such as the Gaussian quadrature) are usually used to evaluate the integrals. Quadrature rules for quadrilateral elements are usually derived from the 1-D quadrature by treating the integration over $\bar{\Omega}$ as a double integral.

Thus, using the 1-D quadrature rule of order N,

$$I = \int_{\bar{\Omega}} G(\xi,\eta) d\xi d\eta = \int_{-1}^{1} \left[\int_{-1}^{1} G(\xi,\eta) d\xi \right] d\eta \approx \sum_{k=1}^{N} \left[\sum_{\ell=1}^{N} G(\xi_\ell, \eta_k) w_\ell \right] w_k$$

For 9-point Gaussian quadrature (1-D of order 3).

$$N = 3,\ w_1 = 5/9,\ w_2 = 8/9,\ w_3 = 5/9,$$

$$\xi_1 = \eta_1 = -\sqrt{3/5},\ \xi_2 = \eta_2 = 0,\ \xi_3 = \eta_3 = \sqrt{3/5}.$$

If $k = k(x, y),\ b = b(x, y)$ and $f = f(x, y)$ are not constant over an element, we may use

$$k(x,y) \approx \sum_{j=1}^{N} k_j \phi_j(x,y),\quad b(x,y) \approx \sum_{j=1}^{N} b_j \phi_j(x,y),\quad f(x,y) \approx \sum_{j=1}^{N} f_j \phi_j(x,y).$$

Then the calculations of a_{ij}^e and f_i^e only require the nodal values of k, b and f.

(4) Boundary Integrals

The calculation of the boundary integrals in (6.8) is carried out by integrating along those sides of $\bar{\Omega}$ that are mapped onto the sides of $\partial\Omega_{2e}$ along which natural boundary conditions are prescribed.

For definiteness, we suppose that the sides $\xi = 1$ of a master square is to be mapped onto $\partial\Omega_{2h}$. Let θ_j denote the restriction of the master-element shape function φ_j to side $\xi = 1$, i.e,

$$\theta_j(\eta) = \varphi_j(1,\eta), \quad j = 1, 2, ..., N.$$

We thus have

$$\int_{\partial\Omega_{2e}} p\phi_i\phi_j\ ds = \int_{-1}^{1} p\theta_i(\eta)\theta_j(\eta)|J|\ d\eta$$

Since

$$ds = \sqrt{\left(\frac{\partial x}{\partial \eta}(1,\eta)\right)^2 + \left(\frac{\partial y}{\partial \eta}(1,\eta)\right)^2}\, d\eta,$$

we have

$$|J(\eta)| = \sqrt{\left(\frac{\partial x}{\partial \eta}(1,\eta)\right)^2 + \left(\frac{\partial y}{\partial \eta}(1,\eta)\right)^2}$$

where $x(\xi, \eta)$ and $y(\xi, \eta)$ are defined in (6.9). The integral can be evaluated numerically.

6.3 Finite Element Program

In general, a finite element program consists of a main program and several subroutines or functions. The main program is used to control the process while each subroutine is used to perform certain operations.

Example 6.2 Let Ω be a square region, consider

$$\begin{cases} \nabla \cdot k(x)\nabla T + Q = 0 & \text{on } \Omega \\ T = x & \text{on } y = 0 \\ T = 3 + x^2 & \text{on } y = 3 \\ T = y & \text{on } x = 0 \\ \frac{\partial T}{\partial x} = 1 - 0.2T & \text{on } x = 3 \end{cases}$$

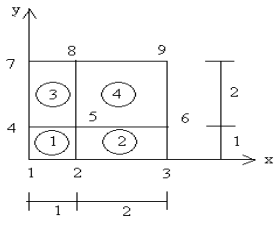


Figure 6.5: Finite element mesh for Example 6.2

1) Find the variational statement of the boundary value problem.

2) Derive the finite element equations.

3) Write a program to solve the problem using the finite element method and the mesh shown in Figure 7.5 and k=1, Q=2.

   (a) Code a subroutine PREP to read data (number of elements, number of nodes, number of boundary points and etc.) necessary for the finite element calculations.

(b) Code a subroutine PROC to construct the finite element equations and solve the equations by the LU factorization method. (Use the 4-nodded iso-parametric element as master element and evaluate integrals using the 9-point Gaussian quadrature).

(c) Code a subroutine POST to print the nodal values of T.

Requirement

- From part 3) print the element matrices (stiffness matrix and load vector) for each element and the global matrices after each assembling of element matrices.
- From part 3) print the global coefficient matrix and load vector after imposing the boundary conditions.
- Print the nodal values of T.
- From part 3) show the finite element solution of $T(x, y)$ along $x = y$.

Sol Variational Statement is

Find $T \in H^1(\Omega)$ such that $T = x$ on $y = 0$, $T = 3 + x^2$ on $y = 3$, $T = y$ on $x = 0$ and

$$a(T, v) = L(v), \qquad \forall v \in H_0^1(\Omega)$$

.

where $H_0^1(\Omega) = \{v : v \in H^1(\Omega) \text{ and } v = 0 \text{ on } x = 0,\ y = 0 \text{ and } y = 3\}$

$a(T, v) = \int_0^3 0.2T(3, y)v(3, y)\, dy + \int_\Omega \nabla T \cdot \nabla v\, d\Omega$

$L(v) = \int_0^3 v(3, y)\, dy + \int_\Omega 2v\, d\Omega$

Finite element equations is a system

$$\mathbf{AT} = \mathbf{F}$$

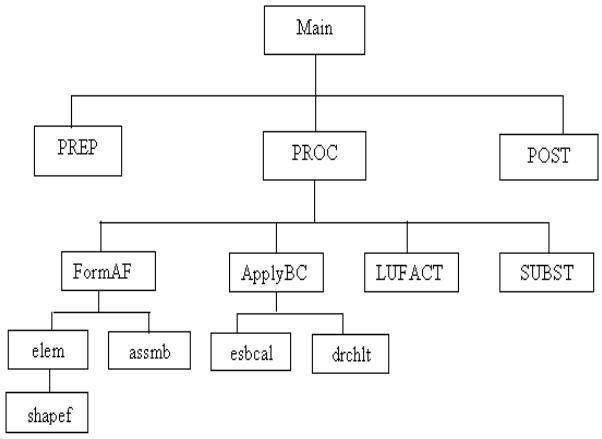
where $\mathbf{A} = \{a_{ij}\}$, with $a_{ij} = \sum_{e=1}^{E} \int_{\Omega_e} \nabla\phi_i \cdot \nabla\phi_j \, d\Omega + \sum_{e=1}^{B} \int_{\partial\Omega_e} 0.2\phi_i\phi_j \, dy$

$\mathbf{F} = \{f_i\}$ with $f_i = \sum_{e=1}^{E} \int_{\Omega_e} 2\phi_i \, d\Omega + \sum_{e=1}^{B} \int_{\partial\Omega_e} \phi_i \, dy$,

B is the number of elements with natural boundary condition.

Program

The following figure is the structure chart of the finite element program for the problem, which outlines the structure of the program.



1) Main function

```
Aim :  Control the process

       Call PREP
       Call PROCESS(T)
       Call POST(T)
```

2) Preprocessing Routine PREP

Aim : Read the data required for the construction of the finite element equations including

(a) Control data

Nele - Number of elements (4)
Nnode - Number of nodes (9)
Nessb - Number of nodes with essential boundary condition (**T** specified) (7)
Gbc - Number of elements with general boundary condition (2)

Topology of elements (elements definition)

Nnode(i,j) - Contains the global node number of the j^{th} node of the i^{th} element

Input : 1 2 5 4 ← element 1
2 3 6 5 element 2
4 5 8 7 element 3
5 6 9 8 element 4

(b) Nodal point coordinate definition

X1(i),Y1(i) - the coordinates of the i^{th} node

Input for $(X1(i),\ i=1,9) = 0,1,3,0,1,3,0,1,3$
$(Y1(i),\ i=1,9) = 0,0,0,1,1,1,3,3,3$

Boundary condition definition

Noess (i) - node no. of the i^{th} node with essential boundary condition.

Tess (i) - T value at the i^{th} node with essential boundary condition.

nogbc(i,1),
nogbc(i,2) - The nodal number of the end points of the i^{th} line segment with general boundary condition.

Input for (noess(i),Tess(i),i=1,7) : 1,0,2,1,3,3,4,1,7,3,8,4,9,12

((nogbc(i,j),j=1,2),i=1,2) : 3,6,6,9

(c) Finite element calculation - Routine PROC

Aim : Solve the finite element equation **AT = F**
for the unknown model values **T**.

Call FormAF ← Construct global matrices A and F

Call ApplyBC ← Modify A and F to account for the B.Cs.

Call Lufact
Call Subst } Solve **AT=F** by the LU Method.

(d) Subroutine FormAF

Steps :

- Initialize the global stiffness matrix **A** and vector **F**
- Call Elem, element by element, to calculate the element matrices
- Call Assmb, to add element matrices to global matrices

Do 30 nel = 1, nele

Call Elem (nel,ea,ef)

Call Assmb (ea,ef,nel)

(e) Subroutine Elem (nel,ea,ef)

Calculate element matrices by Gaussian Quadrature

$$ea_{ij} = \int_{\Omega e} \nabla\phi_i . \nabla\phi_j d\Omega = \sum_{k=1}^{3}\sum_{\ell=1}^{3} \left[|J|(\nabla\phi_i, \nabla\phi_j) \right] (\xi_k, \eta_\ell)\omega_k\omega_\ell$$

$$ef_i = \int_{\Omega e} 2\phi_i d\Omega = \sum_{k=1}^{3}\sum_{\ell=1}^{3} \left[2|J|\phi_i \right] (\xi_k, \eta_\ell)\omega_k\omega_\ell$$

Do 40 k = 1,3

Do 40 l = 1,3

Call shapef (.....shape, dshape.....)

Do 50 i = 1,4

Do 50 j = 1,4

ea(i,j) = ea(i,j) + ...

(f) Subroutine shapef (nel,.....)

Aim : Calculate

$$\phi_i = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i)$$

$$\frac{\partial\phi_i}{\partial x} = \frac{1}{|J|}\left(\frac{\partial\phi_i}{\partial\xi}\sum_{k=1}^{4} y_k\frac{\partial\phi_k}{\partial\eta} - \frac{\partial\phi_i}{\partial\eta}\sum_{k=1}^{4} y_k\frac{\partial\phi_k}{\partial_\xi}\right)$$

$$\frac{\partial\phi}{\partial y} = \frac{1}{|J|}\left(\frac{\partial\phi_i}{\partial\xi}\sum_{k=1}^{4} x_k\frac{\partial\phi_k}{\partial\eta} - \frac{\partial\phi_i}{\partial\eta}\sum_{k=1}^{4} x_k\frac{\partial\phi_k}{\partial_\xi}\right)$$

(g) Assembly Routine - Assmb (nel,ea,ef)

In the subroutine assmb, the contributions to the global **A** and **F** from each single element are added to the accumulated contributions from other elements.

The element matrix is 4 by 4. The list of 4 nodal point numbers, stored in node(nel,j),j = 1,4, specifies the rows and columns of **A** into which the entries of ea are to be accumulated.

Do 10 i = 1,4

```
ig = node(nel,i)
Do 20 j = 1,4
    jg = node (nel,j)
    a(ig,jg) = a(ig,jg)+ea(i,j)
```

(h) Routine Applybc for applying boundary conditions.

- For node with essential boundary condition, call DRCHLT (ui,Nrow) to modify the global matrice **A** and the global vector **F** using the method presented.
- For general boundary condition, call gbcal to calculate the integral on the boundary.

EXERCISES 6

Question 1

Given the list of nodal points and their coordinates and the list of elements and their node numbers below:

(a) sketch the finite element mesh Ω_h

(b) sketch the ξ, η -axes in each element

(c) verify that the maps $T_e, e = 1, 2, ...5$, produce a connected region Ω_h

(d) sketch the global basis function ϕ_e for node 4 of the mesh.

Node	x	y
1	0	1
2	0.7	0.7
3	1	0
4	0	2
5	1.5	1.5
6	2	0
7	0	3
8	1.5	3
9	3	3
10	3	1.5
11	3	0

Element	Nodes
1	1, 2, 5, 4
2	3, 6, 5, 2
3	5, 8, 7, 4
4	5, 10, 9, 8
5	6, 11, 10, 5

Question 2

Code a shape function routine SHAPE for the calculation of the values of the shape function φ_i and their derivatives $\frac{\partial \varphi_i}{\partial \xi}$ and $\frac{\partial \varphi_i}{\partial \eta}$. (use 4 node element).

Code a main program to test routine SHAPE by calling it for several values of element coordinates ξ and η and print the results for comparison with hand calculation.

Question 3

Code a 2-D integration rule routine SETINT. Use the 9-point Gaussian quadrature rule. Code a main program to perform numerical integration over the 2-D domain Ω given by $-1 \leq x \leq 1, -1 \leq y \leq 1$. Calculate

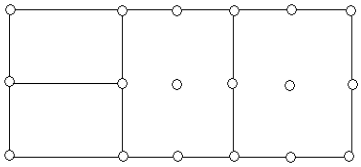
$$I = \int_{\Omega} (3x + 2x^2 y^2)\, dxdy.$$

Question 4

Code an element routine ELEM for the calculation of the element matrix defined in (6.8). Use 4-node element.

Question 5

Why the following FE mesh cannot be used?



Chapter 7

Solution of Linear Systems of Equations

In solving boundary value problems using the finite element method, at certain stage of the solution process, one has to solve one or many equations. A system of linear equations can be written as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n &= b_n \end{aligned} \tag{7.1}$$

or in matrix form $A\mathbf{x} = \mathbf{b}$.

At present, many techniques are available for the solution of linear systems of equations. These techniques are classified into direct methods and iterative methods. In this chapter, we will introduce various direct and iterative methods for solving linear systems arising from finite element formulation of boundary value problems. The rest of the chapter is organized as follows. In section 7.1, we introduce various direct methods. In section 7.2, we introduce the band method for sparse systems based on a direct method. Then in section 7.3, we introduce various iterative methods.

7.1 Direct Methods for Systems of Linear Equations

7.1.1 Gaussian Elimination

Solving a linear system Ax= b by Gaussian elimination includes two phases: eliminating process and backward substitution process.

Elimination Process

The elimination process reduces the system by row operations to an equivalent simpler system $U\mathbf{x} = \mathbf{y}$ in which U is an upper triangular matrix. This process requires (n -1) steps.

Step 1. (Assume $a_{11} = 0$). Eliminate the 1st unknown from equations (2 - N) (i.e., *set the 1st column below the diagonal to zero*). This can be achieved by subtracting suitable multiples of the first row (equation) from the other rows (equations), namely

$$R_i \leftarrow R_i - m_{i1}R_1.$$

The above rule is to be applied to every element of the ith row. Thus we have,

$$\left.\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)} \\ b_i^{(2)} &= b_i^{(1)} - m_{i1}b_1^{(1)} \end{aligned}\right\} \quad (i = 1, 2, 3, ..., n)$$

For $j = 1$, we have

$$a_{i1}^{(2)} = a_{i1}^{(1)} - m_{i1}a_{11}^{(1)}, \quad i = 1, 2, ..., n.$$

Thus to set $a_{i1}^{(2)} = 0$, we only need to choose

$$m_{i1} = a_{i1}^{(1)}/a_{11}^{(1)}.$$

During the process, the first equation is called pivotal equation and its coefficient at the diagonal ($a_{11}^{(1)}$) is called the pivot. After this step, the augmented matrix becomes

$$\left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & \vdots & & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right].$$

Step k. After (k-1) steps of the elimination process, all the elements below the diagonal in columns 1 to (k-1) have been set to zero. In the kth step, assume that the kth pivot $a_{kk}^{(k)} \neq 0$, we deal with column k to set the elements below the diagonal in this column to zero (i.e, eliminate the kth unknown from equations (k+1) to n). This can be achieved by performing the following row operations for $i = k+1$ to n

$$R_i \leftarrow R_i - m_{ik} R_k \ \texttt{with} \ m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}.$$

Applying the above row operation rule to every column of the ith row yields

$$a_{ij}^{k+1} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \ \ b_i^{k+1} = b_i^{(k)} - m_{ik} b_k^{(k)}, \ \ (j = k, n). \tag{7.2}$$

Obviously for the elements in column k, $j = k$ and

$$a_{ik}^{(k+1)} = a_{ik}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kk}^{(k)} = 0,$$

and so the elements below the diagonal in column k will all be set to zero.

After (n -1) steps, the system becomes

$$\begin{bmatrix} a_{11}^{(1)} & \cdots & \cdots & \cdots & a_{in}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ b_n^{(n)} \end{bmatrix}$$

or $A^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$ or $U\mathbf{x} = \mathbf{y}$.

In summary, we have the following recurrence formulae for the elimination process

Recurrence formulae for the Gaussian elimination process

for $k = 1, 2, ..., n-1$

for $i = k+1, n$

$m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$

$b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$

for $j = k+1, n$

$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$

Exercise Write a F95 program segment to implement the above elimination process.

Store the multiples m_{ij} in the lower triangle of A.

Backward Substitution

The backward substitution solves the new equivalent system $U\mathbf{x} = \mathbf{y}$, i.e

$$\begin{bmatrix} u_{11} & & \cdots & & u_{1n} \\ & \ddots & & & \vdots \\ & 0 & u_{kk} & \cdots & u_{kn} \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & & 0 & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_n \end{bmatrix}.$$

From the nth equation, we have

$$u_{nn}x_n \;=\; y_n. \tag{7.3}$$

From the kth equation, we have

$$u_{kk}x_k + \sum_{j=k+1}^{n} u_{kj}x_j \;=\; y_k. \tag{7.4}$$

Thus from 7.3 and 7.4, we have the following recurrence formulae

Recurrence formulae for the backward substitution

$$\begin{aligned} x_n &= \tfrac{y_n}{u_{nn}} \\ x_k &= \tfrac{1}{u_{kk}}\left[y_k - \textstyle\sum_{j=k+1}^{n} u_{kj}x_j\right], \quad (k = n-1, n-2, ..., 1). \end{aligned}$$

Exercise. Write a F95 program segment to implement the above backward substitution process.

Operation Count

The number of operations in each step and consequently the total number of operations required are summarized in Table 7.1. Therefore, for the elimination process

- the number of $*/\div$: $\sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-1} i = \frac{n(n-1)(2n-1)}{6} + \frac{n(n-1)}{2} = \frac{n^3}{3} - \frac{n}{2}$,
- the number of $+/-$: $\sum_{i=1}^{n-1} i^2 = \frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}$;

while for the substitution process

- the number of $*/\div$: $2\sum_{i=1}^{n-1} i + n = n^2$,
- the number of $+/-$: $2\sum_{i=1}^{n-1} i = n^2 - n$.

Table 7.1: Operation Count

	Elimination for U			Forward subs. for y		Backward subs. for x		
Step	$\pm$	$\times$	$\div$	$\times$	$\pm$	$\div$	$\times$	$\pm$
1	$(n-1)^2$	$(n-1)^2$	$n-1$	$n-1$	$n-1$	1	0	0
2	$(n-2)^2$	$(n-2)^2$	$n-2$	$n-2$	$n-2$	1	1	1
k	$(n-k)^2$	$(n-k)^2$	$n-k$	$n-k$	$n-k$	1	$k-1$	$k-1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	1	1	1	1	1	1	$n-2$	$n-2$
n	0	0	0	0	0	1	$n-1$	$n-1$
Total	$\sum_{i=1}^{n-1} i^2$	$\sum_{i=1}^{n-1} i^2$	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n-1} i$	n	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n-1} i$

So if n is large, the elimination process requires about $n^3/3$ operations of $*/\div$, the substitution process requires n^2 operations of $*/\div$.

The elimination process described in this section includes computations of the upper triangular matrix $U(A^{(n)})$ and the right hand side vector $\mathbf{y}(\mathbf{b}^{(n)})$. It will be shown in section 7.1.2 that the determination of $\mathbf{y}$ is in fact through a forward substitution process, namely solving $L\mathbf{y} = \mathbf{b}$ where L is a lower triangular matrix.

Example 7.1 Solve $\begin{cases} x_1 + 2x_2 + x_3 = 0 \\ 2x_1 + 2x_2 + 3x_3 = 3 \\ -x_1 - 3x_2 = 2 \end{cases}$ using Gaussian elimination method.

Solution

$$[A|b] = \left[\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 2 & 2 & 3 & 3 \\ -1 & -3 & 0 & 2 \end{array}\right] = \xrightarrow[R_3 \leftarrow R_3 + R_2]{\begin{array}{c} m_{21} = a_{21}/a_{11} = 2/1 = 2 \\ m_{31} = a_{31}/a_{11} = -1/1 = -1 \end{array}} \left[\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & -2 & 1 & 3 \\ 0 & -1 & 1 & 2 \end{array}\right]$$

$$= \xrightarrow[R_3 \leftarrow R_3 - \frac{1}{2}R_2]{m_{32} = a_{32}/a_{22} = -1/(-2) = 1/2} \left[\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & -2 & 1 & 3 \\ 0 & 0 & 1/2 & 1/2 \end{array}\right]$$

Thus,

$$\begin{aligned} & x_3 = 1 \\ -2x_2 + x_3 = 3 \quad & \Rightarrow \quad x_2 = \tfrac{3-x_3}{-2} = \tfrac{3-1}{-2} = -1 \\ x_1 + 2x_2 + x_3 = 0 \quad & \Rightarrow \quad x_1 = -2x_2 - x_3 = -2(-1) - 1 = 1. \end{aligned}$$

7.1.2 LU Factorization and Its Connection with Gaussian Elimination

Theorem 7.1 (Factorization Theorem)

If the Gaussian elimination procedure can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ without row interchanges, then A can be factored into the product of a lower triangular matrix L and an upper triangular U, i.e., $A = LU$.

Proof (Hint). To prove $A = LU$, let

$$L = \begin{bmatrix} 1 & & & 0 \\ m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ m_{n1} & m_{n2} & \cdots & 1 \end{bmatrix}, \quad U = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ 0 & & & a_{nn}^{(n)} \end{bmatrix}$$

where m_{ij} and $a_{ij}^{(k)}$ are as defined in section 7.1.1. Then show that $(LU)_{ij} = a_{ij}$.

Corollary: $det\ A = a_{11}^{(1)} a_{22}^{(2)} \ldots a_{nn}^{(n)}$.

Proof $det\ A = det\ L\ det\ U$.

Since L and U are triangular, their determinants are the product of their diagonal elements.

Exercise. Write a F95 program segment to read a matrix A and then find its LU factorization using the Gaussian elimination.

LU Method and Its Connection with the Gaussian Elimination Method

If $A = LU$, then $A\mathbf{x} = \mathbf{b}$ becomes $LU\mathbf{x} = \mathbf{b}$.

Put $U\mathbf{x} = \mathbf{y}$, then $L\mathbf{y} = \mathbf{b}$.

Thus, the procedure for solving $A\mathbf{x} = \mathbf{b}$ by the LU method is:

(1) Compute L and U (by the Gaussian elimination process)

(2) Solve $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$ (by the forward substitution to yield $y_1 \to y_2 \to \cdots \to y_n$)

(3) Solve $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$ (by the backward substitution to yield $x_n \to x_{n-1} \to \cdots x_1$).

Advantages of LU Method

- More economic if we need to solve many systems with the same coefficient matrix A but different right hand sides, as we only need to evaluate L and U for one time. Once L and U are saved, only the forward and backward substitutions are needed to solve each system.
- Storage space may be economized. If A is not required after factorization, we can store L and U in A.

7.1.3 Pivoting and Scaling

Pivoting

At each step (say k) of the Gaussian elimination process, we need to use a multiplier

$$m_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}.$$

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{ik}^{(k)}$, m_{ik} will have magnitude much larger than one and thus

- when computing $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)} \quad (j = k+1, n)$, a rounding error introduced in the computation of one of the terms $a_{kj}^{(k)}$ will be multiplied by m_{ik} compounding the original error.

- When performing the backward substitution for the solution x_k, any rounding error in the numerator will be amplified when dividing by $a_{kk}^{(k)}$.

To avoid the above problem, pivoting is performed by selecting a larger element for the pivot.

Partial (maximum column) Pivoting

In the Gaussian elimination process at stage k, determine the smallest $p > k$, such that

$$\left|a_{pk}^{(k)}\right| = \max_{k \le i \le n} \left|a_{ik}^{(k)}\right|$$

and perform $(E_k \leftrightarrow E_p)$ where $E_k =$ the kth equation, then proceed with step k of the elimination process. Thus, all of the multipliers m_{ik} will now satisfy $|m_{ik}| \le 1$.

Exercise. Write a F95 program segment to implement the elimination process with maximum column pivoting.

Example 7.2 Solving $\begin{bmatrix} 0.003 & 59.14 \\ 5.291 & -6.130 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 59.17 \\ 46.78 \end{bmatrix}$ using Gaussian elimination with partial pivoting (with 4 digit arithmetic).

