THREE DIMENSIONAL NUMERICAL SIMULATION OF EXCAVATION AND BACKFILLING IN MINING ENGINEERING①

Gu Desheng, Deng Jian, Li Xibing and Zhang Jiasheng
College of Resourses, Environment and Civil Engineering,
Central South University of Technology, Changsha 410083, P. R. China

ABSTRACT A recently-developed three dimensional (3D) elastoplastic finite element program software on microcomputer was introduced, which finds effective application in mining engineering, such as multi-stage excavation, back-filling, caving, construction, etc. A symbol-identified input data system, parallel module structure and mesh auto-generation are some of the characteristics of the program. Two test examples are presented to verify the program software.

Key words mining engineering excavation back-filling numerical simulation

1 INTRODUCTION

In recent thirty years, with the development of computer technique, it becomes possible to simulate such complicated processes as mining engineering. The finite element method (FEM) is now firmly established as an engineering tool of wide applicability. Scholars from home and abroad have developed many FEM softwares, some of which have greatly contributed to numerical method aplication in practical engineering[1,2]. However, as known to all, those softwares that can correctly simulate dynamic process in mining engineering are difficult to find. It is not only because that mining' object —rock owns various properties and complex physical environment, but also because mining process consists of such complicated activities as excavation and back-filling. A new 3D elastoplastic finite element program is recently developed by the authors to simulate multistage excavation, back-filling, caving, construction, etc in mining engineering, with characteristics of a symbol-identified input data system, parallel module program structure, and mesh auto-generation.

2 SYMBOL-IDENTIFIED INPUT DATA SYSTEM

Fixed form data input system used to be employed in traditional computer programs' input data files. Consequently, users need to know much numerical knowledge and the input data files are not easily understood. Now some programs boast of vivid pre-process ability[1], but usually the users have only the rights to use, and such program often cannot correctly simulate excavation and back-filing. A symbol-identified input data system is adopted in our program software. As a result, FEM data input file with such character can be easily understood and adjusted. If the plot module is loaded, errors in such data files can be easily found on the computer screen.

Some typical subroutines employed to read data file are listed as follows:

CALL FIND(" NAME", KEY)
CALL READI (" NAME", IFLAG, N,

① Project 59625408 supported by the National Natural Science Foundation of China
Received Jun. 29, 1998; accepted Jan. 18, 1999

K_1, K_2, …, K_{22})

CALL READR("NAME", IFLAG, N, R_1, R_2, …, R_{22})

CALL READIA("NAME", IFLAG, IARRAY, N)

CALL READRA ("NAME", IFLAG, RARRAY, N)

where subroutine FIND searches for a leading character string "NAME", KEY is a flag, if KEY = 1 indicates 'not found', if KEY = 0, indicates 'found'. READI and READR are variable input subroutines (Integer and real type, respectively). READIA and READRA are array input subroutines (integer and real type, respectively). NAME is the leading character string, IFLAG is a flag, N is the number of variables, K_i and R_i (i = 1, …, 22) stand for integer and real type variable names respectively. IARRAY and RARRAY are the names of integer and real names respectively.

3 PARALLEL MODULE PROGRAM STRUCTRURE

The chain program structure that used to be employed in traditional computer programs for the finite element method is shown in Fig.1. When a program is running, all subroutines need to be loaded onto computer memory at the same time, and so occupy the memory all the time, which greatly confine the capability of the program. On the other hand, modifing just one algorithm means modifying the whole program.

In order to adapt microcomputer memory and to allow our program software to be rapidly developed or modified in a very short period of time with a minimum cost, the parallel module program structure which is shown in Fig.2 is adopted. One program module is not a subroutine in traditional program, but such a small independent program, which owns its own main program and subroutines, and can run freely on microcomputer. Program module is loaded onto memory and covers former module on the command of command control manager (CCM). All of the program modules are linked with each other by data files on the disk, and can not be loaded on each other. Any module can not only get information from disk data files, but also output necessary data information to disk files used in later modules. As a result, a complete problem solution process and a logic complete program software is integrated.

