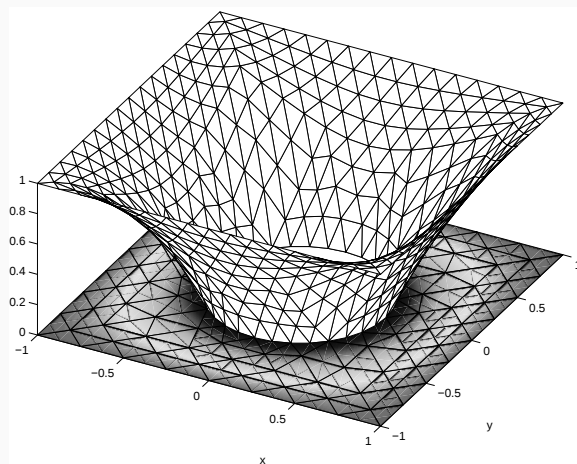


Introduction to Finite and Spectral Element Methods using MATLAB



C. Pozrikidis

© 2005, Chapman & Hall/CRC

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

Preface

Five general classes of numerical methods are available for solving ordinary and partial differential equations encountered in the various branches of science and engineering: finite-difference, finite-volume, finite element, boundary element, spectral and pseudo-spectral methods. The relation and relative merits of these methods are briefly discussed in the “Frequently Asked Questions” section preceding Chapter 1.

An important advantage of the finite element method, and the main reason for its popularity among academics and commercial developers, is the ability to handle solution domains with arbitrary geometry. Another advantage is the ability to produce solutions to problems governed by linear as well as nonlinear differential equations. Moreover, the finite element method enjoys a firm theoretical foundation that is mostly free of *ad hoc* schemes and heuristic numerical approximations. Although this may be a mixed blessing by being somewhat restrictive, it does inspire confidence in the physical relevance of the solution.

A current search of books in print reveals over two hundred items bearing in their title some variation of the general theme “Finite Element Method.” Many of these books are devoted to special topics, such as heat transfer, computational fluid dynamics (CFD), structural mechanics, and stress analysis in elasticity, while other books are written from the point of view of the applied mathematician or numerical analyst with emphasis on error analysis and numerical accuracy. Many excellent texts are suitable for a second reading, while others do a superb job in explaining the fundamentals but fall short in describing the development and practical implementation of algorithms for non-elementary problems.

The purpose of this text is to offer a venue for the rapid learning of the theoretical foundation and practical implementation of the finite element method and its companion spectral element method. The discussion has the form of a self-contained course that introduces the fundamentals on a need-to-know basis and emphasizes the development of algorithms and the computer implementation of the essential procedures. The audience of interest includes students in science and engineering, practicing scientists and engineers, computational scientists, applied mathematicians, and scientific computing enthusiasts.

Consistent with the introductory nature of this text and its intended usage as a practical primer on finite element and spectral element methods, error analysis is altogether ignored, and only the fundamental procedures and their implementation are discussed in sufficient detail. Specialized topics, such as Lagrangian formulations, free-boundary problems, infinite elements, and dis-

continuous Galerkin methods, are briefly mentioned in Appendix F entitled “Glossary,” which is meant to complement the subject index.

This text has been written to be used for self-study and is suitable as a textbook in a variety of courses in science and engineering. Since scientists and engineers of any discipline are familiar with the fundamental concepts of heat and mass transfer governed by the convection–diffusion equation, the main discourse relies on this prototype. Once the basic concepts have been explained and algorithms have been developed, problems in solid mechanics, fluid mechanics, and structural mechanics are discussed as extensions of the basic approach.

The importance of gaining simultaneous hands-on experience while learning the finite element method, or any other numerical method, cannot be overemphasized. To achieve this goal, the text is accompanied by a library of user-defined MATLAB functions and complete finite and spectral element codes composing the software library FEMLIB. The main codes of FSELIB are tabulated after the Contents. Nearly all functions and complete codes are listed in the text, and only a few lookup tables, ancillary graphics functions, and slightly modified codes are listed in abbreviated form or have been omitted in the interest of space.

