Microcomputer-Based Post-Processor for Large Finite Element Analysis

대규모 유한요소해석에 활용되는 소형컴퓨터용 후처리 그래픽 프로그램

이 성 우*
Lee, Sung Woo
이 선 구**
Lee, Sun Goo
이 태 연***
Lee, Tae Yeon

요 약

최근까지만해도 유한요소 모델의 그래픽 후처리는 주로 대형컴퓨터와 이에 수반되는 고가의 도화장비에 의존할 수 밖에 없었다. 그러나 우수한 그래픽 기능을 갖춘 저렴한 소형컴퓨터의 등장에 힘입어 대단히 경제적인 후처리 그래픽 프로그램인 MICRO-POST가 개발되었다. 이 프로그램을 이용하여 무제한 요소망으로 이루어진 대형 유한요소 모델의 기하형태나 해석결과를 각종 저가의 도화장비에 호환하여 도화처리할 수 있게되었다. 본 논문에는 도화장비의 호환을 위한 프로그램 절차와 프로그램의 구성 및 그 기능들을 기술하였다. 또한 소형컴퓨터의 메모리 용량 극복을 위한 효율적인 I/O 기법과 도화처리를 교호작용하면서 관리할 수 있는 대화형식의 입력방식도 기술하였다. 범용 유한요소 해석 프로그램에 접속시킨 이용사례를 통하여 프로그램의 효용성을 입증하였다.

Abstract

Until recently post-processing of finite element model has been heavily relied on expensive graphic peripheral devices. With the aid of inexpensive microcomputers, very economical post-processor graphics program called MICRO-POST has been developed. Model geometry or results of analysis for the unlimited meshes can be easily presented in a number of low-cost graphic devices. The paper presents the procedure obtaining the device-independent graphics, and the structure and functions of the program. It also describes efficient I/O scheme to overcome the memory limitation, and dialogue-type input technique to control the plot operation in an interactive manner. Through the post processing examples for the general purpose finite element programs, it demonstrates the usefulness of the program.

* 정회원, 국민대학교 토목공학과 조교수
** 정회원, 대림엔지니어링(주) 정보사업그룹 전산기술연구개발담당
*** 정회원, 국민대학교 대학원 토목공학과 석사과정

□이 논문에 대한 토론을 1990년 6월 30일까지 본학회에 보내주시면 1990년 12월호에 그 결과를 게재하겠습니다.

INTRODUCTION

Today it is general to utilize computer graphics in performing post-processing the finite element analysis, thereby greatly reduces time and effort in checking the model geometry, or in interpreting the results of analysis through the comprehensive graphical form instead of vast amount of printed output. Until recently the graphic post-processing was done on the mainframe and its expensive peripheral graphic devices. Device-dependent graphic softwares[1-3] used for this purpose were mainly developed to be subordinate to the general purpose finite element packages and usually were costly. For this reason many engineers, researchers, or students in the university could hardly access to the graphics programs. However, apearance of low-cost microcomputers and graphic devices having excellent graphic capability, enables us to develope very economical device-independent post-processor program. If such program is used, all the graphic post-processing can be done on the microcomputer whether the analysis is performed on the mainframe or microcomputer itself. If the problem is analyzed in the mainframe the necessary plot data can be transferred to microcomputer by the appropriate data transmission software.[4-5]

In this study the program called MICRO-POST[6] which can perform the graphical post-processing on the microcomputer for the 2 or 3-dimensional finite element model of unlimited meshes, is developed. The paper presents the structure and functions of the program. Some examples for the large finite element meshes produced by the general purpose finite element programs are also included in the paper.

POST-PROCESSING ON MICROCOMPUTER

1. Plot Files for Post-processing

In any type of finite element program it generally requires geometry data and attribute data such