Fig.1 The chain structure of program

Fig.2 The parallel structure program

It is self-evident that the parallel module program structure and dynamic location of array dimension are characterised by and noted for being highly flexible, conveniently debugged and easily expanded according to the user's requirements.

4 3D MESH AUTO-GENERATION

FEM mesh auto-generation is a vigorous research direction in numerical analysis . Because 3D model is less vivid and less directly perceived than 2D, its geometry is often complex and corresponding errors are difficult to be found. Owing to rock variability and process complexity in mining engineering, 3D mesh auto-generation is especially important.

Twenty nodes or eight nodes per isoparametric element is adopted to simulate mining engineering variable boundaries. 3D mesh is auto-generated by image technique, which can be used to generate arbitrary fine mesh by inputing little original data. This method's essence is to use isoparametric element consistency between natural coordinates and Catesian coordinates:

$$
\begin{aligned}
x &= \sum_{i=1}^{8} N_i x_i \\
y &= \sum_{i=1}^{8} N_i x_i \\
z &= \sum_{i=1}^{8} N_i z_i
\end{aligned}
\tag{1}
$$

where N_i is shape function; x_i, y_i, z_i ($i = 1, 8$) are Catesian coodinates of corner nodes, x, y, z are coordinates which are to be calculated. It is evident that if corner nodes' and few special nodes' coordinates have already known, structure geometry and node coordinates can be auto-generated.

5 EXCAVATION AND BACKFILLING SIMULATION

Rock' stress and deformation are greatly influenced by surrounding pressure and loading history[5]. Excavation and backfilling in mining engineering both belong to mechanical response for structural alteration, which will lead to stress relief or increment. Consequently, middle or final stress and displacement are directly influenced by structural deformation history, e.g. excavation and backfilling activities. So it is very important to correctly simulate excavation and backfilling in mining engineering. Excavation simulation has been paid a great deal attention by lots of scholars[6,7]. However, traditional numerical difficulties have been encountered in finite element simulation of excavation[7]. Numerical simulation of backfilling is rarely discussed. In our program, simulation of excavation and backfilling is treated as a nonlinear problem. The source of nonlinearity is the change of geometry. Therefore, a general solution scheme for nonlinear problems can be used:

$$\boldsymbol{K}_n \Delta \boldsymbol{U}_n = \boldsymbol{R}_n - \boldsymbol{I}_{n-1} \tag{2}$$

where $\boldsymbol{K}$ is structural stiffness matrix; $\boldsymbol{U}$ is vector of incremental displacement, $\boldsymbol{R}$ is load vector, $\boldsymbol{I}$ is the vector of internal resisting forces computed from stresses for the previous step $n-1$,

$$\boldsymbol{I}_{n-1} = \Sigma \int \boldsymbol{B}^{\mathrm{T}} \sigma_{n-1} \mathrm{d} v \tag{3}$$

where $\boldsymbol{B}$ is strain displacement matrix and σ_{n-1} is the stress vector within each element. Displacement, strain, and stress are calculated as follows:

$$
\begin{aligned}
\boldsymbol{U}_n &= \boldsymbol{U}_{n-1} + \Delta \boldsymbol{U}_n \\
\varepsilon_n &= \varepsilon_{n-1} + \Delta \varepsilon_n \\
\sigma_n &= \sigma_{n-1} + \Delta \sigma_n
\end{aligned}
\tag{4}
$$

At step $n-1$, elements in excavation area are activated, their contributions to the stiffness matrix, element load vector, and internal resisting force vector are assembled, while from step n, these elements are deactivated. Backfilling simulation involves a procedure exactly opposite to that just described.

6 TEST EXAMPLES

In order to verify the accuracy and reliability of the elastoplastic finite element program software, its numerical solutions are compared with corresponding theoretical results or those from other literature.