Solution

$E_1 \leftrightarrow E_2$:

$$\left[\begin{array}{cc|c} 5.291 & -6.130 & 46.78 \\ 0.003 & 59.14 & 59.17 \end{array}\right] \xrightarrow[R_2 \leftarrow R_2 - 0.000567R_1]{m_{12} = 0.003/5.291 = 0.000567} \left[\begin{array}{cc|c} 5.291 & -6.130 & 46.78 \\ 0 & 59.14 & 59.14 \end{array}\right]$$

Therefore, $\begin{cases} x_1 = 10.00 \\ x_2 = 1.00. \end{cases}$

`Check`	`Exact solution`	$x_1 = 10,\ x_2 = 1$
	`Gaussian elimination without pivoting`	$x_1 = -10,\ x_2 = 1.001.$

Complete Pivoting

At stage k, determine the smallest $p,\ q > k$, such that

$$\left|a_{pq}^{(k)}\right| = \max_{k \le i,j \le n} \left|a_{ij}^{(k)}\right|$$

and perform $(E_k \leftrightarrow E_p,\ \ C_k \leftrightarrow C_q)$, then proceed with step k of the elimination process.

Scaling and Scaled-column Pivoting

If the elements of A vary greatly in size, the pivoting strategy described above may fail. To deal with this problem, two methods may be used.

a) **Scaling matrix** A so that the elements vary less, usually by multiplying the rows and columns by suitable constants. This process will generally change the choice of pivot elements when pivoting is used with Gaussian elimination.

b) **Scaled - Column Pivoting Technique**

The first step in this procedure is to define for each row a scale factor s_i by

$$s_i = \max_{i \le j \le n} |a_i j|$$

If for some i we have $s_i = 0$, then the system has no unique solution, since all entries in the ith row are zero. If all s_i are not equal to zero, we continue the Gaussian elimination process using matrix A. But we choose the pivot element in step k by determining the smaller $p > k$, such that

$$\frac{\left|a_{pk}^{(k)}\right|}{s_p} = \max_{k \le i \le n} \frac{\left|a_{ik}^{(k)}\right|}{s_i}$$

replacing the definition in partial pivoting. This process is to select the pivotal equation from the available $(n-k)$ candidates as the one that has the absolutely largest coefficient of x_k relative to the size of the equation.

Example 7.3 Solve $\begin{bmatrix} 30.00 & 591400 \\ 5.291 & -6.130 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 591700 \\ 46.78 \end{bmatrix}$ using using scaled column pivoting (the system is obtained by multiplying the 1st equation in example 7.2 by 10^4)

Solution

$$s_1 = \max(|30|, |591400|) = 591400 \qquad s_2 = 6.130$$

$$\frac{|a_{11}|}{s_1} = \frac{30}{591400} = 0.5073 \times 10^{-4}, \quad \frac{|a_{21}|}{s_2} = 0.863$$

$$\frac{|a_{11}|}{s_1} < \frac{|a_{21}|}{s_2}$$

Therefore, $E_1 \leftrightarrow E_2$

$$\left[\begin{array}{cc|c} 5.291 & -6.130 & 48.78 \\ 30.00 & 591410 & 591700 \end{array}\right] \Rightarrow \mathbf{x} = \begin{bmatrix} 10 \\ 1 \end{bmatrix},$$

while the result obtained by Gaussian elimination with partial pivoting is $x = \begin{bmatrix} -10 \\ 1 \end{bmatrix}$.

7.1.4 Permuted LU Factorization

If the matrix A is such that a linear system $A\mathbf{x} = \mathbf{b}$ can be solved using Gaussian elimination that does not require row interchanges, then there exists a direct LU factorization of A and the system can be solved by the LU method presented in 7.1.2. In the following, we will show that if row interchanges are required to control the rounding error resulting from the use of finite-digit arithmetic, there also exists a LU

method, namely the permuted LU factorization method corresponding to the Gaussian elimination with pivoting.

We begin the discussion with the introduction of a class of matrices that are used to rearrange, or permute, rows of a given matrix.

Permutation Matrices

A permutation matrix P has the same form as the identity matrix except that the order of the rows is different,

$$P = \begin{Bmatrix} e_{k1} \\ e_{k2} \\ \vdots \\ e_{kn} \end{Bmatrix} \tag{7.5}$$

where e_{ki} denotes the kith row of the $n \times n$ identity matrix.

For example, if $k1 = 2,\ k2 = 3,\ k3 = 1$, then

$$P = \begin{Bmatrix} e_2 \\ e_3 \\ e_1 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Property of Permutation Matrix

Left multiplying a matrix by a permutation matrix P has the effect of interchanging (permuting) the rows of the matrix. More specifically, let $k_1, k_2, \ldots, k_n$ be a permutation of the integers $1, 2, \ldots, n$, and define P as in (7.5), then

$$PA = \begin{bmatrix} a_{k1,1} & a_{k1,2} & \cdots & a_{k1,n} \\ a_{k2,1} & a_{k2,2} & \cdots & a_{k2,n} \\ \vdots & \vdots & & \vdots \\ a_{kn,1} & a_{kn,2} & \cdots & a_{kn,n} \end{bmatrix} = \{a_{ki,j}\}$$

i.e., the ith row of the new system is the kith row of the original system.

Example 7.4 Let $P = \begin{Bmatrix} e_1 \\ e_2 \\ e_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 3 \\ 0 & 1 & 0 \end{bmatrix}$, $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$. Then $PA = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$. That is, left multiplying A by P has the effect of interchange row 2 and row 3.

Maximum Column Pivoting & Permuted LU Factorization

Application of the elimination phase of the Gaussian elimination with maximum column pivoting is equivalent to the LU factorization of a permuted version of the coefficient matrix. Thus the Gaussian elimination with maximum column pivoting algorithm can be used to find the permuted LU factorization.

Given $A\mathbf{x} = \mathbf{b}$, by pre-multiplying P, we have

$$PA\mathbf{x} \;=\; P\mathbf{b}.$$

Let $PA = LU$, then

$$LU\mathbf{x} \;=\; P\mathbf{b} \quad \Rightarrow \quad \begin{cases} L\mathbf{y} = P\mathbf{b}, \\ U\mathbf{x} = \mathbf{y}. \end{cases}$$

Suppose after row interchanges, the order of equations to be processed is $k1, k2, \cdots, kn$ where ki denotes the kith equation of the original system, then

$$P \;=\; \begin{Bmatrix} e_{k1} \\ e_{k2} \\ \vdots \\ e_{kn} \end{Bmatrix} \;\texttt{and}\; P\mathbf{b} \;=\; \begin{Bmatrix} b_{k1} \\ b_{k2} \\ \vdots \\ b_{kn} \end{Bmatrix}.$$

Thus to determine $P\mathbf{b}$ in solving $L\mathbf{y} = P\mathbf{b}$, we only need to create an array (permuta-

tion vector) $p(1:n)$ to store the values $k1,\ k2,\ \ldots, kn$ and hence

$$P\mathbf{b} = \begin{Bmatrix} b_{k1} \\ b_{k2} \\ \vdots \\ b_{kn} \end{Bmatrix} = \begin{Bmatrix} b(p(1)) \\ b(p(2)) \\ \vdots \\ b(p(n)) \end{Bmatrix}.$$

Program Construction

1) Obtain the permuted LU matrices by the Gaussian elimination with maximum column pivoting. A permutation vector p is also produced to indicate the order in which the original equations are to be processed.

2) Forward substitution $L\mathbf{y} = P\mathbf{b} \quad \Rightarrow \quad y_i = b[p(i)] - \sum_{j=1}^{i-1} l_{ij} y_j$

3) Backward substitution $U\mathbf{x} = \mathbf{y} \quad \Rightarrow \quad x_i = \frac{1}{u_{ii}} \left(y_j - \sum_{j=i+1}^{n} u_{ij} y_j \right).$

7.1.5 LL^T and LDL^T Factorization Methods

For strictly diagonally dominant and positive definite matrices, Gaussian elimination can be performed without row interchanges.

Strictly Diagonally Dominant Matrix (S.D.D)

Definition: A is (strictly) diagonally dominant iff $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ for all i.

Theorem 7.2: If A is a strictly diagonally dominant matrix, then A is nonsingular.

Theorem 7.3: If A is diagonally dominant and nonsingular, then Gaussian elimination can be performed on any linear system $A\mathbf{x} = \mathbf{b}$ to obtain its unique solution without row or column interchanges.

Positive Definite Matrices

Definition: A symmetric $n \times n$ matrix A is positive definite iff $\mathbf{x}^t A\mathbf{x} > 0$ for every $n - D$ column vector $\mathbf{x} \neq \mathbf{0}$.

Properties: If A is an $n \times n$ positive definite matrix, then

(a) A is nonsingular

(b) $a_{ii} > 0$ for all i

(c) A is symmetric, $A = A^t$

Theorem 7.4: The $n \times n$ symmetric matrix A is positive definite iff Gaussian elimination without row interchanges can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ with all pivot elements positive. Moreover, the computations are stable with respect to the growth of rounding error.

(a) **The Cholesky ($\mathbf{LL}^T$) Factorization and Solution of $A\mathbf{x} = \mathbf{b}$**

If A is symmetric,

$$A = LU = A^t = U^t L^t \;\; \Rightarrow \;\; L = U^t,\; U = L^t,\; l_{ii} = u_{ii}.$$

Thus, If A is positive definite, A can be factorized in the form of LL^t where L is a lower triangular matrix with nonzero diagonal entries.

LL^t factorization

From $A = LL^t$ for $i = 1, 2, \ldots, n$ and $j \leq i$, by multiplying the ith row of L and jth column of L^t, we have

$$a_{ij} = \sum_{k=1}^{j} l_{ik} l_{jk} = \sum_{k=1}^{j-1} l_{ik} l_{jk} + l_{ij} l_{jj}. \tag{7.6}$$

From the above, for $i = j = 1$, we have

$$l_{11} = \sqrt{a_{11}}.$$

Now suppose row 1, row 2,..., row $i-1$ of L have been determined, we can derive, from (7.6), the following recurrence formulae to determine the ith row of L.

$$l_{ij} = \frac{1}{l_{jj}} \left[a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right], \quad \texttt{for } j = 1 \texttt{ to } i-1,$$

$$l_{ii}^2 \;=\; a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2.$$

Solving $A\mathbf{x} = \mathbf{b}$

As $A = LL^t$, $A\mathbf{x} = \mathbf{b} \;\rightarrow\; LL^t\mathbf{x} = \mathbf{b} \;\rightarrow \begin{cases} L\mathbf{y} = \mathbf{b} \\ L^t\mathbf{x} = \mathbf{y} \end{cases}$

To solve $L\mathbf{y} = \mathbf{b}$, multiplying the ith row of L with $\mathbf{y}$ yields

$$(l_{i1},\; l_{i2},\; \ldots, l_{i,i-1},\; l_{ii},\; 0, \ldots,\; 0) \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} = b_i.$$

Therefore, $y_i = \frac{1}{l_{ii}} \left[b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right] \quad (i = 1, 2, \ldots, n)$

Remark: From the above, we can determine y_1, then $y_2, \ldots, y_n$.

To solve $L^t\mathbf{x} = \mathbf{y}$, multiplying the ith row of L^t with $\mathbf{x}$ yields

$$(0, \ldots, 0, l_{ii}, l_{(i+1)i}, \ldots, l_{ni}) \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = y_i.$$

Therefore, $x_i = \frac{1}{l_{ii}}\left[y_i - \sum_{j=i+1}^{n} l_{ji}x_j\right] \quad (i = n, n-1, \ldots, 1)$.

Remark: From the above, we can determine x_n , then $x_{n-1}, \ldots, x_1$.

Algorithm (exercises)

Operation Count

Number of $*/\div$ operations: $\frac{n^3}{6} + \frac{n^2}{2} - \frac{2}{3}n$;

Number of $+/-$ operations: $\frac{n^3}{6} - \frac{n}{6}$;

Number of $\sqrt{}$ operations: n.

(b) **LDL^t Decomposition and Solution of $A\mathbf{x} = \mathbf{b}$**

- The square roots in the LL^t decomposition can be avoided by using a slight modification, i.e., find a diagonal matrix D and a new lower triangular matrix L with one's on the diagonal such that $A = LDL^t$.
- This method applies for not only the positive definite matrices but also certain symmetric matrices.

LDL^t decomposition

From $A = LDL^t$, for $i = 1, 2, \ldots, n$ and $j \le i$,

$$a_{ij} = \sum_{k=1}^{j} l_{ik}d_k l_{jk} = \sum_{k=1}^{j-1} l_{ik}d_k l_{jk} + d_j l_{ij} l_{jj},$$

thus we have for

$$\begin{array}{ll} i = 1,\ j = 1, & \to d_1 = a_{11} \\ i = 2, 3, \ldots, n \begin{cases} j = 1, 2, \ldots, i-1 \\ j = i \end{cases} & \begin{array}{l} \to l_{ij} = \frac{1}{d_j}\left[a_{ij} - \sum_{k=1}^{j-1} l_{ik}d_k l_{jk}\right] \\ \to d_i = \left[a_{ii} - \sum_{k=1}^{i-1} d_k l_{il}^2\right] \end{array} \end{array}$$

Solving $A\mathbf{x} = \mathbf{b}$

As $A = LDL^t$, $Ax = b \ \rightarrow \ LDL^t x = b \ \rightarrow \ \begin{cases} L\mathbf{y} = \mathbf{b} \\ L^t\mathbf{x} = D^{-1}\mathbf{y} \end{cases}$

Hence, we can derive

$$y_i \ = \ b_i - \sum_{j=1}^{i-1} l_{ij} y_j \qquad (i = 1,\ 2, \ldots,\ n)$$

$$x_i \ = \ \frac{y_i}{d_i} - \sum_{j=i+1}^{n} l_{ji} x_j \qquad (i = n,\ n-1, \ldots,\ 1)$$

Algorithm (exercise)

Operation Count

Number of $*/\div$ operations: $\frac{n^3}{6} + n^2 - \frac{7}{6}n$; number of $+/-$ operations: $\frac{n^3}{6} - \frac{n}{6}$.

7.2 Solution of Sparse Systems of Linear Equations

In finite element method, the linear systems that arise usually have the property that the nonzero entries of A are often clustered in a small number of diagonals surrounding the main diagonal. Thus, we can use special techniques to

- Improve computation efficiency by avoiding operating involving zero operands;
- Reduce storage requirement.

This section concerns with a unique solution of the finite element linear system

$$A\mathbf{x} \ = \ \mathbf{b}, \tag{7.7}$$

where the $n \times n$ matrix A is large, sparse, and symmetric, $\mathbf{b}$ and $\mathbf{x}$ are both $n \times m$ $(m \geq 1)$ matrices.

Let m_A be the smallest integer such that $a_{ij} = 0$ for $|i - j| > m_A$, then

- the portion of A containing exactly those entries a_{ij} satisfying $|i-j| \leq m_A$ is called the band of A,

- m_A is called the bandwidth by storing mainly the nonzero entries.

In the LDL^T method, none of the entries of the matrix A can ever be nonzero unless $|i-j| \leq m_A$. This fact can be used to modify the algorithm, so that only the entries of the band are ever used.

Algorithm

LDL^T: $d_1 = a_{11}$

For $i = 2, ..., n$

$$\ell_{ij} = \frac{1}{d_j}[a_{ij} - \sum_{k=max(i-m_A, j-m_A)}^{j-1} \ell_{ik}\ell_{jk}d_k], \quad j = 1, 2, ..., i-1$$

$$d_i = a_{ii} - \sum_{k=i-m_A}^{j-1} \ell_{ik}^2 d_k$$

Subst: $y_i = b_i - \sum_{k=i-m_A}^{i-1} \ell_{ik}y_k, \quad i = 1, ..., N$

$$y_i = \frac{y_i}{d_i} - \sum_{k=i+1}^{min(i+m_A, n)} \ell_{ki}x_k, \quad i = n, n-1, ..., 1.$$

Remarks:

Let m_i be the smallest integer such that $a_{ij} = 0$ $(j < i)$ if $i - j > m$ (or $j < i - m$). It can be shown that, the algorithm for such problem is similar to the one above. Except for the backward substitution formula, we can replace $i \pm m_A$ by $i \pm m_i$ and $j \pm m_A$ by $j \pm m_j$.

Storage Structure

Rules:

a) Store the matrix $A_{n\times n}$ row by row into a one-dimensional array $A1$.

b) For each row, store the elements from the 1st nonzero entry to the diagonal element.

- Use $IA(i)_{i=1\ to\ N}$ to identify the storage location of $A(i,i)$ in the 1-D array $A1$. Thus,

  - The elements in ith row (from the 1st nonzero element to the diagonal element) are stored into $A1[IA(i-1)+1], A1[IA(i-1)+2], ..., A1[IA(i)]$
  - The number of nonzero elements (from the 1st nonzero elements to the diagonal but not include the diagonal element)

$$m_i = IA(i) - IA(i-1) - 1$$

  - The element $A(i,j)$ is stored into $A1(ij)$ where

$$ij = IA(i) - (i-j).$$

Example 7.5 For $A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & a_{32} & a_{33} & 0 \\ 0 & a_{42} & 0 & a_{44} \end{bmatrix}$ (assume symmetric)

We will construct $A1$ and IA as follows.

k	1	2	3	4	5	6	7	8
$A1(k)$	a_{11}	a_{21}	a_{22}	a_{32}	a_{33}	a_{42}	0	a_{44}

i	1	2	3	4
$IA(i)$	1	3	5	8

Programming using 1-D Storage Structure

Subroutine LDLT1(N,IA,A)

A: an array, contain matrix A(1-D). On exit, contain LD in 1-D form.

For $i = 2, N$

$m_i = IA(i) - IA(i-1) - 1$

$i1 = max(i - m_i, 1)$

For $j = i1, i$

$m_j = IA(j) - IA(j-1)$

$J1 = max(j - m_j, 1)$

$sum = 0$

For $k = max(i1, j1),\ j - 1$

$ik = Trans(i, k)$

$jk = Trans(j, k)$

$kk = Trans(k, k)$

$sum = sum + A(ik) * A(jk) * A(kk)$

$ij = Trans(i, j)$

$jj = Trans(j, j)$

If $(i \neq j)$ then

$A(ij) = (A(ij) - sum)/A(jj)$

else

$A(ij) = a(ij) - sum$

END

7.3 Iterative Methods for Systems of Linear Equations

For large systems with a high percentage of zero entries, iterative techniques are usually more efficient in terms of both computer storage and computation time than direct methods.

Solving an $n \times n$ linear system by iterative techniques includes the following general steps

- Convert $A\mathbf{x} = \mathbf{b}$ to $N\mathbf{x} = T\mathbf{x} + c$
- Select an initial vector $\mathbf{x}^{(0)}$ approximating $\mathbf{x}$
- Generate a sequence of vectors $\mathbf{x}^{(k)}$ that converges to $\mathbf{x}$ by

$$Nx^{(k)} = Tx^{(k-1)} + c, \quad k = 1, 2, \ldots$$

Jacobi and Gauss-Seidel iterative methods are two basic iterative techniques which convert $A\mathbf{x} = \mathbf{b}$ to $N\mathbf{x} = T\mathbf{x} + \mathbf{c}$ using different ways.

In this section, we first introduce the Jacobi method and Gauss-Seidel method respectively in section 7.3.1 and 7.3.2. Then we present the general convergence condition for iterative methods and particularly the sufficient conditions for convergence of the Jacobi and Gauss-Seidel methods in section 7.3.3 , followed by the topics of error estimate and speed of convergence in section 7.3.4. Finally, the relaxation methods are presented in section 7.3.5 in the notion of improving the Gauss-Seidel method by introducing a scaling (relaxation) factor.

7.3.1 The Jacobi Iterative Method

In this method, the system

$$A\mathbf{x} \;=\; \mathbf{b} \tag{7.8}$$

is converted to $N\mathbf{x} = T\mathbf{x} + \mathbf{c}$ by splitting A into its diagonal and off-diagonal parts

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = D - L - U, \tag{7.9}$$

where

$$D \;=\; \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

$$L \;=\; \begin{bmatrix} 0 & 0 & \dots & 0 \\ -a_{21} & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \dots & -a_{n(n-1)} & 0 \end{bmatrix}$$

$$U \;=\; \begin{bmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -a_{(n-1)n} \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Thus, equation (7.8) becomes

$$D - L - U)\mathbf{x} \;=\; \mathbf{b}$$

which gives

$$D\mathbf{x} - (L + U)\mathbf{x} \;=\; \mathbf{b}$$

and so

$$\mathbf{x} = D^{-1}(L+U)\mathbf{x} + D^{-1}\mathbf{b} \tag{7.10}$$

Hence, we can establish the following iterative formula

$$\mathbf{x}^{(k)} \;=\; T_j\mathbf{x}^{(k-1)} + \mathbf{c} \tag{7.11}$$

where $T_j = D^{-1}(L+U)$ and $\mathbf{c} = D^{-1}\mathbf{b}$. Alternatively, (7.11) can be expressed as

$$\mathbf{x}_i^{(k)} = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1} a_{ij}\mathbf{x}_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}\mathbf{x}_j^{(k-1)} + b_i\right], \quad i = 1, 2, \ldots, n. \tag{7.12}$$

The criterion to stop generating new term is

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|}{\|\mathbf{x}^{(k)}\|} < Tol.$$

Jacobi Iterative Algorithm for Solving $A\mathbf{x} = \mathbf{b}$

To solve $A\mathbf{x} = \mathbf{b}$, given an initial approximation $\mathbf{x}^{(0)}$.

Input: Number of equations N; matrix A, vector $\mathbf{b}$, initial guess x_0, tolerance Tol, maximum number of iterations N_{iter}

Output: The solution $\mathbf{x}$ or a message that the number of iterations was exceeded.

Step 1: $k \leftarrow 1$

Step 2: while ($k \leq N_{iter}$ do steps 3-5

Step 3: for $i = 1$ to N do

$$x_i \;\leftarrow\; \frac{1}{a_{ii}}\left[-\sum_{j=1, j\neq i}^{n} a_{ij}\mathbf{x}_{0j} + b_i\right]$$

Step 4: if $\|\mathbf{x} - \mathbf{x}_0\| < Tol$ then

Output $\mathbf{x}$

STOP

Step 5: else

$x_0 \leftarrow x$

$k \leftarrow k + 1$

then go to step 3

Step 6: Output "Number of iterations was exceeded"

STOP.

Example 7.6 Find the solution to the system

$$\begin{bmatrix} 9 & -1 & -1 \\ -1 & 8 & 0 \\ -1 & 0 & 9 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 7 \\ 7 \\ 8 \end{bmatrix}.$$

Solution

$$\mathbf{x}^{(k)} = \begin{bmatrix} 0 & 1/9 & 1/9 \\ 1/8 & 0 & 0 \\ 1/9 & 0 & 0 \end{bmatrix} \mathbf{x}^{(k-1)} + \begin{bmatrix} 7/8 \\ 7/8 \\ 8/9 \end{bmatrix}.$$

Select $\mathbf{x}^{(0)} = 0$. The numerical results for $k = 1, 2, 3, 4, 5$, obtained by using the Jacobi method, are given in Table 7.1.

7.3.2 The Gauss-Seidel Iteration Method

A possible improvement to Jacobi's scheme is suggested by an analysis of equation (7.12). To compute $\mathbf{x}_i^{(k)}$, the components of $\mathbf{x}_i^{(k-1)}$ are used in Jacobi's method. Since, for $i \geq 1$, $x_1^k, x_2^k, \ldots, x_{i-1}^k$ have already been computed and are likely to be better

Table 7.2: Numerical Solutions by Jacobi and Gauss-Seidel Methods

k	0	1	2	3	4	5
	0	0.7798	0.9738	0.9942	0.9993	0.9998
$\mathbf{x}^k$	0	0.8750	0.9722	0.9967	0.9993	0.9999
by Jacobi	0	0.8889	0.9971	0.9971	0.9993	0.9999
	0	0.7778	0.9942	0.9998	1.000	
$\mathbf{x}^k$	0	0.9722	0.9993	1.0000	1.000	
by Gauss-Seidel	0	0.9753	0.9993	1.0000	1.000	

approximation to the actual solution of $x_1, x_2, \ldots, x_{i-1}$ than $x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$, it is reasonable to compute $\mathbf{x}_i^{(k)}$ using these most recently calculated data.

The Gauss-Seidel iteration scheme is based on this consideration and takes the following form

$$\mathbf{x}_i^{(k)} = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i \right]. \tag{7.13}$$

We can also derive this formula and its matrix form directly from the original equation (7.8). Using (7.9), equation (7.8) can be written as

$$(D - L)\mathbf{x} = U\mathbf{x} + \mathbf{b}. \tag{7.14}$$

Then the Gauss-Seidel iteration scheme is

$$(D - L)\mathbf{x}^k = U\mathbf{x}^{(k-1)} + \mathbf{b}, \tag{7.15}$$

which gives

$$D\mathbf{x}^k = L\mathbf{x}^k + U\mathbf{x}^{(k-1)} + \mathbf{b} \tag{7.16}$$

or

$$\mathbf{x}^k = (D - L)^{-1} U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b}. \tag{7.17}$$

Formulae (7.16) is precisely formulae (7.13).

