FINITE ELEMENT MESH GENERATION
EMPLOYING SATELLITE GRAPHICS

Sze-Keung Lee
Research Assistant and Recent Graduate
The University of Texas at Austin
Austin, Texas

Walter S. Reed
Assistant Professor of Mechanical Engineering
The University of Texas at Austin
Austin, Texas

Abstract

The use of a satellite graphics system is presented as an approach to preprocessing finite element analysis data. The paper discusses the use of curved isoparametric shape functions as the basis of finite element mesh generation and the effects this approach has on the design of a system to support these activities. A minicomputer based design system is presented and its application to a practical three dimensional mesh generation problem is demonstrated.

Introduction

The use of the finite element method for the approximate solution of continuum field problems in mathematical physics has become widespread. While the primary use of the finite element method is in the area of structural mechanics, the technique is also being employed by analysts in other fields including heat transfer, dynamics, fluid mechanics and electromagnetic wave phenomena. The practical use of these discretization techniques for the solution of continuum problems was made possible by the development of large scale computer systems. Three dimensional finite element analyses in particular require the availability of a substantial computing facility.

While many general purpose finite element programs have been developed, several problems [1]* impede the use of these programs by design engineers. One of the most commonly recognized problems by analysts employing the finite element method is that the major cost in the design/analysis cycle comes during the data preparation and output evaluation phases. This cost is primarily a result of the lack of automation during these input/output phases.

The input phase involves the geometric description of the object or field to be analyzed as well as the discretization of this field into elements. This process is usually referred to as finite element mesh generation and is quite time consuming particularly in the case of a three dimensional problem.

The output phase usually involves the post processing of discrete nodal values into a graphical format which is easily interpreted by the designer. While many finite element programs have some form of post processing which results in hard copy graphic output, relatively little has been done in comparison to automate finite element input.

While the finite element analysis is in general best suited for conventional batch processing because of its size, the pre/post processing of data is well suited to an interactive system with a graphic interface. Some of the important reasons are as follows:

1. The amount of input data required for automated mesh generation is small. If the amount were large, the time to input the data in an on-line fashion would be long and very inefficient, so that the processing power of the interactive system would not be fully utilized. In this work, only a minimal set of information on the topology is needed, and most of the data can be input graphically by means of a digitizer.

2. Visual and graphical I/O is essential in mesh generation. Designers usually think and work in terms of a picture and since this process involves mainly the topological layout and geometric connectivities of elements, it is most direct and natural to input the required information graphically. In addition, a graphic feedback is also valuable in that the designer can visually compare the actual design with the computer generated mesh.

3. Only a moderate computational time is required because of the simplicity in the interpolation scheme required for mesh generation. Therefore, in an interactive environment, the designer can have a rapid visual feedback. Since the computation is relatively simple a local minicomputer can be employed for the mesh generation. The graphic minicomputer system acts as a satellite to a large computer system where the time consuming finite element analysis will be processed after the preprocessing is finished and the information is transferred.

4. The designer's interaction and intervention in the design process is essential. The automated mesh generation schemes depend on a set of topological data and some interpolation functions. Therefore the boundaries and the geometry of the generated mesh cannot be conceptually determined before generation and plotting. The designer needs an immediate visual feedback so as to compare and determine its correctness. Furthermore, an important characteristic of the automated mesh generation scheme is the uniformity of the generated mesh grid patterns. The designer may not want uniformaly shaped and placed elements in some area, or he may want to put holes in some other areas. The only way these can be efficiently done is by interactive editing. Therefore, on-line man-machine interaction is essential.

This paper discusses the use of a satellite graphics system for automated finite element mesh generation. This system employs the use of isoparametric shape functions to generate a mesh of elements from a minimal set of input data. In addition to a brief discussion of these shape functions, the hardware/software configuration will be specified as well as a discussion of a data base. Finally, several mesh generation examples will be provided.

