

MODELING OF SOLIDS FOR THREE-DIMENSIONAL
FINITE ELEMENT ANALYSIS

by

Bruce Eric Brown

A dissertation submitted to the faculty of the
University of Utah in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Computer Science

University of Utah

June 1977

PROPERTY OF COMPUTER SCIENCE
LIBRARY 3147 MEB
UNIVERSITY OF UTAH

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a dissertation submitted by

Bruce Eric Brown

I have read this dissertation and have found it to be of satisfactory quality for a doctoral degree.

June 8, 1977
Date

Mart E. Newell
Martin E. Newell
Chairman, Supervisory Committee

I have read this dissertation and have found it to be of satisfactory quality for a doctoral degree.

June 8, 1977
Date

Henry N. Christiansen
Henry N. Christiansen
Member, Supervisory Committee

I have read this dissertation and have found it to be of satisfactory quality for a doctoral degree.

8 June 1977
Date

Richard F. Riesenfeld
Richard F. Riesenfeld
Member, Supervisory Committee

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of The University of Utah:

I have read the dissertation of Bruce Eric Brown in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to the Graduate School.

June 8, 1977
Date

Martin E. Newell
Martin E. Newell
Member, Supervisory Committee

Approved for the Major Department

Zvi Kobavi
Zvi Kobavi
Acting Chairman/Dean

Approved for the Graduate Council

Sterling M. McMurrin
Sterling M. McMurrin
Dean of The Graduate School

ABSTRACT

The geometric modeling of solid objects is a major problem within the design analysis loop of the engineering design process. Models are analyzed by various computer programs to predict their performance. The format of each model is usually different for each analysis routine. The existence of several versions of a model, each of which may be independently modified, results in severe inconsistencies. The solution to this problem is to model the object at a higher level of abstraction. All other models are derived from the one high level model and all modifications are made to this model. The proposed form of the high level model is an extension of the parametric representation used for work with curves and surfaces. Representations of bivariate forms are extended to schemes for trivariate forms.

The analysis of interest to this work has been the finite element method. The formulation of the finite element method was investigated to provide a geometric criterion for evaluation of the model for analysis. A new technique has been developed for deriving improved analysis models from the trivariate representation. Computer programs have been implemented to demonstrate these ideas and four examples are included.

To my Father and Grandfather.

ACKNOWLEDGMENTS

I would first like to thank Dr. Martin Newell for the many hours he has spent with me working on this dissertation. Dr. Henry Christiansen has been a great influence in my life and I would like to thank him for the many years of friendship. The ideas for this research came from my work with him. Dr. Richard Riesenfeld provided me with the tools by teaching me about curves and surface representations and I thank him for the time he has spent. To my fellow students at the University I would like to say thanks for the time spent and the friendships. In particular I thank Russell Athay for always asking challenging questions.

I am grateful to the Department of Computer Science and Mr. Stewart Ogden for the use of the PDP-11/45 computer and the graphics hardware, for this research. Thank you also to the Department's Software Research Laboratory for the use of the Burroughs B1726 computer and the programs for document preparation. Without a good text editor and the Publisher program, I am sure that this work would have taken months longer to accomplish.

Support for this research has come from many sources. I would like to thank the University Research Committee for the Research Fellowship that has allowed me to work full time on this work for the last year. Portions of the research were accomplished during the summers while I was employed by the University of California, Lawrence Livermore Laboratory and Los Alamos Scientific Laboratories and the

Department of the Navy, Civil Engineering Laboratory. I thank the people with whom I spent those summers for having so many interesting problems. In particular I would like to thank Drs. Gerald Goudreau and William Cook.

The largest thank you must go to my family and my wife, Patti, who have always been behind me, helping and pushing for me to finish this work, thank you.

TABLE OF CONTENTS

| | |
|---|----|
| ABSTRACT..... | iv |
| ACKNOWLEDGMENTS..... | vi |
| LIST OF ILLUSTRATIONS..... | x |
| CHAPTER 1 -- INTRODUCTION..... | 1 |
| Overview of the Engineering Design Process..... | 2 |
| Present Process..... | 4 |
| Design Analysis Loop..... | 6 |
| Summary..... | 14 |
| CHAPTER 2 -- PROPOSED SYSTEM..... | 15 |
| Introduction..... | 15 |
| The High Level Model..... | 17 |
| CHAPTER 3 -- MATHEMATICAL FORMULATIONS..... | 21 |
| Introduction..... | 21 |
| Space Curves..... | 21 |
| Surfaces..... | 23 |
| Volumes..... | 29 |
| Higher Order Expressions..... | 35 |
| CHAPTER 4 -- FINIE ELEMENT METHOD..... | 37 |
| Introduction..... | 37 |
| Formulation..... | 37 |
| Calculating Element Stiffnesses..... | 41 |
| Optimum Geometry..... | 42 |
| Geometric Merit..... | 44 |
| CHAPTER 5 -- IMPLEMENTING THE SYSTEM..... | 50 |
| Introduction..... | 50 |
| High Level Model..... | 50 |
| Generating Nodal Coordinates..... | 53 |
| Generating Elements..... | 57 |

| | |
|---|-----|
| Checking for Geometric Merit and Improvement..... | 59 |
| CHAPTER 6 -- RESULTS AND CONCLUSIONS..... | 65 |
| Introduction..... | 65 |
| Two-Dimensional Examples..... | 65 |
| Three-Dimensional Examples..... | 81 |
| Conclusions..... | 92 |
| Future Work..... | 97 |
| REFERENCES..... | 99 |
| APPENDIX..... | 107 |
| VITA..... | 113 |

LIST OF ILLUSTRATIONS

| Figures | page |
|---|------|
| 1. A typical schematic of the engineering design process. | 3 |
| 2. Classifications of finite elements..... | 7 |
| 3. Hand generated model of fiber glass..... | 13 |
| 4. Side view of skull-brain model with face bones..... | 19 |
| 5. Open B-spline curve and polygon..... | 23 |
| 6. Bezier net and resulting surface..... | 24 |
| 7. Boundary curves for patch or surface..... | 25 |
| 8. Boundary curves with P_1 | 27 |
| 9. Boundary curves with P_2 | 27 |
| 10. Surfaces $F(r,s,0)$ and $F(r,s,1)$ | 30 |
| 11. Surfaces $F(r,s,0)$ and $F(r,s,1)$ with P_1 | 30 |
| 12. Surfaces $F(r,0,t)$, $F(r,1,t)$, $F(0,s,t)$, and $F(1,s,t)$.. | 32 |
| 13. Corner points of the volume..... | 32 |
| 14. Curves of the volume..... | 33 |
| 15. Eigenvalues of a 4 node quadrilateral element..... | 46 |
| 16. Quadrilateral element with floating node..... | 48 |
| 17. Trace of the siffness matrix plotted..... | 48 |
| 18. Mapping of elements in x,y space into r,s space..... | 49 |
| 19. Cylinder in space..... | 52 |
| 20. Circular cross section of a cylinder..... | 52 |

| | |
|---|----|
| 21. Plot of the function relating r and r^* with uniform spacing of the resulting nodes..... | 56 |
| 22. Plot of the function relating r and r^* with nodal spacing varying from small to large..... | 56 |
| 23. Plot of the function relating r and r^* with nodal spacing varying small, large and then small..... | 56 |
| 24. Two dimensional mesh numbered largest first..... | 58 |
| 25. Two dimensional mesh numbered smallest first..... | 58 |
| 26. Parallelogram..... | 63 |
| 27. Trace of the stiffness matrix plotted with the sine of the angle..... | 64 |
| 28. Bezier curve bounded reion discretized into 35 elements..... | 67 |
| 29. Parameter space of the model..... | 67 |
| 30. Region discretized into 77 elements..... | 68 |
| 31. Non-uniform distribution of nodes..... | 69 |
| 32. Parameter space non-uniform distribution of nodes ... | 69 |
| 33. Optimized mesh for the 35 element model..... | 70 |
| 34. Parameter space for the optimized model..... | 70 |
| 35. Optimized model using the angle calculation..... | 71 |
| 36. Boundary curves for the second example..... | 72 |
| 37. Second example discretized into 32 elements..... | 72 |
| 38. Parameter space for the second example..... | 73 |
| 39. Second example defined by 96 nodes..... | 73 |
| 40. Non-uniform elements for the second example..... | 75 |
| 41. Non-uniform model's parametric space..... | 75 |
| 42. Optimized uniform mesh..... | 76 |
| 43. Optimized uniform mesh parametric space..... | 76 |
| 44. Optimized non-uniform mesh..... | 77 |

| | |
|--|----|
| 45. Parametric space for the optimized non-uniform mesh.. | 77 |
| 46. Mesh after parameterization by arc length..... | 78 |
| 47. Optimized mesh of the model parameterized by arc length..... | 79 |
| 48. Parametric space for the optimized arc length model.. | 79 |
| 49. Six surfaces defining the skull's exterior..... | 82 |
| 50. Skull-brain model [6 x 6 x 6]..... | 84 |
| 51. Skull-brain model [8 x 7 x 6]..... | 85 |
| 52. Several layers of bricks from the skull-brain model.. | 86 |
| 53. Optimized skull-brain model [6 x 6 x 6] bottom layer..... | 87 |
| 54. Optimized midsagittal layer..... | 88 |
| 55. Twelve curves defining the epoxy matrix model..... | 90 |
| 56. Three views of the epoxy matrix model..... | 91 |
| 57. Selected slices of the epoxy matrix model..... | 93 |
| 58. Optimized epoxy matrix model..... | 94 |
| 59. Optimized first parametric t layer of the epoxy matrix model..... | 95 |

Tables

| | |
|--|----|
| 1. Values of the Stiffness Measure and the Potential Energy for the Second Example..... | 80 |
| 2. Values of the Stiffness Measure for the Skull-Brain Model..... | 89 |
| 3. Values of the Stiffness Measure for the Epoxy Matrix Model..... | 92 |

CHAPTER 1

INTRODUCTION

The focus of this research has been upon the engineering design process for three-dimensional objects. This process includes everything from the conception of an abstract idea to the product and final application of the physical object. Once the abstract thoughts are made somewhat concrete, analysis and modification of them may take place to refine the ideas. It is the design, analysis, re-design loop which is the specific concern of this work. Refining of the proposed design is usually done by analyzing the object using various computer programs. Each of the programs requires a mathematical model of the object in a slightly different format. The design analysis loop progresses each of these separate models to be independently modified. At the end all of these have to be combined back into one. Any changes made along the way to one of the models should be reflected in all of the others. This is not usually done and results in severe problems of consistency. In the next section an overview of the engineering design process is described as it exists today. Then more details of the design analysis loop are given. Finally the process and its problems are summarized.

Chapter 2 details a proposed system to take care of many of the problems which will have been described. The main idea presented is that a higher level of abstraction is needed for the objects that

designed and analyzed. This is called the high level model and from this model all other models are to be derived. In Chapter 3 the underlying mathematics of the high level model are formulated. The ideas which have been developed to represent curves and surfaces are extended to the volume representation.

The analysis of the model plays an important part in the design analysis loop. Reasonably good models must be available to analyze or the answers obtained will be questionable. Chapter 4 deals with the Finite Element Method for analysis of solids. Consideration is given to how the method is formulated and the assumptions which are made concerning the model. Knowing these assumptions, schemes have been developed that provide better models than were previously produced for analysis.

In Chapter 5 the implementation of the high level model is detailed along with the new schemes to be used to generate models for analysis. The focus of this work has been on three-dimensional objects, but two-dimensional approximations still play an important role, so both are discussed. The final chapter displays several examples of the work. The conclusions and thoughts on future work are also given.

Overview of the Engineering Design Process

The engineering design process for structural objects is shown schematically in Figure 1. The general functions to be performed have not been changed for many years. The amount of time and money spent in each has changed significantly. In the past the design analysis loop was minor and was performed empirically. The bulk of the effort and

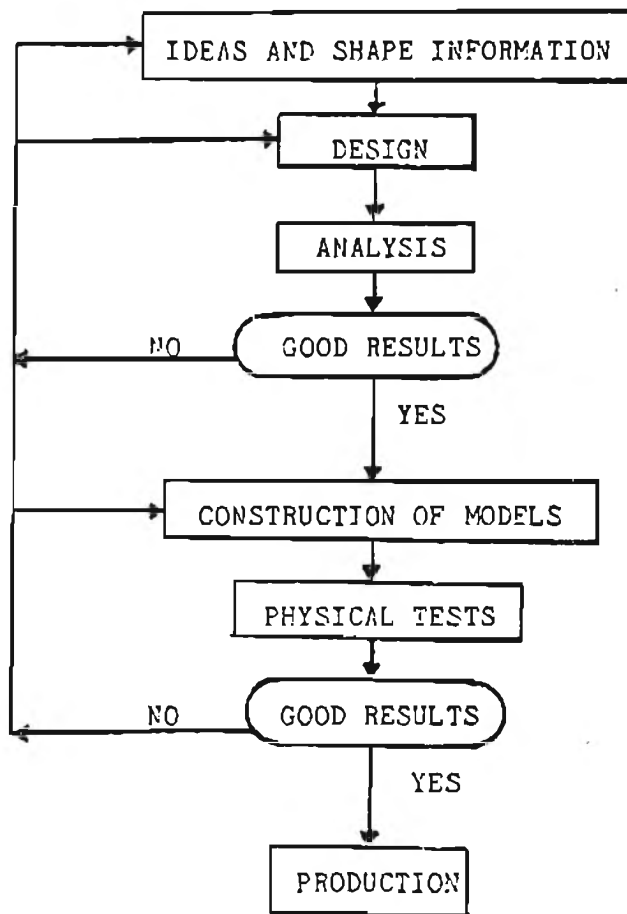


Figure 1. A typical schematic of the engineering design process.

money was spent constructing physical models, testing, and reconstructing the models. As the objects designed became larger and more complex this construction testing loop became very expensive.

