国家出版基金项目
NATIONAL PUBLICATION FOUNDATION

Series in Information and Computational Science —81

Finite Element Language and Its Applications II

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 II)

SCIENCE PRESS
Beijing

国家出版基金项目
NATIONAL PUBLICATION FOUNDATION

Series in Information and Computational Science 81

Finite Element Language and Its Applications II

(有限元语言及应用 II)

Liang Guoping (梁国平) Zhou Yongfa (周永发)

Translated by Gu Quan (古泉峰)

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 978-7-03-046594-8

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

Chapter 6

Solid Mechanics

6.1 Infinitesimal Linear Elastic Deformation

6.1.1 Finite Element Language Program for Static Problems

6.1.1.1 GCN Files

——— test.gcn ———

Comments are on the right

```
defi
a ell ....field a solves displacement, using ellipse algorithm for
   linear steady-state problem.
b str a .....field b solves stress, using the least squares method,
   lumped matrix method str.nfe, and the results of field a

startsin a ....  initialize field a
solvsin a .....  Use a sin solver without storing element stiffness
   but changing bandwidth in field a.
stress b .....  Introduced to solve Field b.  the least squares
   method
```

6.1.1.2 MDI Files

——— test.mdi ———

Comments are on the right

```
3dxyz .....  Solving in three-dimensional Cartesian coordinate system
#a 0 3 u v w .....  field a has no initial values, but has three
   displacement DOFs.
fde delxyz c8g2 .....  Element calculation subroutine of field a is
   delxyz.fde, with eight-node hexahedral element, and second-order
   Gaussian integral.
fbc delxyz q4g2 .....  Calculation subroutine of boundary element is
   delxyz. fbc. with four-node quadrilateral element and second-
   order Gaussian intergral.
#b 0 6 dxx dyy dzz dyz dxz dxy .....  field b has no initial values,
```

```
   but has six stress DOFs.
fde selxyz c8 ....  Element calculation subroutine of field b
   is selxyz.fde, with eight-node hexahedral element, and nodal
   integral
   #
```

6.1.1.3 Displacement Calculation Program

6.1.1.3.1 Theory Text

Balance equations

$$\begin{aligned}
&\frac{\partial \sigma_{xx}}{\partial x}+\frac{\partial \sigma_{xy}}{\partial y}+\frac{\partial \sigma_{xz}}{\partial z}+f_x=0\\
&\frac{\partial \sigma_{xy}}{\partial x}+\frac{\partial \sigma_{yy}}{\partial y}+\frac{\partial \sigma_{yz}}{\partial z}+f_y=0\\
&\frac{\partial \sigma_{xz}}{\partial x}+\frac{\partial \sigma_{yz}}{\partial y}+\frac{\partial \sigma_{zz}}{\partial z}+f_z=0
\end{aligned} \tag{6.1.1}$$

Geometric equations

$$\begin{aligned}
\varepsilon_{xx}&=\frac{\partial u}{\partial x}\\
\varepsilon_{yy}&=\frac{\partial v}{\partial y}\\
\varepsilon_{zz}&=\frac{\partial w}{\partial z}\\
\varepsilon_{yz}&=\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\\
\varepsilon_{xz}&=\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}\\
\varepsilon_{xy}&=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}
\end{aligned} \tag{6.1.2}$$

Constitutive equation

$$\begin{pmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{yz}\\ \sigma_{xz}\\ \sigma_{xy}\end{pmatrix}=\frac{E}{(1+\nu)(1-2\nu)}\begin{pmatrix}1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & 0.5-\nu & & \\ & & & & 0.5-\nu & \\ & & & & & 0.5-\nu\end{pmatrix}\begin{pmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ \varepsilon_{yz}\\ \varepsilon_{xz}\\ \varepsilon_{xy}\end{pmatrix} \tag{6.1.3}$$

Boundary conditions:

The first-type boundary conditions

$$u=u_0,\quad v=v_0,\quad w=w_0 \tag{6.1.4}$$

The second-type boundary conditions

$$T_x = f_1, \quad T_y = f_2, \quad T_z = f_3 \tag{6.1.5}$$

The third-type boundary conditions

$$T_x = f_1(u, v, w), \quad T_y = f_2(u, v, w), \quad T_z = f_3(u, v, w) \tag{6.1.6}$$

Using the principle of Galerkin FEM to solve displacement, from the balance equation we obtain:

$$\begin{aligned}&\int_V \left(\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + f_x \right) \delta u + \left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + f_y \right) \delta v \right. \\ &\left. + \left(\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + f_z \right) \right) \delta w \mathrm{d}V = 0 \end{aligned} \tag{6.1.7}$$