Isoparametric Shape Functions

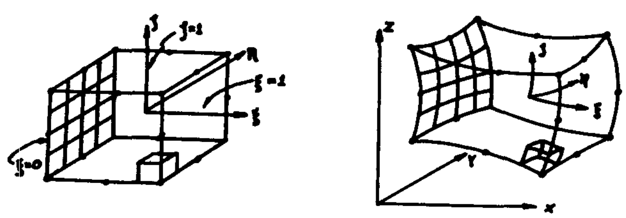
One of the motivating factors for the use of finite element discretization is its applicability to problems with highly irregular boundaries. Therefore one of the initial considerations in the choice of a mesh generation-algorithm is that it can be easily molded to any geometry. Most finite element analysis programs also require the simultaneous use of elements possessing a

*Numbers in brackets designate references at end of paper variety of shapes. Therefore some scheme must be available for specifying and mixing element types.

Many mesh generation schemes are available [2] with no algorithm seemingly best for all cases. The method employed in this work is a scheme described by Zienkiewicz [3] which utilizes curved isoparametric shape functions.

The method is applicable to either two or three dimensional mesh generation and allows an interactive user to choose either linear, quadric or cubic interpolation in an attempt to model a given field. The method may be thought of as the transformation of simple two or three dimensional shapes (such as a triangle or cube) into distorted forms of a more arbitrary shape.

For example a simple cube (Figure 1a) can be distorted into the curved hexahedron shown in Figure 1b. The simple cube is defined in a local (or natural) coordinate system ξ, η, ζ while the resultant transformation results in the required cartesian coordinates. In addition, any subdivision of the cube (in natural coordinates) into smaller cubes or other polyhedra will be mapped by the shape functions into curved subdivisions (or elements) in the cartesian coordinate system.



(1a) (1b)

FIGURE 1 Cube In Natural Coordinate System and Distorted Cube in Cartesian System

Therefore, if the designer can specify the general outline of the region (referred to in this paper as a zone) to be modeled with a select set of nodes on the edges of the zone, then a continuous model of any point on the surface or interior of the zone can be generated employing the shape functions. For example, if quadratic interpolation is to be used, then the x,y,z coordinates of the eight corner and twelve midedge nodes (collectively called key nodes) must be defined as well as the mesh grating (or subdivision of the cube in the natural coordinate system into elements). The shape functions are then used to generate the cartesian coordinates of each element.

A very important consideration in favor of the use of these shape functions results from the conceptual simplicity of the element associativity in the natural coordinate system. Since the elements are cubes or other simple polygons stacked together in a regular fashion, the designer interacting with the system is able to rapidly identify any element by noting its row, column and level numbers in the natural coordinate system. These values are simply the ξ, η, ζ coordinate values. This method of identification greatly facilitates interactive editing with a graphic interface.

A detailed discussion of isoparametric shape functions will not be provided here as it can be found in reference [3].

Design System Consideration

In order to specify the capabilities which must be present in a computer system to support mesh generation, a more detailed discussion of the input process is needed at this point. The necessary computing power, data and program storage capabilities and output capabilities must also be specified.

The general input requirements are determined by the use of the isoparametric shape functions. Each analysis problem is divided into a number of hexahedral zones (3-D) by the designer. Each zone is defined by specifying the coordinates of a set of key nodes. The number of key nodes required is dependent on the degree of the shape function used to approximate the zone. For example, in three dimensions if linear shape functions are used then eight nodes must be defined while for quadratic interpolation twenty nodes are required.

Other zone shapes may easily be created as special cases of the hexahedron. For example, a tetrahedron can be considered to be a hexahedron with several nodes merged together.

From the above discussion therefore it is seen that an automated mesh generation system must be quite flexible and should at a minimum have the following characteristics.

1. It must permit the input of the x,y, and z coordinates of each node defining each zone. This input is best accomplished graphically while other variables such as the degree of interpolation and mesh density are best defined by console input.

2. The computer system must have sufficient speed and power to generate the required mesh through repetitive interpolation.