In the late 1950's methods were developed by engineers to perform much of the structural analysis using the computer [95]. Since then a greater emphasis has been placed on the design analysis loop. The numerous scale models and full size models have just about been replaced by the computer. Other types of analysis have also been developed which use the computer. These computer programs allow the

engineer to simulate almost any type of environment and give results predicting how the object will respond.

Throughout the 1960's and into the 1970's the computer has been used in all phases of the engineering design process. Referring to Figure 1, the generation of the shape information is known as Computer Aided Geometric Design [8,45]. The analysis was originally the only portion which used the computer but now the data preparation (pre-processing) and the viewing of the results of the analysis (post-processing) rely heavily on the computer [16,78,92]. The fabrication and construction of physical models and the actual objects may be performed by the Computer Aided Manufacturing process [96]. It is even possible to have computer controlled robots do the assembly and testing of the object [31].

Present Process

Today's engineering design process begins with a sketch or an idea of a three-dimensional object. This shape information is refined and with the help of a designer/draftsman, preliminary drawings are made. The drawings are looked at and checked by several people, any of whom could and do make changes. When the design is finally approved it goes on for analysis.

Another situation is the re-design of an existing object. The engineer will start with a complete set of drawings and work from these.

The types of analysis to be performed varies extensively. They could include: 1) structural, static and/or dynamic, 2) cost, 3) thermodynamic, 4) magnetic, 5) kinematic, and so forth. Most of the

analyses are done using various computer programs. The input to each program varies in type and form, but most require the shape data, material type and properties.

The structural, thermodynamic, and magnetic analyses are usually performed using the finite difference or finite element methods. Both of these methods require that the domain be discretized into hundreds and possibly thousands of sub-domains. The loading and boundary conditions must also be specified for each of the sub-domains. Then analysis may be performed.

Once any of the analysis routines is run the results need to be interpreted. If the analysis indicates that the design is bad, then the engineer must go back to the appropriate point and make changes. Usually more than one analysis program is run so these changes must be made in all of the input data for the routines, and the programs re-run. This loop continues until the engineer is satisfied with the design, or until time and/or money for analysis runs out.

When the design has made it through the analysis loop, the final drawings (blueprints) may be made. These are usually reconstructions of the three-dimensional object from the analysis models and the original drawings.

The fabrication of the design, if it is a mechanical part rather than a building or other large structure, may be under computer control. If so the drawings must again be digitized for the numerically controlled machines [34]. This data is then fed into the machine and the object is cut or milled. After the object is fabricated, actual testing may be done. If it fails then the design process is re-entered at the appropriate point. When all of the "bugs"

are worked out, the object may go into production.

At this point the engineering design process may be finished but sometimes this is not the case. The object, when put into use, may fail, necessitating a re-analysis. It may also be used in an application not originally considered in the design. For both of these cases the analysis loop must be re-entered.

Design Analysis Loop

Our specific area of concern is the design structural analysis loop. First the structural analysis process is discussed and then how it is performed will be detailed. The problems which exist will also be covered.

Structural Analysis

There are three major types of structural analysis. They are the closed form solution, the finite difference method, and the finite element method. This research will be limited to working with the finite element method [9,35,43,99]. Classification of the analysis within the finite element method will be done by the element type used, as shown in Figure 2.

The two-dimensional elements were the first to be developed. Several assumptions were made to simplify the mathematics. The plane stress element assumes that the object is thin and the distribution of forces through the thickness will be constant. The assumption for the plane strain element is that normal to the element the dimension is very large. This means that any plane may be taken as "typical" or that the change in the distribution of forces does not vary significantly in the normal direction [91].

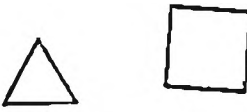
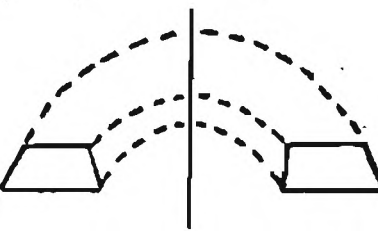
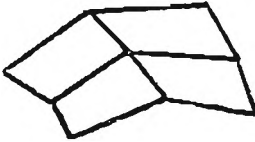
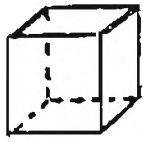
| TYPE | DIMENSION | |
|-------------------------------------|-----------|---|
| PLANE STRESS AND PLANE STRAIN | 2 |  |
| AXISYMMETRIC | 2.5 |  |
| THIN SHELLS | 2.5 |  |
| BRICK | 3 |  |

Figure 2. Classification of finite elements.

The two and a half-dimensional elements are called this because they represent the object in three-dimensions but only two independent variables are needed to define them. The thin shell elements collectively defining an object may be thought of as a bivariate surface. The axisymmetric elements are defined in an R-Z plane, and the geometry does not change in the theta direction.

The first elements used were the two-dimensional elements. The

three-dimensional structures to be analyzed were approximated by taking two-dimensional slices. These slices were analyzed and the engineer then estimated the distribution of forces between the calculated points. Some three-dimensional objects were very accurately analyzed in this fashion. More recently the structures to be designed have been complex and required more accurate results. For these structures the full three-dimensional analysis routines have been developed. Most of the finite element work today is still in the two- and two and a half-dimensional realm but the use of the three-dimensional analysis is growing.

The Two-Dimensional Case

The design analysis loop as previously mentioned consists of the data preparation, data checking, analysis, review of the results, modification of the design, data preparation, etc. The data preparation is performed by first taking the drawings and deciding which portions need to be analyzed. Here the engineer must use his judgment and select the important or significant areas. The specific portions are then discretized into hundreds or possibly thousands of small elements. At first this was a manual procedure. Now computer programs exist to aid in the model development [2,47,63,78]. These programs require that the boundaries be given for each region and some additional input to specify the number of points or elements to be generated. These model generation routines vary in the amount of input required. Some will run from a very small set of data while others will require large sets of data. The engineer must check the results of the generation routines to be sure that the model represents the

original object accurately. His judgement and past experience are used to check that the mesh generated will be well conditioned for the numerical solution. Since only a finite number of elements are used it is desirable to have many elements in the regions where large stress gradients will occur, and fewer elements where the stresses will be low. This will usually give better results without spending more computer time on more elements. The engineer must also choose his type of element and be sure that it is the best suited for his problem. These factors all have some effect on the accuracy of the answers [15,38,80].

Usually each model or mesh generation computer program was developed for a specific analysis routine. Since there has been no standard for the input to the analysis routines, each one to be used requires its own mesh generation program to prepare the model. The editing, or modification, of the mesh is largely a manual operation. Most of the generation programs run as batch jobs and allow for no user interaction. Lately the use of computer graphics has helped ease the burden of model preparation but these routines are not universally available [20,21,84].

It was estimated in 1970 at Lockheed [16] that fifty percent of the time in the analysis loop was spent in the data preparation and checking. At Boeing the same year the estimates were fifty percent of their manpower and forty percent of their computer time in the design analysis loop for data generation [92]. The interpretation of the results were estimated to take forty percent of their manpower and ten percent of their machine time. The situation today has changed somewhat. Since 1970 computers and the analysis programs have improved

considerably. These improvements have allowed more analysis runs to be made using the same model while changing one or two variables. The engineer will now set up his model and then make any number of runs. The analysis routines can now perform dynamic and non-linear analyses of the model. Each of these take a factor of ten times more computer time. The modeling time is the same for a static, dynamic and/or non-linear run. Very little progress has been made in the modeling area during the same time period.

The analysis routines usually output the results on line printers. There may be thousands of pages printed for each run. The engineer must look at these pages to obtain his answers. Computer graphics routines have been written to make pictures of the results. The results are basically pattern information and the pictures will convey this type of data quicker than a page of numbers. The calculated displacements may be displayed by warping the initial geometry. The other results such as pressure, temperature, and stress/strain components may be viewed with contour lines [22,66] or as continuous tone pictures [22,27,28].

The engineer must verify that the solution is good before he believes the results of the analysis. This is a very hard problem to solve. Usually the solution is judged to be good if the computer programs' answers were about what the engineer expected. The interpreted results will indicate changes which should be made to the model. If the modifications are minor they will probably be made to the existing mesh. If the changes are major or a different mesh is needed then the mesh generator must be used again. The classical proof that the answers are good is to prove that they are converging. The

proof is usually performed by refining the mesh size and resolving the problem again. This should show that as the number of elements goes toward infinity the answers converge. Several problems immediately appear. The first is the amount of time which the engineer has to solve the analysis. If the time is short any answer may be taken. If time allows, then an attempt may be made to prove the worth of the answers. Most finite element programs have a definite limit on the size of problem which they can handle. To obtain the best answers the first time, the engineer will normally use as many elements as he can. This means that the program can not handle a mesh with twice as many elements. Alternatives to this classical way of showing convergence are being researched [24,38,48,68,94].

The problems with the design analysis loop are many. The first is in the discretization of the drawings. Here much of the information about the object is discarded by approximating the shape with straight sided polygons. Later when computer graphics routines are used to view the results of the analysis, information about the outline and other properties must be derived. Higher order elements exist which will capture more information about the object's shape, but these are not widely used and some automatic generation routines do not produce them. Usually more than one analysis is performed on a model. Any changes to the model arising from one analysis must be made in all models. This ripple effect is usually not performed unless the change is major. Sometimes the analysis indicates that the model should be subdivided and portions examined in closer detail. Most present model generation routines do not have this facility. The reverse problem of concatenation of models is also a time consuming manual procedure.

The Three-Dimensional Case

When the three-dimensional elements and their usage are discussed there is not as much experience to draw upon as in the two-dimensional case. The analysis routines have not been available for as long and the data preparation and result interpretation routines are very few in number. The finite element routines require that the volumes be represented by hexahedrons (bricks). The mesh or model generation routines most commonly specify the external curves and surfaces, then derive the volume elements [1,3,14,36,61,64]. Another way is by specifying the component volumes [10,17,16,19]. The majority of the model generation routines require that the object be broken into a set of logical cubes. These cubes are then broken down into the brick elements and combined to form the model. The models tend to have fewer total elements than their two-dimensional counterparts because each element in three-dimensions has more degrees of freedom. Each linear hexahedron has twenty-four degrees of freedom as compared with eight degrees of freedom for the two-dimensional linear quadrilateral. A separate equation is generated by the finite element method for each degree of freedom and the number of equations which may be solved for by a given analysis routine is limited.

A major problem with this work is in visualizing the models. There is no natural way to render a three-dimensional object on a two-dimensional medium (i.e. the plotter paper or the cathode-ray tube) [72,75]. The computer programs required to view these three-dimensional models with the hidden portions removed have been the basis of much research [25,74,90,97]. Presently there are a number of routines available to view the three-dimensional models but they

take a large amount of computer time.

Most models used in three-dimensional analysis have been hand generated, see Figure 3.

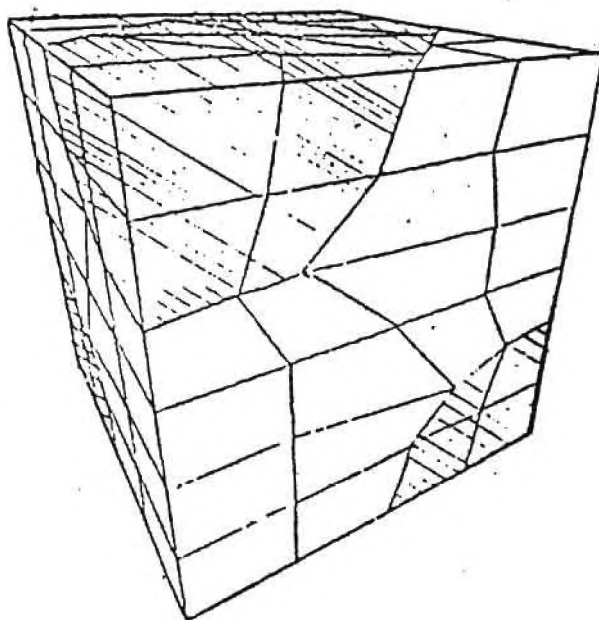


Figure 3. Hand generated model of fiber glass.