Translating into weak form yields:

$$\begin{aligned}&\int_V (\sigma_{xx}\delta\varepsilon_{xx} + \sigma_{yy}\delta\varepsilon_{yy} + \sigma_{zz}\delta\varepsilon_{zz} + \sigma_{yz}\delta\varepsilon_{yz} + \sigma_{xz}\delta\varepsilon_{xz} + \sigma_{xy}\delta\varepsilon_{xy})\,\mathrm{d}V \\ &= \int_V (f_x\delta u + f_y\delta v + f_z\delta w)\,\mathrm{d}V + \int_\Gamma (T_x\delta u + T_y\delta v + T_z\delta w)\,\mathrm{d}\Gamma \end{aligned} \tag{6.1.8}$$

Plugging the constitutive equations into the above, we obtain

$$\begin{aligned}&\int_V \varepsilon_{xx}\delta\varepsilon_{xx}\frac{E}{(1+\nu)(1-2\nu)}(1-\nu)\mathrm{d}V + \int_V \varepsilon_{xx}\delta\varepsilon_{yy}\frac{E}{(1+\nu)(1-2\nu)}(\nu)\mathrm{d}V \\ &+ \int_V \varepsilon_{xx}\delta\varepsilon_{zz}\frac{E}{(1+\nu)(1-2\nu)}(\nu)\mathrm{d}V + \int_V \varepsilon_{yy}\delta\varepsilon_{xx}\frac{E}{(1+\nu)(1-2\nu)}(\nu)\mathrm{d}V \\ &+ \int_V \varepsilon_{yy}\delta\varepsilon_{yy}\frac{E}{(1+\nu)(1-2\nu)}(1-\nu)\mathrm{d}V + \int_V \varepsilon_{yy}\delta\varepsilon_{zz}\frac{E}{(1+\nu)(1-2\nu)}(\nu)\mathrm{d}V \\ &+ \int_V \varepsilon_{zz}\delta\varepsilon_{xx}\frac{E}{(1+\nu)(1-2\nu)}(\nu)\mathrm{d}V + \int_V \varepsilon_{zz}\delta\varepsilon_{yy}\frac{E}{(1+\nu)(1-2\nu)}(\nu)\mathrm{d}V \\ &+ \int_V \varepsilon_{zz}\delta\varepsilon_{zz}\frac{E}{(1+\nu)(1-2\nu)}(1-\nu)\mathrm{d}V + \int_V \varepsilon_{yz}\delta\varepsilon_{yz}\frac{E}{(1+\nu)(1-2\nu)}(0.5-\nu)\mathrm{d}V \\ &+ \int_V \varepsilon_{xz}\delta\varepsilon_{xz}\frac{E}{(1+\nu)(1-2\nu)}(0.5-\nu)\mathrm{d}V \\ &+ \int_V \varepsilon_{xy}\delta\varepsilon_{xy}\frac{E}{(1+\nu)(1-2\nu)}(0.5-\nu)\mathrm{d}V \end{aligned}$$

$$= \int_V (f_x\delta u + f_y\delta v + f_z\delta w)\,\mathrm{d}V + \int_\Gamma (T_x\delta u + T_y\delta v + T_z\delta w)\,\mathrm{d}\Gamma \tag{6.1.9}$$

By spatial discretization, equation (6.1.9) can be written in matrix form, which gives

$$SU = F \tag{6.1.10}$$

Where S is the stiffness matrix, U is the displacement vector, F is the load vector, and the above form is the linearized equation system.

6.1.1.3.2 FDE Files

The volume integral terms of equation (6.1.9) should be written as FDE file, which is described as follows:

——— delxyz.fde ———

```
\................elastic deformation equation .........
\sij,j + fi = 0
\s = D*e, where eij = (ui,j+uj,i)*aij
\aij = 1/2 if i=j else aij=1
\where u denotes displacement, s denote stress, e denotes strain
\D denotes the constitutive matrix
\ .....................................................
\PDE in weak form
\(D*e,de) = (f,du)
\where de denotes the variations of e
\ .....................................................
disp u v w
vect u u v w
coor x y z
func exx eyy ezz eyz exz exy
fmatr fe 3 3
vect ev exx eyy ezz
vect ep eyz exz exy
shap %1 %2
gaus %3
@l singular.xyz n
mate pe pv fu fv fw rou alpha
vect f fu fv fw
matrix sm 3 3
   (1.0-pv) pv pv
pv (1.0-pv) pv
pv pv (1.0-pv)
```

```
func
@l vol.xyz n
$c6 fact=pe/(1.0+pv)/(1.0-pv*2.0)*vol
$c6 shear=(0.5-pv)
@l gradv.xyz f fe
@w ev fe 1 5 9
@a fe_i_j=+[fe_i_j]+[fe_j_i]
@w ep fe 6 3 2
stif
dist=+[ev_i;ev_j]*sm_i_j*fact+[ep_i;ep_i]*shear*fact

load=+[u_i]*f_i*vol

end
```