3. It must have the ability to display zones individually and collectively. As described earlier, the isoparametric mesh generation scheme requires a division of the physical model into hexahedral (3-D) zones and each zone must be created individually. While the exact positions of interior nodes may not be extremely important, the nodes (generated and key) comprising the common boundaries of adjacent zones must coincide with each other. The position of these boundary nodes depend on the positions of the key nodes and the degree of interpolation. Therefore, the exact positions of mesh nodes cannot be conceptually determined before they are actually generated by the computer. It is very important that adjacent zones be displayed collectively to determine the accuracy of the common boundaries. Also, recreation or modification of created zones should be available, so that incorrect boundaries can be easily adjusted.

4. An element editor is essential. Since the mesh nodes are generated by interpolation functions from the key nodes, uniformity and rigidity are unavoidable. However, in most instances, these are not desirable. The designer may want irregularly shaped or placed elements in some area within the zone or may want to put holes in some area. Therefore, in order to have a usable mesh an element editor is necessary. An element editor should at least have the following capabilities:

   a. relocate mesh nodes lying on the boundaries or inside a particular zone.

   b. address any elements within a zone

c. replace any elements within a zone with new elements

d. delete any elements within a zone

e. add new elements to a zone

f. provide some kind of memory management

Graphical editing devices such as a cursor or light pen can drastically enhance the editing process and should be used whenever the situation permits. Mesh nodes can be more easily addressed through the use of a cursor or a light pen than by setting up an addressing scheme from the console. Similarly, element addition can be easily achieved by positioning a cursor or a light pen at the physical location of the corner nodes of the element on the screen. Liberal uses of these devices not only can reduce many man-made errors, but also can greatly speed up the design process.

5. Both an internal and external data base should be provided. An internal data base is needed to provide a vehicle for continuing design activities over several design sessions. An external data base is required to reformat the data into a simple structure compatible with existing finite element analysis programs.

Several other points should also be mentioned which are not a function of the mesh generation scheme but instead are general programming considerations which must be accounted for in the design of the system.

Because of the memory constraint of a minicomputer graphics system, most generated data must be disk resident during execution. This will have a negative effect on the response time due to the data transfer overhead time. To reduce the transfer time and thus improve the response, data blocks should be structured as large as possible and stored on contiguous sectors of a disk.

Automated mesh generation can be used as a truly creative design tool only if the user can interactively control the logical flow of the program. A set of simple commands will facilitate user control. Since not every sequence of commands is meaningful, error checking on non-meaningful command sequences with error messages are needed. Extensive error checking on input data should also be employed.

The design system software should be modularized. This feature will increase machine independence, increase program modifiability and facilitate hierarchical software structure design. For example, a modular structure facilitates the overlay structure necessary to make the mesh generator run on a minicomputer. Also, other graphics routines or interpolation schemes could replace existing routines with little effort. Expandability is also provided for by this approach.

Hardware/Software Organization

The hardware/software system which was developed to test the feasibility of automated mesh generation is shown in Figure 2. The hardware consists of a satellite graphics system and a time shared host computer.

The satellite graphics system employs a Digital Equipment Corporation DEC 11/40 with a Tektronix 4010 terminal as an output device. In addition, an 8K IMLAC PDS-1 graphic minicomputer is linked to the 11/40 via a parallel interface. A console and digitizer form the input devices to the 11/40. The 11/40 is configured with 16K of memory, floating point hardware and a disk. The graphics system is linked to a CDC 6600 which provides the processing power for the finite element analysis.