Gauss-Seidel Algorithm for solving $A\mathbf{x} = \mathbf{b}$

Input: Number of equations N; matrix A, vector $\mathbf{b}$, initial guess x_0, tolerance Tol, maximum number of iterations N_{iter}

Step 1: $k \leftarrow 1$

Step 2: while ($k \leq N_{iter}$ do steps 3-5

Step 3: for $i = 1$ to N do

$$x_i \leftarrow \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij}\mathbf{x}_j + \sum_{j=i+1}^{n} a_{ij}\mathbf{x}_{0j} + b_i \right]$$

Step 4: if $\|\mathbf{x} - \mathbf{x}_0\| < Tol$ then

Output $\mathbf{x}_i$

STOP

Step 5: else

$k \leftarrow k + 1$

$x_0 \leftarrow x$

go to step 3

Step 6: Output "Number of iterations was exceeded"

STOP.

Example 7.7 Find the solution to the following system:

$$\begin{bmatrix} 9 & -1 & -1 \\ -1 & 8 & 0 \\ -1 & 0 & 9 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 7 \\ 7 \\ 8 \end{bmatrix}$$

Solution

Select $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}_3 = 0$ and use the following iterative formula to calculate the value of xi for each cycle

$$\begin{aligned} \mathbf{x}_1 &\leftarrow \tfrac{1}{9}(7 + \mathbf{x}_2 + \mathbf{x}_3) \\ \mathbf{x}_2 &\leftarrow \tfrac{1}{8}(7 + \mathbf{x}_1) \\ \mathbf{x}_3 &\leftarrow \tfrac{1}{9}(8 + \mathbf{x}_1) \end{aligned} \quad .$$

The computed results are shown in Table 7.1.

7.3.3 Convergence Conditions

General Convergence Conditions

An iterative technique to solve the $n \times n$ linear system $A\mathbf{x} = \mathbf{b}$ starts with an initial approximation $\mathbf{x}^{(0)}$ to the solution $\mathbf{x}$, and then generates a sequence of vectors

$$\mathbf{x}_{(k)} = T\mathbf{x}_{(k-1)} + \mathbf{c}, \quad k = 1, 2, \ldots$$

that converges to $\mathbf{x}$.

Questions:

1) Does $\mathbf{x}_{(k)}$ converge to the solution $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ for any case?

2) If not, what is the restriction to $\mathbf{x}^{(0)}$, T or $\mathbf{c}$?

Theorem 7.5 For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c} \quad (k \geq 1 \text{ and } \mathbf{c} \neq \mathbf{0}) \tag{7.18}$$

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ iff $\rho(T) < 1$.

Proof

(i) First, prove that $\rho(T) < 1 \Rightarrow$ the sequence $\mathbf{x}^{(k)}$ generated by (7.18) converges to

the unique solution.

From (7.18)

$$\begin{aligned}\mathbf{x}^{(k)} &= T\mathbf{x}^{(k-1)} + \mathbf{c} \\ &= T(T\mathbf{x}^{(k-2)} + \mathbf{c}) + \mathbf{c} \\ &= T^2\mathbf{x}^{(k-2)} + (T+I)\mathbf{c} \\ &\vdots \\ &= T^k\mathbf{x}^{(0)} + (T^{k-1} + \cdots + T + I)\mathbf{c}.\end{aligned}$$

As $\rho(T) < 1$, we have

$$\lim_{k\to\infty} T^k\mathbf{x}^{(0)}.$$

From the Neumann Lemma

$$\lim_{k\to\infty} \sum_{j=0}^{k-1} T^j = (I-T)^{-1}.$$

Hence,

$$\begin{aligned}\lim_{k\to\infty} \mathbf{x}^{(k)} &= \lim_{k\to\infty} T^k\mathbf{x}^{(0)} + \lim_{k\to\infty} \left(\sum_{j=0}^{k-1} T^j\right)\mathbf{c} \\ &= 0 + (I-T)^{-1}\mathbf{c}.\end{aligned}$$

Thus, $\mathbf{x} = \lim_{k\to\infty} \mathbf{x}^k = (I-T)^{-1}\mathbf{c}$ is the unique solution to $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

(ii) Now prove that $\rho(T) < 1 \Leftarrow \mathbf{x}^{(k)}$ converges to the unique solution $\mathbf{x}$.

If $\mathbf{x}^{(k)} \to \mathbf{x}$ for any $\mathbf{x}^{(0)}$, then from (7.18) $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

So for each k

$$\mathbf{x} - \mathbf{x}^{(k)} = T\mathbf{x} + \mathbf{c} - (T\mathbf{x}^{(k-1)} + \mathbf{c}) = T(\mathbf{x} - \mathbf{x}^{(k-1)}) = \cdots = T^k(\mathbf{x} - \mathbf{x}^{(0)})$$

Hence, $\lim_{k\to\infty} T^k\left(x - x^{(0)}\right) = \lim_{k\to\infty}\left(x - x^{(k)}\right) = 0$.

As $\mathbf{x}^{(0)}$ is arbitrary, $\mathbf{x} - \mathbf{x}^{(0)}$ is also arbitrary. Thus, $\rho(T) < 1$.

Strictly Diagonally Dominant Matrices

Definition: The $n \times n$ matrix A is said to be strictly diagonally dominant when

$$|a_{ij}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

holds for each $i = 1, 2, \ldots, n$.

For example, $A = \begin{bmatrix} 7 & 2 & 0 \\ 3 & 5 & -1 \\ 0 & 5 & -6 \end{bmatrix}$ is strictly diagonally dominant, but

$B = \begin{bmatrix} 6 & 4 & -3 \\ 4 & -2 & 0 \\ -3 & 0 & 1 \end{bmatrix}$ is not.

Theorem 7.6 A strictly diagonally dominant $n \times n$ matrix A is nonsingular.

Proof Assume that D, L and U are as defined by (7.9), then $A = D - L - U$. As A is strictly diagonally dominant, $D_{ii} \neq 0$ and thus D is nonsingular. So we can construct

$$B_1 = D^{-1}(L + U),$$

and

$$\|B_1\|_\infty = \|D^{-1}(L+U)\|_\infty = \max_i \left[\sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| + \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| \right] < 1.$$

Hence,

$$|\lambda| \leq \rho(B_1) < \|B_1\|_\infty < 1.$$

For any eigenvalue λ of B_1, $1 - \lambda$ is an eigenvalue of $I - B_1 = [I - D^{-1}(L + U)]$. Since $\lambda < 1$, it follows that no eigenvalue of $I - B_1$ can be zero and consequently $I - B_1$ is nonsingular.

Therefore, A is a product of two nonsingular matrices D and $(I - B_1)$.

$$A = D(I - B_1) = D - L - U.$$

Hence, $det\ A = det\ D\ det(I - B_1) \neq 0$ and thus A is nonsingular.

Sufficiency Conditions for Convergence of the Jacobi & Gauss-Seidel Methods

By analyzing the iteration matrices for the Jacobi method, T_j given in equation (7.11) and the Gauss-Seidel method, T_g given in equation (7.17), i.e.

$$T_j = D^{-1}(L + U), \quad T_g = (D_L)^{-1}U,$$

we can derive the following sufficiency condition for convergence of the Jacobi and Gauss-Seidel methods.

Theorem 7.7 If A is strictly diagonally dominant, then for any choice of $\mathbf{x}^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\left\{\mathbf{x}_{(k)}\right\}_{k=0}^{\infty}$ that converge to the unique solution of $A\mathbf{x} = \mathbf{b}$.

Proof

(1) For Jacobi method - Exercise

(2) For Gauss-Seidel method

$$\|T_g\|_\infty = \max_{\|x\|_\infty = 1} \|\mathbf{y}\|_\infty$$

where

$$\mathbf{y} = T_g\mathbf{x} = (D - L)^{-1}U\mathbf{x}$$

from which

$$\mathbf{y} = D^{-1}L\mathbf{y} + D^{-1}U\mathbf{x}.$$

Assume

$$y_k = \max_i(y_i),$$

$$\|\mathbf{y}\|_\infty = |y_k| \leq \sum_{j=1}^{k-1} \left|\frac{a_{kj}}{a_{kk}}\right| \|\mathbf{y}\|_\infty + \sum_{j=k+1}^{n} \left|\frac{a_{kj}}{a_{kk}}\right| \|\mathbf{x}\|_\infty$$

$$= r_k\|\mathbf{y}\|_\infty + s_k\|\mathbf{x}\|_\infty,$$

where $r_k = \sum_{j=1}^{k-1} \left|\frac{a_{kj}}{a_{kk}}\right|$, $s_k = \sum_{j=k+1}^{n} \left|\frac{a_{kj}}{a_{kk}}\right|$.

Thus, we have form the above formulae

$$\|\mathbf{y}\|_\infty \leq \frac{s_k}{1-r_k}\|\mathbf{x}\|_\infty,$$

and consequently

$$\|T_g\|_\infty \leq \frac{s_k}{1-r_k}.$$

As $1-(s_k+r_k) > 0$ for strictly diagonally dominant systems, we have $1-r_k > s_k$ and hence

$$\|T_g\|_\infty \leq 1$$

then

$$\rho(T_g) < 1.$$

This, from Theorem 7.5, guarantees that the sequence converges to the unique solution.

7.3.4 Error Bound and Speed of Convergence

We now have the following essential issues:

(i) How to estimate the error?

(ii) How many iterations are needed for a given accuracy requirement?

(iii) When are iterative methods preferable to Gaussian elimination methods in solving $A\mathbf{x} = \mathbf{b}$?

An error bound can be derived from theorem 7.5 and is summarized by the following corollary.

Corollary If $\|T\| < 1$ for any natural matrix norm, then the sequences $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ in (7.18) converges, for any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to a vector $\mathbf{x} \in \mathbb{R}^n$, and the following Error bounds hold

$$\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \leq \|T\|^k \left\|\mathbf{x}^{(0)} - \mathbf{x}\right\|$$

$$\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \leq \frac{\|T\|^k}{1 - \|T\|} \left\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right\|.$$

Remark 1. Since the above formulae hold for any natural matrix norm, it follows that

$$\left\|\mathbf{x}^{(k)} - \mathbf{x}\right\| \approx \rho^k(T) \left\|\mathbf{x}^{(0)} - \mathbf{x}\right\|.$$

Thus, the rate of convergence is essentially $\rho(T)$.

Proof

(a) First prove that $\|T\| < 1 \Rightarrow \left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \leq \|T\|^k \left\|\mathbf{x}^{(0)} - \mathbf{x}\right\|$.

From the iterative formulae (7.18)

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c},$$

we have

$$\mathbf{x} - \mathbf{x}^{(k)} = \mathbf{x} - T\mathbf{x}^{(k-1)} - \mathbf{c}. \tag{7.19}$$

As $\mathbf{x}$ is the exact solution, we have

$$\mathbf{x} = T\mathbf{x} + \mathbf{c},$$

and thus (7.19) becomes

$$\begin{aligned} \mathbf{x} - \mathbf{x}^{(k)} &= T\left(\mathbf{x} - \mathbf{x}^{(k-1)}\right) \\ &= T\left[T\left(\mathbf{x} - \mathbf{x}^{(k-2)}\right)\right] \\ &= T^2\left[\mathbf{x} - \mathbf{x}^{(k-2)}\right] \\ &= T^{(k)}\left[\mathbf{x} - \mathbf{x}^{(0)}\right]. \end{aligned}$$

Hence, $\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| = \left\|T^{(k)}[\mathbf{x} - \mathbf{x}^{(0)}]\right\| \leq \|T\|^{(k)} \left\|\mathbf{x} - \mathbf{x}^{(0)}\right\|$.

(b) Now prove that $\|T\| < 1 \;\Rightarrow\; \left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \leq \frac{\|T\|^k}{1-\|T\|} \left\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right\|$.

As $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$, we have

$$\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)} = T\left(\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}\right) = T^2\left(\mathbf{x}^{(n-1)} - \mathbf{x}^{(n-2)}\right) = \cdots = T^n\left(\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right).$$

Thus for $n > k \geq 1$,

$$\begin{aligned} \mathbf{x}^{(n)} - \mathbf{x}^{(k)} &= \left(\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}\right) + \left(\mathbf{x}^{(n-1)} - \mathbf{x}^{(n-2)}\right) + \cdots + \left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right) \\ &= T^{n-1}\left[\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right] + T^{n-2}\left[\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right] + \cdots + T^k\left[\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right] \\ &= \left(I + T + T^2 + \cdots + T^{n-k-1}\right) T^k\left[\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right] \end{aligned} \tag{7.20}$$

From which and the Neumann lemma, we have

$$\lim_{n\to\infty}\left[\mathbf{x}^{(n)}-\mathbf{x}^{(k)}\right]=\left(\lim_{n\to\infty}\sum_{j=0}^{n-k-1}T^j\right)T^k\left[\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right]=(I-T)^{-1}T^k\left[\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right] \tag{7.21}$$

Hence

$$(I-T)\left(\mathbf{x}-\mathbf{x}^{(k)}\right)=T^k\left(\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right) \tag{7.22}$$

or

$$\left(\mathbf{x}-\mathbf{x}^{(k)}\right)=T\left(\mathbf{x}-\mathbf{x}^{(k)}\right)+T^k\left(\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right) \tag{7.23}$$

Thus

$$\begin{aligned}\left\|\mathbf{x}-\mathbf{x}^{(k)}\right\| &\leq \left\|T\left(\mathbf{x}-\mathbf{x}^{(k)}\right)+T^k\left(\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right)\right\| \\ &\leq \left\|T\left(\mathbf{x}-\mathbf{x}^{(k)}\right)\right\|+\left\|T^k\left(\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right)\right\| \\ &\leq \|T\|\left\|\mathbf{x}-\mathbf{x}^{(k)}\right\|+\|T\|^k\left\|\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right\|\end{aligned} \tag{7.24}$$

Therefore, $\left\|\mathbf{x}-\mathbf{x}^{(k)}\right\|\leq\frac{\|T\|^k\left\|\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right\|}{1-\|T\|}$.

Remark 2. If the initial error is to be reduced to its ε multiple, then

$$\left\|\mathbf{x}-\mathbf{x}^{(k)}\right\|\leq\varepsilon\left\|\mathbf{x}-\mathbf{x}^{(0)}\right\|,$$

and thus we require from the corollary

$$(\rho(T))^k\leq\varepsilon$$

from which the iteration number needed can be determined by

$$k\geq\frac{+ln\,\varepsilon}{+ln\,\rho}\quad(ln\,\rho<0).$$

Example 7.8 Solve a dense linear system by iteration with accuracy up to about six digits.

Solution

Assume that $\mathbf{x}^{(0)} = \mathbf{0}$, then we require

$$\frac{\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_\infty}{\|\mathbf{x}\|_\infty} \leq \varepsilon = 10^{-6}.$$

If A has order n, the number of operations (multiplications) per iteration is n^2 . To obtain the required accuracy, the necessary number of iterations is

$$k^* = \frac{+\mathtt{ln}(10^{-6})}{\mathtt{ln}\rho} = \frac{6\ \mathtt{ln}\ 10}{-\mathtt{ln}\ \rho}$$

and the number of operations is

$$k^* n^2 = 6\ \mathtt{ln}\ 10 \frac{n^2}{-\mathtt{ln}\rho}.$$

If Gaussian elimination is used to solve $A\mathbf{x} = b$, the number of operations is about $\frac{n^3}{3}$. Therefore, the iterative method will be more efficient than the Gasuuian elimination method if

$$k^* n^2 < \frac{n^3}{3},\ \mathtt{that\ is}\ \ k^* < \frac{n}{3}.$$

Table 7.2 shows the k^* value corresponding to different ρ values. Obviously, if n is large and ρ is small, the iterative method is more efficient.

7.3.5 Relaxation Method

Relaxation method is an alternative form of Gauss-Seidel iterative formulae. Firstly, we examine the Gauss-Seidel iterative formulae

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i \right] \tag{7.25}$$

Table 7.3: The k^* value corresponding to different ρ

ρ	k^*
0.9	131
0.8	62
0.6	27
0.4	15
0.2	9

and develop an alternative form of the formulae.

Denote $$r_i^{(k)} = \left(r_{1i}^{(k)},\ r_{2i}^{(k)}, \ldots, r_{ni}^{(k)}\right)$$

as the residual error corresponding to the approximate solution at the end of the $(i-1)$th step of the kth iteration cycle

$$x_i^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)}\right)^T.$$

Then, the mth component of is

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_j^{(k-1)}. \tag{7.26}$$

For m=i, (7.26) becomes

$$r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} - a_{ii} x_i^{(k-1)},$$

from which we have

$$x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right].$$

It is noted that the right hand side of the above formula gives of the Gauss-Seidel method as shown by (7.25). Hence, we can determine $x_i^{(k)}$ by

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}, \tag{7.27}$$

which is an alternative formula for calculating the ith component of x in iteration k.

Characteristics of Gauss-Seidel Method

From (7.25) and (7.26) the component of residual vector for the ith equation $r_{i(i+1)}^{(k)}$ after the calculation of $x_i^{(k)}$ and $x_i^{(k-1)}$ becomes

$$\begin{aligned} r_{i(i+1)}^{(k)} &= b_i - \sum_{j=1}^{i} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \\ &= \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right] - a_{ij}x_i^{(k)} = \mathbf{0} \end{aligned}$$

which indicates that the characteristics of Gauss-Seidel method is that at each step of calculation, one component of the residual vector is reduced to zero.

Relaxation Method

In general, the basis of all relaxation methods is to calculate the residual vector $r = b - A\mathbf{x}$, and to modify (or relax) one or more components of the approximate solution $\mathbf{x}$ in order to reduce to zero one or more components of r. The Gauss-Seidel method is an example of relaxation methods.

Successive Over Relaxation (SOR) Method and Under Relaxation Method

Reducing one component of the residual vector to zero is not generally the most efficient way to reduce the norm of the vector r. The procedure (7.27) can be modified by

$$x_i^{(k)} = x_i^{(k-1)} + \omega\frac{r_{ii}^{(k)}}{a_{ii}}. \tag{7.28}$$

For certain choice of positive ω, the speed of convergence of $\mathbf{x}^{(k)} \to \mathbf{x}$ can be accelerated. For choice of $\omega < 1$, the procedures are called *under-relaxation methods*, and can be used to obtain convergence of some systems that are not convergent by the Gauss-Seidel method. For choice ? greater than 1, the procedures are called *over-relaxation methods*, which are used to accelerate convergence for systems that are convergent

by the Gauss-Seidel technique. These methods are called *Successive Over- Relaxation* (SOR).

The system of equations (7.28) can be written as

$$x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right]$$

or in matrix form

$$(D-\omega L)\mathbf{x}^{(k)} = [(1-\omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega b,$$
$$\mathbf{x}^{(k)} = (D-\omega L)^{-1}[(1-\omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega(D-\omega L)^{-1}b.$$

Choose of ω

In order to make $\mathbf{x}^{(k)}$ converge to $\mathbf{x}$ as rapidly as possible, the ω is to be chosen to minimize $\rho(T_\omega)$ where $T_\omega = (D-\omega L)^{-1}[(1-\omega)D+\omega U]$. Although *no complete answer* to this question is known for our general $n \times n$ linear systems, the following results can be used in certain situations.

Theorem 7.8 If $a_{ii} \neq 0$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that $\rho(T_\omega < 1$ only if $0 < \omega < 2$.

Theorem 7.9 (Ostrowski-Reich). If A is a positive definite matrix and $0 < \omega < 2$, then the SOR converges for any choice of initial approximate solution vector $\mathbf{x}^{(0)}$.

Theorem 7.10 If A is positive definite and tri-diagonal, then $\rho(T_g) = [\rho(T_j)]^2 < 1$ and the optimal choice of ω for the SOR is

$$\omega = \frac{2}{1+\sqrt{1-\rho(T_g)}} \quad \texttt{and} \quad \rho(T_\omega) = \omega - 1.$$

SOR Algorithm for Solving $A\mathbf{x} = \mathbf{b}$

Input: Number of equations N, A, $\mathbf{b}$, $\mathbf{x}_{0i}$, ω, Tol, N_{iter}

Output: The solution $\mathbf{x}_i$ or a message that the number of iterations was exceeded.

Step 1: $k \leftarrow 1$

Step 2: while $(k < N_{iter})$ do steps 3 to 5

Step 3: for $i = 1$ to N do

$$x_i \leftarrow (1-\omega)x_{0i} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_{0j}\right]$$

Step 4: If $\|\mathbf{x} - \mathbf{x}_0\| < Tol$ then

Output x and the procedure completed successfully

STOP

Step 5: else

$k \leftarrow k + 1$

$\mathbf{x}_{0i} \leftarrow \mathbf{x}_i$

go to step 3

Step 6: Output "maximum number of iterations was exceeded".

End

EXERCISES

Question 1 Solve the following linear systems using

a) Gaussian elimination (GE.), (2- digit rounding arithmetic) without pivoting,

b) GE. with maximal column pivoting (2- digits),

c) GE. with scaled-column pivoting (2- digits),

d) Exact arithmetic and determine which part, (a), (b), or (c) is the most accurate.

$$\begin{cases} 10^{-2}x + y = 1 \\ x + y = 2 \end{cases} \qquad \begin{cases} x_1 + 2x_2 + 3x_3 = 1 \\ 2x_1 + 3x_2 + 4x_3 = -1 \\ 3x_1 + 4x_2 + 6x_3 = 2 \end{cases}$$

Question 2 Find a factorization of the form $A = LDL^T$ for matrix

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Programming

Question 3 The attached subroutines LUFACT and SUBST are for solving $A\mathbf{x} = \mathbf{b}$. The routine LUFACT computes a LU factorization of A with scaled-column pivoting $PA = LU$. The routine SUBST calculates $\mathbf{x}$ by forward and backward substitutions: $L\mathbf{y} = P\mathbf{b}$, $U\mathbf{x} = \mathbf{y}$. Document and modify the program such that (1) all real values are stored with at least 12 digits of precision; (2) whole array operations are used whenever possible; (3) use assumed-shape arrays in procedures. Then write a main program which reads entries of A and b from a data file and calls the subroutine to solve the following system

$$\begin{aligned} 3.3330x_1 + +15920x_2 - 10.333x_3 &= 15913 \\ 2.2220x_1 + 16.710x_2 + 9.6120x_3 &= 28.544 \\ 1.5611x_1 + 5.1791x_2 + 1.6852x_3 &= 8.4254 \end{aligned}$$

Algorithm for Subroutine LUFACT

This algorithm uses the Gaussian elimination process with scaled-column pivoting to find the permuted LU factorization of A (namely, to find P and the LU factorization of PA) where U is the upper triangular matrix obtained from the elimination process; L is the lower triangular matrix which is the collection of the multiples m_{ij}.

Step 1. Set $s(i) = \max_{1\le j\le n} |a_{ij}|$, (determine the size of each equation).

Step 2. For $k = 1$ to $N-1$ do step 3 to step 6 (set 1st,...,(n-1)th column below diagonal to zero)

Step 3. Find the (smaller) $P \ge k$ such that $\left|\frac{a_{pk}}{s_p}\right| = \max_{k\le i\le n}\left|\frac{a_{ik}}{s_i}\right|$.

`(select pivot element for the step)`

Step 4. If $a_{pk} = 0$ then

write '(IERR=1, A is singular)' then return.

Step 5. Else

$E_k \leftrightarrow E_p, \quad P_k \leftrightarrow P_p$ (row interchange)

(P_k records the order in which the equations are to be processed)

Step 6. For $i = k+1$ to n (do usual Gauss elimination process for the kth step)

Set $m_{ik} = \frac{a_{ik}}{a_{kk}} \quad \Rightarrow \quad a(I,k))$

Set $a_{ij} = a_{ij} - m_{ik}a_{kj} \quad (j = k+1, ..., n)$

Step 7. If $a_{nn} = 0$, return "(IERR=1: A is singular)"

Return

Algorithm for SUBST

`For` $i = 1$ `to` n `do`
$y_i = b(p_i) - \sum_{j=1}^{i-1} a_{ij} y_j$ Forward substitution (as $Pb = b(p_i)$)

`For` $i = n$ `to` 1 `by` -1 `do`
$x_i = \frac{1}{a_{ii}} \left[y_i - \sum_{j=i+1}^{n} a_{ij} x_j \right]$ Backward substitution

End

Question 4 Based on the subroutines LUFACT and SUBST, write subroutines LU1 and SUB1, respectively for computing the LU factorization with partial pivoting (without scaling) and for finding solution of $A\mathbf{x} = \mathbf{b}$ by forward and backward substitution. Then, write a main program to call the subroutines to solve the linear system in Question 1.

Question 5 Write a subroutine for LDL^T factorization of a square matrix A, and a separate subroutine for solving equations $A\mathbf{x} = \mathbf{b}$ by forward and backward substitutions using the LDL^T factorization. Then write a main program to call the subroutines to solve the following linear system

$$\begin{aligned} 4x_1 + x_2 - x_3 &= 7 \\ x_1 + 3x_2 - x_3 &= 8 \\ -x_1 - x_2 + 5x_3 + 2x_4 &= -4 \\ 2x_3 + 4x_4 &= 6 \end{aligned}$$

Algorithm for factorizing an $n \times n$ matrix A into LDL^T decomposition

Input n and matrix A

Output L

Set $d_1 = a_{11}$

For $i = 2$ to N do

$$\text{set } l_{ij} = \frac{1}{d_j}\left[a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}\right] \quad j = 1, 2, ..., i-1.$$

$$\text{set } d_i = \left[a_{ii} - \sum_{k=1}^{i-1} d_k l_{ik}^2\right]$$

Return

Question 6 Find the first two iterations of the Jacobi method for the following system, using $\mathbf{x}^{(0)} = \mathbf{0}$

$$\begin{bmatrix} 10 & -1 & 0 \\ -1 & 10 & -2 \\ 0 & -2 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 7 \\ 6 \end{bmatrix}.$$

Question 7 Repeat Question 6 using the Gauss-Seidel method.