The owner of this book can freely download and use the library subject to the conditions of the GNU public license from the web site:

<http://dehesa.freeshell.org/FSELIB>

For instructional reasons and to reduce the overhead time necessary for learning how to run the codes, the library is almost completely free of .dat files related to data structures. All necessary parameters are defined in the main code, and finite element grids are generated by automatic triangulation determined by the level of refinement for specific geometries. With this book as a user guide, the reader will be able to immediately run the codes as given, and graphically display solutions to a variety of elementary and advanced problems.

The MATLAB language was chosen primarily because of its ability to integrate numerical computation and computer graphics visualization, and also because of its popularity among students and professionals. For convenience, a brief MATLAB primer is included in Appendix G. Translation of a MATLAB code to another computer language is both straightforward and highly recommended. For clarity of exposition and to facilitate this translation, hidden operations embedded in the intrinsic MATLAB functions are intentionally avoided as much as possible in the FSELIB codes. Thus, two vectors are added explicitly component by component rather than implicitly by issuing a symbolic vector addition, and the entries of a matrix are often initialized and manipulated in a double loop running over the indices, even though this makes for a longer code.

Further information, a list of errata, links to finite element recourses, and updates of the FSELIB library are maintained at the book web site:

<http://dehesa.freeshell.org/FSEM>

Comments and corrections from readers are most welcome and will be communicated to the audience with due credit through the book web site.

I owe a great deal of gratitude to Todd Porteous for providing a safe harbor for the files, and to Mark Blyth and Haoxiang Luo for insightful comments on the manuscript leading to a number of improvements.

C. Pozrikidis
San Diego, California

Contents

Fselib software library	xv
Frequently Asked Questions	xxi
1 The finite element method in one dimension	1
1.1 Steady diffusion with linear elements	1
1.1.1 Linear element interpolation	2
1.1.2 Element grading	4
1.1.3 Galerkin projection	7
1.1.4 Formulation of a linear algebraic system	12
1.1.5 Relation to the finite difference method	16
1.1.6 Flux at the Dirichlet end	17
1.1.7 Galerkin finite element equations by way of Dirac's delta function	18
1.1.8 Element diffusion and mass matrices	21
1.1.9 Algorithm for assembling the linear system	25
1.1.10 Thomas algorithm for tridiagonal systems	25
1.1.11 Finite element code	27
1.2 Variational formulation and weighted residuals	35
1.2.1 Rayleigh-Ritz method for homogeneous Dirichlet bound- ary conditions	35
1.2.2 Rayleigh-Ritz method for inhomogeneous Dirichlet bound- ary conditions	38
1.2.3 Rayleigh-Ritz method for Dirichlet/Neumann boundary conditions	40
1.2.4 Rayleigh-Ritz method for Neumann/Dirichlet boundary conditions	42
1.3 Steady diffusion with quadratic elements	44
1.3.1 Thomas algorithm for pentadiagonal systems	50
1.3.2 Element matrices	51
1.3.3 Finite element code	56
1.3.4 Node condensation	61
1.3.5 Relation to the finite difference method	63
1.3.6 Modal expansion	68

1.4	Unsteady diffusion in one dimension	71
1.4.1	Galerkin projection	71
1.4.2	Integrating ODEs	73
1.4.3	Forward Euler method	73
1.4.4	Numerical stability	74
1.4.5	Finite element code	79
1.4.6	Crank-Nicolson integration	84
1.5	One-dimensional convection	86
1.5.1	Linear elements	87
1.5.2	Quadratic elements	91
1.5.3	Integrating ODEs	92
1.5.4	Nonlinear convection	94
1.6	One-dimensional convection–diffusion	94
1.6.1	Steady linear convection–diffusion	95
1.6.2	Nonlinear convection–diffusion	97
1.7	Beam bending	101
1.7.1	Euler-Bernoulli beam	101
1.7.2	Cubic Hermitian elements	106
1.7.3	Galerkin projection	108
1.7.4	Element stiffness and mass matrices	111
1.7.5	Code for cantilever beam with nodal loads	116
1.8	Beam buckling	121
1.8.1	Buckling under a compressive tip force	123
2	High-order and spectral elements in one dimension	129
2.1	Nodal bases	130
2.1.1	Lagrange interpolation	130
2.1.2	Element matrices	133
2.1.3	C^0 continuity and shared element nodes	133
2.1.4	Change of nodal base	134
2.2	Spectral interpolation	138
2.2.1	Lobatto nodal base	138
2.2.2	Discretization code	146
2.2.3	Legendre polynomials	146
2.2.4	Chebyshev second-kind nodal base	151
2.3	Lobatto interpolation and element matrices	153
2.3.1	Lobatto mass matrix	155
2.3.2	Lobatto integration quadrature	156
2.3.3	Computation of the Lobatto mass matrix	157
2.3.4	Computation of the Lobatto diffusion matrix	165
2.4	Spectral code for steady diffusion	172
2.4.1	Node condensation	179
2.5	Spectral code for unsteady diffusion	186
2.5.1	Crank-Nicolson discretization	187
2.5.2	Forward Euler discretization	192