The model generation is the most difficult part of the design analysis loop. There are not many figures available as to the amount of time spent on the model generation and checking, but an estimate would be over eighty percent of the effort. The number of elements used in the three-dimensional analysis is small so the accuracy of the representation must be checked carefully.

After the analysis is run the interpretation of the results may begin. The amount of printed output produced by the three-dimensional analysis routine is greater than that produced by a two-dimensional run. Here again there are several graphics routines available to display the results [22,28,64,78]. Most of the viewing of the results

using the hidden portion removal packages will use more computer time than it took to obtain the results. This expense has not helped spread the use of three-dimensional finite element analyses.

Another problem with any of the design analysis loops is the re-entering of the loop at a future time. Since the exit from the loop may not be when the analysis is completed but rather when time or money runs out, the models used may be discarded or at least abandoned. To restart the analysis loop may be as hard as beginning afresh.

Summary

The engineering design process and its problems as it exists today have been described. The major problem is that there are too many partial models in the process. When changes are made to any particular model they need to be made in all, which is usually not performed. The models are also specialized to the particular application, so there is no common data form. Communication between the engineer and the model must be improved to increase the effectiveness of his work and to reduce the time requirements. In the next chapter a system is proposed which will help improve many of the problem areas of the engineering design process.

CHAPTER 2

PROPOSED SYSTEM

Introduction

For any given design problem, one high level model rich enough in information could solve many problems in the engineering design process. This model would have all other models and representations derived from it. It could be generated using an interactive computer graphics terminal and be available to a wide number of engineers. The engineering design process might then be performed as follows.

The engineer would sit down at an interactive graphics terminal and begin the program. He could start fresh or he could use what was previously generated by retrieving it from secondary storage. The shape information could be displayed on the graphics device at his will. He could then add to the existing definition or modify it until it met the specifications. At any time plots (hard copy) of the current view on the display could be made. The data preparation for analysis would be started by specifying a few global parameters. The program would then start discretizing the model into the selected type of elements. At any time the engineer could interact with the discretization and make changes globally or locally. If only a subset of the model needed to be analyzed then the engineer could specify this by interacting with a light pen or tablet and pointing at the area wanted. The program would then just deal with this subsection.

The loading and the boundary conditions would be given logically by pointing at the areas and the values input from the keyboard. There would also be a process for recovering previous states of the model if a mistake were made.

Once a model was ready for analysis the input to the specific routine, in the required format, would be prepared. It is probable that the analysis would not be performed on the same computer, so the input data and the results of the analysis would have to be communicated between machines. The same type of graphics programs used to view the model in the generation stage would be used to view the results of the analysis. Various plots of the results could be made and viewed interactively or hard copy plots could be made to be viewed at the engineer's leisure.

From the various views of the results the engineer could then decide what printed output he needs. Instead of getting thousands of printed pages he could specify the needed output and get it. The amount of time spent interpreting the results could be reduced by using the graphics first.

If modifications in the design are indicated by the results, the engineer will make them to the high level model. By making the changes in this high level model, all subsequent models would have the changes reflected in them. Any number of specific models for analysis could be derived from the high level model. When changes are needed to be made it is only this one model which gets changed.

Once the design analysis loop is complete, the final blueprints for construction may be made from the high level model. The high level model could also be used to produce data for numerically controlled

milling machines. Each part could then be cut or milled for assembly. It would even be possible to use the model to derive information to control robots in the assembly of the parts.

The engineering design process would then be complete and the high level model could be stored away until it was needed. There would be no loss of information through time by the engineer cleaning out his office and throwing some of the data away because the model would be stored on secondary storage of the computer.

The High Level Model

As presented the proposed system could be realized if a representation of complex three-dimensional objects can be expressed as a high level model. There are several different choices for the representation of the model. The model could be represented as a set of hexahedrons. This would be advantageous to the analysis routine since hexahedrons are the required input. The choice could be made to represent the object as a display file [75]. Most of the interaction with the model will be done via a graphics terminal so the model could be described in a form easily interpreted to generate the graphical display. Alternatively the model could be the commands required to mill the object on a numerically controlled machine, if it is a mechanical part rather than a whole building. Eventually it has to be built, so why not start at the beginning with a representation which describes how to construct the object? This is by no means a complete list of possibilities. The high level model will have to provide information to each of the processes, the analysis, the display, and the production. The representation or rendering of the model for each

is different so it could be concluded that the high level model must be independent of the ways in which it will be used. This has been expressed by P. F. Riesenfeld as the separation of the model from the rendering [82].

An approach which might work would be to represent components of the three-dimensional object in their natural form. Referring to Figure 3, this concept can be demonstrated. The object being modeled is a composite structural material which could be fiberglass [65]. Layers of fibers are set in an epoxy matrix with each layer placed at $\pi/2$ radians to the previous layer. The model itself represents two fibers from different layers in the epoxy matrix. The cube is approximately 25 microns on an edge. The fibers are cylindrical and could be represented as such. The cube of the epoxy matrix is just that, a cube, missing the volume which the fibers take up. The most natural representation for this model would be two cylinders and a cube. This representation is very similar to that which has been proposed by I. C. Braid [17].

A second example may be obtained from Figure 4. Here the human head is being modeled [87]. The model includes the skull, the brain and face. The skull is composed of three layers, the outer table bone, the middle layer of diploe, and the inner table bone. The ratio of adjacent layer's thicknesses is almost constant. Around the brain there is also the subarachnoid space, a fluid layer, which could be considered constant in thickness. The skull-brain model could be represented as a surface enclosing a volume. There are a wide variety of surface representations to choose from. One choice is a local approximating surface. The face in the model was an afterthought. If

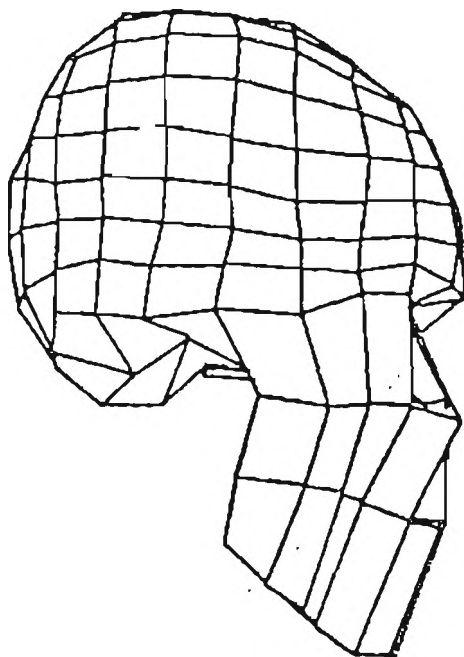


Figure 4. Side view of skull-brain model with face bones.

was added to provide the mass that exists in the actual skull. Since it is a polyhedral representation it could be left as such.

For the two objects presented, the computer program to implement the high level model would have to handle cylinders, cubes, polyhedrons, and approximating surfaces. If other examples are considered then spheres, interpolating surfaces, and some free form volume representations might be added. For each of these the program would have to provide the natural representation and be able to handle the display of one at a time or all together, and produce hexahedrons for analysis. Such a program needs to be structured so that it is modular and easily changed. A procedure could be written to take the data for each of the primitives and do the necessary operations. The head will be modeled by a surface procedure and a polyhedron

procedure. The crossed fibers would be represented by a cylinder procedure and a cube procedure. This concept is not new and has been used successfully by M. E. Newell to display complex three-dimensional objects [73]. It is known as Procedure Models. The mathematics behind the high level model will be presented in the following chapter.

CHAPTER 3

MATHEMATICAL FORMULATIONS

Introduction

This chapter develops a mathematical representation for the high level model. The requirements of the representation for the high level model are that the object being modeled must be represented in a straightforward, natural form. The choice has been made to use a trivariate representation. This has many advantages and seems to be a general enough form for many applications. The generation of finite elements from the trivariate representation is convenient. When the model needs to be displayed the outline may be found easily or the boundary surfaces derived very quickly. If the instructions for a cutting tool are needed they can be derived and displays of sections for blueprints are possible. The formulation is developed by first discussing space curves, surfaces, and then volumes. These are presented in their parametric form. Throughout the discussion linear functions are used to simplify the presentation. In the last section extensions of the derived expressions to higher orders will be discussed.

Space Curves

The representation of a space curve may take many forms. For work with the computer in the design and manipulation of geometry, the most convenient form of representation has been parametric. The arguments

for and against parametric representation have been discussed many times and can be found in [37,50,51,83]. Primarily parametric form is used because it is axis independent. There are several other reasons which make it even more attractive for use. The first is that the form may be multiple valued with respect to the reference coordinate system. Second it may be piecewise defined with specified continuity, so no attempt need be made to represent the complete structure with one equation. Finally it is convenient for generation of the nodal coordinates needed for the finite element models, this will be discussed in Chapter 5. For a space curve the parametric form is

$$F(r) = [X(r), Y(r), Z(r)] \quad (3.1)$$

where r is the independent variable whose usual domain is $[0,1]$.

In the use of this form to represent an arbitrary curve in space defined by a set of points, the $X(r)$ are often formulated as:

$$X(r) = \sum_1 A_i(r) X_i \quad (3.2)$$

where $A_i(r)$ is an appropriate function and the X_i are the points in space. Two similar equations exist for $Y(r)$ and $Z(r)$. The choices for $A_i(r)$ are many and varied. If interpolation is required they could be the coefficients of a Vandermonde matrix [33,41]. For approximation the X_i could be the defining polygon for a B-spline or a Bezier scheme. For B-splines, see Figure 5, the $A_i(r)$ are the basis functions [42,81]. If Bezier curves are used the $A_i(r)$ are the Bernstein polynomials whose general form is given below [12].

$$A_i(r) = \frac{m!}{i!(m-i)!} r^i (1-r)^{m-i} \quad (3.3)$$

The alternatives are numerous and the reader is referred to references

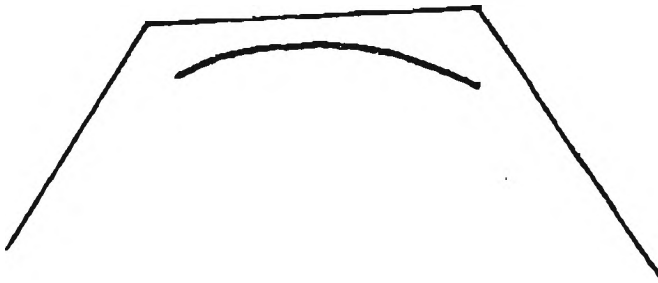


Figure 5. Open B-spline curve and polygon.

[26,44,49,77] for some of them.

Surfaces

The parametric form of a surface is

$$F(r,s) = [X(r,s), Y(r,s), Z(r,s)] \quad (3.4)$$

where r and s normally have the range from 0.0 to 1.0. For interpolation and approximation the expression for $X(r,s)$ is

$$X(r,s) = \sum_i \sum_j A_{ij}(r,s) X_{ij}. \quad (3.5)$$

Here $A_{ij}(r,s)$ is an appropriate function and the X_{ij} are a set of points, also called a net. Similar equations exist for $Y(r,s)$ and $Z(r,s)$ [86]. An example of a net of points and the resulting Bezier surface may be seen in Figure 6.

Often the set of points for the surface will not be available. Instead the three or four boundary curves will be given as in Figure 7. Here interpolation of the data given is needed to define a surface.