Question 8 Repeat Question 6 using the SOR method with $\omega = 1.2$.

Question 9 Write a computer subroutine to implement the SOR iterative scheme for $A\mathbf{x} = \mathbf{b}$. Then solve Question 8 using $Tol = 10^{-2}$, maximum number of iteration $N_{iter} = 25$, and $\omega = 0.5$ and $\omega = 1.1$, respectively.

Chapter 8

Stokes Problem and Incompressible Flows

In this chapter, we firstly present, in section 8.1, the fundamental equations governing the flow of incompressible Newtonian fluids. Then in section 8.2, we present the finite element solution for a special case, namely the steady state flow of Stokes fluids. Then in section 8.3, we give the general finite element formulation for the solution of the transient flow of incompressible Newtonian fluids.

8.1 Fundamental Equations for the Flow of Fluids

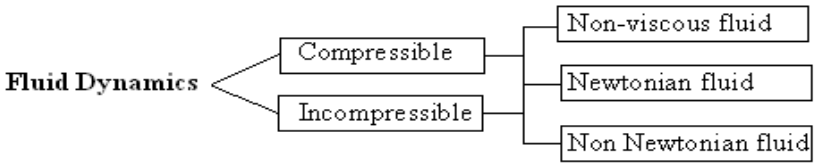
Fluid dynamics is concerned with the study of fluids in motion. More specifically, it concerns

- Kinematics of the flow field
- Stress distribution throughout the field

It is convenient to broadly classify fluid dynamics on the basis of constitutive equations for the fluid and in terms of compressibility of the fluid, as illustrated in the following

figure.

For example, the subject dealing with incompressible Newtonian fluids is referred to as fluid dynamics for incompressible Newtonian fluids.



Remaks

- For most problems, liquids can be treated as incompressible fluids and, in general, gases (except for low speed gas flows) must be considered as compressible.
- Many common fluids such as air and water can be modeled as Newtonian fluids.

Governing field Equations

The equations governing the flow of an incompressible Newtonian fluid consist of the equations of motion

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho X_i \;=\; \rho \frac{Dv_i}{Dt}, \tag{8.1}$$

the equation of continuity

$$div(\mathbf{v}) \;=\; \frac{\partial v_j}{\partial x_j} \;=\; 0, \tag{8.2}$$

and the constitutive equations

$$\sigma_{ij} \;=\; -p\delta_{ij} + 2\mu d_{ij}, \tag{8.3}$$

where

$$\frac{Dv_i}{Dt} = \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} = a_i$$

while the d_{ij} is the rate of deformation (also known as strain rate) and is related to the velocity by

$$d_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

The above equations are all field equations which must be satisfied at all points within a continuum.

To solve the above equations, substituting (8.3) into (8.1), we obtain

$$\rho\frac{Dv_i}{Dt} = \rho X_j - \frac{\partial p}{\partial x_j}\delta_{ij} + \mu\left(\frac{\partial^2 v_i}{\partial x_j \partial x_j} + \frac{\partial^2 v_j}{\partial x_i \partial x_j}\right). \tag{8.4}$$

Using the continuity equation (8.2), we have

$$\frac{\partial^2 v_j}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}\left(\frac{\partial v_j}{\partial x_j}\right) = 0$$

and thus equations (8.4) becomes

$$\frac{Dv_i}{Dt} = X_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\mu}{\rho}\nabla^2 v_i \tag{8.5}$$

which are the so called **Navier-Stokes** equations for incompressible Newtonian fluids. The Navier-Stokes equations, written out in unbridged form, are

$$\begin{aligned} \tfrac{Du}{Dt} &= X - \tfrac{1}{\rho}\tfrac{\partial p}{\partial x} + \tfrac{\mu}{\rho}\nabla^2 u, \\ \tfrac{Dv}{Dt} &= Y - \tfrac{1}{\rho}\tfrac{\partial p}{\partial y} + \tfrac{\mu}{\rho}\nabla^2 v, \\ \tfrac{Dw}{Dt} &= Z - \tfrac{1}{\rho}\tfrac{\partial p}{\partial z} + \tfrac{\mu}{\rho}\nabla^2 w. \end{aligned} \tag{8.6}$$

Remarks

a) The Navier-Stokes equations for other kinds of fluids can be derived using the same process but with different constitutive equations.

b) The Navier-Stokes equations (8.5) together with the continuity equation (8.2) constitute a system of four partial differential equations for four unknown variables

u, v, w and p and thus is solvable in principle . These four partial differential equations define all possible motions of an incompressible Newtonian fluid. The feature which distinguishes one flow situation from another is the nature of the boundary conditions satisfied by the velocity field v and p.

Boundary Conditions

Normal component of fluid velocity

When a fluid adheres to rigid, but possibly moving, surfaces bounding the fluid, evidently at the rigid surfaces the normal component of fluid velocity must be the same as for the rigid surfaces, as fluids cannot penetrate the solid.

Tangential component of fluid velocity

Two different boundary conditions may be used for the tangential velocity components.

(i) Assume that the tangential velocity component is likewise the same as that of the rigid surface , which is known as no-slip condition.

(ii) Assume that slip can occur between the fluid and the rigid body, which is known as slip condition.

It has been found that the no-slip condition accords with experimental observation on real materials.

Traction boundary condition

In some applications, the traction forces on the surface of the fluid are prescribed, namely

$$t_i = \sigma_{ji} n_j = \bar{t}_i.$$

8.2 Stokes Problem

Basic Equations

From section 8.1, the steady-state motion of an incompressible Newtonian fluid with viscosity μ enclosed in the domain $\Omega \in \mathcal{R}^3$ and acted upon by the volume load f is governed by

$$\begin{array}{lll} \texttt{Stokes equations:} & \mu\triangle u_i - p_{,i} + f_i = 0 & \texttt{in } \Omega \ \ (i = 1,2,3) \\ \texttt{Continuity equation:} & u_{i,i} = 0 & \texttt{in } \Omega \\ \texttt{Boundary condition:} & u_i = 0 & \texttt{on } \partial\Omega \ \ (\texttt{fixed boundary}) \end{array} \tag{8.7}$$

where we have used the index notation with repeated lateral index representing summation over the index range and $()_{,i}$ representing differentiation with respect to x_i.

Variational Statement

Let $v \in V = \{v \in [H_0^1(\Omega)]^3 \mid div\ v = 0 \texttt{ on } \Omega\ \}$ be a test function. To derive the finite element equations, we set

$$\int_\Omega (\mu\triangle u_i - p_{,i} + f_i)\ v_i\ d\Omega \ = \ 0. \tag{8.8}$$

The above integral equation can be simplified by noting that

(i) $\nabla \cdot (v_i \nabla u_i) = \nabla u_i \cdot \nabla v_i + v_i \nabla \cdot \nabla u_i \quad (\nabla \cdot \nabla = \triangle^2)$

$\Rightarrow \quad v_i \triangle u_i = \nabla \cdot (v_i \nabla u_i) - \nabla u_i \cdot \nabla v_i$

Therefore,

$$\begin{array}{rcl} \int_\Omega v_i \triangle u_i\ d\Omega & = & \int_\Omega \nabla \cdot (v_i \nabla u_i)\ d\Omega - \int_\Omega \nabla u_i \cdot \nabla v_i\ d\Omega \\ & = & \int_{\partial\Omega} v_i \nabla u_i \cdot \mathbf{n}\ ds - \int_\Omega \nabla u_i \cdot \nabla v_i\ d\Omega \\ & = & 0 - \int_\Omega \nabla u_i \cdot \nabla v_i\ d\Omega. \end{array}$$

(ii) $p_{,i}v_i = (pv_i)_{,i} - pv_{i,i} = (pv_i)_{,i}$

Therefore,

$\int_\Omega p_{,i}v_i \, d\Omega = \int_\Omega (pv_i)_{,i} \, d\Omega = \int_{\partial\Omega} pv_i n_i \, ds = 0.$

Hence 8.8 becomes

$$\mu \int_\Omega \nabla u_i \cdot \nabla v_i \, d\Omega \; = \; \int_\Omega f_i v_i \, d\Omega. \tag{8.9}$$

Thus, the variational statement for the problem is:

Find $u \in V$ such that

$$a(u,v) = L(v) \quad \forall v \in V \tag{8.10}$$

where $a(u,v) = \mu \int_\Omega \nabla u_i \cdot \nabla v_i \, d\Omega$
$L(v) = \int_\Omega f_i v_i \, d\Omega$
$V = \{v \in [H_0^1(\Omega)]^3 \mid div \; v = 0 \; \texttt{in} \; \Omega\}.$

Finite Element Formulation

We need to construct a finite-dimensional subspace V_n of V. This is not so easy as we have to satisfy the condition $div \; v = 0$. For simplicity, consider 2-D cases in which

$$V = \{v = (v_x, v_y) \in [H_0^1(\Omega)]^2 \mid div \; v = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0 \;\; \texttt{in} \;\; \Omega\}.$$

From calculus, it follows that if Ω does not contain any holes, then $div \; v = 0$ in Ω iff $v = rot \; \varphi \equiv \left(\frac{\partial \varphi}{\partial y} - \frac{\partial \varphi}{\partial x} \right)$.

Thus,

$v \in V \quad \Leftrightarrow \quad v = rot \; \varphi, \;\; \varphi \in H_0^2(\Omega)$

where φ is called stream function connected with the velocity field v.

Now, let W_h be a finite-dimension subspace of $H_0^2(\Omega)$ and define

$$V_h \;=\; \{v \mid v = rot\ \varphi,\ \varphi \in W_h\}.$$

Then, we can formulate a finite element method in the usual way by replacing V by V_h in the equation (8.10).

8.3 Flow of Incompressible Fluids

In this section, we present the finite element method for 2-D flows of incompressible Newtonian fluids. From section 8.1, the governing equations for the problem is

$$\begin{array}{lll}
 & \rho\left(\frac{\partial u_i}{\partial t} + u_j u_{i,j}\right) = -p_{,i} + [\mu(u_{i,j} + u_{j,i})]_{,j} & \texttt{in } \Omega \times I \\
 & u_{i,i} = 0 & \texttt{in } \Omega \times I \\
B.C. & u_i = \bar{u}_i & \texttt{on } \Gamma_u \\
 & t_i = \sigma_{ij} n_j = \bar{t}_i & \texttt{on } \Gamma_t \\
\texttt{where} & \sigma_{ij} = -p\delta_{ij} + \mu(u_{i,j} + u_{j,i}) & \texttt{in } \Omega \\
I.C. & u_i(\mathbf{x}, 0) = u_i^0(\mathbf{x}) & \texttt{in } \Omega \\
\texttt{where} & I \in [0, T]. &
\end{array} \tag{8.11}$$

Variational Statements

Let $r_i(\mathbf{x}, t) = \rho\left(\frac{\partial u_i}{\partial t} + u_j u_{i,j}\right) + p_{,i} - [\mu(u_{i,j} + u_{j,i})]_{,j}$. The method of weighted residuals seeks for $(u, p) \in V \times Q$ such that for every $t \in I$

$$\begin{array}{lll}
(r_i, v_i) = 0 & \forall v_i \in V \ \texttt{and} & v_i = 0 \ \texttt{on } \Gamma_u \\
(u_{i,i}, q) = 0 & \forall q \in Q & \\
u(\mathbf{x}, 0) = u^0 & \texttt{in } \Omega & \\
u_i = \bar{u}_i & \texttt{on } \Gamma_u &
\end{array} \tag{8.12}$$

where V and Q are velocity and pressure spaces and $(\cdot, \cdot)$ is the inner product defined by $(\mathbf{a}, \mathbf{b}) = \int_\Omega \mathbf{a} \cdot \mathbf{b}\ d\Omega$.

The detailed manipulations involving the integrals defined above are presented as follows.

First, consider $(r_i, v_i) = 0$

Let $\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j u_{i,j}$. Then we have from (8.12)

$$\int_\Omega \rho \frac{Du_i}{Dt} v_i d\Omega + \int_\Omega [(pv_i)_{,i} - (pv_{i,i})] d\Omega - \int_\Omega \{[\mu(u_{i,j} + u_{j,i})v_i]_{,j} - \mu(u_{i,j} + u_{j,i})v_{i,j}\} d\Omega = 0$$
$$\Rightarrow$$
$$\int_\Omega \rho \frac{Du_i}{Dt} v_i d\Omega + \int_\Omega (-pv_{i,i} + \mu(u_{i,j} + u_{j,i})v_{i,j}\ d\Omega + \int_{\partial\Omega} [pv_j n_j - \mu(u_{i,j} + u_{j,i})v_i n_j]\ ds = 0 \tag{8.13}$$

As

$$\begin{aligned} -pv_j n_j + \mu(u_{i,j} + u_{j,i})v_i n_j &= [-p\delta_{ij} v_i + \mu(u_{i,j} + u_{j,i})v_i] n_j \\ &= \sigma_{ij} n_j v_i = t_i v_i, \end{aligned}$$

we have from (8.13)

$$\int_\Omega \rho \frac{Du_i}{Dt} v_i\ d\Omega + \int_\Omega [-pv_{i,i} + \mu(u_{i,j} + u_{j,i})v_{i,j}] d\Omega = \int_{\partial\Omega} \bar{t}_i v_i\ ds$$

As $\partial\Omega = \Gamma_u \cup \Gamma_t$ and u is specified on Γ_u, we choose v_i such that $v_i = 0$ on Γ_u.

Therefore, the variational statement of the problem is:

Find $(u, p) \in V \times Q$ such that for every $t \in I$,

$$\begin{array}{ll} \left(\rho \frac{Du_i}{Dt}, v_i\right) - (p, v_{i,i} + (\mu[u_{i,j} + u_{j,i}], v_{i,j}) = b(\bar{t}_i, v_i) & \forall v_i \in V_0 \\ (u_{i,i}, q) = 0 & \forall q \in Q \\ u(\mathbf{x}, 0) = u^0 & \texttt{in } \Omega \\ u_i = \bar{u}_i & \texttt{on } \Gamma_u \end{array} \tag{8.14}$$

$$\texttt{where} \quad V = \{\mathbf{v} | \mathbf{v} \in [H^1(\Omega)]^2\} \quad V^0 = \{\mathbf{v} | \mathbf{v} \in V \ \texttt{and}\ \mathbf{v} = 0 \ \texttt{on}\ \Gamma_u\}$$
$$Q = \{v | v \in H^1(\Omega)\}.$$

Finite Element Formulation

Let $V_h \subset V$ be a N-D subspace of V with basis functions $\{\phi_1,\ \phi_2, ..., \phi_N\}$. Approximating v_i and q in (8.14) by

$$v_{ih} = \sum_{k=1}^{N} \phi_k v_{ik} \ \texttt{and}\ q = \sum_{p=1}^{M} \varphi_p q_p,$$

we have

$$\begin{aligned}&\textstyle\sum_{k=1}^{N}\{(\rho\frac{Du_i}{Dt},\phi_k)+(\mu[u_{i,j}+u_{j,i}],\phi_{k,j})-(p,\phi_{k,i})-b(\bar{t}_i,\phi_k)\}v_{ik} \;=\; 0\\ &\textstyle\sum_{p=1}^{M}(u_{i,i},\varphi_p)q_p \;=\; 0\end{aligned}$$

$$\Rightarrow\begin{cases}(\rho\frac{\partial u_i}{\partial t},\phi_k)+(\rho u_j\frac{\partial u_i}{\partial x_j},\phi_k)+(\mu[u_{i,j}+u_{j,i}],\phi_{k,j})-(p,\phi_{k,i}) \;=\; b(\bar{t}_i,\phi_k)\\ (u_{i,i},\varphi_p) \;=\; 0.\end{cases}\tag{8.15}$$

Approximating u_i and p respectively by

$$u_{ih}=\sum_1^N\phi_\ell u_{i\ell},\quad p_h=\sum_1^M\psi_p p_p,$$

we have from (8.15) that

$$\begin{aligned}&\textstyle\sum_{\ell=1}^{N}\{(\rho\phi_\ell,\phi_k)\dot{u}_{i\ell}+(\rho u_j\phi_{\ell,j},\phi_k)u_{i\ell} \quad +(\mu\phi_{\ell,j},\phi_{k,j})u_{i\ell}+(\mu\phi_{\ell,i},\phi_{k,j})u_{j\ell}\}\\ &\qquad\qquad\qquad\qquad\qquad\qquad -\textstyle\sum_{p=1}^{M}(\psi_p,\phi_{k,i})p_p \;=\; b(\bar{t}_i,\phi_k)\\ &\textstyle\sum_{k=1}^{N}(\phi_{k,i},\psi_p)u_{ik} \;=\; 0\end{aligned}$$

which can be expressed in matrix form by

$$\begin{aligned}M\dot{U}_i+AU_i-CP \;&=\; F\\ -C_1^TU_1-C_2^TU_2 \;&=\; 0\end{aligned}$$

or

$$\begin{bmatrix}M&0&0\\0&M&0\\0&0&0\end{bmatrix}\begin{bmatrix}\dot{U}_1\\\dot{U}_2\\\dot{P}\end{bmatrix}+\begin{bmatrix}2K_{11}+K_{22}+D&K_{12}&-C_1\\K_{21}&K_{11}+2K_{22}+D&-C_2\\-C_1^T&-C_2^T&0\end{bmatrix}\begin{bmatrix}U_1\\U_2\\P\end{bmatrix}=\begin{bmatrix}F_1\\F_2\\0\end{bmatrix},\tag{8.16}$$

where $M=(m_{k\ell})$ with $m_{kl}=(\rho\phi_k,\phi_\ell)$ $(k,\ell=1,2,...,N)$

$K_{ij}=(K_{ij\,k\ell})$ with $K_{ij\,k\ell}=(\mu\frac{\partial\phi_k}{\partial x_i},\frac{\partial\phi_\ell}{\partial x_j})$ $(k,\ell=1,2,...,N;\ i,j=1,2)$

$D=(D_{k\ell})$ with $D_{k\ell}=(\rho u_j\frac{\partial\phi_\ell}{\partial x_j},\phi_k)$

$C_i=(C_{ikp})$ with $C_{ikp}=(\psi_p,\phi_{k,i})$

$F_i=(F_{ik})$ with $F_{ik}=b(\bar{t}_i,\phi_k)=\int_{\Gamma_t}\phi_k\,ds$

Time Integration

Two different kinds of integration schemes, implicit and explicit, can be utilized to solve the system (8.16).

eg. `Backward Euler:` $\dot{M}\frac{U_{n+1}-U_n}{\triangle t} + A(U_{n+1})U_{n+1} = F_{n+1}$ `-- implicit`

`Forward Euler:` $M\frac{U_{n+1}-U_n}{\triangle t} + A(U_n)U_n = F_n$ `-- explicit`.

Note: In constructing a time integration scheme, questions of numerical stability and accuracy must be considered.

EXERCISES

Question

Develop a variational statement for the stokes problem with boundary condition $u_i = 0$ on $\partial\Omega_1$ and $u_i = u_i^0$ on $\partial\Omega_2$.

Chapter 9

Coupled Heat Transfer & Turbulent Flows

This chapter is written based on our recent research results published in the paper (Wiwatanapataphee, Wu, Archapitak & Siew 2004). In our work, a numerical algorithm, based on the Galerkin finite element method and the enthalpy formulation, is developed for solving the coupled turbulent fluid flow and heat transfer problem arising from an industrial process - the continuous steel casting process. The governing equations consist of the continuity equation, the Navier-Stokes equations, the energy equation and the modified $K-\varepsilon$ equations. The formulation of the method is cast into the framework of the Bubnov-Galerkin finite element method. The rest of the chapter is organized as follows. Firstly, the continuous casting process is introduced in section 9.1 together with a brief introduction of the problem to be investigated. In section 9.2, the mathematical model for the solidification process and the turbulence phenomena are presented. In section 9.3, the numerical algorithm is presented followed by some numerical results given in section 9.4.

9.1 The Continuous Casting Process

In this section, we consider a coupled turbulent flow and heat transfer problem arising from the study of the continuous steel casting process. Figure 1 shows the essential feature of the continuous casting process. Molten steel is poured from a tundish through a submerged entry nozzle into a water-cooled mould, where intense cooling causes a thin solidified steel shell to form around the edge of the steel. The solidified steel shell with a liquid pool in the center is then continuously extracted from the bottom of the mould at a constant speed. The product is supported by a set of rollers, after leaving the mould, cooled down by water sprays and then subsequently cooled through radiation. When the completely solidified casting has attained the desired length, it is cut off with a cutter.

The process involves many complex phenomena such as formation of oscillation marks, heat transfer with a moving phase-change boundary and turbulent flow in the mould. The understanding and control of heat transfer and fluid flow is essential for the success of the process. Improper rate of heat extraction from the steel may lead to surface cracks and break-outs of molten steel from the bottom of moulds. The flow field of fluid affects the distribution of inclusion particles and of entrained slag particles which can significantly influence the quality of products. Over the last few decades, intensive studies have been carried out to model various aspects of the process, in particular the flux flow (Fowkes & Woods 1989), the heat transfer and molten steel flow (Bennon & Incropera 1987, Brimacombe, Samarasekera & Laid 1983, Flint 1990, Hill & Wu 1994*a*, Lally, Biegler & Henein 1990, Wu, Hill & Flint 1994). Initial attempts are mainly on heat transfer modeling, but recent work includes both turbulent fluid flow

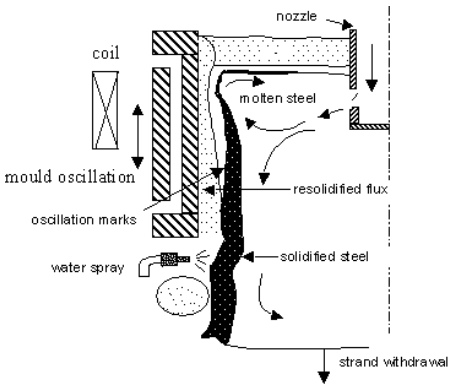


Figure 9.1: The continuous steel casting process

and heat transfer modeling. Most of these models use a velocity field pre-determined by solving the momentum equations independently. Various commercial fluid dynamic packages such as FIDAP (Thomas, Najjar & Mika 1990) and SIMPLE have also been used for the analysis of heat flow and solidification in the casting process.

In spite of extensive modeling studies on fluid flow and heat transfer in the upper region of continuous casting systems, little work has been done to solve the strongly coupled problem of turbulent fluid flow and heat transfer with solidification and to address the techniques for dealing with rapid change of temperature and fluid velocity near the boundary layer. In this work, we focus on this aspect and develop a fully coupled heat transfer - turbulent flow model based on an enthalpy formulation.

9.2 Heat Transfer-Turbulent Flow Model

(a) Governing Equations for Heat Transfer & Solidification

To simulate the process of heat transfer with phase change, a single domain enthalpy method is used. The enthalpy of the material is defined as the sum of sensible heat

$h = cT$ and latent heat H, namely

$$H_t = h + H. \tag{9.1}$$

The latent heat H in general can be expressed as

$$H = f(T)L, \tag{9.2}$$

where L is the latent heat of steel and $f(T)$ is the local liquid fraction with value one representing liquid state and zero for solid state. The liquid fraction generally is a nonlinear function of temperature T but for simplicity, it is approximated here by the following linear function

$$f(T) = \begin{cases} 0 & T \leq T_s \\ \dfrac{T - T_s}{T_L - T_s} & T_s < T < T_L \\ 1 & T \geq T_L \end{cases} \tag{9.3}$$

where T_L and T_s are respectively the melting temperature and solidification temperature of steel.

From the principle of energy conservation, we have the following equation for the region undergoing a phase change

$$\rho \left(\frac{\partial H_t}{\partial t} + u_j H_{t,j} \right) = (k_0 T_{,j})_{,j}, \tag{9.4}$$

where we have used and will continue to use the suffix notation with repeated literal index representing summation over the index range and $(.)_{,j}$ denoting differentiation with respect to x_j (Sokolnikoff 1986), u_j represents the velocity component of fluid in the x_j direction, ρ and k_0 are respectively the density and molecular thermal conductivity of steel. Now on using the definition for the enthalpy H_t, we have

$$\rho c \left(\frac{\partial T}{\partial t} + u_j T_{,j} \right) = (k_0 T_{,j})_{,j} - S_T \tag{9.5}$$

with the source term S_T representing the rate of change of the volumetric latent-heat given by

$$S_T = \rho \left(\frac{\partial H}{\partial t} + u_j H_{,j} \right). \tag{9.6}$$

Obviously, $S_T = 0$ everywhere except in the region where phase change occurs. Now with equation (5), there is no need to consider the solidified region and the liquid region separately and there are no conditions to be satisfied at the phase-change boundary. The equation can be applied to all the regions including the solid region, the mushy region and the liquid region. Thus, the heat transfer - phase change problem can be solved by using a single domain approach.