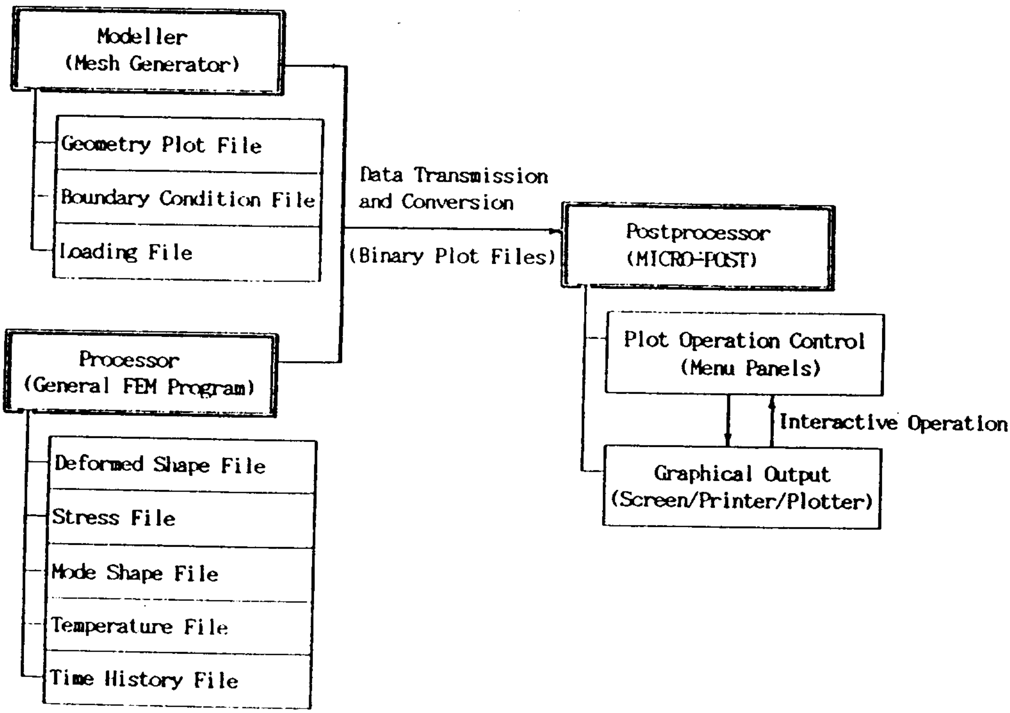
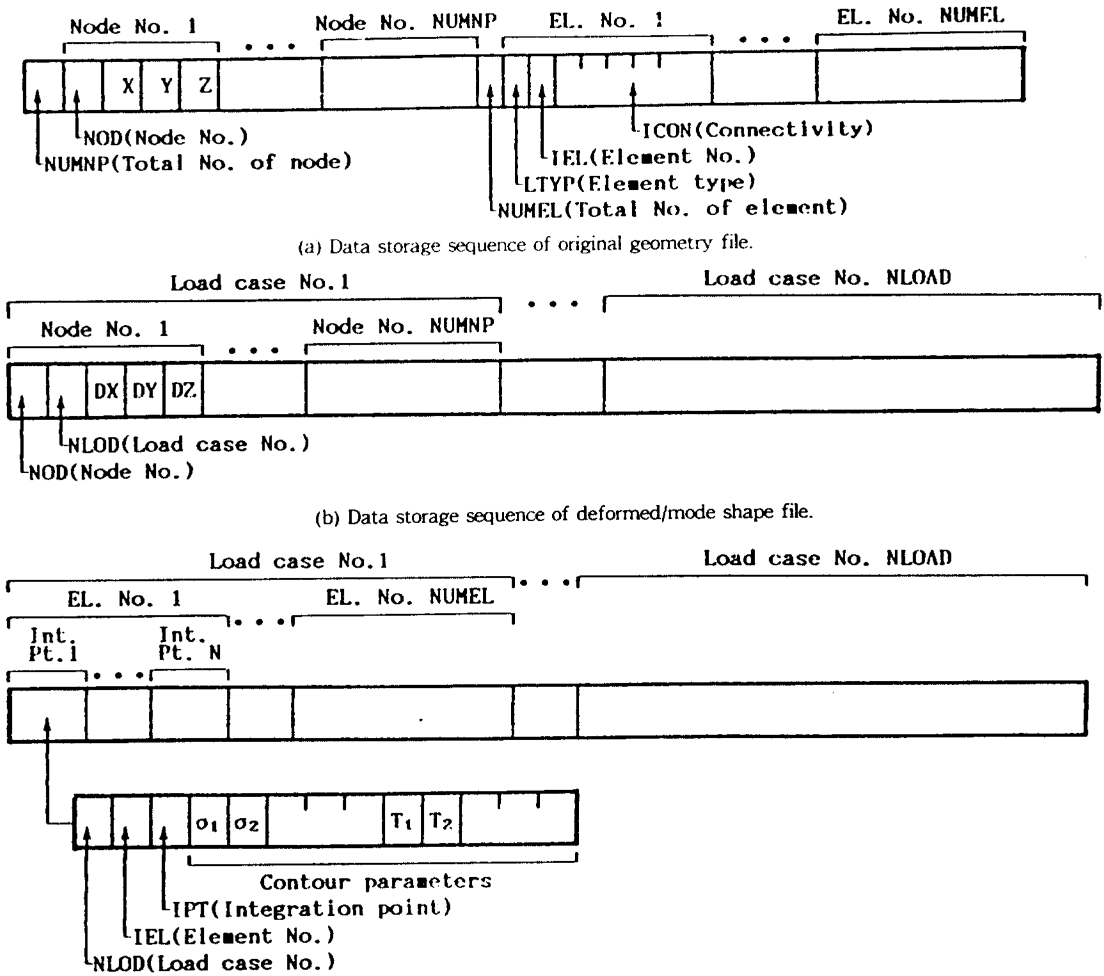