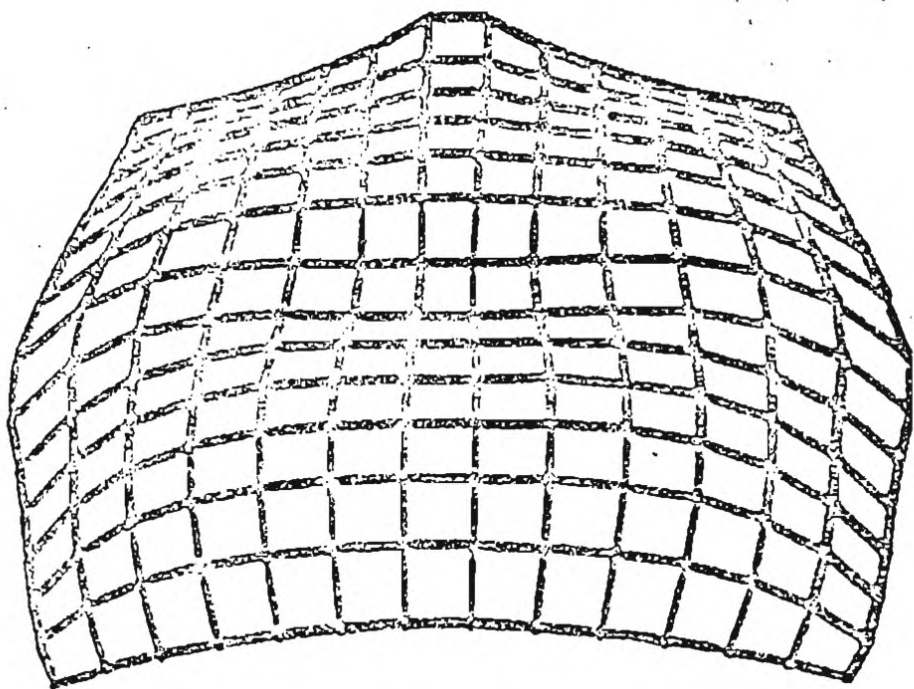
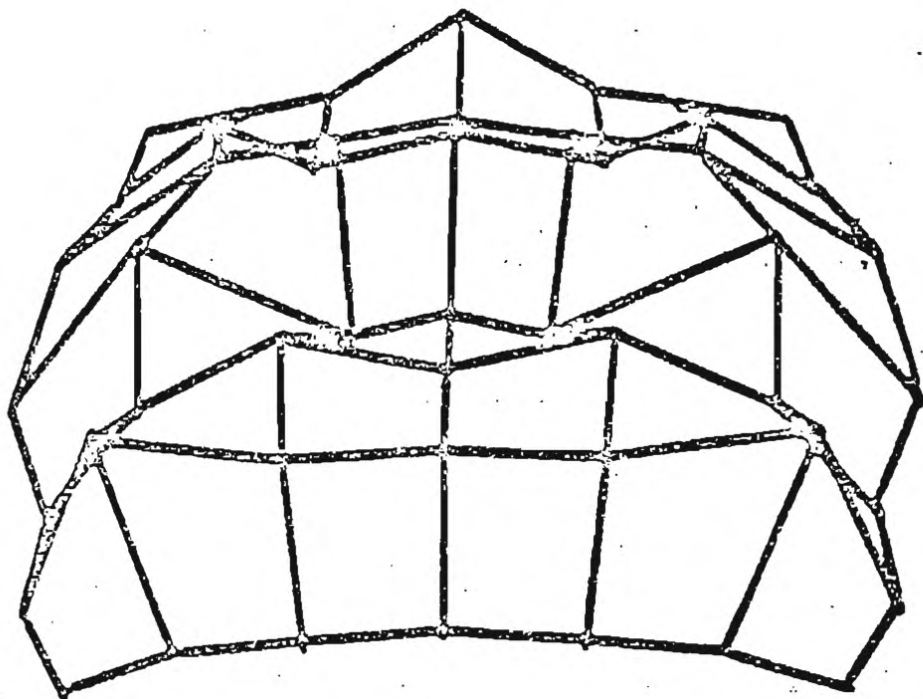


Figure 6. Bezier net and resulting surface.

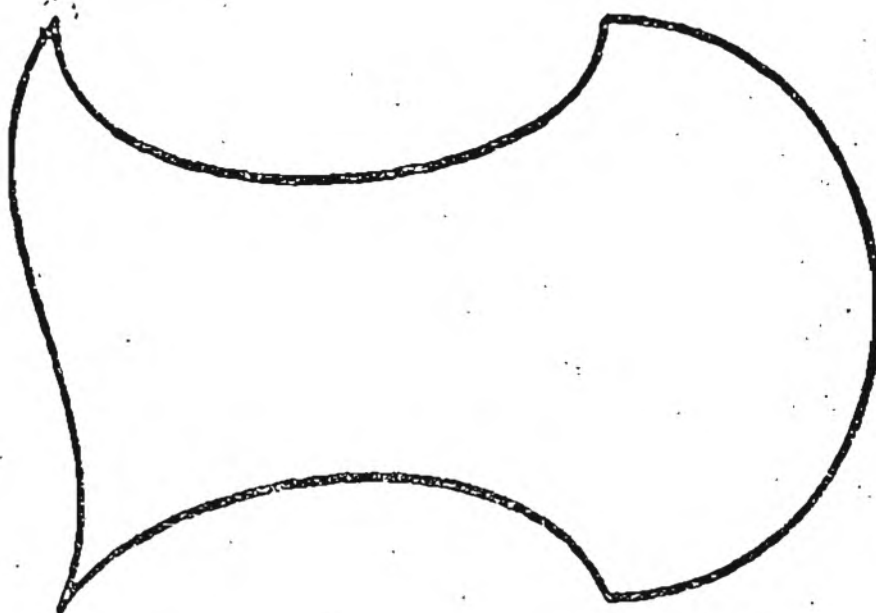


Figure 7. Boundary curves for patch or surface.

In [37] S. A. Coons proposed his now well known solution. Later his work was expanded upon and formalized by many others [51,57,58,81]. A brief summary of the work will be given here.

W. J. Gordon in [56] presented the projector notation which will be used in the discussion that follows. A projector P_1 (a linear, idempotent operator) for the univariate case, interpolating two points $F(0)$ and $F(1)$, is defined as

$$P_1 F(r) = (1-r)F(0) + rF(1). \quad (3.6)$$

The $(1-r)$ and r are called weights or blending functions, and $F(0)$ and $F(1)$ are positional data. As r varies from 0 to 1, $P_1 F(r)$ varies linearly from $F(0)$ to $F(1)$.

If four points $F(0,0)$, $F(0,1)$, $F(1,0)$, and $F(1,1)$ are given, a similar bilinear scheme can be developed

$$PF(r,s) = (1-r)(1-s)F(0,0) + (1-r)sF(0,1) \\ + r(1-s)F(1,0) + rsF(1,1). \quad (3.7)$$

This equation is known as the bilinear tensor product or cartesian product surface.

When two space curves $F(0,s)$ and $F(1,s)$ are given as in Figure 8, a linear bivariate interpolant, between them can be defined, also known as a lofting projector [5,16]

$$P_1F(r,s) = (1-r)F(0,s) + rF(1,s). \quad (3.8)$$

If the $F(r,0)$ and $F(r,1)$ curves given instead, (see Figure 9) a dual equation in s may be expressed as

$$P_2F(r,s) = (1-s)F(r,0) + sF(r,1). \quad (3.9)$$

Taking the product $P_1P_2F(r,s)$

$$P_1P_2F(r,s) = (1-r)(1-s)F(0,0) + (1-r)sF(0,1) \\ + r(1-s)F(1,0) + rsF(1,1) \quad (3.10)$$

which is equivalent to (3.7). W. J. Gordon terms this the minimal operator on $F(r,s)$ because it interpolates only those pieces of data common to P_1 and P_2 , namely the corners [57]. This interpolation property is demonstrated below for one corner. The same procedure can be used to show that $P_1P_2F(r,s)$ also interpolates the other three corners.

Example for corner $F(0,1)$

$$P_1P_2F(0,1) = (1-0)(1-1)F(0,0) + (1-0)(1)F(0,1) \\ + (0)(1-1)F(1,0) + (1)(0)F(1,1) \\ = F(0,1)$$

Another way of combining P_1 and P_2 to interpolate the given data is by taking the so called Boolean sum symbolized by the exclusive-or symbol +

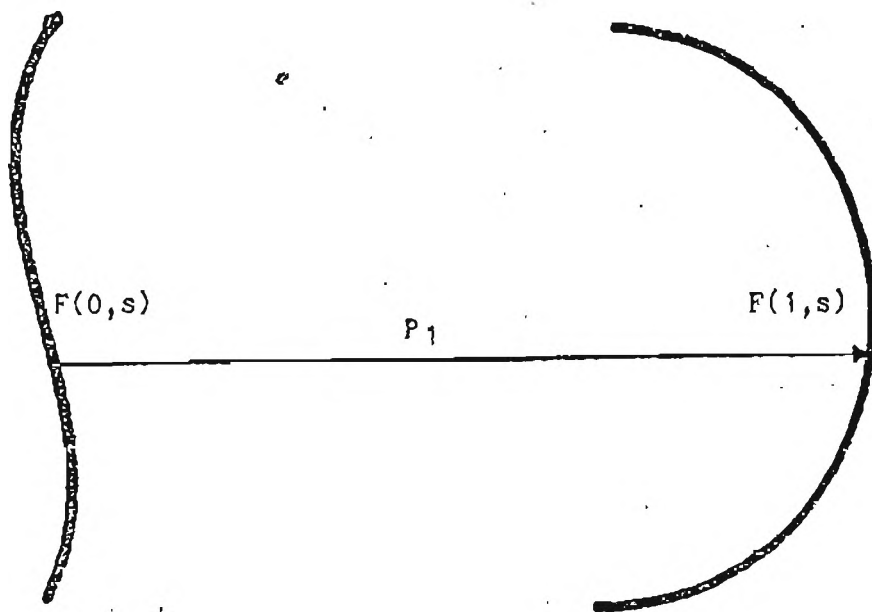


Figure 8. Boundary curves with P_1 .

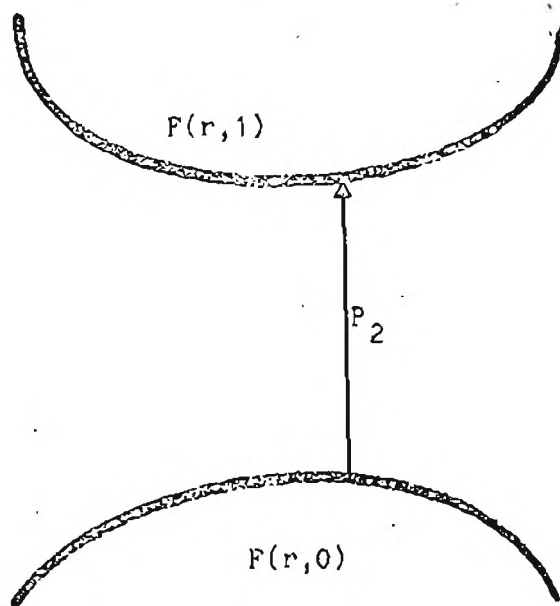


Figure 9. Boundary curves with P_2 .

$$(P_1 + P_2)F(r,s) = P_1F(r,s) + P_2F(r,s) - P_1P_2F(r,s) \quad (3.11)$$

or written out

$$\begin{aligned} (P_1 + P_2)F(r,s) &= (1-r)F(0,s) + rF(1,s) \\ &+ (1-s)F(r,0) + sF(r,1) \\ &- (1-r)(1-s)F(0,0) - (1-r)sF(0,1) \\ &- r(1-s)F(1,0) - rsF(1,1). \end{aligned} \quad (3.12)$$

This interpolates the boundary curves and therefore the corners are also interpolated.

Example for curve $F(r,1)$

$$\begin{aligned} (P_1 + P_2)F(r,1) &= (1-r)F(0,1) + rF(1,1) \\ &+ (1-1)F(r,0) + (1)F(r,1) \\ &- (1-r)(1-1)F(0,0) - (1-r)(1)F(0,1) \\ &- r(1-1)F(1,0) - r(1)F(0,0) \\ &= (1-r)F(0,1) + rF(1,1) + (1)F(r,1) \\ &- (1-r)(1)F(0,1) - r(1)F(1,1) \\ &= F(r,1) \end{aligned}$$

Gordon has call this the maximal operator on $F(r,s)$ because it interpolates everything which P_1 and P_2 each do. Note that the tensor product is a term of the Boolean sum.

The equation (3.12) should be recognized as the simplest form of a "Coons Patch". It is also known as the transfinite bilinear Lagrange interpolant. The word transfinite is used because it interpolates a nondenumerable set of points on the boundaries [58]. For the rectangular patch the following usually hold

$$P_1P_2F(r,s) = P_2P_1F(r,s) \quad (3.13)$$

$$(P_1 + P_2)F(r,s) = (P_2 + P_1)F(r,s). \quad (3.14)$$

There have been many implementations of such equations in design

systems [1,4,29,58,77,85]. Only a skeleton of the work which has been done for surface representation has been presented. The projectors P_1 and P_2 as presented are linear expressions but this is not to prevent them from being higher order expressions, which are discussed later in this chapter. With this as background the volume representation will next be considered.

Volumes

The trivariate form of the volume is

$$F(r,s,t) = [X(r,s,t), Y(r,s,t), Z(r,s,t)]. \quad (3.15)$$

The ranges of r , s , and t will be from 0.0 to 1.0. The $X(r,s,t)$ will describe the x coordinate in the real space of the volume, as the r , s , and t vary in the parametric space. The interpolation or approximation formula to a set of points in space is expressed for $X(r,s,t)$ as

$$X(r,s,t) = \sum_i \sum_j \sum_k A_{ijk}(r,s,t) X_{ijk}. \quad (3.16)$$

These types of equations have not been used in general to represent volumes. A problem with this interpolation or approximation scheme is that points are needed from within the volume. These may or may not be easy to obtain. Having the background of surface representation, volumes will be developed in a similar fashion.

