国家出版基金项目
NATIONAL PUBLICATION FOUNDATION

Series in Information and Computational Science —80

Finite Element Language and Its Applications I

Liang Guoping (梁国平) Zhou Yongfa (周永发)
Translated by Gu Quan (古泉译)

(有限元语言及应用 I)

SCIENCE PRESS
Beijing

国家出版基金项目
NATIONAL PUBLICATION FOUNDATION

Series in Information and Computational Science 80

Finite Element Language and Its Applications I

(有限元语言及应用 I)

Liang Guoping (梁国平) Zhou Yongfa (周永发)

Translated by Gu Quan (古泉译)

SCIENCE PRESS
Beijing, China

Responsible Editors: Li Xin, Zhao Yanchao

Published by Science Press
16 Donghuangchenggen North Street
Beijing 100717, P. R. China

Printed in Beijing

ISBN 978-7-03-046517-7

Preface to the Series in Information and Computational Science

Since the 1970s, Science Press has published more than thirty volumes in its series Monographs in Computational Methods. This series was established and led by the late academician, Feng Kang, the founding director of the Computing Center of the Chinese Academy of Sciences. The monograph series has provided timely information of the frontier directions and latest research results in computational mathematics. It has had great impact on young scientists and the entire research community, and has played a very important role in the development of computational mathematics in China.

To cope with these new scientific developments, the Ministry of Education of the People's Republic of China in 1998 combined several subjects, such as computational mathematics, numerical algorithms, information science, and operations research and optimal control, into a new discipline called Information and Computational Science. As a result, Science Press also reorganized the editorial board of the monograph series and changed its name to Series in Information and Computational Science. The first editorial board meeting was held in Beijing in September 2004, and it discussed the new objectives, and the directions and contents of the new monograph series.

The aim of the new series is to present the state of the art in Information and Computational Science to senior undergraduate and graduate students, as well as to scientists working in these fields. Hence, the series will provide concrete and systematic expositions of the advances in information and computational science, encompassing also related interdisciplinary developments.

I would like to thank the previous editorial board members and assistants, and all the mathematicians who have contributed significantly to the monograph series on Computational Methods. As a result of their contributions the monograph series achieved an outstanding reputation in the community. I sincerely wish that we will extend this support to the new Series in Information and Computational Science, so that the new series can equally enhance the scientific development in information and computational science in this century.

Shi Zhongci
2005.7

Preface

This book is a new edition of the previous one *Finite Element Language.* A new part 'applications' has been added in the current version, as shown explicitly in the title. Finite Element Language (FEL) is a state of art modeling language used to solve partial differential equations (PDEs) by using finite element method (FEM) or finite volume method (FVM). This language is used to generate computer programs of FEM/FVM, by simply creating system-defined expressions for PDEs and its corresponding algorithms. The computer program of FEM/FVM (e.g., C or Fortran code) can be automatically generated by using the generator of this language.

When programming using FEL, the amount of code generated is reduced by more than 90 percent compared with that generated by other advanced language generators, thus it tremendously improves the efficiency of programming. Moreover, the system-defined expressions for PDFs and its algorithms are extremely easy for user to read, modify, update, re-use and maintain. FEL helps engineers and researchers to mainly focus on understanding their physics problems and creating the appropriate mathematical models by making them free from the tedious, time-consuming and error-prone coding work.

This book is organized as follows: Part I includes 5 chapters and appendix A - F. Chapter 1 discusses the description language for creating the expressions of PDFs. These expressions are used by the system to generate the element subroutines in FEM/FVM; Chapter 2 presents the fundamental method to create the FEM algorithms for solving problems in single-physics field; Contents presented in Chapter 3 are similar to that in Chapter 2 but involed with coupled problems in multi-physics fields; Chapter 4 introduces the strategy for building FEL which is based on the component-based-programming method. Details about five most commonly used component programs are also given; The FEM data structure is presented and discussed in Chapter 5. Appendix A to C provide some fundamental concepts and knowledge for finite element shap function and element types as well as the coordinate transformations and numerical integration. Appendix D and E contain the collection of keywords and some specific statements defined in FEL.

Part II includes six chapters, introducing the applications of FEL in solid mechanics, Navier-Stokes equation, Darcy flow, electromagnetic field, structural mechanics and thermal field problems, respectively. It is worth mentioning that the analytical examples in the book are only used to illustrate the specific applications of FEL, some of the application results have not been strictly benchmarked so it is user's responsibility to perform the verification. The pre- and post-processing work is done based on FEPG.GID platform.

The history of FEL and the developed software FEPG can be tracked back to the 1980s. FEPG has been involved from the early version of only working on single CPU to the latest version which works on HPC and internet and provides user with very friendly GUI, thanks to the rapid development of modern simulation and high performance computing technologies.

The early users of FEPG have become its 'fans' and strong supporters or even participants. However, limited by the current situation of CAE industry in China, the promotion of FEL and FEPG face big challenges. We would like to take the opportunity when this book being published, to invite people in scientific computing community in China to join us in promoting FEPG and developing our own high performance finite element software.