6.1.1.3.3 VDE Files

The VDE file generated by FDE file in the previous section is as follows:

——— delxyz.vde ———

```
disp u v w
vect u u v w
coor x y z
func exx eyy ezz eyz exz exy
vect ev exx eyy ezz
vect ep eyz exz exy
shap %1 %2
gaus %3
mate pe pv fu fv fw rou alpha
vect f fu fv fw
matrix sm 3 3
(1.0-pv) pv pv
pv (1.0-pv) pv
pv pv (1.0-pv)

func
$c6 vol=1.0
$c6 fact=pe/(1.0+pv)/(1.0-pv*2.0)*vol
$c6 shear=(0.5-pv)
exx=+[u/x]

eyy=+[v/y]
```

```
ezz=+[w/z]

eyz=+[v/z]+[w/y]

exz=+[u/z]+[w/x]

exy=+[u/y]+[v/x]

stif
   dist=+[ev_i;ev_j]*sm_i_j*fact+[ep_i;ep_i]*shear*fact

load=+[u_i]*f_i*vol
end
```

6.1.1.3.4 FBC Files

Boundary integral terms in the equation (6.1.9) should be written in the FBC file of boundary element, which is described as follows:

——— delxyz.fbc ———

```
defi
disp u,v,w
coor x,y
shap %1 %2
gaus %3
mate fu fv fw 0.0;0.0;100.0;

stif
dist=+[u;u]*0.0

load=+[u]*fu+[v]*fv+[w]*fw

end
```

Local coordinate system are used for the boundary elements, the followings are the method of the local coordinate system: the x-axis of the local coordinate system is defined by the unit vector of whose direction is defined as starting from the first element node to the second; the unit vector which is vertical to the boundary unit and meet the right-hand screw rule with the order of the nodes of the boundary element is defined as the z-axis of the local coordinate system; taking the vector which is vertical to the x-axis and z-axis, moreover, can constitute the right-hand screw rule with the x-axis and z-axis as y-axis of local coordinate system. Therefore, f_x, f_y, f_z in the boundary element FBC file represent the boundary force in the local coordinate system.

6.1.1.4 Stress Calculation Program

6.1.1.4.1 Theory Text

Linear elastic constitutive equation

$$\begin{pmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{yz}\\ \sigma_{xz}\\ \sigma_{xy}\end{pmatrix}=\frac{E}{(1+\nu)(1-2\nu)}\begin{pmatrix}1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & 0.5-\nu & & \\ & & & & 0.5-\nu & \\ & & & & & 0.5-\nu\end{pmatrix}\begin{pmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ \varepsilon_{yz}\\ \varepsilon_{xz}\\ \varepsilon_{xy}\end{pmatrix}$$

The above equation can be abbreviated as

$$\sigma = D\varepsilon \tag{6.1.11}$$

Using the least squares method to solve stress with known displacement, the following equation is obtained by using the above formula (6.1.11):

$$\int_V \sigma\delta\sigma \mathrm{d}V = \int_V D\varepsilon\delta\sigma \mathrm{d}V \tag{6.1.12}$$

After spatial discretization (6.1.12) can be solved by utilizing the least squares method.

6.1.1.4.2 FDE Files

The volume integral terms of the equation (6.1.12) should be written as FDE file, described as follows:

——— selxyz.fde ———

```
\................stress computation .........
\stress equation
\s = D*e, where eij = (ui,j+uj,i)*aij
\aij = 1/2 if i=j else aij=1
\where u denotes displacement, s denotes stress, e denotes strain
\D denote the constitutive matrix
\.......................................................
\PDE in weak form
\(s,ds) = (D*e,ds)
\where ds denotes the variations of s
\.......................................................
disp dxx,dyy,dzz,dyz,dxz,dxy
coor x,y,z
coef u,v,w
```