Initially two surfaces $F(r,s,0)$ and $F(r,s,1)$ as shown in Figure 10, will be given. Similar to what was done for surfaces defined by boundary curves a projector $P_1 F(r,s,t)$ will be defined to linearly interpolate between the surfaces as shown in Figure 11 [13,56].

$$P_1 F(r,s,t) = (1-t)F(r,s,0) + tF(r,s,1) \quad (3.17)$$

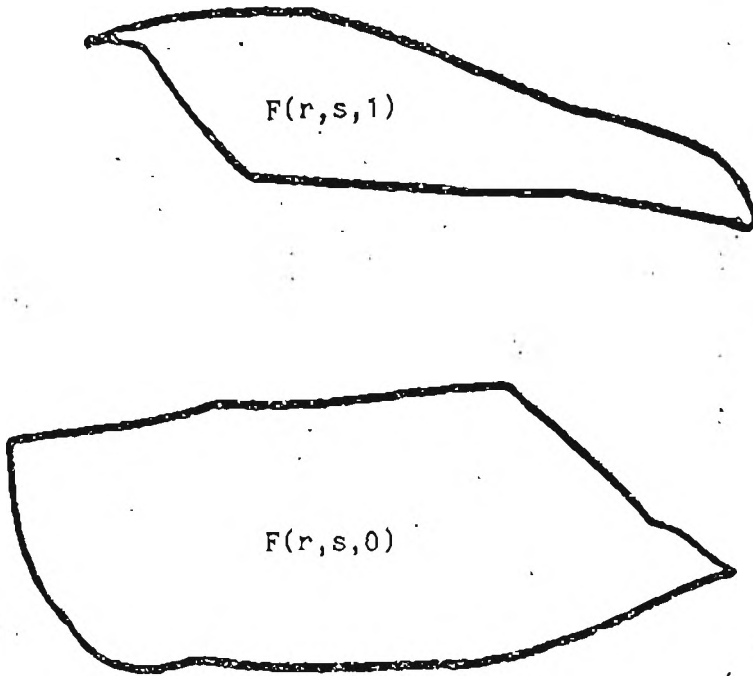


Figure 10. Surfaces $F(r,s,0)$ and $F(r,s,1)$.

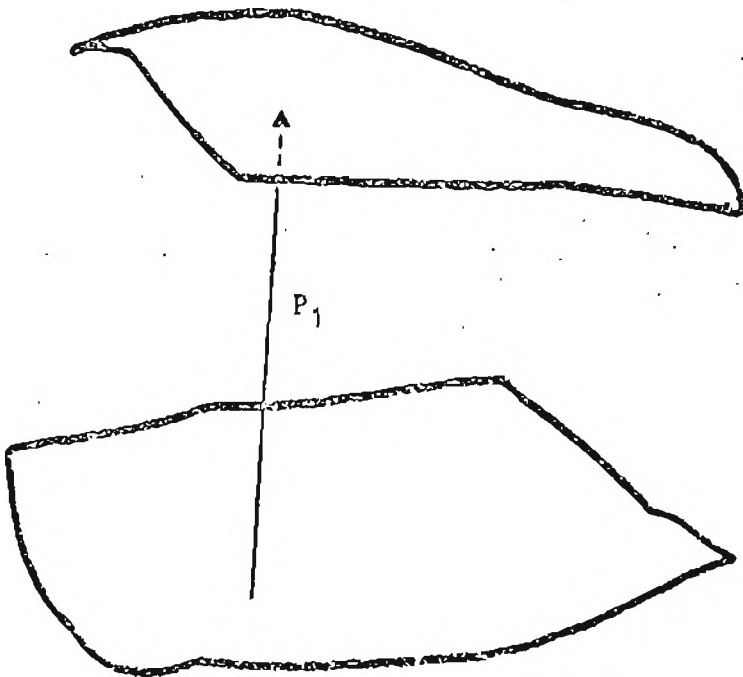


Figure 11. Surfaces $F(r,s,0)$ and $F(r,s,1)$ with P_1 .

If the other four surfaces $F(r,0,t)$, $F(r,1,t)$, $F(0,s,t)$, and $F(1,s,t)$, as shown in Figure 12, are given then two more projectors may be defined

$$P_2 F(r,s,t) = (1-s)F(r,0,t) + sF(r,1,t) \quad (3.18)$$

$$P_3 F(r,s,t) = (1-r)F(0,s,t) + rF(1,s,t). \quad (3.19)$$

These three projectors can be combined to give a trilinear tensor product volume or the minimal operator on $F(r,s,t)$

$$\begin{aligned} (P_1 P_2 P_3) F(r,s,t) &= (1-r)(1-s)(1-t)F(0,0,0) + (1-r)(1-s)tF(0,0,1) \\ &+ (1-r)s(1-t)F(0,1,0) + (1-r)stF(0,1,1) \\ &+ r(1-s)(1-t)F(1,0,0) + r(1-s)tF(1,0,1) \\ &+ rs(1-t)F(1,1,0) + rstF(1,1,1). \end{aligned} \quad (3.20)$$

Here only the eight corner points are used (Figure 13). They are the only points in common with each pair of surfaces. A proof of the interpolation by substitution may be seen in the appendix.

The Boolean sum interpolant, the maximal operator on $F(r,s,t)$ is defined as

$$\begin{aligned} (P_1 + P_2 + P_3) F(r,s,t) &= (P_1 + P_2 + P_3 - P_1 P_2 - P_2 P_3 \\ &- P_3 P_1 + P_1 P_2 P_3) F(r,s,t). \end{aligned} \quad (3.21)$$

If each term is written out then equation (3.21) becomes

$$\begin{aligned} (P_1 + P_2 + P_3) F(r,s,t) &= (1-t)F(r,s,0) + tF(r,s,1) \\ &+ (1-s)F(r,0,t) + sF(r,1,t) \\ &+ (1-r)F(0,s,t) + rF(1,s,t) \\ &- (1-s)(1-t)F(r,0,0) - s(1-t)F(r,1,0) - (1-s)tF(r,0,1) - stF(r,1,1) \\ &- (1-r)(1-s)F(0,0,t) - r(1-s)F(1,0,t) - (1-r)sF(0,1,t) - rsF(1,1,t) \\ &- (1-r)(1-t)F(0,s,0) - r(1-t)F(1,s,0) - (1-r)tF(0,s,1) - rtF(1,s,1) \end{aligned}$$

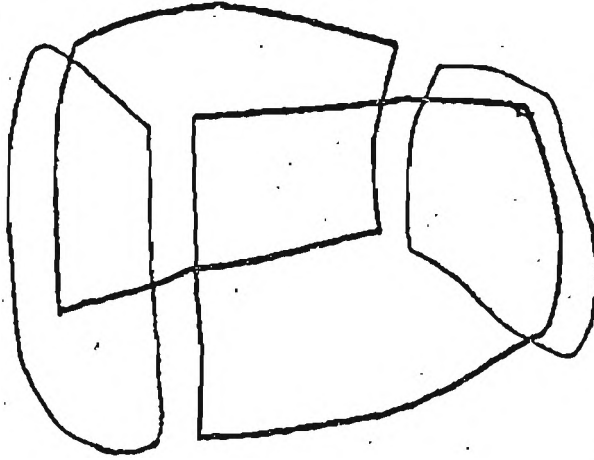


Figure 12. Surfaces $F(r,0,t)$, $F(r,1,t)$, $F(0,s,t)$ and $F(1,s,t)$

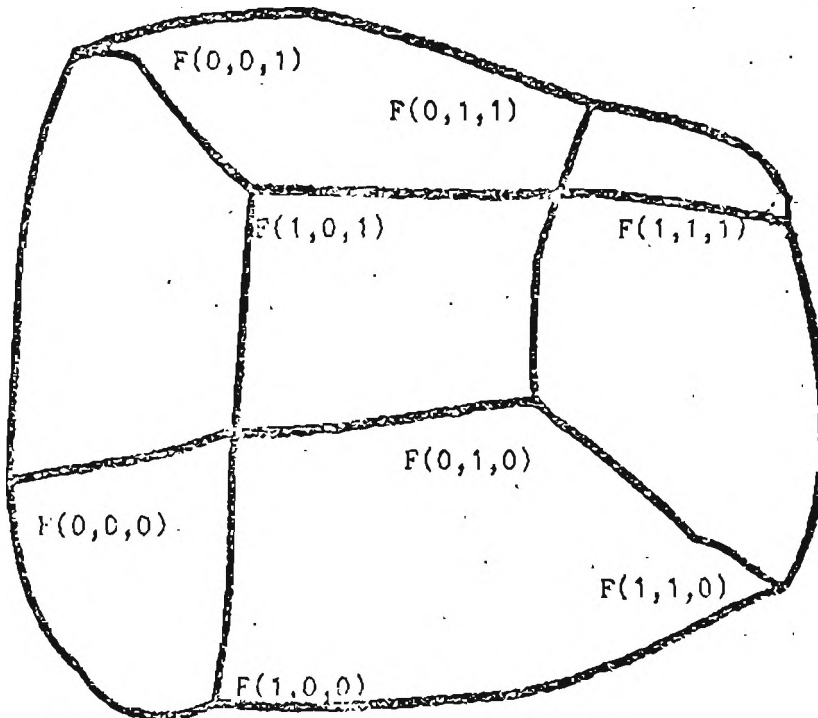


Figure 13. Corner points of the volume.

$$\begin{aligned}
& + (1-r)(1-s)(1-t)F(0,0,0) + (1-r)(1-s)tF(0,0,1) + (1-r)s(1-t)F(0,1,0) \\
& + (1-r)stF(0,1,1) + r(1-s)(1-t)F(1,0,0) + r(1-s)tF(1,0,1) \\
& + rs(1-t)F(1,1,0) + rstF(1,1,1). \tag{3.22}
\end{aligned}$$

The Boolean sum interpolant can be shown to interpolate the six surfaces, and therefore the twelve edge curves, see Figure 14,.

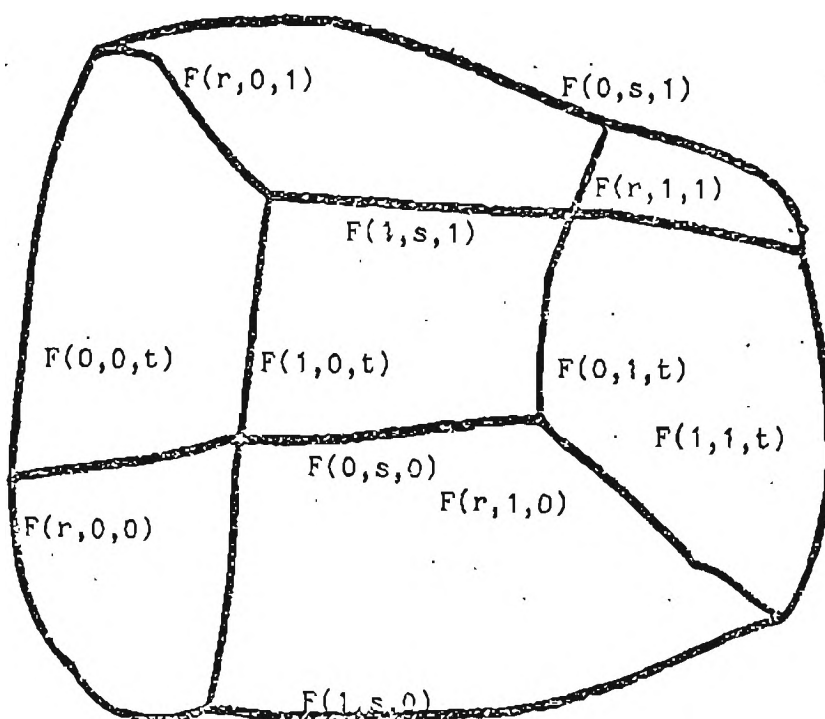


Figure 14. Curves of the volume.

and the eight corner points will also be reproduced. This may be seen by examining the terms in the expression. For interpolation of surfaces one of the first three terms will give the surface, while the other two will approximate it using the boundary curves. The second three terms will subtract off the approximation to the surface by the curves and will subtract an approximation to the surface using the four corner values. The tensor product term adds an approximation to the surface by the corner values which produces surface interpolation.

These interpolation properties are demonstrated in the appendix by substitution.