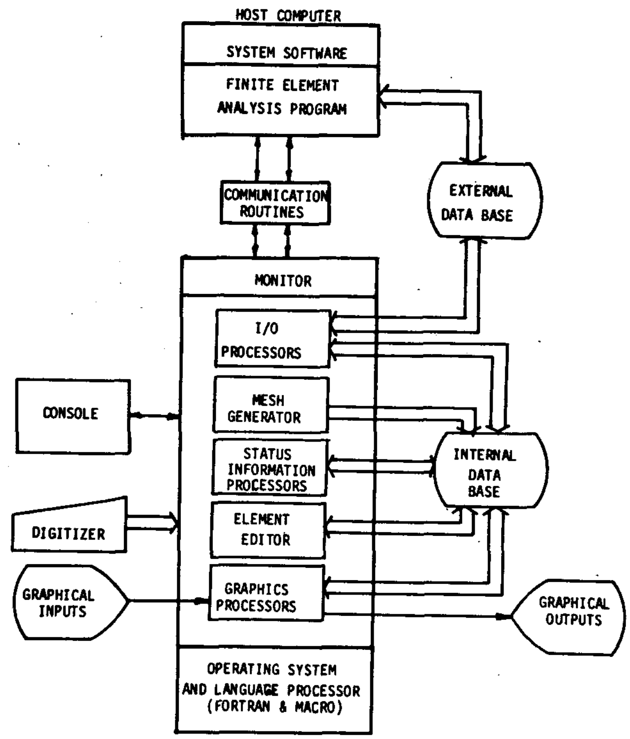


FIGURE 2 Satellite Graphics System

The software consists of five classes of processors executing under the supervision of a monitor. The monitor permits the user to enter design oriented commands and thereby control the execution of program modules.

The program modules are usually disk resident and accessed by the monitor as a function of designer activity. The five processors have the following functions.

1. The I/O processors accept console and digitizer input of zone, node and mesh information. Also these processors generate the display instructions for the IMLAC and Tektronix display devices.

2. The mesh generator provides the x,y,z values of each element from the key node input data and mesh grating description.

3. The status information processors provide the user with a record of the current mesh generation status. For example, how many zones have been generated and edited, etc.

4. The element editor provides a general capability for adding, deleting or modifying element descriptions.

5. The graphics processors provide a general three dimensional transformation and clipping capability [4]. Available transformations include rotation, translation and scaling. Also highlighting and individual display of elements and/or zones is provided.

Data Structure

The selection of a data base to support automated mesh generation and serve as input to a finite element analysis program is one of the more important design considerations. The major criteria for a data base impose somewhat contrasting constraints on the selection process. Among the most important considerations are:

1. The data structure must be general enough to hold zone and element information and also show relationships between various zones and elements within a zone.

2. The data structure must be well suited to being modified. Elements must be easily added or deleted and nodes and nodal connectivity must also be easily modified.

3. The data structure must serve as a source of information from which to create pictures using computer graphics. A display file compiler [4] must scan the data base and easily generate a user requested view of any zones and elements within the zones.

4. The data base must serve as an input for finite element analysis programs.

5. Since the data base could undergo a large number of element additions and deletions, the data base must permit some kind of memory management.

The overall structure of the data base is shown in Figure 3a. A zone directory and information block is established for each zone as it is created. These zone blocks form a linked list as shown in Figure 3a and each zone block also points to the edge description block for that zone and the level 1 element block for that particular zone.

The edge description block for each zone is a detailed set of x,y,z, coordinate values which describe the twelve edges of the zone. This information can be used by a display file compiler to rapidly generate a view of any set of zones without elements.

A detailed description of the linked element blocks for each zone is shown in Figure 3b. The element blocks are subdivided into levels which relate to a physical layer or strata of elements within each zone. Each element block contains the following information:

1. A pointer to the element block in the next lower level. In the case of a thin shell there may only be a single level of elements within a zone.

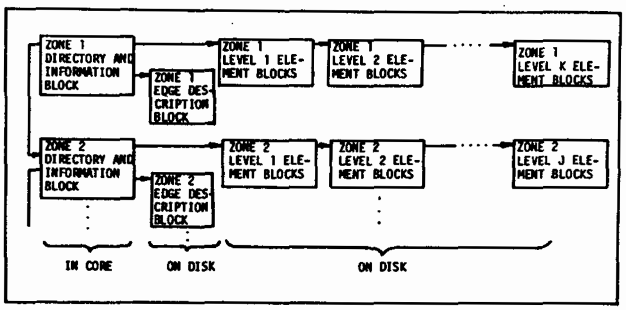
2. An element ID number.