(b) Governing Equations for Turbulent Flows

The influence of turbulence on the transport of momentum and energy is modeled by the addition of the turbulent viscosity μ_t to the laminar viscosity μ_0 and the turbulent conductivity $k_t = c\mu_t/\sigma_t$ to the molecular conductivity k_0, yielding the effective viscosity μ and the effective thermal conductivity k given by

$$\mu = \mu_0 + \mu_t, \qquad k = k_0 + \frac{c\mu_t}{\sigma_t}, \tag{9.7}$$

where σ_t is the turbulent Prandtl number (Launder 1988). Thus, the unified field equations governing the multiphase heat transfer and fluid flow with turbulence, for all the regions with or without phase change, are as follows

$$u_{i,i} = 0, \tag{9.8}$$

$$\rho \left(\frac{\partial u_i}{\partial t} + u_j u_{i,j} \right) + p_{,i} - (\mu(u_{i,j} + u_{j,i}))_{,j} = F_i(u_i, x_i, t), \tag{9.9}$$

$$\rho c\left(\frac{\partial T}{\partial t}+u_j T_{,j}\right)=(kT_{,j})_{,j}-S_T, \tag{9.10}$$

where Darcy's law for porous media (Reddy & Reddy 1992) has been used for modeling the flow in the mushy region and $F_i(u_i, x_i, t)$ is thus determined by

$$F_i(u_i,x_i,t)=C\frac{\mu\left[1-f(T)\right]^2}{\rho f(T)^3}(u_i-U_i), \tag{9.11}$$

in which $U=(0,U_2)$ denotes the downward velocity of the solidified steel shell.

Equations (8)-(10) do not constitute a closed system as both μ and k are related to an unknown function μ_t. Various models, such as the mixing-length type model, the one-equation model and the two-equation $(K-\varepsilon)$ model, have been proposed for calculating μ_t. Ferziger (Ferziger 1987) and Launder (Launder & Spalding 1974), based on a critical review, suggested that the simple mixing-length type model is suitable for most boundary-layer type flows in the absence of recirculation; the one-equation model can be used to model simple recirculation flows; while for more complex flow fields, the two-equation model should be used. As the flow field in the continuous casting mould is complex with circulation, we use the two-equation $(K-\varepsilon)$ model for calculating μ_t.

With the standard $K-\varepsilon$ model, the turbulent viscosity μ_t is determined (Ferziger 1987, Launder & Spalding 1974) by

$$\mu_t=\frac{\rho C_\mu K^2}{\varepsilon}, \tag{9.12}$$

where C_μ is suggested to be 0.09, the turbulent kinetic energy K and its dissipation rate ε are determined by

$$\rho\left(\frac{\partial K}{\partial t}+u_j K_{,j}\right)-(\frac{\mu_t}{\sigma_K}K_{,j})_{,j}=-\frac{\mu_t}{\sigma_t}\beta g_j T_{,j}+\mu_t G-\rho\varepsilon, \tag{9.13}$$

$$\rho\left(\frac{\partial\varepsilon}{\partial t}+u_j\varepsilon_{,j}\right)-(\frac{\mu_t}{\sigma_\varepsilon}\varepsilon_{,j})_{,j} = C_1(1-C_3)\frac{\varepsilon\mu_t}{K\sigma_t}\beta g_j T_{,j} + C_1\frac{\varepsilon}{K}\mu_t G-\rho C_2\frac{\varepsilon^2}{K}, \quad (9.14)$$

where $G=2\epsilon_{ij}\epsilon_{ij}$ with $\epsilon_{ij}=\left(u_{i,j}+u_{j,i}\right)/2$. The constants involved in equations (7) - (14) are empirical constants. Extensive examination of turbulent flows has resulted in a recommended set of values for these constants (Launder & Spalding 1974), namely $\sigma_t=0.9, \sigma_k=1,\ \sigma_\varepsilon=1.25, C_\mu=0.09, C_1=1.44, C_2=1.92, C_3=0.8$.

It has been well established that the above standard $K-\varepsilon$ model is applicable only to the highly turbulent region (far-wall region) and cannot be applied to the near-wall region where viscous effects become dominant. In the modelling of the continuous casting process, due to the phase change, the computational region typically includes three different sub-regions including the solidified steel region, the mushy region and the molten steel region. Obviously, the standard $K-\varepsilon$ model is only applicable to the region far from the solidified steel layer. Thus, in order to have a unified model applicable to all the three regions, some modifications to the standard $K-\varepsilon$ model are needed. Through intensive research over the last few decades, various techniques have been proposed for modelling flows near solid boundary such as the wall function approach and the low-Reynolds number $K-\varepsilon$ model (Chien 1982, Driest 1996, Jaeger & Dhatt 1992, Jones & Launder 1973, Lam & Bremhorst 1981, Nagano & Hishida 1987, Reddy & Reddy 1992). In the continuous steel casting, the boundary of the solidified steel shell is not known a priori, thus the wall-function technique is not applicable to the problem. We therefore use the low-Reynolds number $K-\varepsilon$ model to accommodate the region with relatively low local turbulent Reynolds number, in which some damping functions are added

into the standard $K-\varepsilon$ equations to reduce the effect of turbulence across the viscous sub-layers. Firstly, based on the work in (Lam & Bremhorst 1981, Patel, Rodi & Scheuerer 1985), the constant C_μ is modified to

$$C_\mu = 0.09 f_\mu, \tag{9.15}$$

where f_μ represents the generalized damping mechanism of turbulent transport in both the liquid and mushy regions and is determined by

$$f_\mu = \sqrt{f(T)} exp(-3.4/(1+R_t/50)^2), \tag{9.16}$$

where $f(T)$ is the liquid fraction as defined before in (3), R_t denotes the local turbulent Reynolds number defined by

$$R_t = \frac{\rho K^2}{\mu\varepsilon}. \tag{9.17}$$

To ensure that all the terms in equations (9.13) and (9.14) will not tend to infinity as K approaches zero in the near-wall region, the last term of the right hand side of equation (14) is multiplied by a damping function f_ε defined by

$$f_\varepsilon = 1 - A_\varepsilon e^{-R_t^2}, \tag{9.18}$$

where A_ε is a constant and is chosen as one if $K < 10^{-4}$ or otherwise $A_\varepsilon = 0.3$ (Jaeger & Dhatt 1992, Jones & Launder 1973). By choosing $A_\varepsilon = 1$ in (9.18), the new term $\rho C_2 f_\varepsilon \varepsilon^2/K$ in (9.14) approaches zero as K becomes small.

9.3 Finite Element Solution

For two dimensional problems, equations (7) - (14) can be manipulated to yield a closed system of six partial differential equations in terms of six coordinate and time-

dependent unknown functions (u_1, u_2, p, T, K and ε). The system, once supplemented by the initial and boundary conditions, can be solved numerically to yield the velocity and temperature fields and to determine the phase-change boundary. In this work, the boundary conditions considered for each field variable include the Dirichlet type and the Neumann/Robin type,i.e,

$$q = \bar{q}, \text{ on } \partial\Omega_q,$$
$$\frac{\partial q}{\partial n} = g(q), \text{ on } \partial\Omega_{q_2},$$

where $\partial\Omega = \partial\Omega_q \bigcup \partial\Omega_{q_2}$ denotes the boundary of the computation domain Ω, q referes to u_1, u_2, p, T, K and ε.

To solve the problem, firstly, the penalty function method is used to weaken the continuity requirement (9.8) by the following equation

$$u_{j,j} = -\delta p, \tag{9.19}$$

where δ is a small positive number. The effect of the penalization is simply to relax the incompressibility condition (8). For more details on the mathematical aspect of the method, the reader is referred to references (Falk 1975). Thus, the pressure variable can be eliminated from the system, overcoming the difficulty associated with proper imposition of the pressure boundary condition. Hence, we end up with a system of five partial differential equations in terms of five unknown functions u_1, u_2, T, K and ε. To develop the variational statement for the boundary value problem, we consider the following integral representation of the problem:

> Find u_1, u_2, T, K and $\varepsilon \in H^1(\Omega)$ such that for all test functions $w^1 \in H^1_{ou_1}(\Omega), w^2 \in H^1_{ou_2}(\Omega), w^T \in H^1_{oT}(\Omega), w^K \in H^1_{oK}(\Omega)$ and $w^\varepsilon \in H^1_{0\varepsilon}(\Omega)$, all

the Dirichlet boundary conditions for the unknown functions are satisfied

and

$$
\begin{aligned}
&(\frac{\partial u_i}{\partial t}, w^i) + (u_j u_{i,j}, w^i) - ((\frac{\mu}{\rho}(u_{i,j} + u_{j,i}))_{,j}, w^i) - (\frac{1}{\rho\delta} u_{j,ji}, w^i) = (\frac{1}{\rho} F_i, w^i),\\
&(\frac{\partial T}{\partial t}, w^T) + (u_j T_{,j}, w^T) - ((\frac{k}{\rho c} T_{,j})_{,j}, w^T) = -\frac{1}{c}\left\{(\frac{\partial H}{\partial t}, w^T) + (u_j H_{,j}, w^T)\right\},\\
&(\frac{\partial K}{\partial t}, w^K) + (u_j K_{,j}, w^K) - ((\frac{\mu_t}{\rho\sigma_K} K_{,j})_{,j}, w^K) = -(\frac{\mu_t}{\rho\sigma_t}\beta g_j T_{,j} - \frac{\mu_t}{\rho} G + \varepsilon, w^K),\\
&(\frac{\partial \varepsilon}{\partial t}, w^\varepsilon) + (u_j \varepsilon_{,j}, w^\varepsilon) - ((\frac{\mu_t}{\rho\sigma_\varepsilon}\varepsilon_{,j})_{,j}, w^\varepsilon) = (C_1(1-C_3)\frac{\varepsilon\mu_t}{K\rho\sigma_t}\beta g_j T_{,j}\\
&\qquad\qquad + C_1\frac{\varepsilon\mu_t}{K\rho} G - C_2 f_\varepsilon \frac{\varepsilon^2}{K}, w^\varepsilon),
\end{aligned}
\tag{9.20}
$$

where $(\cdot,\cdot)$ denotes the inner product on the square integrable function space $L^2(\Omega)$, $H^1(\Omega)$ is the Sobolev space $W^{1,2}(\Omega)$ with norm $||\cdot||_{1,2,\Omega}$, $H^1_{0q}(\Omega) = \{v \in H^1(\Omega) | v = 0 \text{ on } \partial\Omega_q\}$. A standard procedure is then carried out to reduce the second order derivatives involved in the above problem into the first order ones using integration by parts. For example, by using integration by parts and the Neumann/Robin type boundary condition, we have

$$
((\frac{\mu}{\rho} u_{i,j})_{,j}, w^i) = -(\frac{\mu}{\rho} u_{i,j}, w^i_{,j}) + (\frac{\mu}{\rho} g(u_i), w^i)_B, \tag{9.21}
$$

where $(\cdot,\cdot)_B$ denotes the inner product on $L^2(\partial\Omega_{u_{i2}})$. Through this process, all second order derivatives in (20) are reduced to first order ones ensuring that all integrals involved are well defined.

To find the numerical solution of the problem, we pose the variational problem into an N-dimension subspace. The computation domain Ω is discretized into a finite

number of elements connected by N nodes. Let $\mathbf{U}, \mathbf{T}, \mathbf{K}$ and $\mathbf{E}$ denote respectively the global vectors with each *ith* entry representing the value of the corresponding unknown function at the *ith* node of the finite element mesh. Then, by using the Galerkin finite element formulation, we obtain the following sets of ordinary differential equations

$$\begin{aligned} M_u\dot{\mathbf{U}} + A_u\mathbf{U} &= \mathbf{F}, \\ M\dot{\mathbf{T}} + A_T\mathbf{T} &= M'\dot{\mathbf{H}} + A'\mathbf{H} + \mathbf{F}_b, \\ M\dot{\mathbf{K}} + A_K\mathbf{K} &= \mathbf{F}_K, \\ M\dot{\mathbf{E}} + A_\varepsilon\mathbf{E} &= \mathbf{F}_\varepsilon, \end{aligned} \tag{9.22}$$

where the superposed dot represents differentiation with respect to time and all coefficient matrices are global matrices assembled from element matrices. Matrices M_u, M and M' correspond to the transient terms, matrices A_u,A_T, A', A_K and A_ε correspond to the advection and diffusion terms, and vector $\mathbf{F_b}$ are due to heat-flow at the boundary, vector $\mathbf{F}$ provides forcing functions for the Navier-Stokes equations, and vectors $\mathbf{F}_K$ and $\mathbf{F}_\varepsilon$ are respectively due to $K-$production dissipation and $\varepsilon-$production dissipation. To keep details to minimum, formulae for calculation of the global coefficient matrices and vectors are not given here.

From (9.22) a time integration scheme is then developed to find the finite element solutions of the temperature and velocity fields at any instant of time. For more detail, the reader is referred to reference [45].

9.4 Numerical Investigation

A test example is given here to demonstrate the validity of the mathematical model. The example under consideration is a slab caster, with a mould width of 1750 mm, a

narrow-face width of 236 mm and a depth of 800 mm. The submergence depth of an entry nozzle is 230 mm, the nozzle ports are rectangular with a height of 76 mm and a width of 54 mm. The port angle is 15^o downward. Other system parameters are as follows. The molten steel delivery velocity $U_{in} = 1.08244\ m/s$, the melting temperature of steel $T_L = 1525^oC$, solidification temperature $T_S = 1465^oC$, temperature of cooling water $T_\infty = 20^oC$, external temperature $T_{exit} = 100^oC$, density of steel $\rho = 7800\ kg/m^3$, laminar viscosity $\mu_0 = 0.001\ pa \cdot s$, specific heat $c = 465\ J/kg^oC$, thermal conductivity of steel $k_0 = 35\ W/m^oC$, latent heat $L = 2.72 \times 10^5\ J/kg$, surface heat transfer coefficient $h_\infty = 1079.45\ W/m^{2o}C$, emissivity of solid steel $\varpi = 0.4$, Stefan-Boltzmann constant $\sigma = 5.66 \times 10^{-8}\ W/m^2K^4$, morphology constant $C = 1.8 \times 10^6 = m^{-2}$, casting speed $U_2 = 0.02167m/s$. The molten steel has 5^oC of super-heat above the melting temperature. The delivery turbulent kinetic energy and its dissipation rate are respectively 0.0502 m^2/s^2 and 0.457 m^2/s^3.

Figure 2(a) shows the velocity vectors in the upper part of the solution domain. The flow pattern shows that there exist two circulation zones in the top part of the casting. The flow become parallel further downstream. Figure 2(b) shows the temperature distribution in the first 3 metres below the meniscus. The temperature profiles clearly outline the path of the hot steel and show how the fluid carries heat with it.

Distribution of the turbulence quantities K and ε are shown in Figure 3. The values of K and ε are very high near the nozzle opening. Close to the solid boundary, the level of turbulence approaches zero. Values of turbulent kinetic energy and its dissipation rate rapidly decrease in the circulation region and then reach the smallest level in the solidified steel shell.

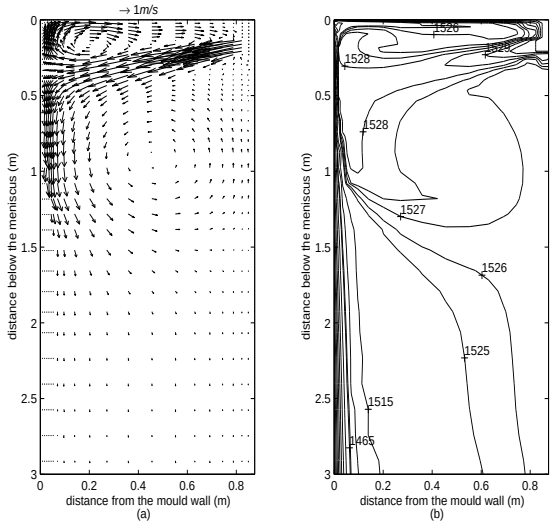


Figure 9.2: Velocity and temperature profile (a) velocity vectors (m/s), (b) temperature contours (^oC)

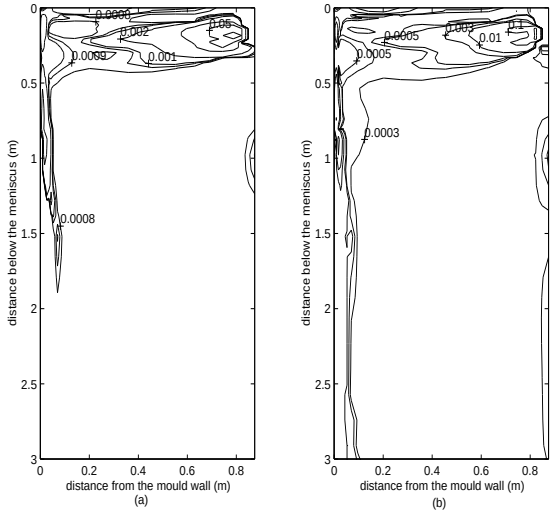


Figure 9.3: Contour plot of (a) turbulent kinetic energy $K(m^2/s^2)$ and (b) dissipation rate $\varepsilon(m^2/s^3)$

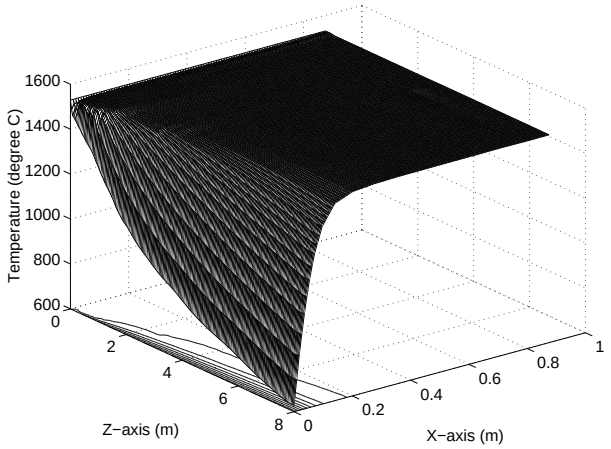


Figure 9.4: Temperature distribution in the computational region

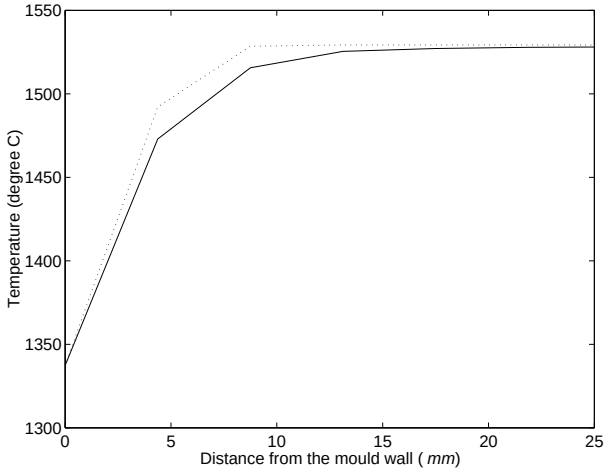


Figure 9.5: Comparison of temperature profiles at the bottom of the mould obtained by models with turbulence effect (solid line) and with no turbulence effect (dotted line)

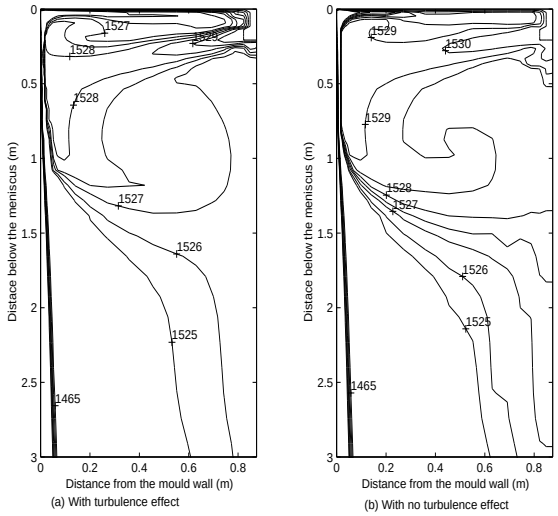


Figure 9.6: Comparison of temperature profiles in the first 3 metres below meniscus obtained by models with turbulence effect and with no turbulence effect

Figure 4 shows the distribution of temperature field in the computational region. It indicates that temperature drops very fast near the strand surface.

Figure 5 shows comparison of the temperature profiles at the bottom of the mould obtained by models with turbulence effect and with no turbulence effect. It is noted that the temperature profiles at the bottom of the mould obtained by the model with turbulence effect is slightly lower than that with no turbulence effect.

Figure 6 shows the comparison of temperature contours in the first 3 metres below the meniscus obtained by the models with and without turbulence effect. It is noted that the average temperature from the model with turbulence effect is lower than that with no turbulence effect in this region.

Chapter 10

Multi-Phase Flows under EM Force

This chapter is written based on our recent research results published in the paper (Wu & Wiwatanapataphee 2007). In our work, we develop a mathematical model and finite element based numerical technique to study the coupled turbulent flow and heat transfer process in continuous steel casting under electromagnetic force. The complete set of field equations are established and solved numerically. The influences of electromagnetic field on flow pattern of molten steel and temperature field as well as steel solidification are presented in the paper. The rest of the chapter is organized as follows. In section 10.1, a brief introduction of the problem to be investigated is given. In section 10.2, the governing boundary value problem is presented followed by numerical formulation in section 10.3 and numerical results in section 10.4.

10.1 Steel Casting with Electromagnetic Stirring

Continuous steel casting is a heat extraction process for casting steel products from molten steel. In this process, molten steel is poured continuously from a tundish through a submerged entry nozzle into a water-cooled mould where intensive cooling results in a thin solidified steel shell to form around the edge of the casting. The solidified steel shell with a liquid pool in the center is then continuously withdrawn from the bottom of the mould at a constant speed, as shown in Figure 9.1. To control the fluid flow pattern and the steel solidification process, an electromagnetic field, generated from the source current through the coil, is imposed to the system. The magnetic field induces electric currents in molten steel and consequently generates a body force, namely the electromagnetic force or Lorentz force. This body force acts on the molten steel and consequently influences the flow of the molten steel and the steel solidification process.

Over the last few decades, extensive studies have been carried out worldwide to model various aspects of the continuous casting process, in particular the heat transfer and steel solidification process (Hill & Wu 1994*b*, Wu et al. 1994), the electromagnetic stirring (Jenkins & Hoog 1996), the flow phenomena (Thomas 1990) and the formation of oscillation marks. However, as analyzed by Thomas (Thomas 2001), the continuous casting process involves a staggering complexity of at least eighteen interacting phenomena at the mechanistic level. Due to this complexity, in the past, research was focussed mainly on the modelling of each individual phenomenon or interaction of two or three phenomena only. Hence, it is a worthwhile undertaking to develop a sophisticated model capable of dealing with the staggering complexity of interacting phenomena including turbulence, convection heat transfer, phase change and

electromagnetic stirring.

In this work, we further develop our coupled heat transfer - turbulent flow model by incorporating the effect of the electromagnetic field. The rest of the paper is organized as follows. In section two, a complete set of field equations are presented. In section three, a brief description of the solution method is given. In section four, a numerical study is presented to demonstrate the influence of the electromagnetic field on the flow of molten steel in the central liquid pool and the distribution of temperature as well as solidification of steel.

10.2 Mathematical Model

The continuous casting process involves many complex phenomena including turbulent flow, heat transfer with phase change and electromagnetic stirring. These phenomena interact one with another and thus the modelling of the continuous casting process constitutes one of the most outstanding mathematical modelling problems.

A single domain enthalpy method is used to simulate the heat transfer and steel solidification. Based on the principle of energy conservation and the formulation in section 9.2(a) the equation for the heat transfer process in the continuous casting is

$$\rho c \left(\frac{\partial T}{\partial t} + u_j T_{,j} \right) = (k_0 T_{,j})_{,j} - \rho \left(\frac{\partial H}{\partial t} + u_j H_{,j} \right) \tag{10.1}$$

It should be remarked here that the last term on the right hand side of equation (10.1) is equal to zero everywhere except in the region where phase change occurs. Hence, equation (10.1) can be applied to all the regions including the solid region, the mushy region and the molten steel region.