2.6	Modal expansion	194
2.6.1	Implementation	197
2.6.2	Lobatto modal expansion	197
3	The finite element method in two dimensions	205
3.1	Convection–diffusion in two dimensions	206
3.1.1	Galerkin projection	208
3.1.2	Domain discretization and interpolation	209
3.1.3	GFEM equations	211
3.1.4	Implementation of the Dirichlet boundary condition	213
3.1.5	Split nodes	215
3.1.6	Variational formulation	216
3.2	3-node triangles	219
3.2.1	Element matrices	222
3.2.2	Computation of the element diffusion matrix	224
3.2.3	Computation of the element mass matrix	226
3.2.4	Computation of the element advection matrix	228
3.3	Grid generation	231
3.3.1	Successive subdivisions	231
3.3.2	Delaunay triangulation	236
3.3.3	Generalized connectivity matrices	243
3.3.4	Element and node numbering	243
3.4	Code for Laplace’s equation with the Dirichlet boundary condition in a disk-like domain	247
3.5	Code for steady convection–diffusion with the Dirichlet boundary condition	254
3.6	Code for Helmholtz’s equation with the Neumann boundary condition	260
3.7	Code for Laplace’s equation with Dirichlet and Neumann boundary conditions	266
3.8	Bilinear quadrilateral elements	272
4	Quadratic and spectral elements in two dimensions	277
4.1	6-node triangular elements	277
4.1.1	Integral over the triangle	282
4.1.2	Isoparametric interpolation and element matrices	282
4.1.3	Element matrices and integration quadratures	286
4.2	Grid generation and finite element codes	292
4.2.1	Triangulation of an L-shaped domain	299
4.2.2	Triangulation of a square with a square or circular hole	299
4.2.3	Triangulation of a rectangle with a circular hole	299
4.2.4	Code for Laplace’s equation with the Dirichlet boundary condition	311
4.2.5	Code for steady convection–diffusion	316
4.3	High-order triangle expansions	321

4.3.1	Computation of the node interpolation functions	324
4.3.2	Appell polynomial base	328
4.3.3	Prorior polynomial base	331
4.3.4	The Lebesgue constant	335
4.3.5	Node condensation	335
4.4	High-order node distributions	336
4.4.1	Node distribution based on a 1D master grid	336
4.4.2	Uniform grid	338
4.4.3	Lobatto grid on the triangle	341
4.4.4	The Fekete set	343
4.4.5	Further nodal distributions	345
4.5	Modal expansion on the triangle	346
4.6	Surface elements	352
4.7	High-order quadrilateral elements	355
4.7.1	8-node serendipity elements	356
4.7.2	12-node serendipity elements	358
4.7.3	Grid nodes via tensor-product expansions	359
4.7.4	Modal expansion	362
5	Applications in solid and fluid mechanics	367
5.1	Plane stress-strain analysis	367
5.1.1	Elements of elasticity theory	368
5.1.2	Plane stress	372
5.1.3	Plane strain	375
5.2	Finite element methods for plane stress/strain	378
5.2.1	Code for plane stress	380
5.3	Plate bending	398
5.3.1	Tensions and bending moments	399
5.3.2	Tension and bending moment tensors	400
5.3.3	Equilibrium equations	403
5.3.4	Boundary conditions	404
5.3.5	Constitutive and governing equations	405
5.3.6	Buckling and wrinkling of a stressed plate	408
5.3.7	Buckling and wrinkling of a circular plate	410
5.4	Hermite triangles	412
5.4.1	4-node, 10-dof triangle	413
5.4.2	The Morley element	423
5.4.3	6-node, 21-dof triangle	427
5.4.4	The Hsieh-Clough-Tocher (HCT) element	428
5.4.5	An echelon of plate bending elements	435
5.5	Finite element methods for plate bending	436
5.5.1	Formulation as a biharmonic equation	437
5.5.2	Formulation as a system of Poisson-like equations	448
5.6	Viscous flow	454
5.6.1	Velocity and pressure interpolation	456