Fig. 1 Post-processing procedure in the finite element analysis.



(a) Data storage sequence of original geometry file.

(b) Data storage sequence of deformed/mode shape file.

(c) Data storage sequence of contour information file.

Fig. 2 File structure of MICRO－POST acceptable plot files.

as boundary conditions, property data or loading conditions prior to analysis. These are usually created by modeller, and the correctness of data is checked by graphics before the analysis. Then the analysis is performed with this confirmed data through the processor, and the results such as deformed geometry, stresses or other parameters from the various load cases are obtained. The files containing these informations are the plot files for graphic post-processing. The plot files generated from modeller and processor are shown in Fig. 1. As shown in Fig. 1, these files are transmitted, if analysis is done on the mainframe, and converted to binary form acceptable to post-processor program MICRO－POST. During the process the plot operation is interactively controlled by dialogue-type menu panels, and the user can obtain the graphics in the display, plotter or printer at his or her own will.

To convert plot data to MICRO-POST acceptable binary input form, a simple data conversion program called CONVBIN is used. The conversion from ASCII decimal data to binary data is necessary to speed up I/O, thereby to accelerate the graphic process. Three files generated through the process are : 1) original geometry file with '.BPL' extension ; 2) deformed shape or mode shape file with '.BDE' extension ; 3) contour information file containing the stresses or other desired parameters with extension '.BST'. The plot information stored in these files are shown in Fig. 2. If user provide these binary plot files with the same sequence as shown in Fig. 2., MICRO–POST can be used as a post–proccesor for any type of finite element program created or owned by user. This provides a great flexibility to the program in interfacing with other programs. Thus even the student who is working on the class project in the finite element course can utilize MICRO–POST in conjuction with his or her own analysis program without getting involved in the graphic programming. Currently CONVBIN program interfaces with SAP6[7], SAP7[8], COSMOS7[9]and CFEP[10] finite element programs. The plot files generated by these programs can be converted by CONVBIN to MICRO-POST acceptable binary files without any modifications.

2. Device - independent Graphic Program

The graphic programs used in the mainframe system are generally device–dependent and this gives user a great trouble when his or her organization does not have the supporting graphic devices. Eliminating this inconvenience, device – independency of the graphic programs is obtained owing to graphic standardization. To perform graphics independently at different devices the MICRO-POST program uses high-level interface graphic libraries[11-12] which are developed using the CGI (Computer Graphics Interface) software[13], one of standardized graphics programs. These interface libraries are the programs which are developed to facilitate application programmer in developing the menu-driven input system, and 2 or 3-dimensional application graphic programs. In Fig. 3 device-independent graphic system implemented in the MICRO-POST program is shown. As shown in Fig. 3, since CGI treats the various type of displays, plotters and printers as a single virtual device, the program operates independently in a number of different graphic devices.