To model the flow of molten steel in the central liquid pool, the molten steel

is assumed as an incompressible Newtonian fluid. The flow in the mushy region is modeled by Darcy's law for porous media. Thus, the unified field equations governing the fluid flow for all the regions with or with no phase change are as follows

$$u_{i,i} \;=\; 0, \tag{10.2}$$

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j u_{i,j}\right) + p_{,i} - (\mu_f(u_{i,j} + u_{j,i})_{,j} \;=\; F_i(u_i, x_i, t) + \rho g_i + F_{emi}, \tag{10.3}$$

where F_i is determined by

$$F_i(u_i, x_i, t) \;=\; C\frac{\mu_f[1 - f(T)]^2}{\rho f(T)^3}(u_i - (U_{cast})_i) \tag{10.4}$$

The influence of electromagnetic field on the transport of momentum is modeled by the addition of the electromagnetic force in the momentum conservative equation (10.3). Based on on our previous work in (Archapitak, Wiwatanapataphee, Wu & Tang 2004), the electromagnetic force can be determined by

$$\mathbf{F}_{em} = \mathbf{J} \times (\nabla \times \mathbf{A}) \tag{10.5}$$

where $\mathbf{A}$ is the magnetic vector potential which is governed by the following equation

$$\frac{1}{\mu}\nabla \times (\nabla \times \mathbf{A}) \;=\; \mathbf{J} \tag{10.6}$$

where $\mathbf{J} = \mathbf{J}_s - \sigma\frac{\partial \mathbf{A}}{\partial t} - \sigma\nabla\phi$, μ and σ denote the magnetic permeability and electroconductivity, $\mathbf{J}_s$ is the source current density, ϕ is a scalar potential function. It should be addressed that in deriving equation (9) from the Maxwell's equations, we have neglected the influence of the displacement current and flow induced current on the magnetic field generated by the source current.

The influence of turbulence on the transport of momentum and energy is modeled by the addition of the turbulent viscosity μ_t to the laminar viscosity μ_0 and the

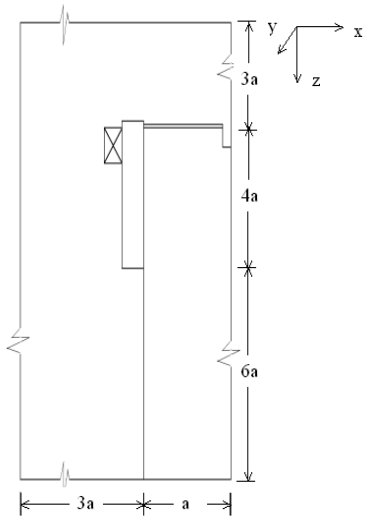


Figure 10.1: Computation domain (a = 0.1 m.)

turbulent conductivity $k_t = \frac{c\mu_t}{\sigma_t}$ to the molecular conductivity k_0 as described in section 9.2(b).

10.3 Method of Solution

As shown in section two, the electromagnetic field problem can be uncoupled from the fluid flow - heat transfer problem. Thus, the electromagnetic field problem is solved first to yield the electromagnetic force for subsequent fluid flow and heat transfer analysis.

For the electromagnetic field, in this work, we are concerned with two-dimensional problems with $\mathbf{A}$, $\mathbf{J}$ and ϕ taking the following forms in the coordinate system as shown in Figure 10.1,

$$\mathbf{A} = (0,\ A_2(x,z,t),\ 0),\ \ \mathbf{J} = (0,\ J_2(x,z,t),\ 0),\ \ \phi = \text{constant}. \tag{10.7}$$

Substituting equation (14) into (9), we have

$$A_{2,jj} \;=\; \mu\sigma\frac{\partial A_2}{\partial t} - \mu J_{s2}. \tag{10.8}$$

For the case of sinusoidal source current, i.e. $J_{s2} = j_s e^{i\omega t}$, the above equation admits solution of the following form

$$A_2 \;=\; a(x,z)e^{i\omega t} \tag{10.9}$$

and equation (15) becomes

$$a_{,jj} - \beta^2 a = -\mu j_s, \tag{10.10}$$

where $\beta^2 = \sigma\mu\omega i$ and $i = \sqrt{-1}$.

To solve equation (17) numerically, we firstly develop the following associated variational boundary value problem:

Find $a \in H_0^1(\Omega)$ such that $\forall w \in H_0^1(\Omega)$

$$(a_{,j},\; w_{,j}) + \beta^2(a,\; w) \;=\; \mu(j_s,\; w) \tag{10.11}$$

where w is the so called weight function or test function, $(\cdot,\cdot)$ denotes the inner product on the square-integrable function space $L^2(\Omega)$ and $H_0^1(\Omega)$ is defined as follows

$$H_0^1(\Omega) = \left\{ v | v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial z} \in L^2(\Omega) \;\text{ and }\; v = 0 \;\text{ on }\; \partial\Omega \right\}. \tag{10.12}$$

The Galerkin finite element method is then used to discretize the problem in space to yield the following system of equations

$$\sum_{i=1}^{N} \left[(\psi_{i,j}, \psi_{k,j}) + \beta^2(\psi_i, \psi_k) \right] a_i \;=\; \mu(j_s, \psi_k) \quad (k = 1, 2, ..., N). \tag{10.13}$$

Once the solution of the above system is obtained we can determine the magnetic vector potential $\mathbf{A}$ by

$$\mathbf{A} \;=\; e^{i\omega t}[0,\; a(x,z),\; 0] \tag{10.14}$$

and consequently we can calculate $\mathbf{J}$ and $\mathbf{F}_{em}$.

For the coupled fluid flow - heat transfer problem, firstly, the penalty function method is used to weaken the continuity requirement by

$$u_{j,j} = -\delta p \tag{10.15}$$

where δ is a small positive number. Thus, the pressure variable can be eliminated from the system. Hence, for two dimensional cases, we have a closed system of partial differential equations, (10.1)-(10.6), in terms of five coordinate and time-dependent unknown functions (u_1, u_2, T, K and ε). To find the numerical solution, the governing partial differential equations are discretized in space by the Galerkin finite element method to yield the following system of nonlinear ordinary differential equation

$$M\dot{\mathbf{q}} + K\mathbf{q} \;=\; \mathbf{f}(\mathbf{q}), \tag{10.16}$$

where $\mathbf{q} = \{(u_{1i},\ u_{2i},\ T_i,\ K_i,\ \varepsilon_i)\}_{i=1}^{N}$ represent the values of $u_1,\ u_2,\ T,\ K$ and ε on the finite element nodes ($i = 1, 2, ..., N$). The matrix M corresponds to the transient terms in the governing partial differential equations, the matrix K corresponds to the advection and diffusion terms, and the vector $\mathbf{f}$ depends nonlinearly on $u_i,\ T,\ K$ and ε. To keep details of the paper to minimum, the specific form of each of the matrices and vectors are omitted here.

The numerical solutions to the nonlinear discretization system with appropriate boundary conditions are then obtained by using an iterative scheme. The following convergence condition was used in the simulation

$$\frac{\|R_i^{m+1} - R_i^m\|}{\|R_i^m\|} \;\leq\; tol, \tag{10.17}$$

Table 10.1: Parameters used in numerical simulation

Parameter	Symbol	Value	Unit
Pouring temperature	T_m	1530	^oC
Molten temperature	T_L	1525	^oC
Solidification temperature	T_S	1465	^oC
Density	ρ	7850	kg/m^3
Viscosity	μ_0	0.001	$pa \cdot s$
Specific heat	c	465	J/kg^oC
Thermal conductivity of steel	k_0	35	W/m^oC
Latent heat	L	2.72×10^5	J/kg
Morphology constant	C	1.8×10^6	m^{-2}
Casting speed	U_{cast}	0.028	m/s
Magnetic permeability of vacuum	μ	$4\pi \times 10^{-7}$	Henry/m
Electric conductivity	σ		$\Omega^{-1}m^{-1}$
- steel		4.032×10^6	
- coil		1.163×10^7	
Electric permittivity of vacucm	ϵ	8.8540×10^{-12}	Farad/m

where the superscript $m+1$ and m denote iterative computation steps, R_i is residual and *tol* is a small positive constant.

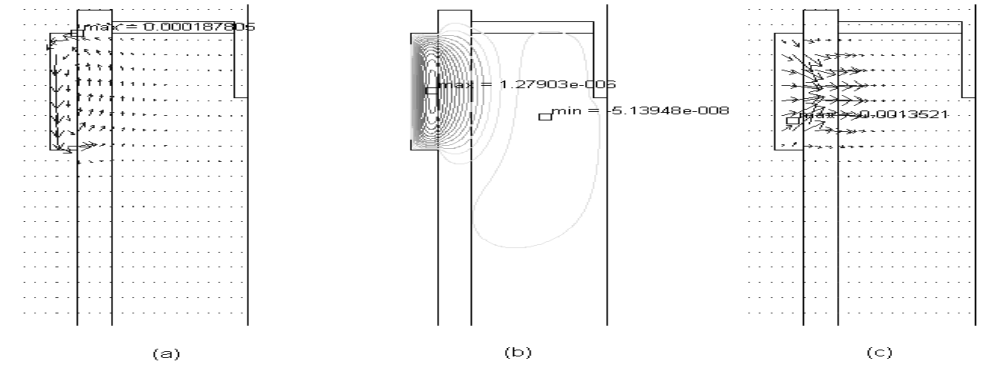
10.4 Numerical Investigation and Discussion

The influence of electromagnetic field on the coupled turbulent flow and steel solidification is investigated in the present study. The example under investigation is a rectangular caster which has a width of 0.1 m and a depth of 0.4 m in the $x-z$ plane. The computation region is as shown in Figure 2. The finite element mesh, used in this study, consists of 15,104 tetrahedron elements with a total of 99,889 degrees of freedom. The system parameters are as listed in Table 1.

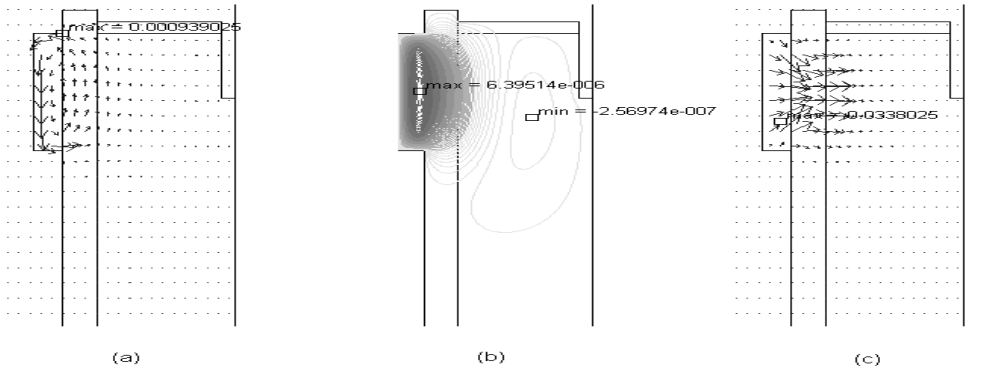
Figure 3 shows the magnetic flux density vector $\mathbf{B}$ (i.e. $\nabla \times \mathbf{A}$), the contour plot of $a(z,t)$ and the electromagnetic force $\mathbf{F}_{em}$ corresponding to different external source current

density j_s. The results show that the electromagnetic force acts on the molten steel basically in the horizontal direction toward the central line. This force will contribute to preventing molten steel from sticking to the mould wall and smoothing the steel casting surface. The results have also shown that the magnitude of the force can be controlled by controlling the imposed source current density. The magnitude of the force increases as the current density increases as shown in Figure 5.

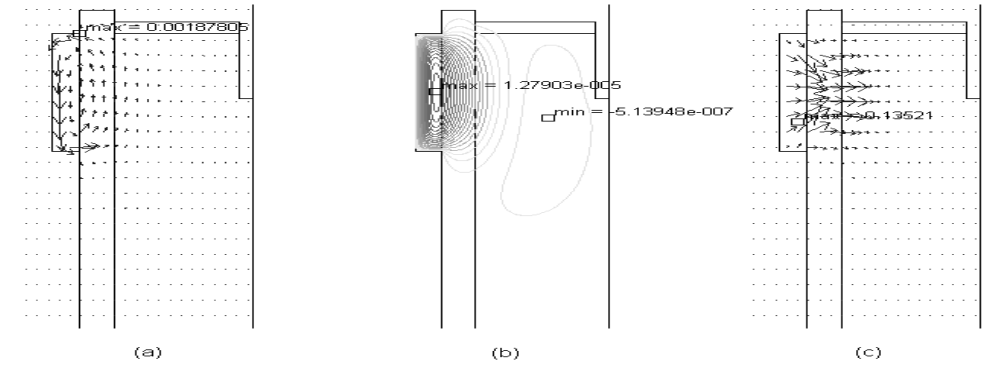
Figure 4 shows the influences of source current density on velocity and temperature fields in molten steel. The electromagnetic field applied to the system suppresses the melt flow and results in reduction of velocity in the mould region and leads to more uniform melt flow below the mould. The suppression of the jet melt flow, by the imposed electromagnetic field, causes the reduction of advective heat transfer to the casting surface. Therefore, superheat is not removed sufficiently on the casting surface, resulting in the increase of energy level in the overall liquid region and the increase in temperature gradients near the solidified shell. The increased temperature gradient near the solidifying shell increases the diffusion heat flux to the shell surface, resulting in thicker solidified shell. The temperature profiles on a horizontal section 0.4 m below the meniscus (i.e. at exit of the mould) for various different current densities are shown in Figure 6. With the increase of current density, the thickness of the solidified shell increases significantly.



(1) $J_s^e = 10,000A/m^2$

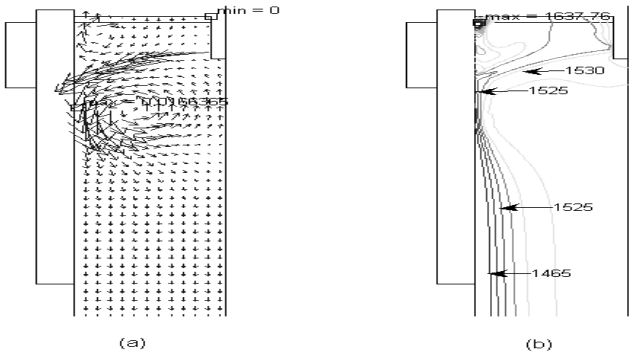


(2) $J_s^e = 50,000A/m^2$

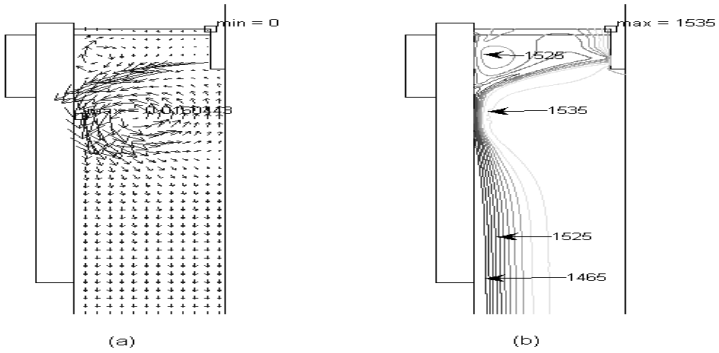


(3) $J_s^e = 100,000A/m^2$

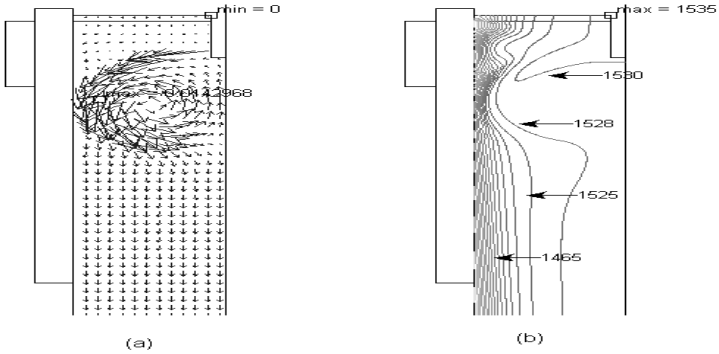
Figure 10.2: Influence of external current density on (a) the magnetic flux density $\mathbf{B}$; (b) the magnetic potential $\mathbf{A}_z$; (c) The electromagnetic force $\mathbf{F}_{em}$.



(1) $J_s^e = 10,000A/m^2$



(2) $J_s^e = 50,000A/m^2$



(3) $J_s^e = 100,000A/m^2$

Figure 10.3: Influence of external current density on the fluid flow and heat transfer (a) velocity field of molten steel; (b) Temperature profiles.

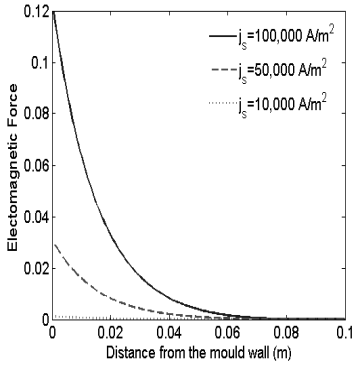


Figure 10.4: Influence of source current density on the magnitude of electromagnetic force at the horizontal section 0.055 m below the meniscus.

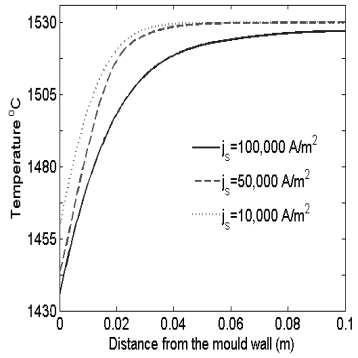


Figure 10.5: Influence of source current density on the temperature profile at the horizontal section 0.4 m below the meniscus.

Chapter 11

Blood Flows in Stenosed Arteries

This chapter is written based on our recent research results published in the papers (Wiwatanapataphee, Poltem, Wu & Lenbury 2006, Wiwatanapataphee 2008). In our work, a numerical technique based on the finite element method is developed to simulate the flow of blood through stenosed coronary arteries taking into account of arterial wall deformation under pulsatile flow condition. In section 11.1, a brief introduction is given to describe the background of the problem investigated. In section 11.2, a complete set of equations is presented for the flow of blood through stenotic arteries followed by the solution method in section 11.3 and numerical results in section 11.4.

11.1 Stenosis and Cardiovascular Disease

The blood circulatory system consists of various parts such as the heart, the arterial and the venous systems as well as the microcirculatory systems. The heart contracts

and relaxes about 70 times per minute to push blood through the thousands of arteries and veins. Fig. 11.1 shows the blood circulation in the heart. The right ventricle of the heart pumps blood through the arteries to the lungs where the red blood cells absorb the oxygen and release the carbon dioxide. Then the bright red, freshly-oxygenated blood goes to the left ventricle of the heart from which blood is pumped into the arteries and goes around the body. As it goes along the body, the blood transfers oxygen to the body cells and receive the waste product. Then, the impure blood flows into the right auricle through veins. The live arteries can change and adapt to new hemodynamic condition.

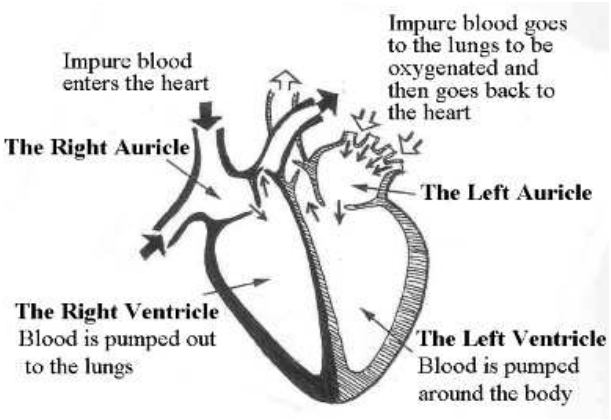


Figure 11.1: The blood circulation in the heart

The presence of unusual hemodynamic condition in the arteries often creates abnormal biological responses. Skewing of the blood speed in some region could cause oscillating direction of wall shear stress which can create pockets leading to atherosclerotic disease. The skewing of velocity tends to be localized and results in the narrowing of the artery lumen - a stenosis. In the arteries with high grade stenoses, very high shear stresses near the throat of the stenosis can activate the platelets. This induces thrombosis, and

may totally block the blood flow to the heart and lead to heart attacks and stokes.

Cardiovascular disease is one of the major causes of death in developed countries. Most of the cases are associated with some form of abnormal blood flow in arteries due to the existence of stenoses. When the coronary artery is affected by a stenosis, critical flow conditions occur, such as flow separation, high wall shear stress and wall compression, which are believed to be the significant factors at the onset of coronary heart diseases. In recent years, surgical treatments of cardiovascular diseases have been developed rapidly, and coronary artery bypass grafting (CABG) has been widely used for patients with severe stenosis. A large number of bypass grafts are implanted worldwide each year. However, up to 25 percents of the grafts fail within one year and up to 50 percents fail within ten years after surgery. Today, it has been recognized that one of the most important determinations in a successful bypass operation is the information of the rheological behavior of blood, the flow speed, the pressure distribution, the wall shear stress in the stenotic artery, and the wall deformation in cardiac cycles.

In order to understand the pathogenesis of coronary diseases, a number of in-vivo and vitro experiments have been conducted using animal models. Due to the difficulty in determining the critical flow conditions for both in-vivo and vitro experiments, the exact mechanism involved is not well understood. On the other hand, mathematical modeling and numerical simulation can lead to better understanding of the phenomena involved in vascular diseases. Thus, over the last two decades, various mathematical models based on the finite element method have been proposed to describe the rheological behavior of blood in stenotic arteries. However, some of the studies describe the fluid flow without taking into account of the motion of the

arterial wall, while some others concentrate on the behavior of the structure without taking into account of the fluid flow (Chandran, Mun, Choi, Chen, Hamilton, Nagaraj & McPherson 2003, Tada & Tarbell 2000, Tada & Tarbell 2004, Karner & Perktold 2000, Stangeby & Ethier 2002, Jung, Choi & Park 2004, Simon, Kaufmann, McAfee & Baldwin 1993, Holzapfel, Gasser & Stadler 2002).

It is well established that the fluid-structure interaction determines the behavior of blood flow through arteries. Recently, various studies have focused on the coupled fluid flow - arterial wall deformation problem (McCracken & Peskin 1980, Chahboune & Crolet 1998, Gerbeau, Vidrascu & Frey 2005). Chahboune and Crolet (1998) proposed a two-dimensional mathematical model and a numerical algorithm based on the finite element method for the fluid-structure interaction during the cardiac cycle. The model is used to couple the flow of blood with the motion of the arterial wall of the left ventricle. A three dimensional model has been proposed for the fluid-structure interaction in the arteries (Gerbeau et al. 2005). Queen (1992) developed a three-dimensional model of the heart including the four chambers, the four valves system and the exiting vessels. Gerbeau, Vidrascu and Frey (2005) proposed a three-dimensional model and a numerical algorithm to simulate the fluid-structure interaction in large compliant vessels where large displacement occurs but the biological interpretation of the results are not given. So far, none of the models seems to be completely satisfactory for all kinds of flow regimens.

In this study, we consider the flow of blood through coronary arteries with an unsymmetrical stenosis. A mathematical model is developed to study the unsteady state blood flow through a stenotic artery and the deformation of the arterial wall

in a cardiac cycle. Human blood is considered as an incompressible non-Newtonian fluid and the arterial wall is modelled as a poro-elastic material. Using three different geometry domains of a curved artery with three different size of stenosis, 25%, 50% and 75%, numerical simulations based on the finite element method are carried out for the flow field, pressure field, internal wall shear stress and the wall deformation in a cardiac cycle. Dependence of the flow field on the severity of stenosis and wall-interaction will be discussed.

11.2 Mathematical Model

The structure of a typical cross section of an arterial wall is shown schematically in Fig. 11.2. It is composed of three embedded layers including the tunica intima, the tunica media and the adventitia. The innermost layer is the tunica intima which consists of a thin layer of endothelial cells, connective tissue and basement membrane. The middle layer is the tunica media which comprises the smooth muscle cells and a continuous interstitial fluid phase of proteoglycan and collagen fiber. The outermost layer is the adventitia which is made up mostly of stiff collagenous fibers having an elastic modulus of $10^8 - 10^9\ dyn/cm^2$. Blood is transported mainly in the artery lumen but some could be transported through the endothelial and intimal layers and the media. Precise blood flow analysis requires simulating the flow of blood through the lumen and the various layers in deforming blood vessels. To make the model simple and more tractable, the entire arterial wall is assumed as one poro-elastic layer.

The model in this study uses two coordinate systems. One is a fixed mesh system Ω^F, in which the fluid model in the lumen region is solved. Another system is a moving

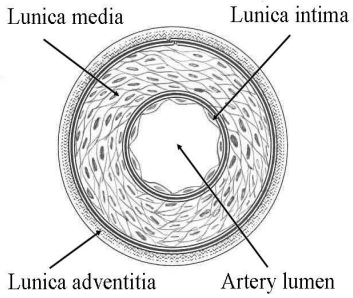


Figure 11.2: The cross section of an artery

mesh $\Omega^S(t)$, corresponding to the deformed geometry of the structure, in which the fluid model in the arterial wall is solved. Blood is assumed to be an incompressible non-Newtonian fluid. The non-Newtonian Carreau model is used to determine the viscosity of blood. Blood flow in the lumen region is governed by the continuity equation and the Navier-Stokes equations, while blood flow in the porous wall is described by the Brinkman equations. The wall deformation is modelled by the equations of classical elastodynamics. The velocity fields $\mathbf{u}$ in the luminal channel and $\mathbf{v}$ in the wall and the displacement $\mathbf{d}(\mathbf{x}, t)$ of the arterial wall are computed in a fully coupled manner through the use of the fluid-structure interface condition.