5.6.2	Galerkin projections	457
5.7	Stokes flow	459
5.7.1	Galerkin finite element equations	460
5.7.2	Code for Stokes flow in a rectangular cavity	464
5.7.3	Triangularization	471
5.8	Navier-Stokes flow	474
5.8.1	Steady state	476
5.8.2	Time integration	476
5.8.3	Formulation based on the pressure Poisson equation	478
6	Finite and spectral element methods in three dimensions	481
6.1	Convection–diffusion in three dimensions	481
6.1.1	Galerkin projection	483
6.1.2	Galerkin finite element equations	484
6.1.3	Element matrices	485
6.1.4	Element types	486
6.2	4-node tetrahedral elements	487
6.2.1	Parametric representation	488
6.2.2	Integral over the volume of the tetrahedron	491
6.2.3	Element matrices	492
6.2.4	Mapping the tetrahedron to a cube	496
6.2.5	Orthogonal polynomials over the tetrahedron	497
6.3	High-order and spectral tetrahedral elements	501
6.3.1	Uniform node distribution	502
6.3.2	Arbitrary node distributions	508
6.3.3	Spectral node distributions	508
6.3.4	Modal expansion	510
6.4	Hexahedral elements	513
6.4.1	Parametric representation	514
6.4.2	Integral over the volume of the hexahedron	516
6.4.3	High-order and spectral hexahedral elements	517
6.4.4	Modal expansion	518
	Appendices	523
A	Function interpolation	525
A.1	The interpolating polynomial	525
A.1.1	Vandermonde matrix	526
A.1.2	Generalized Vandermonde matrix	528
A.1.3	Newton interpolation	529
A.2	Lagrange interpolation	530
A.2.1	Cauchy relations	531
A.2.2	Representation in terms of the Lagrange generating polynomial	531
A.2.3	First derivative and the node differentiation matrix	532
A.2.4	Representation in terms of the Vandermonde matrix	535

A.2.5	Lagrange polynomials corresponding to polynomial roots	538
A.2.6	Lagrange polynomials for Hermite interpolation	539
A.3	Error in polynomial interpolation	541
A.3.1	Convergence and the Lebesgue constant	542
A.4	Chebyshev interpolation	544
A.5	Lobatto interpolation	546
A.6	Interpolation in two and higher dimensions	550
B	Orthogonal polynomials	553
B.1	Definitions and basic properties	553
B.2	Gaussian integration quadratures	563
B.3	Lobatto integration quadrature	566
B.4	Chebyshev integration quadrature	567
B.5	Legendre polynomials	569
B.6	Lobatto polynomials	571
B.7	Chebyshev polynomials	572
B.8	Jacobi polynomials	575
C	Linear solvers	577
C.1	Gauss elimination	577
C.1.1	Pivoting	579
C.1.2	Implementation	580
C.1.3	Gauss elimination code	583
C.1.4	Multiple right-hand sides	584
C.1.5	Computation of the inverse	584
C.1.6	Gauss-Jordan reduction	589
C.2	Iterative methods based on matrix splitting	589
C.2.1	Jacobi's method	590
C.2.2	Gauss-Seidel method	591
C.2.3	Successive over-relaxation method (SOR)	591
C.2.4	Operator- and grid-based splitting	592
C.3	Iterative methods based on path search	592
C.3.1	Symmetric and positive-definite matrices	593
C.3.2	General methods	601
C.4	Finite element system solvers	601
D	Mathematical supplement	603
D.1	Index notation	603
D.2	Kronecker's delta	603
D.3	Alternating tensor	604
D.4	Two- and three-dimensional vectors	604
D.5	Del or nabla operator	605
D.6	Gradient and divergence	605
D.7	Vector identities	606
D.8	Divergence theorem in three dimensions	606
D.9	Divergence theorem in two dimensions	607