3. The x, y, z coordinates of the eight nodes of the element. Several of the nodes could have the same coordinate values indicating that the element is a lower order polyhedron.

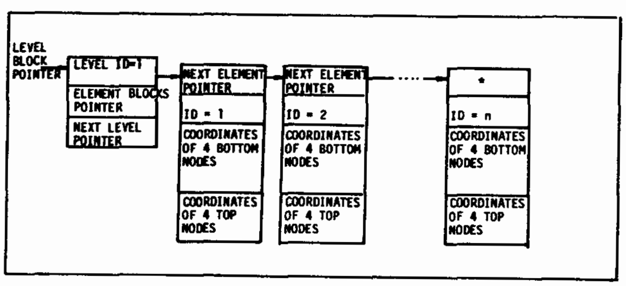
This data structure has evolved as a compromise to all of the criteria presented at the beginning of this section.

A Mesh Generation Example

As an example of the mesh generation capabilities of this system, a journal bearing housing (Figure 4) was divided into five zones and digitized. Since the housing is a three dimensional structure, two orthogonal views were used to input the X-Y and the X-Z coordinates of the twenty key nodes in each zone. A mesh grating was then specified and the projection of the entire set of generated elements with no transformation is shown in Figure 5 (each of these example Figures was produced by a Tektronix 4010 hard copy unit).



(3a)



(3b)

FIGURE 3 The Basic Data Structures and the Level 1 Element Blocks

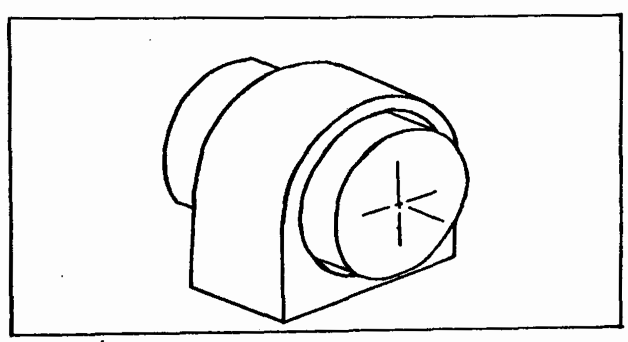


FIGURE 4 Journal Bearing with Housing

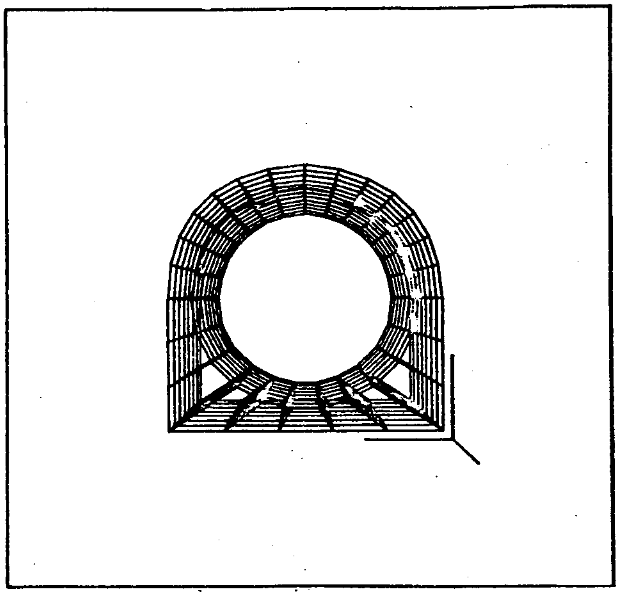


FIGURE 5 Three Dimensional Model of the Housing

The same information is shown in Figure 6 following a rotation, translation and scaling of the housing. This display also illustrates the automatic clipping against a viewing pyramid in three dimensions. The use of the graphic processors to show only the element information in two zones is demonstrated in Figure 7.