(a) Governing Equations for Blood Flows

Human blood consists of plasma fluid, red blood cells, white blood cells and thrombocyte. The blood plasma is made up of about 90-95% water and contains numerous dissolved materials such as proteins, lipoproteins and ions by which nutrients and wastes are transported. The elements of blood seem to be continuous, with no empty space

between the cells. Blood can therefore be assumed as a continuum medium. The small semisolid particles of red blood cells create the viscosity of blood. When the red blood cells clump together into larger particles at low shear rate, the non-Newtonian behavior becomes most evident. It has been generally accepted that when the shear rate is greater than 100 s^{-1} (Bonert, Myers, Fremes, Williams & Ethier 2002, Fei, Thomas & Rittgers 1994, Song, Sato & Ueda 2000), human blood behaves as an incompressible Newtonian fluid and thus the stress - deformation rate relations are described by the Newtonian model :

$$\sigma_{ij}^F \;=\; -p^F\delta_{ij} + 2\mu D_{ij}, \tag{11.1}$$

where μ is the blood viscosity and D denotes the rate of deformation tensor

$$D_{ij} \;=\; \frac{1}{2}(u_{i,j} + u_{j,i})$$

However, when the shear rate is lower than 100 s^{-1}, blood behaves as a non-Newtonian fluid and the stresses depend nonlinearly on the deformation rate. Various non-Newtonian models have been proposed including the Power law model, the Carreau model and etc. In all these models, the stresses are related to the deformation rate by

$$\sigma_{ij} \;=\; -p^F\delta_{ij} + 2\mu_n(\dot{\gamma})D_{ij}, \tag{11.2}$$

which is similar to the Newtonian model except that the viscosity is a function of shear rate instead of a constant. In different non-Newtonian models, the relation between the viscosity μ_n and the shear rate $\dot{\gamma} = \sqrt{2\mathbf{D}:\mathbf{D}}$ is different. In the Carreau model, $\mu_n = \mu_\infty + (\mu_0 - \mu_\infty)[1 + (\lambda\dot{\gamma})^2]^{(n-1)/2}$ for the constant values of $\mu_0,\ \mu_\infty,\ \lambda$ and n;

in the power law model, $\mu_n = m\dot{\gamma}^{n-1}$ for the constant values of m and n, while in the generalized power law model, $\mu_n = \lambda|\dot{\gamma}|^{n-1}$ in which the functions λ and n depend on the shear rate, $\dot{\gamma}$ (Johnston, Johnston, Corney & Kilpatrick 2004). So far, there is no universally accepted non-Newtonian model for blood. Thus, analysis of the effect of using different models on blood flow is still a worthwhile undertaking.

In the luminal region Ω^F, the equations governing the flow of blood include the constitutive equation (11.2), and the continuity equation as well as the stress equations of motion as detailed below

$$u_{i,i} = 0, \tag{11.3}$$

$$\rho^F \left(\frac{\partial u_i}{\partial t} + u_j u_{i,j} \right) = \frac{\partial \sigma^F_{ji}}{\partial x_j} + F^F_i, \tag{11.4}$$

where ρ^F denotes the blood density which is $1.06g\ cm^{-3}$, u_i represents the component of velocity vector in the ith direction, and F^F is the volume force acting on the fluid.

In the porous media of the cardiac wall Ω^S, blood flow is described by the following continuity equation and the Brinkman equations,

$$v_{i,i} = 0, \tag{11.5}$$

$$\rho^F \frac{\partial v_i}{\partial t} + \frac{\mu}{\kappa} v_i = -p^S_i + (\mu(v_{i,j} + v_{j,i}))_{,j} + F^S_i, \tag{11.6}$$

where μ denotes the viscosity in porous layer, κ is permeability, v_i represents the component of velocity vector in the ith direction, p^S denotes pressure, and F^S is the body force acting on the fluid in the wall.

(b) Governing equations for arterial wall deformation

The arterial wall is assumed as an elastic material. During a cardiac cycle, blood pressure acting on the inner wall surface varies with time, and thus the arterial wall

deformation is a function of time. The dynamic wall deformation can be modeled by the equations of classical elastodynamics:

$$\rho^s \frac{\partial^2 \mathbf{d}}{\partial t^2} = G\nabla^2 \mathbf{d} + (\lambda + G)\nabla(\nabla \cdot \mathbf{d}) \tag{11.7}$$

where ρ^s is the density of the structure, $\mathbf{d}$ denotes the displacement vector, λ and G are the Lamé constants which are related to the material Young's modulus E and Poisson's ratio ν by

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad G = \frac{E}{2(1+\nu)}.$$

(c) Boundary and interface conditions

To specify the boundary conditions for the blood flow, we consider precisely the blood flow mechanism. The heart is a two-step pump: first the atria, then the ventricles contract. The heart ejects and fills with blood in alternating cycles known as systole and diastole. Blood is ejected from the left ventricle into the arterial system during systole. The heart rests during diastole in which no blood is ejected. The cyclic nature of the heart pump creates pulsatile conditions in all the arteries. The pulsatile characteristic of pressure varies in different part of the arterial system.

Fig.11.3 shows the periodic blood pressure and flow rate waveforms oscillating between the systolic and diastolic levels with cardiac period T. Ignoring the variation in different cardiac cycles, the pulsatile pressure and flow rate are given by

$$p(t) = p(t+nT) \text{ and } Q(t) = Q(t+nT),\ n = 0, \pm 1, \pm 2, \ldots.$$

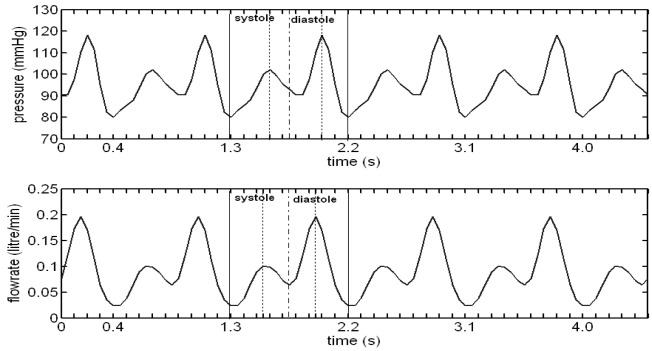


Figure 11.3: The periodic blood pressure and flow rate waveforms of the right coronary artery oscillating within systolic and diastolic levels with cardiac period $T = 0.9s$.

Mathematically, a periodic function can be expressed as a Fourier series. Based on a typical set of data, we obtain the following Fourier series representations

$$Q(t) \;=\; \bar{Q} + \sum_{n=1}^{4} \alpha_n^Q cos(\frac{2n\pi t}{T}) + \beta_n^Q sin(\frac{2n\pi t}{T}) \tag{11.8}$$

and

$$p_0(t) = \bar{p} \;+\; \sum_{n=1}^{4} \alpha_n^p cos(\frac{2n\pi t}{T}) + \beta_n^p sin(\frac{2n\pi t}{T}), \tag{11.9}$$

where $\bar{Q} = 0.0896\ litre/mimute$ and $\bar{p} = 95.3333\ mmHg$ are respectively the mean flow rate and mean pressure respectively, T is the cardiac period, and the values of α_n^Q, α_n^p, β_n^Q and β_n^p are as in Table below.

We therefore impose a pulsatile flow rate condition on the inlet boundary and a corresponding pulsatile pressure condition on the outlet boundary of the computation region. The boundary conditions for the velocity and pressure fields thus include both Dirichlet

Table 11.1: Values of parameters used in computation

n	α_n^Q	β_n^Q	α_n^p	β_n^p
1	0.0393	0.0241	5.9369	3.6334
2	-0.0360	0.0342	-11.1997	2.1255
3	-0.0131	0.0026	-2.2778	-3.7528
4	-0.0035	-0.0041	2.7333	-0.6375

type and Neumann/Robin type, i.e, for $i, j = 1, 2, 3$

$$\begin{array}{rcll} u_1 = 0, \;\; u_2 = \bar{u}_0(t) * cos(\frac{\pi}{3}), \;\; u_3 = \bar{u}_0(t) * sin(\frac{\pi}{3}) & & \text{on} & \partial\Omega_{in}^F \\ p = p_0(t), \quad (\mu_n(u_{i,j} + (u_{j,i})) \cdot \mathbf{n} = 0 & & \text{on} & \partial\Omega_{out}^F \end{array} \tag{11.10}$$

where $\bar{u}_0(t) = \frac{Q(t)}{A}$, A denotes the inlet cross-section area of the artery, $Q(t)$ is the pulsatile flow rate and $p_0(t)$ represents the pulsatile pressure.

On the interface between the lumen and the arterial wall $\Gamma^{F/S}$, the expression for the velocity must be continuous across the interface. We thus set

$$\mathbf{v} = \mathbf{u} = \frac{\partial \mathbf{d}}{\partial t}. \tag{11.11}$$

The movement of the inflow boundary $\partial\Omega_{in}^S$ of the structure is assumed to be restricted in all directions,

$$\mathbf{d}(\mathbf{x}, t) = 0. \tag{11.12}$$

The movement of the outflow boundary $\partial\Omega_{out}^S$ and other boundaries of the structure is moved freely in all directions.

11.3 Method of Solution

By substituting equations (11.2) into (11.4), we obtain the Navier-Stokes equations for the flow of blood in the lumen region. The Navier-Stokes equations, together with the Brinkman equations (6), the continuity equations (11.3) and (11.5), and the elastodynamic equations, constitute a system of eleven equations in terms of eleven unknown functions $p^F,\ u_1,\ u_2,\ u_3,\ p^S,\ v_1,\ v_2,\ v_3,\ d_1,\ d_2$ and d_3. To solve the problem, firstly the penalty function method is used to weaken the continuity requirement (11.3) and (11.5) by the following equations

$$u_{j,j} \ = \ -\delta p^F \tag{11.13}$$

$$v_{j,j} \ = \ -\delta p^S \tag{11.14}$$

where δ is a small positive number. Thus, the pressure variables can be eliminated from the system.

To develop the variational statement for the boundary value problem, we consider the following integral representation of the problem.

Find $\mathbf{u} \ \epsilon \ [H^1(\Omega^F)]^3,\ \mathbf{v} \ \epsilon \ [H^1(\Omega^S)]^3$ and $\mathbf{d} \ \epsilon \ [H^1(\Omega^S)]^3$ such that for all test functions $\hat{\mathbf{u}} \ \epsilon \ [H^1_{0\mathbf{u}}(\Omega^F)]^3,\ \hat{\mathbf{v}} \ \epsilon \ [H^1_{0\mathbf{v}}(\Omega^S)]^3,\ \hat{\mathbf{d}} \ \epsilon \ [H^1_{0\mathbf{d}}(\Omega^S)]^3$, all the Dirichlet boundary conditions for the unknown functions are satisfied and

$$(\rho^F \frac{\partial u_i}{\partial t},\ \hat{u}_i) + (\rho^F u_j u_{i,j},\ \hat{u}_i) - ((\mu(u_{i,j} + u_{j,i}))_{,j},\ \hat{u}_i) - (\frac{1}{\delta}(u_{j,ji},\ \hat{u}_i) \ = \ (F_i^F,\ \hat{u}_i), \tag{11.15}$$

$$(\rho^F \frac{\partial v_i}{\partial t},\ \hat{v}_i) + (\rho^F \frac{\mu}{\kappa} v_i,\ \hat{v}_i) - ((\mu(v_{i,j} + v_{j,i}))_{,j},\ \hat{v}_i) - (\frac{1}{\delta}(v_{j,ji},\ \hat{v}_i) \ = \ (F_i^S,\ \hat{v}_i), \tag{11.16}$$

$$(\rho^S \frac{\partial^2 d_i}{\partial t^2}, \hat{d}_i) - (G d_{i,jj} + (\lambda + G) d_{k,ki},\ \hat{d}_i) \ = \ 0, \tag{11.17}$$

where $(\cdot, \cdot)$ denotes the inner product on the square integrable function space $L^2(\Omega)$, $H^1(\Omega)$ is the Sobolev space $W^{1,2}(\Omega)$ with norm $\| \cdot \|_{1,2,\Omega}, H^1_{0u}(\Omega) = \{v \epsilon H^1(\Omega) | v = 0 \text{ on } \partial\Omega\}$. A standard procedure is then carried out to reduce the second-order spatial derivatives involved in the above problem into the first-order ones using integration by parts to ensure that all integrals involved are well defined.

To find the numerical solution of the problem, we pose the variational problem into an N-dimension subspace and the computation domain Ω is discretized into a finite number of elements connected by N nodes. By using the Galerkin finite element formulation, we obtained the following systems of ordinary differential equations

$$\mathbf{M}^f \frac{\partial U}{\partial t} + \mathbf{K}^f U \ = \ \mathbf{F}^f, \tag{11.18}$$

$$\mathbf{M}^s \frac{\partial V}{\partial t} + \mathbf{K}^s U \ = \ \mathbf{F}^s, \tag{11.19}$$

$$\mathbf{M}^w \frac{\partial^2 \mathbf{d}}{\partial t^2} + \mathbf{K}^w \mathbf{d} \ = \ \mathbf{F}^w, \tag{11.20}$$

where the matrices $\mathbf{M}^f$, $\mathbf{M}^s$ and $\mathbf{M}^w$ correspond to the transient terms, while $\mathbf{K}^f$, $\mathbf{K}^s$ and $\mathbf{K}^w$ correspond to the advection and diffusion terms. To keep details to minimize, we omit the specific form of each matrix and vector here.

For convenience in using the interface boundary calculation, we denote $\frac{\partial d}{\partial t}$ by q and use the following approximation

$$q(t) \ = \ \frac{\partial \mathbf{d}}{\partial t} \ = \ \frac{\mathbf{d}(t) - \mathbf{d}(t - \triangle t)}{\triangle t}. \tag{11.21}$$

Thus, at a typical instant of time t, system (11.20) in term of $q(t)$ becomes

$$\mathbf{M}^w \frac{dq}{dt} + \bar{\mathbf{K}}^w q \ = \ \bar{\mathbf{F}}^w, \tag{11.22}$$

where $\bar{\mathbf{F}}^w = \mathbf{F}^w - \mathbf{K}^w\mathbf{d}_n$ and $\bar{\mathbf{K}}^w = \triangle t\mathbf{K}^w$.

Now we are ready to couple the fluid flows and the wall deformation. To implement the compatibility conditions, the equations in each domain are partitioned into two parts including those corresponding to the interface boundary and others. Thus, from (11.18), (11.19) and (11.20), we have

$$\mathbf{M}\dot{\mathbf{U}} \ + \ \mathbf{K}\mathbf{U} \ = \ \mathbf{F}, \tag{11.23}$$

where

$$\mathbf{M} \ = \ \begin{bmatrix} \mathbf{M}^s_{ss} & \mathbf{M}^s_{sB} & 0 & 0 \\ \mathbf{M}^s_{Bs} & \mathbf{M}^s_{BB} + \mathbf{M}^f_{BB} + \mathbf{M}^w_{BB} & \mathbf{M}^f_{Bf} & \mathbf{M}^w_{Bw} \\ 0 & \mathbf{M}^f_{fB} & \mathbf{M}^f_{ff} & 0 \\ 0 & \mathbf{M}^w_{wB} & 0 & \mathbf{M}^w_{ww} \end{bmatrix}$$

$$\mathbf{K} \ = \ \begin{bmatrix} \mathbf{K}^s_{ss} & \mathbf{K}^s_{sB} & 0 & 0 \\ \mathbf{K}^s_{Bs} & \mathbf{K}^s_{BB} + \mathbf{K}^f_{BB} + \bar{\mathbf{K}}^w_{BB} & \mathbf{K}^f_{Bf} & \bar{\mathbf{K}}^w_{Bw} \\ 0 & \mathbf{K}^f_{fB} & \mathbf{K}^f_{ff} & 0 \\ 0 & \bar{\mathbf{K}}^w_{wB} & 0 & \bar{\mathbf{K}}^w_{ww} \end{bmatrix}$$

$$\mathbf{U} \ = \ \begin{bmatrix} \mathbf{v}_s \\ \mathbf{v}_{sB} \\ \mathbf{u}_f \\ \mathbf{q}_w \end{bmatrix} \ \text{and} \ \mathbf{F} \ = \ \begin{bmatrix} \mathbf{F}_s \\ \mathbf{F}^s_B + \mathbf{F}^f_B + \bar{\mathbf{F}}^w_B \\ \mathbf{F}_f \\ \bar{\mathbf{F}}_w \end{bmatrix}.$$

A standard backward Euler scheme is then used to solve the above system of ordinary differential equations to determine the velocity and pressure fields at any instant of time.

11.4 Numerical Results and Discussion

A test example is given here to study the flow of blood through a stenosed artery. Fig. 11.4 shows the angiogram of a stenosed coronary artery. The examples under consideration are stenotic arteries with 25%, 50% and 75%-area severity. The computation

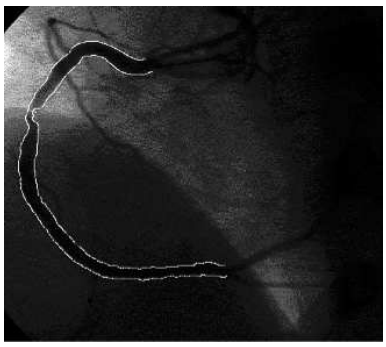


Figure 11.4: The right coronary artery with stenosis.

region, as shown in Figure 11.5, represents the right coronary artery with a 50% stenosis with spherical curvature located at 2.35 cm from the inlet boundary. The diameter of the lumen is 0.2 cm and the wall thickness is 0.025 cm. The arc length of the artery is 6.7 cm for a typical coronary artery in this investigation.

The computation domains for the cases with 25%, 50% and 75%-area severity are respectively discretized into 8,536 tetrahedron elements with 68,265 degrees of freedom, 7,946 tetrahedron elements corresponding to 63,455 degrees of freedom and 8,898 tetrahedron elements corresponding to 70,607 degrees of freedom ($\mathbf{u}$, p^F in the lumen region, $\mathbf{v}$, p^S, and $\mathbf{d}$ in the wall region). The solutions were computed for 5 cardiac cycles (t=0.0 to 4.5 second) to ensure reproducibility of the pulsatile characteristic flow.

To determine the inlet pulsatile flow rate and outlet pulsatile pressure, the parameters listed in Table 11.1 are used. In this study we assume that (1) pressure at the exit boundary in the lumen region is always between $80 - 120 mmHg$ and (2) The mean flow rate $\bar{Q}$ at the inlet boundary depends on the percentage of area-severity of the stenotic artery. From numerical experiments, we found that the corresponding

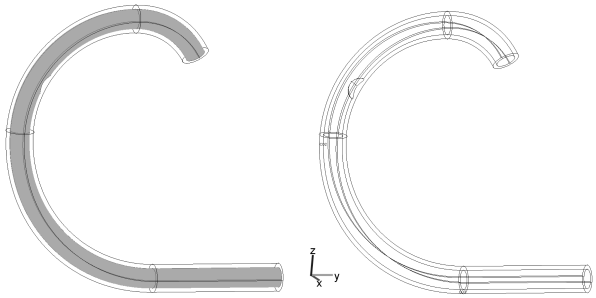


Figure 11.5: The 3-D geometry of the 50% stenotic artery.

mean flow rate at the inlet boundary of the 25% and 50% stenotic arteries is $\bar{Q}$ =0.0896 $litre/min$ while $\bar{Q}$ = 0.0514 $litre/min$ for the 75% stenotic artery.

Figure 11.6 depicts the velocity field in the luminal channel with 75%-area severity at the peak of diastole (t=1.95s). The plot clearly shows the flow pattern. In upstream from the stenosis, the velocity profile in the flow direction is parabolic and the fluid passes through the stenosis at jet speed, especially at the throat of the stenosis. Figures 11.7(a)-(b) show the pressure distributions along a longitudinal line of the 50% and 75% stenotic arteries during the systolic and diastolic periods. It shows that the pressure drops very significantly near the stenosis site and creates a jet flow at the throat of the stenosis. Higher area severity generates higher pressure drop around the stenosis site as expected.

The pulsatile patterns of blood flow, pressure field and shear rate at a point around the stenosis site in the lumen region of the stenotic artery with 50% and 75%-area severity are demonstrated in Figures 11.8(a) and 11.8(b), respectively. The results show that at a point around the 50% stenosis, blood speed varies between 18 - 141.261 cm/s and shear rate is between 200-3000 s^{-1} while at a point around the 75% stenosis,

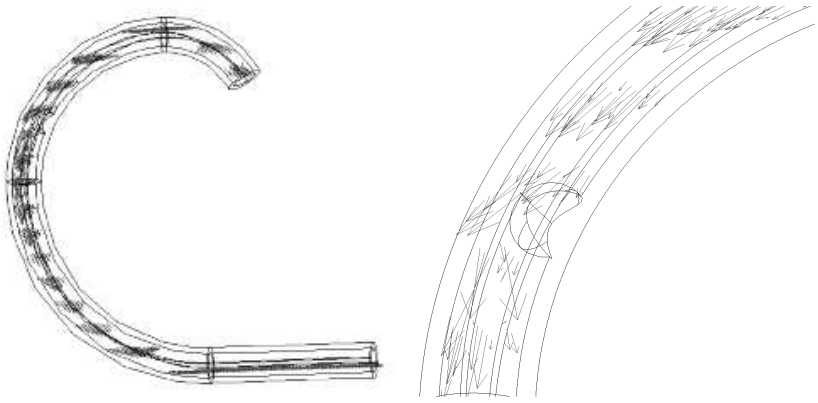
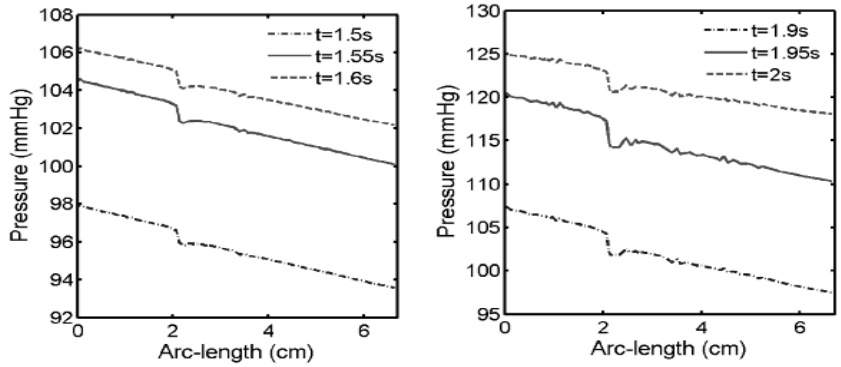


Figure 11.6: Velocity filed in the luminal channel of the 75% stenotic artery at the peak of diastole

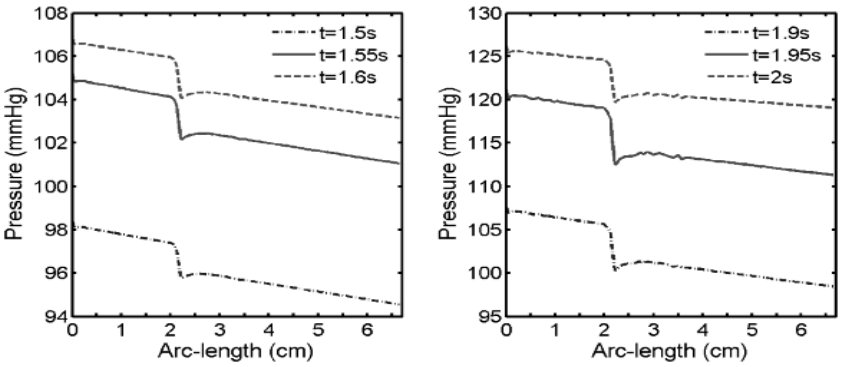
blood speed varies between 340 - 450 cm/s and shear rate is between 2.8 - 4 $\times 10^4 s^{-1}$. In the wall region at a point near the exit boundary, wall deformation varies between 0.01-0.05 cm, velocity field is between 0-2.25 $\times 10^{-8} cm/s$ and pressure field is between 15-23 $mmHg$.

The wall displacement during a cardiac cycle is investigated. Figure 11.9 shows the wall displacement at the peak of diastole (t=1.95s). The results indicate that the wall displacement varies periodically between 0 and 0.049 cm and the highest displacement is present at the exit boundary.

To capture the effect of wall-interaction on wall shear stresses (WSS) and wall shear rate (WSR), we investigate the models with 50% and 75%-area severity at various instants including the beginning of systole (t=1.3s), the peak of the systole(t=1.55s), the beginning of diastole (t=1.75s) and the peak of diastole (t=1.95s). The solutions are plotted along a longitudinal line on the interface between the lumen region and the wall region. The results as shown in Figures 11.10(a)-(b) indicate that at the peak of



(a) 50% stenotic artery



(b) 75% stenotic artery

Figure 11.7: Pressure distribution along a longitudinal line on the interface between the lumen region and the arterial wall during a cardiac cycle.