3. Structure of the Post–Processor Program

A. Graphic Control Procedure

Control flow of the MICRO–POST program is shown in Fig. 4. As can be seen in Fig. 4, graphic operation is controlled by the 3 menu panels. The first panel contains the input fields related to the overall graphic operations, and the next panel is for the geometry or for the contour plot. Since these panels controlled by the program execution commands, the movement from following panel to previous one or vice versa, or termination of program execution can be easily done during the operation. Once the graphic mode is determined through this control process, the corresponding picture will be displayed on the screen. At this mo-

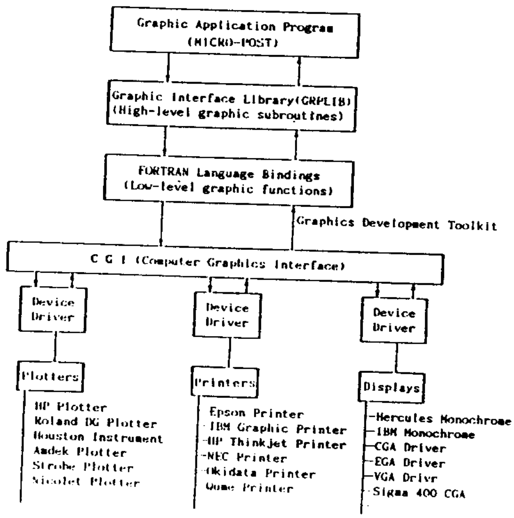


Fig. 3 Device-independent graphic system implemented in MICRO-POST.

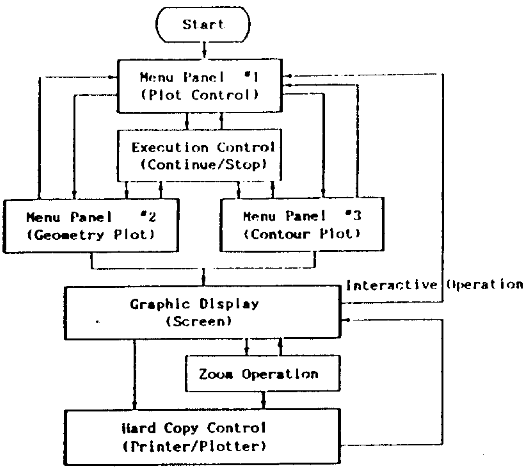


Fig. 4 Control flow of MICRO-POST program.

ment if the displayed picture is not the desired one, the user can interact with computer by going back again to the control panels and modifying the plot selections. If the magnified view of certain portion of the displayed picture is desired, it can be obtained using the zoom function of the program. The selection of zooming area is also interactively done on the picture by the direction keys. This operation can be repeatedly performed until desired maginification of the portion is obtained. The picture displayed on the screen, either maginified or not, can be directed to the plotter or printer to obtain the hardcopy of good quality. Once the hardcopy operation is done, the same picture will be displayed on the screen and it allows the user again to do any of interactive operations mentioned above.

B. Overcoming the Memory Limitation

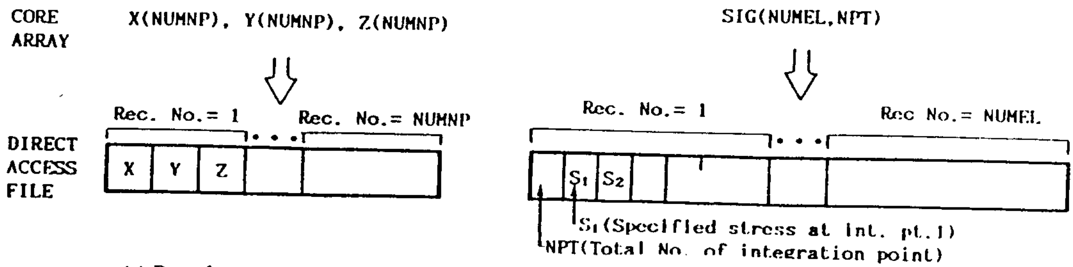
In the process of graphic operations, large amount of data manipulations are required. If the mainframe is used in this operation, many of them are done in the memory core using arrays of large dimension. However if microcomputer is mobilized for this purpose, the memory limitation must be overcome through efficient disk I/O. In the MICRO-POST program direct access files are extensively used to replace the core arrays, thus the finite element model of unlimited meshes can be handled without any difficulty.

Fig. 5(a) shows how the arrays of large dimension for the nodal coordinates are replaced by the disk files. The similar scheme is applied to the operation in obtaining static deformed shape or dynamic mode shape. Arrays required for the transformation of coordinate systems, which is inevitable in the process of projections of 3-dimensional model having various viewing parameters onto the 2-dimensional display, are also replaced by disk I/O.

In Fig. 5(b) it shows the scheme applied to the contouring operation. The file, shown in Fig. 5(b), containing the contour parameters such as stresses of the specified component for the given load case, is created with retrieved data from the binary contour information file, mentioned in the previous section. Smoothed contour parameters at nodal points are then stored for contouring. In this way contour map for any component of parameters for any load case can be obtained unless the disk storage space is full.