D.10 Stokes' theorem	607
E Element grid generation	609
F Glossary	611
G Matlab primer	615
G.1 Programming in MATLAB	616
G.1.1 Grammar and syntax	616
G.1.2 Precision	617
G.1.3 MATLAB commands	619
G.1.4 Elementary examples	621
G.2 MATLAB functions	625
G.3 Numerical methods	628
G.4 MATLAB Graphics	629
References	639
Index	645

FSELIB *software library*

The FSELIB software library accompanying this book contains miscellaneous functions and complete finite and spectral element codes written in MATLAB, including modules for domain discretization, system assembly and solution, and graphics visualization. The codes are arranged in directories corresponding to the book chapters and appendices. The owner of this book can download the library freely from the Internet site:

<http://dehesa.freeshell.org/FSELIB>

The software resides in the public domain and should be used strictly under the terms of the GNU General Public License, as stated on the GNU Internet page cited below.

The following tables list selected FSELIB finite and spectral element codes for problems in one and two dimensions, corresponding to Chapters 1–5 and appendices. A complete list of the FSELIB functions and codes can be found in the first part of the subject index.

FSELIB Finite and Spectral Element Software
Copyright ©2005 C. Pozrikidis

This library is free software; you can redistribute it and/or modify it under the terms of the current GNU General Public License as published by the Free Software Foundation (<http://www.gnu.org/copyleft/gpl.html>.)

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Chapter 3: The finite element method in two dimensions

Directory: 03

<i>Code:</i>	<i>Problem:</i>	<i>Element type:</i>
hlm3_n	Helmholtz's equation in a disk-like domain, with the Neumann boundary condition	3-node triangles
lap13_d	Laplace's equation in a disk-like domain, with the Dirichlet boundary condition	3-node triangles
lap13_dn	Laplace's equation in a disk-like domain, with Dirichlet and Neumann boundary conditions	3-node triangles
lap13_dn_sqr	Laplace's equation in a square domain, with Dirichlet and Neumann boundary conditions	3-node triangles
scd3_d	Steady convection–diffusion in a disk-like domain, with the Neumann boundary condition	3-node triangles

Chapter 4: Quadratic and spectral elements in two dimensions

Directory: 04-05

<i>Code:</i>	<i>Problem:</i>	<i>Element type:</i>
lap16_d	Laplace's equation in a disk-like domain, with the Dirichlet boundary condition	6-node triangles
lap16_d_L	Laplace's equation in an L-shaped domain, with the Dirichlet boundary condition	6-node triangles
lap16_d_rc	Laplace's equation in a rectangular domain with a circular hole, with the Dirichlet boundary condition	6-node triangles
lap16_d_sc	Laplace's equation in a square domain with a circular hole, with the Dirichlet boundary condition	6-node triangles
lap16_d_ss	Laplace's equation in a square domain with a square hole, with the Dirichlet boundary condition	6-node triangles
scd6_d	Steady convection–diffusion in a disk-like domain, with the Neumann boundary condition	6-node triangles
scd6_d_rc	Steady convection–diffusion in a rectangular domain with a circular hole, with the Neumann boundary condition	6-node triangles