S.A. Coons described an "n-variate" form of his surface work which he called "hypersurfaces" [37]. Working out his expressions and substituting the projectors into it the expression becomes when $n=3$

$$(P_1P_2 + P_2P_3 + P_3P_1)F(r,s,t) = (P_1P_2 + P_2P_3 + P_3P_1 - 2P_1P_2P_3)F(r,s,t). \quad (3.23)$$

Expanding each term

$$\begin{aligned} (P_1P_2 + P_2P_3 + P_3P_1)F(r,s,t) = & \\ & (1-s)(1-t)F(r,0,0) + s(1-t)F(r,1,0) + (1-s)tF(r,0,1) + stF(r,1,1) \\ & + (1-r)(1-s)F(0,0,t) + r(1-s)F(1,0,t) + (1-r)sF(0,1,t) + rsF(1,1,t) \\ & + (1-r)(1-t)F(0,s,0) + r(1-t)F(1,s,0) + (1-r)tF(0,s,1) + rtF(1,s,1) \\ & - 2((1-r)(1-s)(1-t)F(0,0,0) + (1-r)(1-s)tF(0,0,1) + (1-r)s(1-t)F(0,1,0) \\ & + (1-r)stF(0,1,1) + r(1-s)(1-t)F(1,0,0) + r(1-s)tF(1,0,1) \\ & + rs(1-t)F(1,1,0) + rstF(1,1,1)). \end{aligned} \quad (3.24)$$

As before values of r , s , and t may be substituted into (3.24) to see what it interpolates and this is done in the appendix. For interpolation of curves, one of the first three terms will interpolate the given curve. The other two terms will approximate the curve using the corner values. The tensor product will also approximate the curve using the corner value and the minus two multiplier makes the sum equal to one curve. When a surface is considered, two of the first three terms will approximate it using boundary curves and the other will approximate the surface using the corner values. The tensor product term subtracts off two approximations to the surface using the corner data. Equation (3.24) has interpolated the curves and therefore the corners, but not the surfaces. The surface equation produced is

equivalent to (3.12) for the face of the logical cube in question. So equation (3.24) is called an intermediate operator on $F(r,s,t)$.

These three interpolants can be used to represent most all volumes which are needed. The implementation of these will be discussed in Chapter 5.

Higher Order Expressions

If in addition to positional data derivative data is available then all of the previous equations may be extended to reflect the higher order. Alternatively higher order equations may be developed if additional positional data within the body are given. In the area of design the next higher order expressions of interest to many are the cubics. Cubics provide C^1 continuity between segments of piecewise defined curves, surfaces and volumes. For the cubic univariate case six pieces of data at each end point would be provided. In the linear case, equation (3.6), only three pieces of data, the coordinates, at each end point were supplied. The positional data will be called $F(i)$, and the derivative data $F'(i)$. The i will take on values of 0 and 1 depending upon which endpoint is being used. Equation (3.6) is often written as

$$P_1 F(r) = \sum_i A_i(r) F(i) \quad (3.25)$$

where the $A_i(r)$ is the blending function. The expression for the cubic projector is

$$P_1^3 F(r) = \sum_i A_i(r) F(i) + \sum_i B_i(r) F'(i) \quad (3.26)$$

in which $A_i(r)$ is the blending function for positional data and $B_i(r)$ is the blending function for the derivative data.

Likewise if derivative data is available for the curves defining

the boundary of the surface, equation (3.11), the Boolean sum interpolant may be expanded. The interpolation properties of the Boolean sum will also increase to include the derivative data. In a similar fashion the trivariates may be extended.

The next chapter presents the finite element method. The high level model will be used to derive the input for analysis by this method. The method makes several assumptions about the model which need to be taken into consideration when models for analysis are generated.

CHAPTER 4

FINITE ELEMENT METHOD

Introduction

This chapter considers the formulation of a displacement based finite element method. While other bases have been developed, this was the first and is the most commonly used. This displacement based formulation considers a body divided into numerous sub-domains. Within each sub-domain, or finite element, an approximation to the displacement field is made. From these displacements the strains and stresses are then calculated. The development of the isoparametric element is briefly given along with its numerical integration. The optimum geometry, or placement of nodes, for a given number of degrees of freedom is also derived. Alternatives to this optimum are discussed in the last section.

Formulation

For a linear elastic material the expression for potential energy may be written as

$$PE = I_p + E_p \quad (4.1)$$

where I_p is the potential energy due to the internal work and E_p is the potential energy due to the external work. We assume, in the initial state, that the stresses and strains are zero. When the external loads are applied, the body will deform so work has been done on the body by these external loads. As deformation of the body takes

place internal work or strain energy is stored within the body. The expression for this internal energy is

$$I_p = 0.5 \int_V [e]^T [T] dV \quad (4.2)$$

in which

[e] is the column vector listing of the independent components of the strain tensor,

[T] is the corresponding listing for the stress tensor,

and V is the body's volume.

The expression for the external work, E_p is

$$E_p = - \int_V [u] [F_b] dV - \int_S [u_s] [F_s] dS - \sum_i U_i F_i \quad (4.3)$$

in which

[u] is the displacement field,

[F_b] is the body force,

[u_s] is the displacement field due to the surface traction,

[F_s] is the surface traction,

[U_i] is the displacement field due to the concentrated load,

F_i is the concentrated load,

and S is the boundary over which the surface traction is applied.

Substituting (4.2) and (4.3) into (4.1) produces

$$PE = 0.5 \int_V [e] [T] dV - \int_V [u] [F_b] dV - \int_S [u_s] [F_s] dS - \sum_i U_i F_i \quad (4.4)$$

The principle of minimum potential energy states: The displacement fields satisfying internal compatibility, the boundary conditions, and equilibrium, have a stationary value of potential energy. Further if the stationary value is a minimum then the equilibrium is stable [54]. So the solution to this variational

problem is obtained by finding a displacement field $[u]$ which minimizes the potential energy. We will approximate the displacement field $[u]$ by a function $[H]$ and a set of parameters $[U]$ (normally a set of nodal displacement components) or

$$[u] = [H][U] \quad (4.5)$$

This should be recognized as the classical Ritz approach to the solution of variational problems [70]. The $[H]$ function will be an interpolant between displacements calculated at a discrete set of points. It is also called a shape or blending function.

Two other relationships are needed to solve the problem. The first is the stress strain relationship or

$$[T] = [C][e] \quad (4.6)$$

in which $[C]$ is a square symmetric (for linear elastic materials) constitutive matrix, also called the elasticity matrix. Also required is the relationship between the element strains and the displacement field in the form

$$[e] = [B][U] \quad (4.7)$$

in which $[B]$ is a matrix of linear operators and algebraic expressions.

Substituting (4.5), (4.6), and (4.7) into (4.4)

$$\begin{aligned} PE = & [U]^T 0.5 \int_V [B]^T [C] [B] dV [U] - [U]^T \int_V [H]^T [F_b] dV \\ & - [U]^T \int_S [H]^T [F_s] dS - \sum_i U_i^T F_i \end{aligned} \quad (4.8)$$

If the body is then divided into m finite elements, an energy equation for each element can be written. The total potential energy for the body would then be the sum of all elemental energies or

$$\begin{aligned}
 PE = & [U]^T \sum_m \int_v [B]^T [C] [B] dV [U] - [U]^T \sum_m \int_v [H]^T [F_{bm}] dV \\
 & - [U]^T \sum_m \int_s [H_m] [F_{sm}] dS_m - \sum_i U_i^T F_i
 \end{aligned} \quad (4.9)$$

The [H] and the set of displacements [U] must be chosen such that element compatibility is achieved and the approximation converges [9,99].

To obtain a minimum potential energy implies that the partial derivatives with respect to the parameters U are equal to zero, that is

$$\frac{\partial PE}{\partial U} = 0 \quad (4.10)$$

which applied to (4.9) is

$$\begin{aligned}
 \frac{\partial PE}{\partial U} = & \sum_m 0.5 \int_{v_m} [B] [C] [B] dV [U] - \sum_m \int_v [H_{bm}] [F_{bm}] dV \\
 & - \sum_m \int_s [H_{sm}]^T [F_{sm}] dS_m - \sum_i F_i = 0.0.
 \end{aligned} \quad (4.11)$$

Rearranging (4.11)

$$\begin{aligned}
 \sum_m 0.5 \int_{v_m} [B] [C] [B] dV [U] = & \sum_m \int_{v_m} [H_m] [F_{bm}] dV_m \\
 + \sum_m \int_{s_m} [H_{sm}]^T [F_{sm}] dS_m + \sum_i F_i.
 \end{aligned} \quad (4.12)$$

Letting the right hand side be equal to [R] which we call the applied loads, and setting

$$[K] = \sum_m 0.5 \int_{v_m} [B_m]^T [C] [P_m] dV_m \quad (4.13)$$

then

$$[K][U] = [R] \quad (4.14)$$

in which [K] is known as the global stiffness matrix.

The steps required to perform finite element analysis are

- 1) Divide the domain into m finite elements

- 2) Determine the applied loads [R]
- 3) Using material properties calculate the element stiffness
- 4) Combine the element stiffnesses to form the global stiffness matrix
- 5) Solve equations (4.14) for [U]
- 6) Use equations (4.7) and the (4.6) to find the element strains and stresses.

This then is the basic procedure for the finite element method. The next section is concerned with one general family of elements and how they are used.

Calculating Element Stiffnesses

As shown in the previous section, to develop the element stiffness matrix $[K_e]$ the formula is

$$[K_e] = 0.5 \int_{V_e} [B_e]^T [C_e] [B_e] dV_e \quad (4.15)$$

where the e subscript denotes element related quantities. If the same interpolating function is used to represent the coordinates, as is used for the displacements (4.5), then the formulation is called "isoparametric" [46,62]. Isoparametric elements are the most commonly used elements for solving three-dimensional problems. Assuming we have isoparametric elements, and choosing to represent our element in a curvilinear coordinate system we could write

$$[x] = [H][X] \quad (4.16)$$

where the [X] are the nodal coordinates in the x,y,z coordinate space, and possibly other parameters, [H] is an interpolation function, and the [x] is a function of our coordinates r,s,t. Then dV may be

represented as

$$dV = dx dy dz = |J| dr ds dt \quad (4.17)$$

in which $|J|$ is the determinant of the Jacobian matrix (the transformation between derivatives in x,y,z and r,s,t spaces).

Substituting (4.17) into (4.15) we have

$$[K_e] = 0.5 \int_r \int_s \int_t [B_e]^T [C_e] [B_e] |J| dt ds dr. \quad (4.18)$$

If r , s and t are restricted to the range -1 to 1 and Gauss-Legendre quadrature is used to numerically perform the integration, then (4.18) can be written as

$$[K_e] = 0.5 \sum_i \sum_j \sum_k [B_e]^T [C_e] [B_e] |J| w(r,s,t)_{ijk} \quad (4.19)$$

where $w(r,s,t)_{ijk}$ is an appropriate weighting function [9,41]. In this expression the matrix $[B_e]$ and the determinant of the Jacobian matrix, $|J|$, are both functions of r,s,t , therefore we have

$$G(r,s,t)_{ijk} = [B_e]^T [C_e] [B_e] |J| \quad (4.20)$$

which, when substituted into (4.19) produces a formula for the element stiffness matrix

$$[K_e] = \sum_i \sum_j \sum_k G(r,s,t)_{ijk} w(r,s,t)_{ijk}. \quad (4.21)$$

The evaluation of the element stiffness is then reduced to calculating a summation of two functions.

Optimum Geometry

When a body is discretized into finite elements, the placement of the nodes is somewhat arbitrary. From experience it is known that some nodal configurations produce better answers than others for the same number of degrees of freedom [15,67]. We could think of the placement

of the nodes within a body as variable. Then (4.9) would be a function of both the displacement field and the nodal positions. For a true minimum potential energy to be achieved both of the following equations must be satisfied.

$$\frac{\partial PE}{\partial U} = 0 \quad (4.10)$$

and

$$\frac{\partial PE}{\partial X} = 0 \quad (4.22)$$

in which X are the nodal positions. Equation (4.10) produced (4.14) and (4.22) produces

$$\frac{1}{2} [U]^T \frac{\partial [K]}{\partial X} [U] - [U] \frac{\partial [R]}{\partial X} = 0.0. \quad (4.23)$$

We now have a system of nonlinear equations to solve.

The situation is similar to the chicken and the egg problem. To find the displacements we need to have specified the nodal positions. To solve (4.23) for the nodal positions we need to have the displacements. E. R. A. Oliveira was one of the first to work on this problem [76]. He found that the solution of (4.23) required the nodes to be placed on contours of constant strain energy, called isoenergetics. G. M. McNeice and others have been developing algorithms to accomplish this [68,94]. Basically they have used an iterative scheme. First they solve for the displacements with a regular mesh. Using the displacements they find the new nodal positions along the isoenergetics. Then they go back and solve for the displacements again. The loop may be repeated any number of times and the answers for each successive try will be better. The expense in solving for the new nodal positions is about equal to the solution

time for the displacements so each iteration costs twice as much as the usual analysis. This is for static problems in the elastic range [93]. D. J. Turke and G. M. McNeice suggest that a coarse mesh be first analyzed. The isoenergetics are then found and a finer mesh is generated [94]. These methods will all produce optimum geometry for the specific analysis problem. If the loading or boundary conditions change, so must the nodal coordinates. The next section considers an alternative to this procedure.