C. Speeding up the Graphic Processing

To speed up the graphic processing several techniques are implemented in the program. Conversion of formatted decimal plot files to binary files, as mentioned earlier, takes a share in the speed up procedure. Another acceleration of procedure is corresponding disk I/O from the second run of plotting session, if the viewing parameters are not changed. This situation is occuring when the following operations are involved : 1) zooming the portion of the picture ; 2) obtaining the superposed static deformed or dynamic mode shape with different scaling factor for exaggerated view : 3) obtaining the contour map for the different component of parameters for any specified load case



(a) Data for nodal coordinates

(b) Data for contour parameters

Fig. 5 Replacement of core arrays with direct access disk files.

; 4) reducing the picture by scaling down the viewport ; 5) obtaining the hardcopy of the picture.

Moreover, if the memory is expanded and this part of memory is used as a virtual disk, the speed of plot operation on this virtual disk is remarkably fast-even faster than on many graphic devices connected to the mainframe. This is truly exciting fact and the value of inexpensive microcomputer is highly rated.

4. Functions of the Program

The funcions of the program can be classified in two categories ; the functions related to the geometry plot and the ones for the contour map. The picture for the original geometry, static deformed geometry or dynamic mode shape can be obtained by the former functions. Using the latter functions the contour map for the stresses, member forces, moments or other desired parameters can be obtained.

As mentioned earlier the control of plot operation is done by the three menu panels and they are shown in the Figs. 6 to 8. In these menu panels, the input parameters and the various plot functions corresponding to them are self-explanatory. Since these manu panels consist of dialogue-type input fields, the user can input the required data in a highly interactive manner. At the beginning, the default values are automatically displayed in the input fields. These values can be modified if they

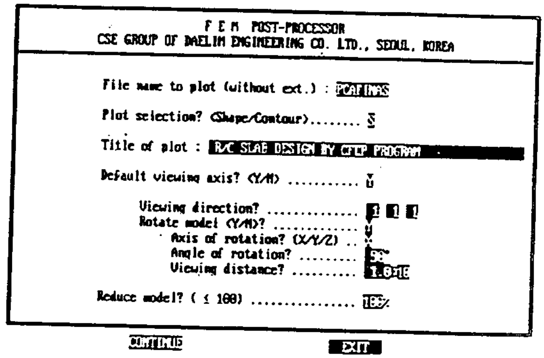


Fig. 6 Menu panel for plot control

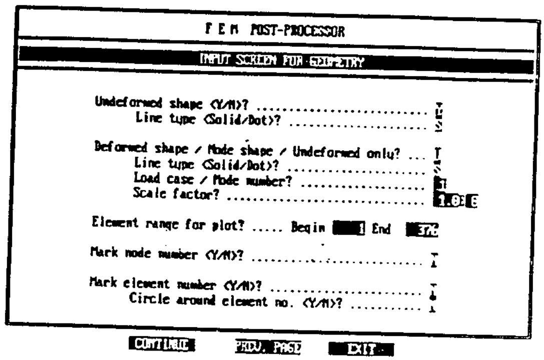


Fig. 7 Menu panel for geometry plot

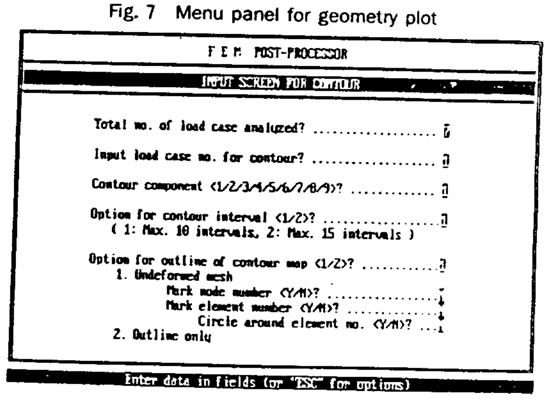


Fig. 8 Menu panel for contour plot

are not the desired ones, and the changed values will be displayed at the next run. Input can be done from the top by typing in the appropriate values. However, if the user does not want to change the displayed input and wants to skip the following input fields, he or she can directly access to the fields of execution command, which is shown at the bottom part of the Figs. 6 to 8. At this stage the user can continue the process or, if necessary, can go back to the previous panel for the desired modifications.