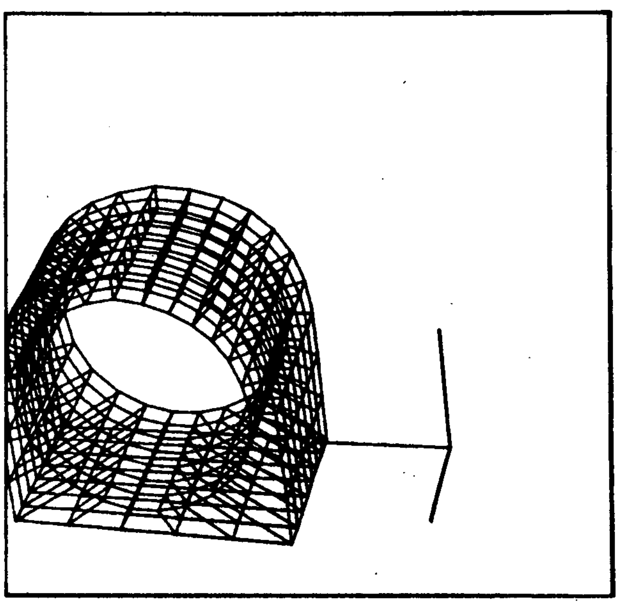


FIGURE 6 Transformed and Clipped Housing

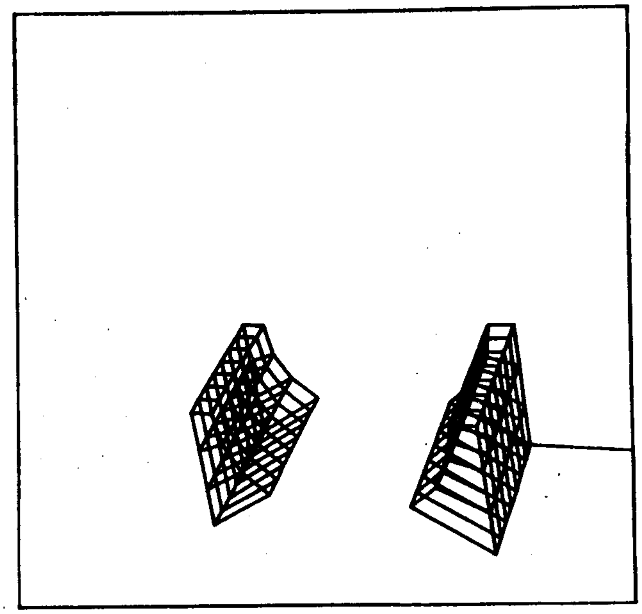


FIGURE 7 Individual Zones and Elements

Figure 8 shows an edge description of three selected zones while Figures 9 and 10 show the same zones after some transformation with elements added. It is easy to see that the display shown in Figure 10 provides the necessary detail for element checking and editing.