Geometric Merit Function

The idea of a geometric merit function is developed as an alternative to the more formal geometric optimization presented in the previous section. Here we want to consider criteria based solely upon geometry and thus independent of possible loading systems. This decoupling of the geometry from the loading is done because at the outset the engineer may not know what loads and boundary conditions he will impose upon the model. In fact, he will probably wish to consider several loading and boundary conditions during this phase of the design process. Since the isoenergetics are functions of the loads and the optimal placement of the nodes a function of the isoenergetics, the engineer will probably not want to generate a separate mesh for each test case. To take care of the modeling requirements we want a model for analysis which will produce good results independent of the loading. There have been several papers published which indicate that the geometry of an individual element has significant effect on the reliability of the answers [15,23,53,59,89]. Distorted elements, or elements departing from a rectangular shape, have been shown to give

very poor answers especially when the isoparametric formulation is used ([15] and references cited therein). We now will consider how the geometry affects the displacements calculated.

Since the approximated body can only be deformed into shapes which are superpositions of the displacement function terms, if the displacement function includes terms which coincide with the actual body deformation, there will be no error. Otherwise the potential energy minimum which we obtain will be higher than the actual potential energy of the body with resulting behavior of excessive stiffness.

R. W. Clough suggests that a measure of the stiffness of an element may be obtained from the eigenvalues of the element stiffness matrix [30]. The lower the eigenvalues, the more flexible the element. These eigenvalues represent the element deforming in its orthogonal modes of free vibration. For a four node planar quadrilateral element the stiffness matrix is an $[8 \times 8]$, having eight eigenvalues. These are shown in Figure 15. The first three eigenvalues are zero corresponding to the three rigid body modes associated with plane motion (i.e. in plane translations and rotation about an axis normal to the plane). The fourth and fifth are flexural modes, the sixth is a shear mode and the seventh is a stretching mode. The eighth and final is the uniform extension mode. Instead of calculating the eigenvalues for each stiffness matrix, Clough proposes that the sum of the eigenvalues be used as a measure of the element's stiffness. The sum of the eigenvalues is equal to the trace of the coefficient matrix in the eigenvalue problem. This trace is equivalent to the trace of the stiffness matrix (for homogeneous materials) because the matrices are

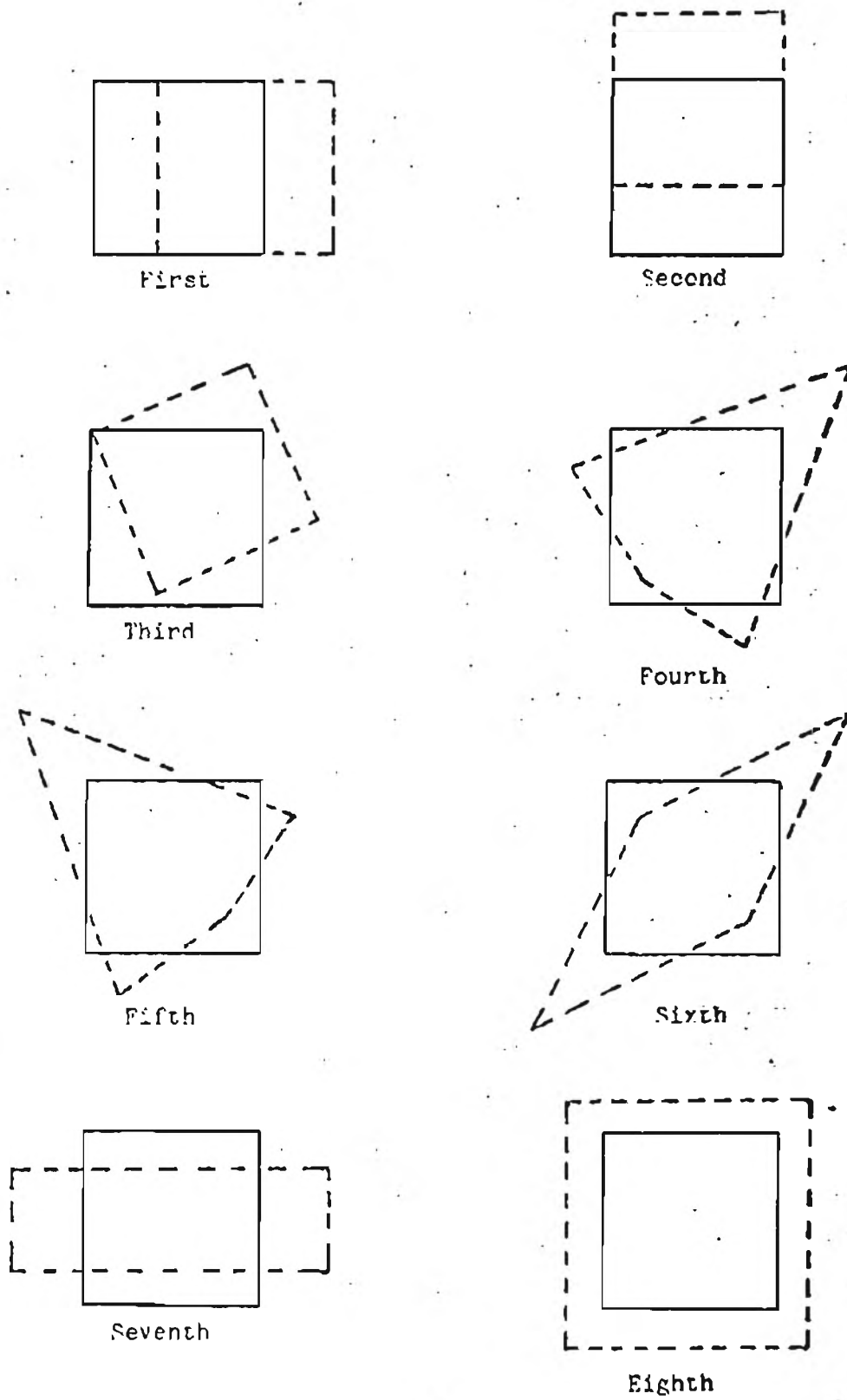


Figure 15. Eigenvalues of a 4 node quadrilateral element.

similar in a linear algebra sense and the trace is an invariant property of similar matrices (theorems 3.7.1 and 6.8.9 of [71]). Figure 16 shows an element in which the position of node number three is variable. The trace of the stiffness matrix as a function of the node's position is plotted in Figure 17. We see from Figure 17 that the stiffness of the element is at a minimum when the element is a square and it increases as the element's shape departs from the square.

This increase of the stiffness may be accounted for when the displacement field representation is considered. For linear isoparametric elements the assumption is that the displacements will vary linearly in the r and s directions. Whatever the shape in the x,y space, the element is always mapped to the square in r,s space, see Figure 18. To find the displacement functions in terms of the x,y space the mapping must be inverted. For the rectangular element in Figure 18 the displacement function will correspond to a linear function in our x,y space. When we consider the distorted element, the resulting function in x,y is very non-linear and unnatural for displacements of linear materials.

Figure 17 has shown that the trace of the stiffness matrix for the rectangular element has a lower value than the trace for the distorted element. If the trace of the stiffness matrix is used as the geometric merit function the analysis model can be checked and the very stiff elements found. Having a stiff element the location of the nodes defining the element can be modified to minimize this trace. The next chapter discusses how the high level model is used to produce initial nodal coordinates for the analysis model. The iterative

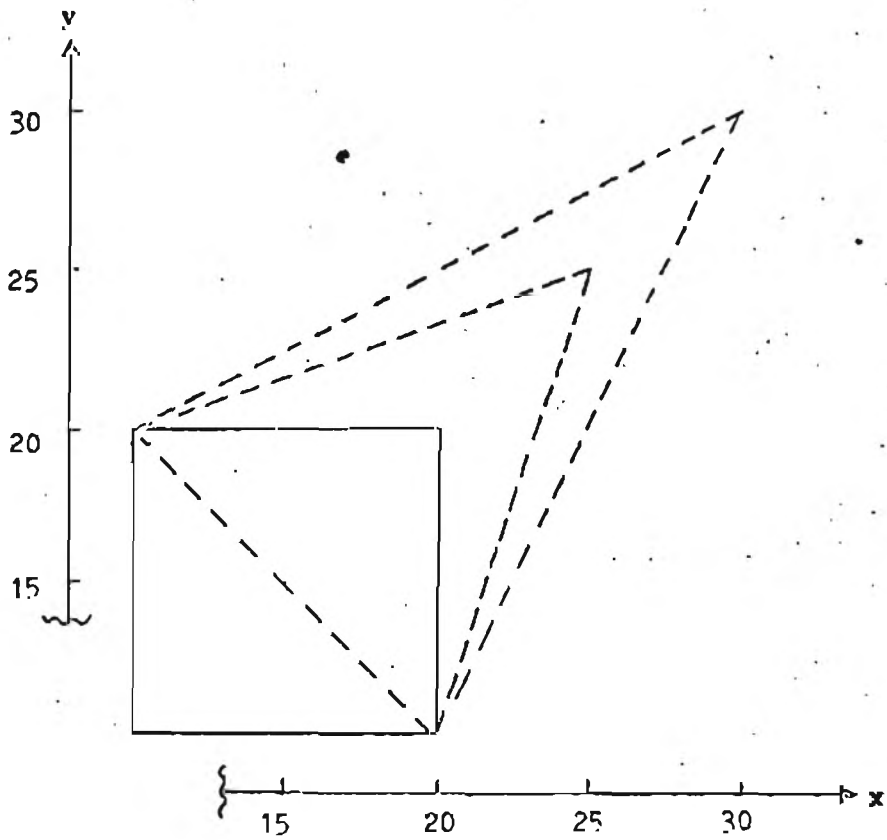


Figure 16. Quadrilateral element with floating node.

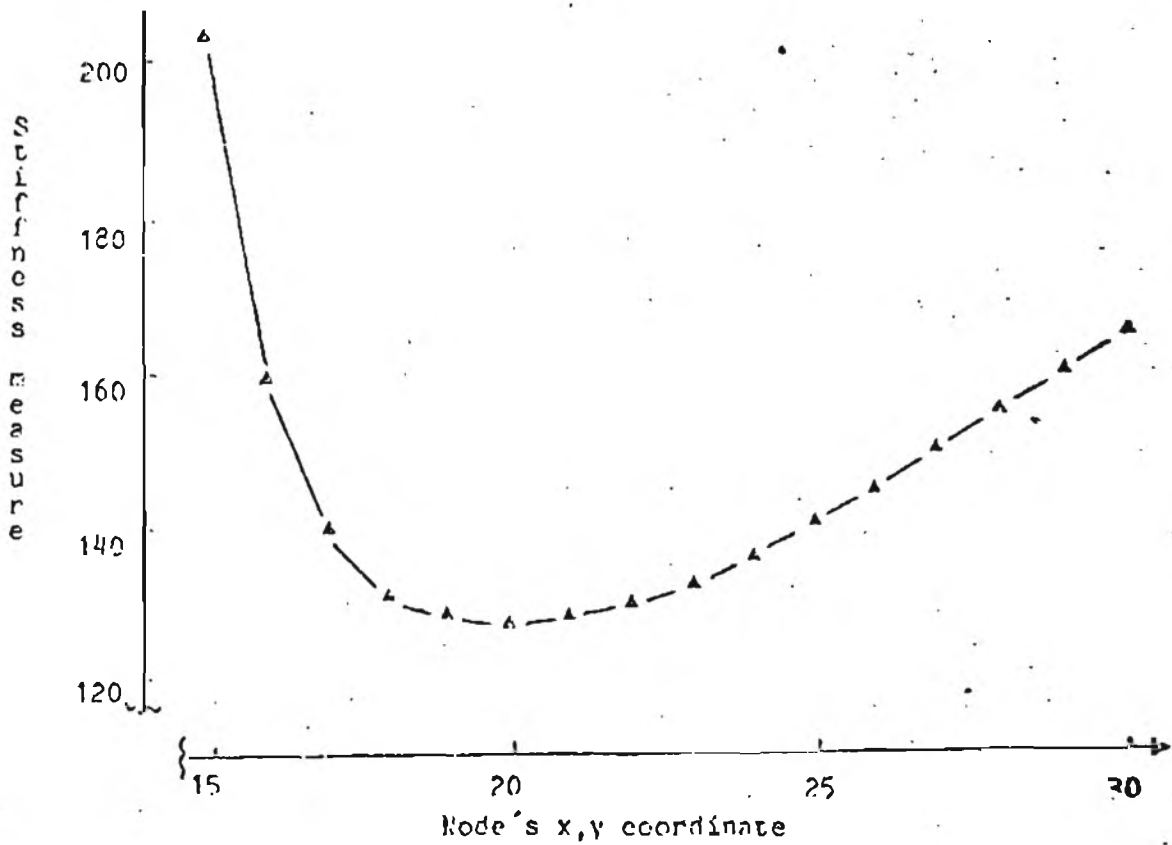


Figure 17. Trace of the stiffness matrix plotted.

procedure for model improvement by relocation of nodal positions is then defined.

