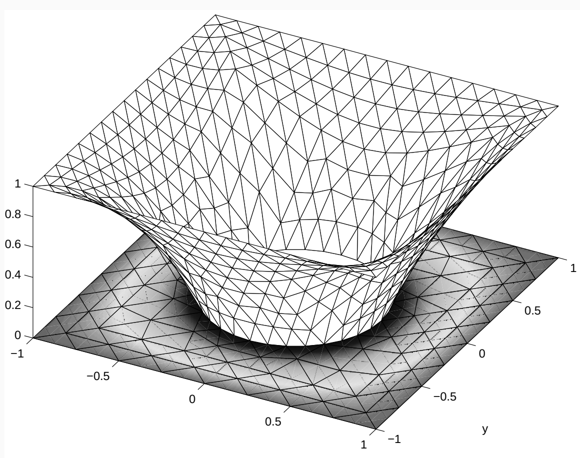
Introduction to Finite and Spectral Element Methods using MATLAB



C. Pozrikidis

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 discontinuous 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*

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.

Chapter 3: The finite element method in two dimensions

Directory: 03

Code:	*Problem:*	*Element type:*
`hlm3_n`	Helmholtz's equation in a disk-like domain, with the Neumann boundary condition	3-node triangles
`lapl3_d`	Laplace's equation in a disk-like domain, with the Dirichlet boundary condition	3-node triangles
`lapl3_dn`	Laplace's equation in a disk-like domain, with Dirichlet and Neumann boundary conditions	3-node triangles
`lapl3_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

Code:	*Problem:*	*Element type:*
`lapl6_d`	Laplace's equation in a disk-like domain, with the Dirichlet boundary condition	6-node triangles
`lapl6_d_L`	Laplace's equation in an L-shaped domain, with the Dirichlet boundary condition	6-node triangles
`lapl6_d_rc`	Laplace's equation in a rectangular domain with a circular hole, with the Dirichlet boundary condition	6-node triangles
`lapl6_d_sc`	Laplace's equation in a square domain with a circular hole, with the Dirichlet boundary condition	6-node triangles
`lapl6_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

Code:	*Problem:*	*Element type:*
`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

Code:	*Problem:*
`gel`	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`