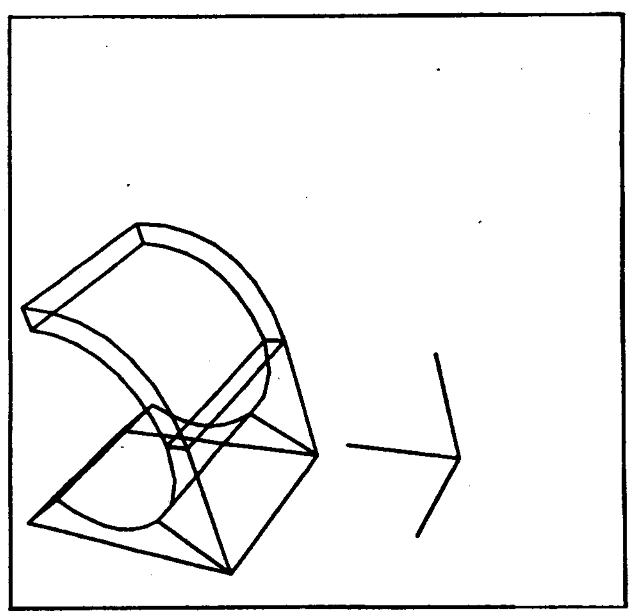


FIGURE 8 Selected Zone Display

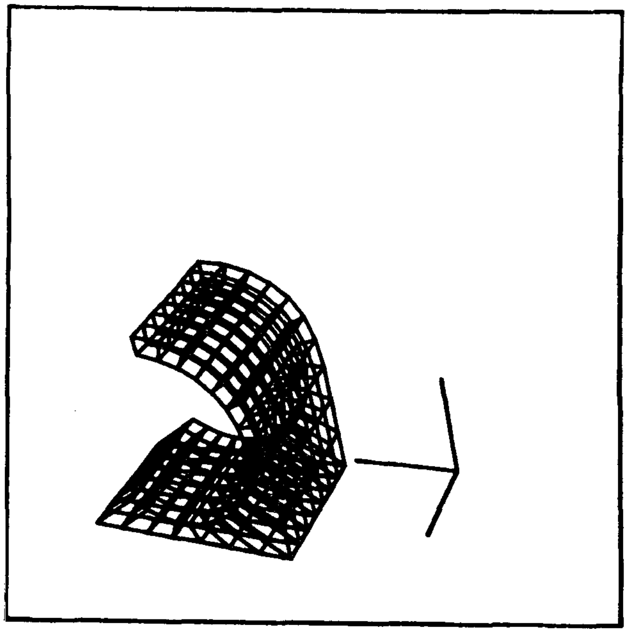


FIGURE 9 Zone and Mesh Display

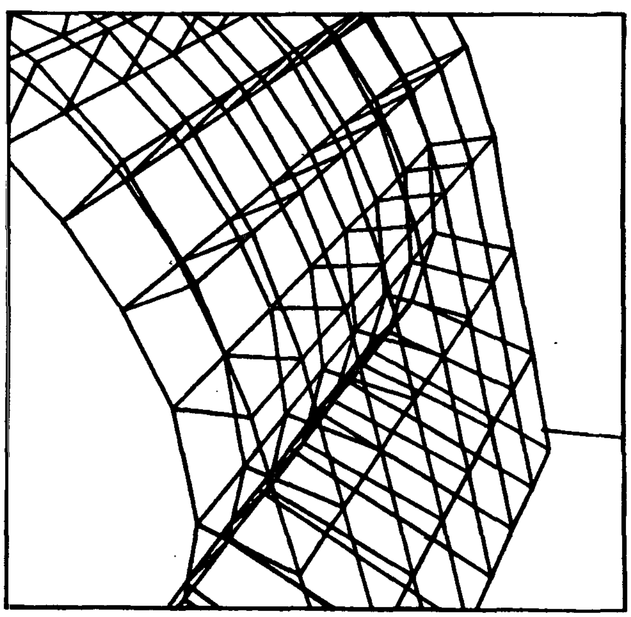


FIGURE 10 Enlarged View of Selected Zones

Conclusion

A finite element mesh generator applicable in both two and three dimensions was presented and its use demonstrated on a practical problem. The use of curved isoparametric shape functions was presented as the basis of the mesh generation scheme. The considerations which played an important role in the design of the mesh generation system were also presented in addition to the hardware/software system employing satellite graphics. A data base was also discussed which evolved during this work.

This research project has demonstrated that practical three dimensional mesh generation can be done on a small interactive computer. Further work is progressing on linking the preprocessor, postprocessor and finite element analysis programs together in a manner which is both efficient in terms of computer resources and human resources.

References

1. Pilkey, W., Schaeffer, H., and Saczalski, K., Structural Mechanics Computer Programs: Surveys, Assessments and Availability, University of Virginia Press, Charlottesville, Va., 1974.

2. Buell, W. R. and Bush, B. A., "Mesh Generation - A Survey," ASME paper 72-WA/DE-2.

3. Zienkiewicz, O. C. and Phillips, C. V., "An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Coordinates," International Journal of Numerical Methods, Eng. 3, 1971.

4. Newman, W. M. and Sproul, R. F., Principles of Interactive Computer Graphics, McGraw Hill, S. F., California 1973.