The advantages of dialogue-type input are many and some of them are : 1) owing to the self-explantory input descriptions, the user does not need to discuss with manual and can interact immediately with computer as if he or she is talking with computer, 2) since the input selections are controlled by the program internally, it is possible for the user to take only acceptable characters or numbers within the permissible range, thereby input errors can be greatly reduced.

EXAMPLES FOR THE GENERAL PURPOSE FINITE ELEMENT PROGRAMS

1. Static Anaysis of 30-story Frame

In Fig. 9, the deformed and undeformed shape of three dimensional 30-story frame, analyzed by CFEP program[10], is shown. This picture is obtaind by the low-cost plotter. Of course this model can be plotted with different viewing directions, or with other desirable reduced scale. Original model and deformed model can be plotted seperately with different line type and color. The exaggeration of displaced geometry can also be adjusted at will. By specifying the range of elements to be plotted, user is able to obtain only the selected portion of the picture. In addition, the node numbers or element numbers can be written on the model. The magnified view of the portion of the model can be also obtained using the zooming function.

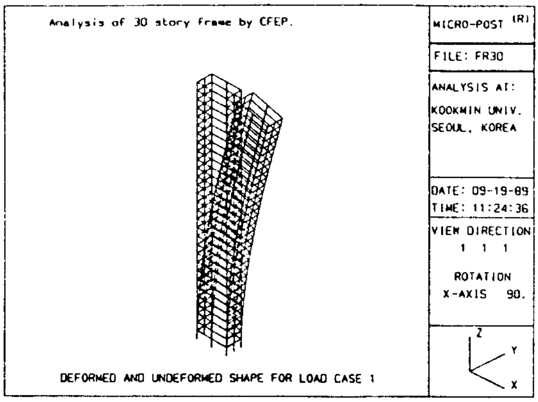
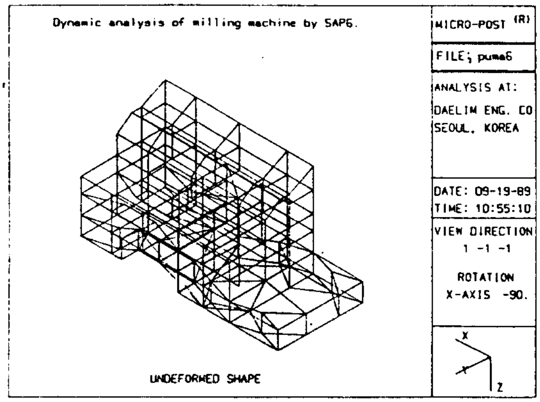
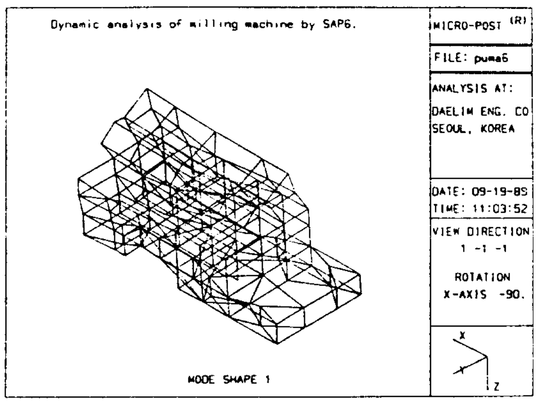


Fig. 9 Deformed and undeformed shape of 30-story frame.



(a)



(b)

Fig. 10 Original geometry and first mode shape of milling machine.

2. Free Vibration of Milling Machine

The first vibration mode and original geometry of milling machine, analyzed by SAP6 program[7], is shown in Fig. 10. The other mode shape can also be plotted with option of either mode shape alone or mode shape superposed onto the original geometry. All the functions described in the previous section can be exploited in this model too.

3. Design of Reinforced Concrete Slab

In Fig. 11, the contour map of the moment in X-direction for the reinforced concrete flat plate model, analyzed and designed by CFEP program, is shown. This model can be plotted with different viewing directions as in the previous case. Besides, the contour for the moment in Y-direction can also be obtained for the various load cases. In addition, the contour map for stresses, forces, other parameters for the different load cases, or designed steel areas can be obtained. The interval of the contour can be adjusted if desired. The option whether to draw the outline only or to superpose undeformed meshes on the contour plot is also available. When the user take the latter choice, the node numbers or element numbers can be written on the plot. Of course, this numbers can be different colors to distinguish with the contour numbers. This option is very useful when the portion of high stress concentration is magnified and to see the related elements or nodes. The Fig. 12 is the magnified portion of the picture outlined in Fig. 11.