Chapter 5: Applications in solid and fluid mechanics

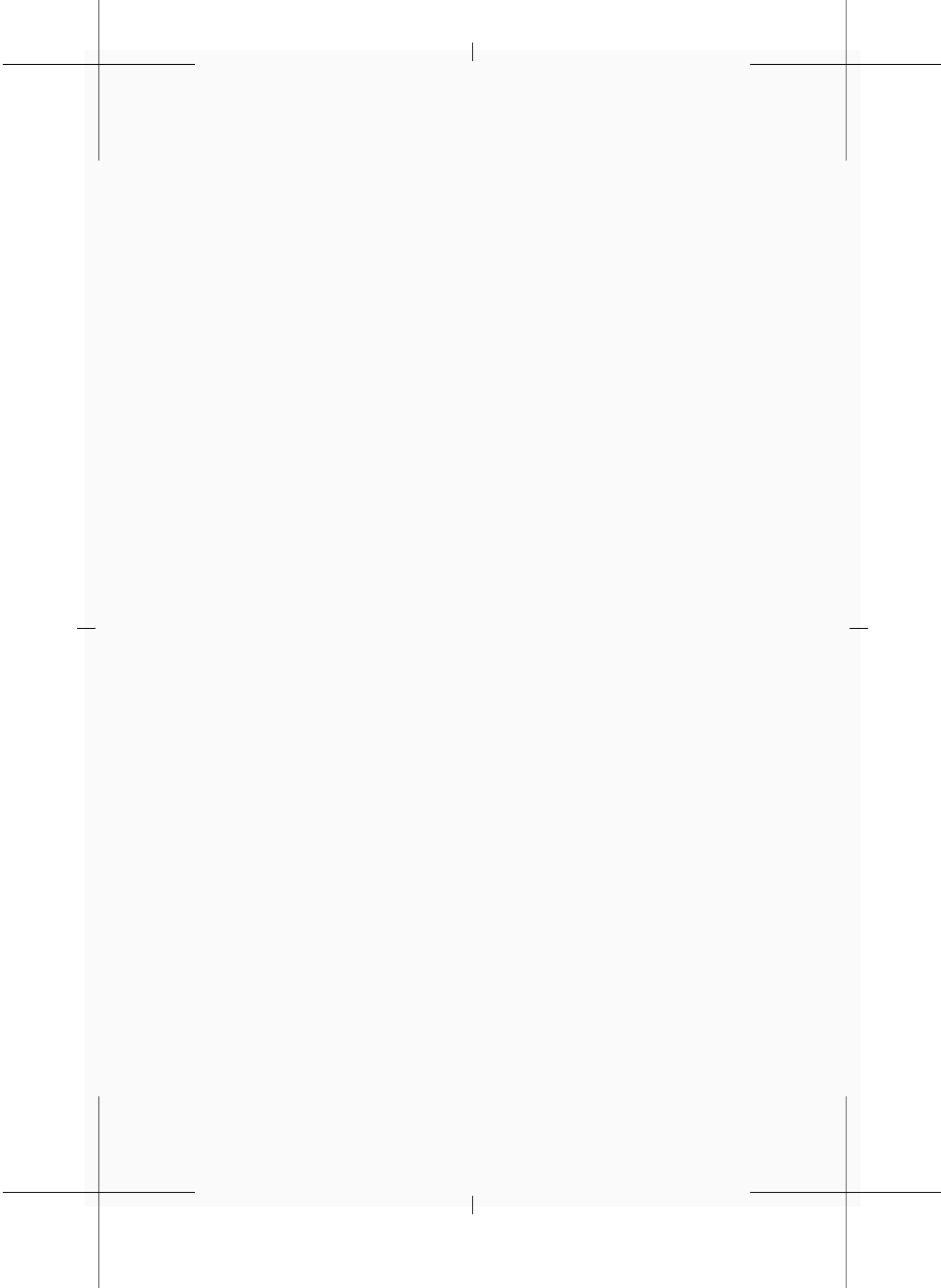
Directory: 04-05

<i>Code:</i>	<i>Problem:</i>	<i>Element type:</i>
bend_HCT	Bending of a clamped plate based on the biharmonic equation	HCT triangles
cvt6	Stokes flow in a rectangular cavity	6-node triangles
membrane	In-plane deformation of a membrane patch under a constant body force in plane stress analysis	6-node triangles
psa6	Plane stress analysis in a rectangular domain possibly with a circular hole	6-node triangles

Appendix C: Linear solvers

Directory: AC

<i>Code:</i>	<i>Problem:</i>
ge1	Solution of a linear system by Gauss elimination
cg	Solution of a symmetric system by the method of conjugate gradients



Frequently Asked Questions

- *What is the finite element method (FEM)?*

The finite element method is a numerical method for solving partial differential equations encountered in the various branches of mathematical physics and engineering. Examples include Laplace's equation, Poisson's equation, Helmholtz's equation, the convection–diffusion equation, the equations of potential and viscous flow, the equations of electrostatics and electromagnetics, and the equations of elastostatics and elastodynamics.

- *When was the finite element method conceived?*

The finite element method was developed in the mid-1950s for problems in stress analysis under the auspices of linear elasticity. Since then, the method has been generalized and applied to solve a broad range of differential equations, with applications ranging from fluid mechanics to structural dynamics.