The first numerical example considered is illustrated in Fig.3. It is an excavation problem to test the program's simulation of excavation in

Fig.3 Test excavation simulation model
(Dimension unit: m)

mining engineering. The supposed stratum material's elastic modulus $E = 100.031$ MPa, poissons ratio $\mu = 0.25$, body weight density $\rho = 19.614$ kN/m^3, cohesion $C = 117.684$ kPa, friction angle $\theta = 40°$, strain hardening parameter $H' = 0.0$ MPa. Initial stress is only caused by gravity, lateral pressure parameter $\zeta = \mu/(1 - \mu) = 0.3333$. The Mohr-coulomb yield criterion and plane strain condition are assumed. Numerical solutions are compared with Ref.[6], listed in Table 1, in which load case 1 ~ 5 are calculated by various excavation model simulation. Load case 6 is our results. According to Ref.[6], load case 1 is more correct than the other 4 cases. From Table 1, results of load cases 1 and 6 are similar. The main difference is the displacement of point 18. It seems that negative result more corresponds to the practical situation.

The second example is a thick cylinder subjected to internal pressure[8]. It is a theoretical problem to verify the program's elastoplastic simulation. Fig.4 is a 3D FEM mesh of quarter of this problem (because of symmetry). Fig.5 shows the circumferential (hoop) stress distributions, and good agreement between the numerical and analytical solutions is evident.

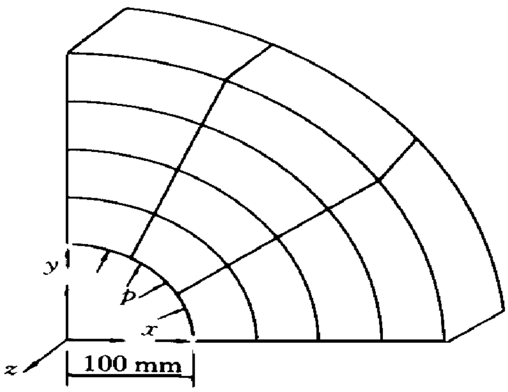

Fig.4 Thick cylinder model

Fig.5 Thick cylinder hoop stress distribution

Table 1 Test 1 main node displacement

Load case	Specific nodes			
	11	12	14	18
	Vertical disp/m	Vertical disp/m	Vertical disp/m	Vertical disp/m
1	-0.346221E-2	-0.922820E-2	0.927288E-2	0.110428E-2
2	0.117041E-1	0.103447E-1	0.741769E-2	0.211669E-4
3	-0.151663E-1	-0.195726E-1	0.185521E-2	0.108312E-2
4	-0.117041E-1	-0.103447E-1	-0.741767E-2	-0.211669E-4
5	-0.388269E-2	-0.103787E-1	0.149130E-1	-0.859066E-2
6	-0.3409E-2	-0.7035E-2	0.4229E-2	-0.2477E-2

7 CONCLUSIONS

A 3D elastoplastic finite element program, which is recently developed, is introduced. Our attention is focused on the program structure, input and output system, mesh autogeneration, and numerical simulation of excavation and backfilling. The program is correct and reliable after tested by examples. Parallel structure make the program easily understand and low-costly expanded or developed. Symbol-identified input data system is advanced and mesh autogeneration greatly reduces the volume of data file. Excavation and backfilling are treated as a non-linear problem, which makes it possible to correctly simulate such complicated dynamic process in mining engineering.

REFERENCES

1 Zu Yiwen *et al*. Microcomputer Finite Element Pre-post Process System Vizicad and Its Application, (in Chinese). Beijing: Scientific Technology Bibliology Press, 1993: 1 - 10.

2 Prat P C. DRAC: Rock Mechanic Analysis Computer Software, Xi′an. 1993.

3 Cai Yuejun. PhD thesis, (in Chinese). Changsha: Central South University of Technology, 1990.

4 Zhang R Q and Dong M. Structral Analysis Programming, (in Chinese). Chongqing: Chongqing Press, 1988.

5 Brady B H G and Brown E T. Rock Mechanics for Underground Mining. London: Geoge Allen & Unrrin Ltd, 1985.

6 Feng Z L. Rock and Soil Engineering, 1991, 3(13): 38 - 40.

7 Christian J T and Wong I H. Soils Foundation, 1972, 13: 1 - 10.

8 Yang G T. Mechanics of Elastoplastics, (in Chinese). Beijing: People Education Press, 1981.

(Edited by He Xuefeng)