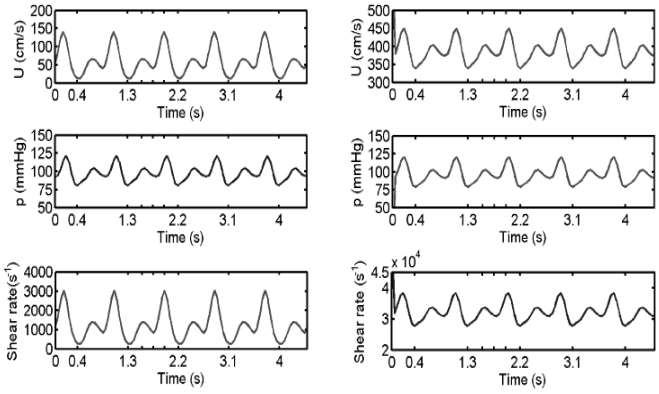


Figure 11.8: Pulsatile patterns of blood flow, pressure field and shear rate around the stenosis site in the lumen region (a) 50%-area severity (b) 75%-area severity.

diastole (t=1.95s), for the 50% stenotic artery, wall shear stresses and wall shear rate around the stenosis site vary between 100 - 330 dyn/cm^2, and 2,000 - 9,000 s^{-1}. For the 75% stenotic artery, wall shear stresses and wall shear rate around the stenosis site varies between 100 - 580 dyn/cm^2, and 2,000 - 17,500 s^{-1} as shown in Figure 8(c)-(d). It indicates that high wall shear stresses and high wall shear rate appear around the stenosis site. Higher area severity leads to higher wall shear stresses and higher wall shear rate around the stenosis site.

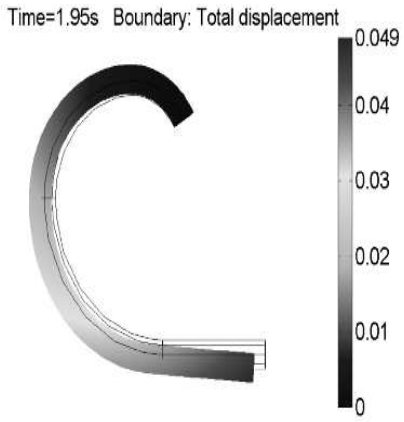


Figure 11.9: Wall displacement during a cardiac cycle at the peak of diastole.

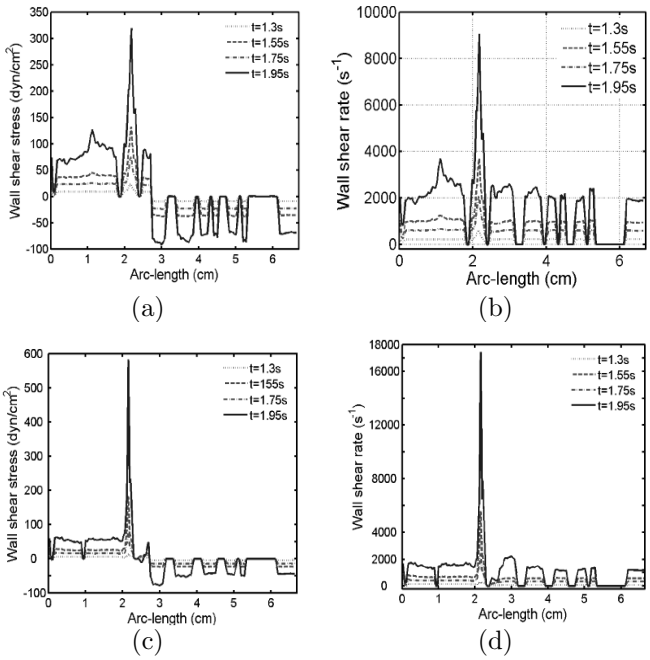


Figure 11.10: Wall shear stresses and wall shear rate along a longitudinal line on the interface between the lumen region and the arterial wall of stenotic arteries: (a-b) 50%-area severity; (c-d) 75%-area severity.

Bibliography

Archapitak, J., Wiwatanapataphee, B., Wu, Y. & Tang, I. (2004), 'A finite element scheme for the determination of electromagnetic force in continuous steel casting', *Int. J. Computational and Numerical Analysis and Applications* **5**(1), 81–96.

Bennon, W. & Incropera, F. (1987), 'A continuum model for momentum, heat and species transport in binary solid-liquid phase change systems - i. model formulation', *Int. J. Heat and Mass Transfer* **30**, 2161–2170.

Bonert, M., Myers, J., Fremes, S., Williams, J. & Ethier, C. (2002), 'A numerical study of blood flow in coronary artery bypass graft side-to-side anatomoses', *In Annals of Biomedical Engineering* **30**, 599–611.

Brimacombe, J., Samarasekera, I. & Laid, J. (1983), 'Continuous casting : Heat flow, solidification and crack formation', *Iron and Steel Society, AIME* **2**.

Chahboune, B. & Crolet, J. (1998), 'Numerical simulation of the blood-wall interaction in the human left ventricle', *The European Physical Journal Applied Physics* **2**, 291–297.

Chandran, K., Mun, J., Choi, K., Chen, J., Hamilton, A., Nagaraj, A. & McPherson, D. (2003), 'A method for in-vivo analysis for regional arterial wall material property alterations with atheroclerosis: preliminary results', *Medical Engineering and Physics* **25**, 289–298.

Chien, K.-Y. (1982), 'Predictions of channel and boundary-layer flows with low-reynolds number turbulence model', *AIAA Journal* pp. 33–38.

Driest, E. (1996), 'On turbulence flow near a wall', *Journal of the Aeronautical Sciences* **23**, 1007–1011.

Falk, R. (1975), *An analysis of the penalty method and extrapolation for the stationary Stokes problem*, Advances in Computer Methods for Partial Differential Equations (edited by R. Vichnevetsky), AICA, New Brunswick, pp. 66–69.

Fei, D.-Y., Thomas, J. & Rittgers, S. (1994), 'The effect of angle and flow rate upon hemodynamics in distal vascular graft anatomoses: a numerical model study', *J. Biomechanical Engineering* **116**, 331–336.

Ferziger, J. (1987), 'Simulation of incompressible turbulent flows', *Computational Physics* **69**, 1–48.

Flint, P. (1990), A three-dimensional finite difference model of heat transfer, fluid flow and solidification in the continuous slab caster, *in* 'Steelmaking Conference Proceedings', Dubrovnik, Yugoslavia, pp. 481–490.

Fowkes, N. & Woods, A. (1989), The flux of flux in a continuous steel caster, Technical report, Department of Mathematics, University of Western Australia.

Gerbeau, J.-F., Vidrascu, M. & Frey, P. (2005), 'Fluid-structure interaction in blood flows on geometries based on medical imaging', *Computers and structures* **83**, 155–165.

Hill, J. & Wu, Y. (1994*a*), 'On a nonlinear stefan problem arising in the continuous casting of steel', *Acta Mechanica* **107**, 183–198.

Hill, J. & Wu, Y. (1994*b*), 'On a nonlinear stefan problem arising in the continuous casting of steel', *Acta Mechanica* **107**, 183–198.

Holzapfel, G., Gasser, T. & Stadler, M. (2002), 'A structural model for the viscoelastic behavior of arterial walls: Continuum formulation and finite element analysis', *European Journal of Mechanics A,Solids* **21**, 441–463.

Jaeger, M. & Dhatt, G. (1992), 'An extended $k - \epsilon$ finite element model', *Journal of Numerical Methods in Fluids* **14**, 1325–1345.

Jenkins, D. & Hoog, F. D. (1996), *Numerical methods in laminar and turbulent flow*, John Wiley and Sons Ltd, pp. 332–336.

Johnston, B., Johnston, P., Corney, S. & Kilpatrick, D. (2004), 'Non-newtonian blood flow in human right coronary arteries: steady state simulations', *J. Biomechanics* **37**, 709–720.

Jones, W. & Launder, B. (1973), 'The calculation of low-reynolds number phenomena with a two-equation model of turbulence', *Int. J. Heat and Mass Transfer* **16**, 1119–1130.

Jung, H., Choi, J. & Park, C. (2004), 'Asymmetric flows of non-newtonian fluids in symmetric stenosed artery', *Korea-Australia Rheology Journal* **16**, 101–108.

Karner, G. & Perktold, K. (2000), 'Effect of endothelial injury and increased blood pressure on albumin accumulation in the arterial wall: a numerical study', *J. Biomechanics* **33**, 709–715.

Lally, B., Biegler, L. & Henein, H. (1990), 'Finite difference heat transfer modelling for continuous casting', *Metallurgical Transactions* pp. 761–770.

Lam, C. & Bremhorst, K. (1981), 'A modified form of the $k-\varepsilon$ model for predicting wall turbulence', *ASME Journal of Fluid Engineering* **103**, 456–460.

Launder, B. (1988), 'On the computation of convective heat transfer in complex turbulent flows', *ASME Journal of Heat Transfer* **110**, 1112–1128.

Launder, B. & Spalding, D. (1974), 'The numerical computation of turbulent flows', *Computer Methods in Applied Mechanics and Engineering* **3**, 269–289.

McCracken, M. & Peskin, C. (1980), 'A vortex method for blood flow through heart values', *J. Computational Physics* **35**, 183–205.

Nagano, Y. & Hishida, M. (1987), 'Improved form of the $k-\varepsilon$ model for wall turbulent shear flows', *Transactions of ASME* **109**, 156–160.

Patel, V., Rodi, W. & Scheuerer, G. (1985), 'Turbulence models for near-wall and low reynolds number flows: A review', *AIAA Journal* **13**, 1308–1319.

Reddy, M. & Reddy, J. (1992), 'Numerical simulation for forming processes using a coupled fluid flow and heat transfer model', *Int. J. Numerical Methods for Engineering* **35**, 807–833.

Simon, B., Kaufmann, M., McAfee, M. & Baldwin, A. (1993), 'Finite element models for arterial wall mechanics', *J. Biomechanical Engineering* **115**, 489–496.

Sokolnikoff, I. (1986), *Mathematical Theory of Elasticity*, Robert E. Krieger Publishing Company, Malabar, Florida.

Song, M.-H., Sato, M. & Ueda, Y. (2000), 'Three-dimensional simulation of coronary artery bypass grafting with the use of computational fluid dynamics', *Surgery Today* **30**, 993–998.

Stangeby, D. K. & Ethier, C. (2002), 'Computational analysis of coupled blood-wall arterial ldl transport', *J. Biomechanical Engineering* **124**, 1–8.

Tada, S. & Tarbell, J. (2000), 'Interstitial flow through the internal elastic lamina affects shear stress on smooth muscle cells in the arterial wall', *The Americal Journal of Physiology - heart and Circulatory Physiology* **278**, 1589–1597.

Tada, S. & Tarbell, J. (2004), 'Internal elastic lamina affects the distribution of macromolecules in the arterial wall: a computational study', *The American Journal of Physiology - Heart and Circulatory Physiology* **287**, 905–913.

Thomas, B. (1990), *Metallurgical Transactions B* **21**, 387–400.

Thomas, B. (2001), *Continuous casting: modelling*, Pergamon Elsevier Science Ltd, UK.

Thomas, B., Najjar, F. & Mika, L. (1990), *in* '29th CIM Proceeding of F. Weinberg International Symposium on Solidification Processing', Hamilton.

Wiwatanapataphee, B. (2008), 'Modelling of non-newtonian blood flow through stenosed coronary arteries', *DCDIS B: Applications and Algorithms* **15**(5), 619–634.

Wiwatanapataphee, B., Poltem, D., Wu, Y. & Lenbury, Y. (2006), 'Simulation of pulsatile flow of blood in stenosed coronary artery bypass with graft', *Journal of Mathematical Biosciences and Engineering* **3**(2), 317–383.

Wiwatanapataphee, B., Wu, Y., Archapitak, J. & Siew, P. (2004), 'Numerical study of the turbulent flow of molten steel in a domain with a phase-change boundary', *Journal of Computational and Applied Mathematics* **166**(1), 307–319.

Wu, Y., Hill, J. & Flint, P. (1994), 'A noval finite element method for heat transfer in the continuous caster', *J.Austral. Math. Soc. Ser. B* **35**, 263–288.

Wu, Y. & Wiwatanapataphee, B. (2007), 'Modelling of turbulent flow and multi-phase heat transfer under electromagnetic force', *Discrete and Continuous Dynamical Systems-Series B* **8**(3), 695–706.

APPENDICES

A. Linear Space

Definition A.1.1 A *linear space* consists of a set V and two operations:

$$(u, v) \to u + v \texttt{ from } V \times V \texttt{ into } V;$$

$$(k, v) \to kv \texttt{ from } \Re \times V \texttt{ into } V$$

such that the following properties hold for every $u, v, w \in V$ and $\alpha, \beta \in \Re$

1. $u + v = v + u$ (commutativity)

2. $u + (v + w) = (u + v) + w$ (distributivity)

3. There exists a unique element in V, denoted by 0, such that

$$0 + v = v + 0 = v$$

4. For every $v \in V$ there exists a unique element in V, denoted by $-v$, such that

$$-v + v = 0 = v + (-v)$$

5. $\alpha(u + v) = \alpha u + \alpha v$

6. $(\alpha + \beta)u = \alpha u + \beta u$

7. $\alpha(\beta u) = (\alpha\beta)u$

8. $1 \cdot u = u$

Definition A.1.2 Let V be a linear space. A nonempty set $V_0 \subset V$ is called a *linear subspace* of V if for $u, v \in V_0$ `and` $\alpha \in \Re$, $u + v$ and $\alpha v \in V_0$.

Definition A.1.3 Let V be a linear space and $\{v_i\}$ be a finite subset of V. A linear combination of v_i is a vector of the form $\sum \alpha_i v_i$ with any scalar α_i. For any nonempty subset E of V, we set

$$\texttt{span}\ E = \{\sum \alpha_i v_i \ : \ \alpha_i \in \Re,\ v_i \in E\}$$

Definition A.1.4 Let V be a linear space and A be a nonempty subset of V. A is *dependent* iff for distinct vectors $v_i \in A$, there exist some nonzero coefficients such that $\sum \alpha_i v_i = 0$. A is *independent* iff $\sum \alpha_i v_i = 0$ only when all coefficients are zero.

Definition A.1.5 Let V and W be linear spaces. A map $L : V \to W$ is linear from V into W, iff

$$L(v_1 + v_2) = L(v_1) + L(v_2), \quad \forall v_1, v_2 \in V$$

$$L(\alpha v) = \alpha L(v) \quad \forall \alpha \in \Re, \forall v \in V$$

We call a linear operator from V to $\Re$ as a *linear functional* on V or a linear form on V.

Definition A.1.6 Let V_1, V_2 and W be linear spaces. A map $a : V_1 \times V_2 \to W$ is a *bilinear operator* from $V_1 \times V_2$ into W iff

$$\forall \bar{v}_1 \in V_1,\ v_2 \to a(\bar{v}_1, v_2) \ \texttt{is linear on}\ V_2$$

$$\forall \bar{v}_2 \in V_2,\ v_1 \to a(v_1, \bar{v}_2) \texttt{ is linear on } V_1.$$

For $W = \Re$, a mapping a is a bilinear form on $V_1 \times V_2$.

For linear spaces V and W, if V_0 and W_0 are subspaces of V and W respectively and L is a linear operator from V into W, then $L(V_0)$ and $L^{-1}(W_0)$ are subspaces of W and V, respectively. The range and the kernel of L defined by

$$R(L) = L(V) \texttt{ and ker } L = L^{-1}\{0\}.$$

are also subspaces of W and V, respectively.

Definition A.1.7 Let a be a linear operator from $V \times V$ into W. The symmetric part a_s of a is defined by

$$a_s(u, v) \ = \ \frac{1}{2}(a(u, v) + a(v, u)).$$

A bilinear operator a is said to be *symmetric* iff $a \equiv a_s$

Definition A.1.8 If V is a linear space, a map

$$\phi \ : \ V \to \Re$$

is called a *quadratic form* on V iff there exists a bilinear form a on $V \times V$ such that

$$\phi(v) \ = \ a(v, v) \quad \forall v \in V.$$

Definition A.1.9 For linear spaces V and W, a map $L : V \to W$ is an *algebraic isomorphism* from V onto W iff

$$\begin{cases} L \texttt{ is linear}; \\ \texttt{ker } L = \{0\}; \\ R(L) = W. \end{cases}$$

For example, if Ω is an open subset of $\Re^n$, $\Im(\Omega)$ is the set of all real functions, then the subset of all $v \in \Im(\Omega)$ which are continuous is a subspace of $\Im(\Omega)$ and is denoted by $C^0(\Omega)$.

Definition A.1.10 A seminorm on a linear space V is a map $v \to |v|$ from v into $[0, \infty)$ such that

$$|u + v| \leq |u| + |v| \ \texttt{and} \ |\lambda u| = |\lambda||u| \quad \forall u, v \in V, \ \forall \lambda \in \Re$$

We call a linear space with a norm on it as a *normed space* and denote seminorms and norms by $|\cdot|$ and $\|\cdot\|$, respectively.

Definition A.1.11 Let V be a normed space with the norm $\|\cdot\|$. If $A \subset V$, $v_0 \in V$, and a sequence $\{v_m\} \in V$, then

- a ball of center $v_0 \in V$ and radius r is the set

$$B_r(v_0) = \{v \in V \ : \ \|v - v_0\| < r\};$$

- $\{v_n\}$ converges strongly to $v_0 \in V$ iff $\texttt{lim} \ \|v - v_0\| = 0$;
- $\{v_n\}$ satisfies the Cauchy condition iff $\forall \varepsilon > 0, \ \exists n_\varepsilon \ \texttt{such that} \ \forall n, m > n_\varepsilon \ \ \|v_n - v_m\| < \varepsilon$;
- A is open iff $\forall v \in A, \ \exists r > 0 \ \texttt{such that} \ B_r(v) \subset A$;
- A is closed iff $v_n \in A$ and mapping $v_n \to v$ implies $v \in A$;
- A is dense iff $\forall v \in V$ there exists a sequence $\{v_n\} \in A$ such that $v_n \to v$;
- A is bounded iff there exists a constant M such that $\|v\| \leq M \ \ \forall v \in A$;
- The closure $\bar{A}$ of A is the smallest closed set containing A;
- The interior $\mathring{A}$ of A is the largest open set contained in A;

Definition A.1.12 Let V and W be normed spaces and $f : A \to W$ a map defined on a subset A of V. Then f is *continuous* at a point $v_0 \in A$ iff $v_n \in A$ and $v_n \to v_0$ in V imply $f(v_n \to f(v_0)$ in W. f is called continuous in A iff it is continuous at every point in A.

Definition A.1.13 An open subset Ω of $\Re^n$ is called smooth iff it is connected and bounded and satisfies the following conditions:

For every $x_0 \in \partial\Omega$ there exist $r(x_0) > 0$ and a coordinate system $(\xi_1, \ldots, \xi_n)$ with origin at x_0 such that

$$\Omega \bigcap \mathbf{B}_{r(x_0)}(x_0) = \{(\xi_1, \ldots, \xi_n) \in \mathbf{B}_{r(x_0)} \ : \ \xi_n > \zeta_{x_0}(\xi_1, \ldots, \xi_{n-1})\} \text{ and}$$

$$\partial\Omega \bigcap \mathbf{B}_{r(x_0)}(x_0) = \{(\xi_1, \ldots, \xi_n) \in \mathbf{B}_{r(x_0)} \ : \ \xi_n = \zeta_{x_0}(\xi_1, \ldots, \xi_{n-1})\},$$

where ζ_{x_0} is a function defined on $\Re^{n-1}$ continuous with all the derivatives of every order.

Definition A.1.14 An open subset Ω of $\Re^n$ is called a convex polygon iff it is bounded and there exists a finite number m of scalars c_i and linear maps $L_i : \Re^n \to \Re$ such that

$$\Omega = \{x \in \Re^n \ : \ L_i x > c_i, \ i = 1, \ldots, m\}.$$

Definition A.1.15 An open subset Ω of $\Re^n$ is called a polygon iff it is connected and there exists a finite number of convex polygons Ω_j such that $\Omega = \texttt{interior of} \bigcup \bar{\Omega}_j$.

Let u be a smooth function defined on domain Ω and $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ be an n-tuple of nonnegative integers and denote $|\alpha| = \sum \alpha_i$. Then the αth derivative of u is denoted by

$$D^\alpha u = \frac{\partial^{\alpha_1+\alpha_2+\ldots+\alpha_n} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots} \partial x_n^{\alpha_n} = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots} \partial x_n^{\alpha_n}.$$

B. Operations on Vectors

Let $\underline{u}$ and $\underline{v}$ be vectors and q be a real-valued function.

$$\nabla \cdot (q\underline{v}) \;=\; q\nabla \cdot \underline{v} + \nabla q \cdot \underline{v}, \tag{B.1}$$
$$\nabla \times (q\underline{v}) \;=\; q\nabla \times \underline{v} + \nabla q \times \underline{v}, \tag{B.2}$$
$$\nabla \cdot (\underline{u} \times \underline{v}) \;=\; (\nabla \times \underline{u}) \cdot \underline{v} - \underline{u} \cdot (\nabla \times \underline{v}), \tag{B.3}$$
$$\nabla \times \nabla \times \underline{v} \;=\; \nabla(\nabla \cdot \underline{v}) - \triangle \underline{v}, \tag{B.4}$$
$$\underline{v} \times (\nabla \times \underline{v}) \;=\; \tfrac{1}{2}\nabla(\underline{v}^2) - (\underline{v} \cdot \nabla)\underline{v}, \tag{B.5}$$
$$\nabla \times (\underline{u} \times \underline{v}) \;=\; (\underline{v} \cdot \nabla)\underline{u} - \underline{v}(\nabla \cdot \underline{u}) - (\underline{u} \cdot \nabla)\underline{v} + \underline{u}(\nabla \cdot \underline{v}). \tag{B.6}$$

C. Green's Formula

For $\underline{u},\ \underline{v}$ and q are smooth function,

Integrating (B.1) and using the Gauss divergence theorem lead to
$$(\nabla \cdot \underline{v},\ q) + (\underline{v},\ \nabla q) \;=\; <\underline{n} \cdot \underline{v},\ q> . \tag{C.1}$$
Substituting $\underline{v} = \nabla p$ into (C.1) yields
$$(\triangle p,\ q) + (\nabla p,\ \nabla q) \;=\; <\underline{n} \cdot \nabla p,\ q> . \tag{C.2}$$
Substituting $q = \nabla \cdot \underline{u}$ into (C.1) yields
$$(\nabla \cdot \underline{v},\ \nabla \cdot \underline{u}) + (\underline{v},\ \nabla(\nabla \cdot \underline{u})) \;=\; <\underline{n} \cdot \underline{v},\ \nabla \cdot \underline{u}> . \tag{C.3}$$
Replacing $\underline{v}$ by $\nabla \times \underline{v}$ in (C.1) leads to
$$(\nabla \times \underline{v},\ \nabla q) \;=\; <n \cdot (\nabla \times \underline{v}),\ q> . \tag{C.4}$$
Integrating (B.3) and using the Gauss divergence theorem lead to
$$(\nabla \times \underline{u},\ \underline{v}) - (\underline{u},\ \nabla \times \underline{v}) \tag{C.5}$$
Substituting $u = \nabla q$ into (C.5) yields
$$(\nabla \times \underline{v},\ \nabla q) \;=\; -<\underline{n} \times \nabla q,\ \underline{v}> = <\nabla q,\ \underline{n} \times \underline{v}> . \tag{C.6}$$
Replacing $\underline{v}$ by $\nabla \times \underline{v}$ in (C.5) yields
$$(\nabla \times \underline{u},\ \nabla \times \underline{v}) - (\underline{u},\ \nabla \times \nabla \times \underline{v}) \;=\; <\underline{n} \times \underline{u},\ \nabla \times \underline{v}> . \tag{C.7}$$

Index

admissible functions, 34
approximate error, 63
asymptotic error, 63

Backward difference scheme, 93
bandwidth, 60
basis function, 43,72
boundary conditions, 23
boundary element method, 29
boundary operator, 27
boundary value problem, 23
Bubnov-Galerkin, 41

classical statement, 33
consistency, 88
convergence, 63
Crank-Nicolson Scheme, 91

differential operator, 27
diffusion equation, 25
direct approach, 32
Dirichlet type, 26,61
discretization, 54
divergence operator, 68

eigenvalues, 90
eigenvectors, 90
elliptic equation, 26,68
energy balance approach, 32

finite difference method, 29
finite element equations, 62
finite element method, 29
finite-dimensional subspace, 70
forward difference scheme, 87

Galerkin, 41
gradient operator, 68

hyperbolic equation, 26

ill-conditioned system, 43

interpolating function, 45
interpolation error, 77
iso-parametric map, 103

lagrange polynomial, 56
Laplace operator, 68
linear equation, 24

master element, 56,95
mesh, 54
Monte Carlo Method, 29

natural boundary conditions, 70
Neumann type, 26
non-linear, 24

parabolic equation, 26,85
partial differential equations, 24
Pascal's triangle, 75
perturbation method, 29
piecewise function, 46
poisson equation, 25
power series method, 29

quadrilateral element, 104
quasi-linear, 24

rectangular element, 75
residual, 34,51
Robin type, 26

shape function, 100
shape functions, 54
smoothness, 36
Sobolev spaces, 47
square-integrable, 36,63
stability, 88
sub-parametric map, 103
super-parametric map, 103

topology, 54,81,109
transformation, 55,96,103
trial functions, 34
triangular element, 73

unconditional stability, 91

variational approach, 32
variational Statement, 107
variational statement, 33,52,68

wave equation, 25
weak statement, 35

weighted residual, 32,37

weighting function, 34