- *What is the Galerkin finite element method (GFEM)?*

The GFEM is a particular and most popular implementation of the FEM, in which algebraic equations are derived from the governing differential equations by a process called the Galerkin projection.

- *What are the advantages of the finite element method?*

The most significant practical advantage is the ability to handle solution domains with arbitrary geometry. Another important advantage is that transforming the governing differential equations to a system of algebraic equations is performed in a way that is both theoretically sound and free of *ad hoc* schemes and heuristic numerical approximations. Moreover, the finite element method is built on a rigorous theoretical foundation. Specifically, for a certain class of differential equations, it can be shown that the finite element method is equivalent to a properly-posed functional minimization method.

- *What is the origin of the terminology “finite element?”*

In the finite element method, the solution domain is discretized into elementary units called finite elements. For example, in the case of a two-dimensional domain, the finite elements can be triangles or quadrilateral elements. The

discretization is typically unstructured, meaning that new elements may be added or removed without affecting an existing element structure, and without requiring a global element and node relabeling.

- *Is there a restriction on the type of differential equation that the finite element method can handle?*

In principle, the answer is negative. In practice, the finite element method works best for diffusion-dominated problems, and has been criticized for its inability to handle convection-dominated problems occurring, for example, in high-speed flows. However, modifications of the basic procedure can be made to overcome this limitation and improve the performance of the algorithms.

- *How does the finite element compare with the finite difference method (FDM)?*

In the finite difference method, a grid is introduced, the differential equation is applied at a grid node, and the derivatives are approximated with finite differences to obtain a system of algebraic equations. Because grid nodes must lie at boundaries where conditions are specified, the finite difference method is restricted to solution domains with simple geometry, or else requires the use of cumbersome boundary-fitted coordinates and artificial body forces for smearing out the boundary location.

- *How does the finite element method compare with the finite volume method (FVM)?*

In the finite volume method, the solution domain is also discretized into elementary units called finite volumes. The differential equation is then integrated over the individual volumes, and the divergence theorem is applied to derive equilibrium equations. In the numerical implementation, solution values are defined at the vertices, faces, or centers of the individual volumes, and undefined values are computed by neighbor averaging. Although the finite volume method is also able to handle domains with arbitrary geometrical complexity, the required *ad-hoc* averaging puts it at a disadvantage.

- *How does the FEM compare with the boundary element method (BEM)?*

Because the boundary element method (BEM) requires discretizing only the boundaries of a solution domain, it is significantly superior (e.g., [44]). In contrast, the finite element method requires discretizing the whole of the solution domain, including the boundaries. For example, in three dimensions, the BEM employs surface elements, whereas the FEM employs volume elements. However, the BEM is primarily applicable to linear differential equations with constant coefficients. Its implementation to more general types of differential equations is both cumbersome and computationally demanding.

- *How does the FEM compare with the spectral and pseudo-spectral method?*

In one class of spectral and pseudo-spectral methods, the solution is expanded in a series of orthogonal basis functions, the expansion is substituted in the differential equation, and the coefficients of the expansion are computed by projection or collocation. These methods are suitable for solution domains with simple geometry.

- *What should one know before one is able to understand the theoretical foundation of the FEM?*

The basic concepts are discussed in a self-contained manner in this book. Prerequisites are college-level calculus, numerical methods, and a general familiarity with computer programming.

- *What should one know before one is able to write a FEM code?*

Prerequisites are general-purpose numerical methods, including numerical linear algebra, function interpolation, and function integration. All necessary topics are discussed in this text, and summaries are given in appendices. Familiarity with a computer programming language is another essential prerequisite.

- *What is the spectral element method?*

The spectral element method is an advanced implementation of the finite element method in which the solution over each element is expressed in terms of *a priori* unknown values at carefully selected spectral nodes. The advantage of the spectral element method is that stable solution algorithms and high accuracy can be achieved with a low number of elements under a broad range of conditions.

- *How can one keep up with new developments in the finite and spectral element method?*

Several Internet sites provide current information on various aspects of the finite and spectral element methods. Links are provided at the book web site:

<http://dehesa.freeshell.org/FSEM>