Contents

Introduction

In the 1950's, scientists in structural mechanics in the united states invented finite element method. The main principles of this method are originally based on variational method and piecewise low-order interpolation polynomial. However these two basic principles no longer meet user's needs due to the vast finite element applications in physics and engineering over the past decades. They have been then developed based on partial differential equation weak form and arbitrary approximation function space.

In physics and engineering, boundary value problem of partial differential equations (PDEs) in general can be expressed as follows.

In the solving domain Ω, the partial differential is satisfied by

$$A(u) = \left\{ \begin{array}{c} A_1(u) \\ A_2(u) \\ \vdots \end{array} \right\} = 0 \quad (\Omega) \tag{1}$$

on the boundary $\partial\Omega$, the boundary value condition is satisfied

$$B(u) = \left\{ \begin{array}{c} B_1(u) \\ B_2(u) \\ \vdots \end{array} \right\} = 0 \quad (\partial\Omega) \tag{2}$$

Where A and B are differential operators, u is unknown function.

In general, these partial differential equations have no analytical solution and usually are solved by using numerical methods. Finite element method is one of these numerical methods that plays a major role in solving various PDEs effectively by providing the weak form solution.

The PDE (1) is multiplied by the test function δu and then integrated on Ω, this gives

$$\int_\Omega A^{\mathrm{T}}(u)\delta u \mathrm{d}\Omega = \int_\Omega A_i(u)\delta u \mathrm{d}\Omega = 0 \tag{3}$$

Where δu is the arbitrary function.

Assuming $A(u)$ is a smooth function and the integral equation (3) is satisfied for any δu, then PDE (1) is satisfied at any point of Ω. This is because if at some points or some subdomains of Ω $A(u)$ doesn't satisfy (1), namely $A(u) \neq 0$, an appropriate function δu can be found to make the integral form of equation (3) doesn't to equal zero. It is obvious that when $A(u)$ is a smooth function, equations (1) and (3) are equivalent. But in some cases, for example, the problem of 3-D harmonic electromagnetic field, these two are not equivalent. In many cases, by performing integration by parts to equation (3) and using boundary value condition (2), another form can be obtained:

$$\int_{\Omega} D^{\mathrm{T}}(u)C(\delta u)\mathrm{d}\Omega + \int_{\partial\Omega} F^{\mathrm{T}}(u)E(\delta u)\mathrm{d}S = 0 \tag{4}$$

Where C, D, E, F are differential operators. Note that the derivative order of unknown functions may be lower than the differential operator in equation (1), therefore only the continuity of lower order of function u is required. Equation (4) is called the weak form of differential equation (1) with the boundary condition (2). Weak form (4) may also be obtained using Petrov-Galerkin method.

Finite element approximate solution u is written as

$$u = \sum_{i=1}^{n} N_i u_i \tag{5}$$

The variational variable (namely dummy variable) is written as

$$\delta u = \sum_{i=1}^{n} N_i \delta u_i \tag{6}$$

Where u_i is the unknown variable to be solved, δu_i is the dummy variable corresponding to u_i, N_i is the basis function and the finite element space is composed of the basis functions $N_i\,(i = 1, 2, \cdots, n)$.

By Pluging (5) and (6) into (3), finite dimensional algebra equations can be obtained from the arbitrary selection of dummy variable δu_i, therefore the problem of solving PDEs is converted to the problem of solving finite dimensional algebra equations.

As seen from (5) and (6), the approximating solution space and test function space have the same basic functions. In this book, methods with different basis functions for these two spaces generally are not considered.

Chapter 1

Description Language of Differential Equation Expression

The description of differential equation, a basic content of Finite Element Language (FEL), is presented in this chapter. File that uses pdf as its extension name, referred to as PDE file, is adopted by FEL to describe the differential equation expression based on weak form. The automatic generating system of FEL generates element subroutines to calculate element stiffness matrix, element damping matrix, element load vector, etc, by using this file.

GES file, the most fundamental file, provides all formulas of FEM such as shape function, numerical integration, etc. It is used to generate element subroutine. PDE file allows use to obtian shape function and the formula of numerical integration from formula library, thus makes the creation easy and simple.

The files of CDE, VDE and FDE are designed to save time for creating PDE expressions. The main content of CDE file is for complex variable differential equation expression. And the main content of VDE file is for tensor expression of differential equation. FDE file can access operator formula library for operator expression.

The main content of FBC file is for differential equation expression of boundary condition (the second-type and third-type boundary conditions).The automatic generating system of element subroutine generates subroutines to calculate element stiffness matrix, damping matrix, load vector, etc for boundary conditions based on this file. The following sections are dedicated to the creation of these files.

1.1 Writing PDE Files

The main content of PDE file is to prepare differential equation expression. System will automatically generate element subroutines for calculating element stiffness matrix, element mass matrix, element damping matrix, etc. according to this file. The structure of PDF file is demonstrated as follows.

(1) User needs to create 6 segments at most whose keywords of information segments are DEFI, FUNC, STIF, MASS, DAMP and LOAD respectively. According