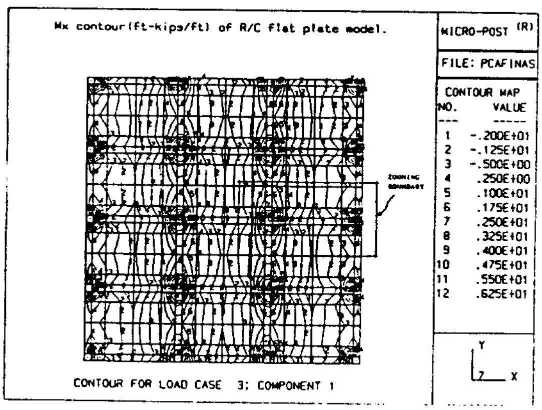


Fig. 11 Mx contour of R/C flat plate model.

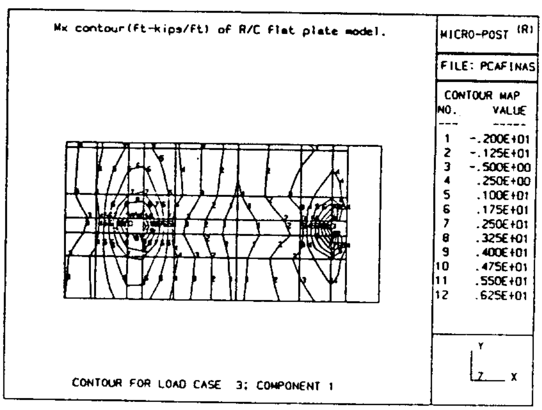


Fig. 12 Magnified portion of the picture outlined in Fig. 11.

CONCLUSIONS

To replace expensive graphic post-processing of large finite element model on the mainframe system, an economical microcomputer-based post-processor program called MICRO-POST has been developed.

By using the efficient I/O scheme to overcome the memory limitation, it is able to handle the model of unlimited meshes.

Error-preventive dialogue-type input scheme allows the user to control the plot operation in a highly interactive manner.

Due to the device-independent program structure, the graphic output is obtained in a number of low-cost graphic devices.

Through the post-processing examples for the general purpose finite element programs, it demonstrates the usefulness of the program.

ACKNOWLEDGMENTS

This study was suponsored partly by the Korea Science and Engineering Foundation through Grant No. 893－1301－004－1. This support is greatly appreciated. The authors also like to thank Daelim

Engineering Co., Ltd. for providing the computing facilities and example materials.

REFERENCES

1. Gelman A. P., POST – A Post–processor Graphics Program to Accompany the COSMOS Finite Element Programs, SRAC, Santamonica, California, (1984).
2. ADINA Engineering, Inc., ADINA – PLOT(A Post - processor for ADINA Program), Watertown, Massachusetts,(1986).
3. TDA Engineering, PATRAN(CAD/CAE Software System), Costamesa, California, (1987).
4. Persoft Inc., SMARTERM 240(Data Transmission Software), USA,(1986).
5. Samsung Data System Inc., VTPC(Data Transmission Software, Seoul,(1987).
6. Lee, S. W. and Daelim Engineering Co., Ltd., MICRO-POST – A Microcomputer–Based Interactive Post–processor Graphics Program for Finite Element Programs, Seoul, (1989).
7. SAP Users Group, SAP6 – A Structural Analysis Program for Static and Dynamic Analysis, University of Southern California, (1982).
8. SAP Users Group, SAP7 – Static and Dynamic Analysis Program for Linear and Nonlinear Systems, University of Southern California, (1982).
9. Structural Research and Analysis Corp., COSMOS 7 – Structural Analysis System, Santa Monica, California, (1987).
10. Lee, S. W., CFEP – Compact Finite Element Program, University of Southern California, California,(1987).
11. Lee, S. W., GRPLIB – A FORTRAN Graphic Interface Library, Kook Min University, Seoul, (1989).
12. Lee, S. W., SCRLIB – A FORTRAN Screen Management Library, Kook Min University, Seoul, (1989).
13. Graphic Software System, GSS Graphic Development Toolkit, Wilsonville, Oregon, (1985).

(접수일자 : 1989. 10. 4)