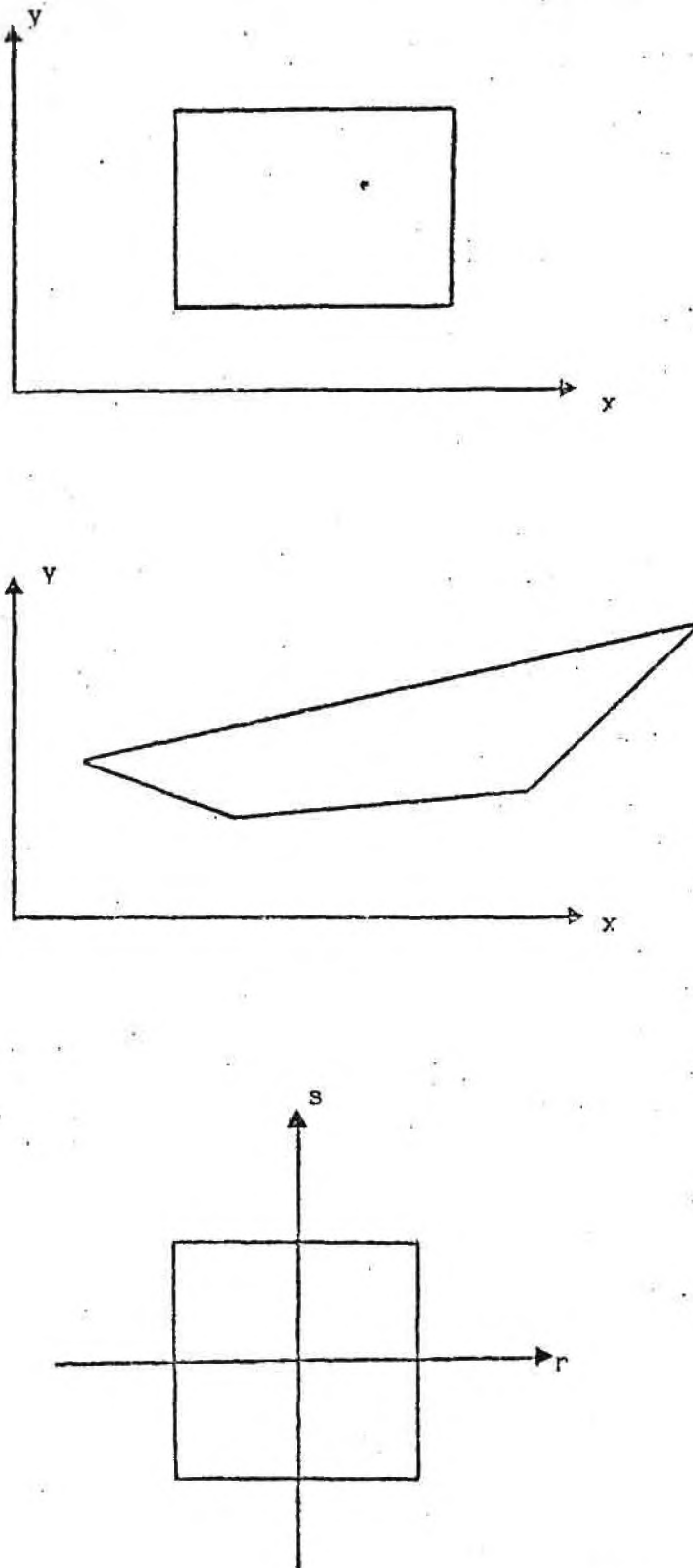


Figure 18. Mapping of elements in x, y space to r, s space.

CHAPTER 5

IMPLEMENTING THE SYSTEM

Introduction

In this chapter the high level model creation is discussed using the equations developed in Chapter 3. The high level model is used to produce the initial nodal coordinates in a uniform or non-uniform way. The numbering of the nodal coordinates contributes to the efficiency in obtaining the eventual solution and this effect is illustrated. The final section deals with the checking of the elements for their geometric merit based upon the criteria developed in the previous chapter. Having a check the nodal coordinates are modified to improve the element's geometric merit.

High Level Model

This section deals with the implementation of the high level model. The first type of volume representation discussed is the use of the component volumes. Next the more general free form volume representations are covered.

Component Volumes

Examples of the component volumes are spheres, cylinders, cones, and the like for which analytical expressions exist to describe positions within the volume. These expressions are usually given in a "natural" coordinate system, such as spherical coordinates or

cylindrical coordinates. To use these component volumes they need to be transformed into the parametric representations, equation (3.15). This will be demonstrated using the cylinder shown in Figure 19. The cylindrical coordinates are R (radius), THETA (angle), and H (height). We will express $Z(r,s,t)$ as simply

$$Z(r,s,t) = t H_{\max} \quad (5.1)$$

To develop $X(r,s,t)$ and $Y(r,s,t)$ we consider the circular cross section of the cylinder shown in Figure 20. We need to find an expression for the $X(r,s,t)$ and $Y(r,s,t)$ in terms of r and s , t will be constant for any one cross section. To do this we will use the equation of a circle in polar form and the Boolean sum interpolant (bivariate form) equation (3.11).

$$X(r,s,t_c) = (P_1 + P_2)F(r,s,t_c) \quad (5.2)$$

The c subscript on t is to remind us that t is constant. We recall that the expressions for the curves $F(0,s,t_c)$, $F(1,s,t_c)$, $F(r,0,t_c)$ and $F(r,1,t_c)$ are used in (3.12). For $X(r,s,t_c)$ they are

$$\begin{aligned} F(0,s,t_c) &= R_{\max} \cos(0.5\text{PI}(-s) + 1.25\text{PI}) \\ F(1,s,t_c) &= R_{\max} \cos(0.5\text{PI}(s) - 0.25\text{PI}) \\ F(r,0,t_c) &= R_{\max} \cos(0.5\text{PI}(r) + 1.25\text{PI}) \\ F(r,1,t_c) &= R_{\max} \cos(0.5\text{PI}(-r) + 0.75\text{PI}) \end{aligned} \quad (5.3)$$

The four positions (corner values) needed $F(0,0,t_c)$, $F(1,0,t_c)$, $F(1,1,t_c)$, and $F(0,1,t_c)$ may be found by substituting values of r and s appropriately set to 0 and 1 into the above expressions. The equation for $Y(r,s,t_c)$ is found by substituting the sine function for the cosine function in the above equations. This is just one of many ways which the cylinder could be represented. If the structure is more complex

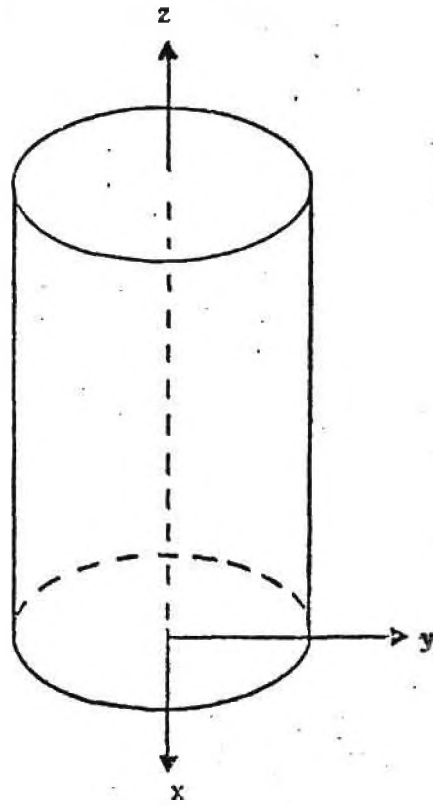


Figure 19. Cylinder in space.

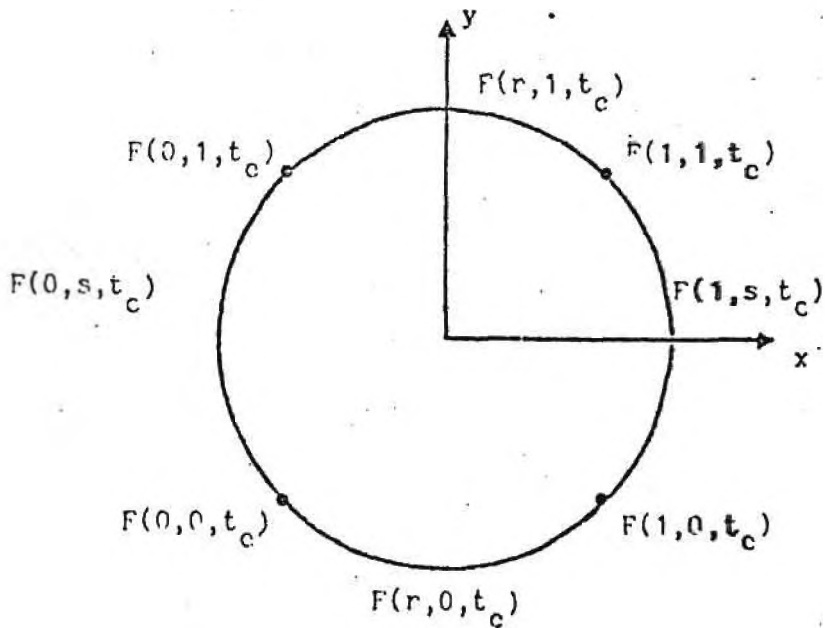


Figure 20. Circular cross section of a cylinder.

piecewise logical cubes to represent the object.

Other Volumes

If explicit equations do not exist to describe positions within the volume then we need to interpolate or approximate the data given. When only positional data is available then the trivariate should take the form of the tensor product (3.20). The cross fiber epoxy model model, Figure 3, could be developed by giving the twelve edge curves and then using the Coons volume equation (3.23) to interpolate them. The skull model, Figure 4, uses the trivariate Boolean sum (3.21). Each of the six surfaces was defined as a Bezier surface. The net of points for each surface consisted of 49 points (7 x 7). One of the six surfaces defining the skull in Figure 4, is shown in Figure 6 with its control net.

Generating Nodal Coordinates

Uniform

The volume has been parametrized in an approximately uniform manner if a uniform step of delta, taken anywhere in the body, produces an approximately uniform step in our x, y, and z space. The nodal coordinates can be generated by taking uniform steps in each of the r, s, and t directions, plugging these into our trivariate expression and calculating our x, y, and z coordinates. This is a major advantage of the parametric formulations, it provides a simple way to generate nodal coordinates. For example if uniform steps in the r direction are desired then an expression for r could be

$$r_{n+1} = r_n + dr \quad (5.4)$$

in which $dr = 1.0/(\text{number of nodes in } r - 1)$. The number of nodes calculated in each direction is under user control. If the elements generated are to be approximately cubes then the number of nodes should be a function of the arc length in each parametric direction. This also assumes that the arc length of all curves in the same parametric direction are of approximate equal length. When the parameterization is non-uniform then the next section's ideas can be used to obtain uniform spacing of the nodes.

Non-uniform

There are several ways in which a non-uniform spacing of nodes in x , y , and z may be specified using our r , s , and t coordinates. Probably the most common in mesh generators has been to make each succeeding nodal point spaced progressively further away or closer to, than the preceding nodal point [36,63]. An expression for r to achieve this is

$$r_{n+1} = r_n + (r_n - r_{n-1}) * FAC \quad (5.5)$$

in which FAC is greater than 1.0 to space the nodes further away, or less than 1.0 to space them closer.

J. I. Gill in his research on interactively designing bivariate surfaces uses the idea of an element density function [55]. The user of his system provides a value for the element density at each corner of the surface. The spacing of the nodes is then calculated by linearly interpolating between the corner values. When the density function is high in a region then more nodes and hence more elements are generated. This allows the user to specify more nodes in the regions where he thinks that high stress gradients will occur. Both of

these methods only allow for the nodal spacing to vary from small to large or from large to small. If the variation desired is small spacing then larger spacing and finally smaller spacing, or the inverse, these methods can not produce this directly.

The method implemented in this system was developed to allow for the above variation and the previous linear variation. We know that our r , s , and t can only vary from 0.0 to 1.0. We can define another set of variables r^* , s^* , and t^* which will always vary from 0.0 to 1.0 in a uniform manner, and a function to relate our r , s , and t to them. A graph of the relationship for r and r^* when it is linear is shown in Figure 21. If more elements were wanted when r was close to zero the function could look like that in Figure 22. For the small spacing, large, and small again, it would look like Figure 23. The ideal situation would allow the user to sketch in the relationship wanted. The program would then use the defined function and produce the required nodal spacing. For strictly two-dimensional problems this may be the best way to go because of the relatively large number of elements used. In three-dimensions we have to use fewer elements because of the large number of degrees of freedom for each element. A practical compromise between complete freedom and uniform spacing might be a cubic relationship. A planar cubic requires four pieces of data, four positions or two positions and two derivatives. If we choose the latter we could use the planar Hermite formula. The positions are already specified at (0,0) and (1,1) so the only input we require are the derivatives at each end point. The equation for r in terms of these derivatives and r^* is

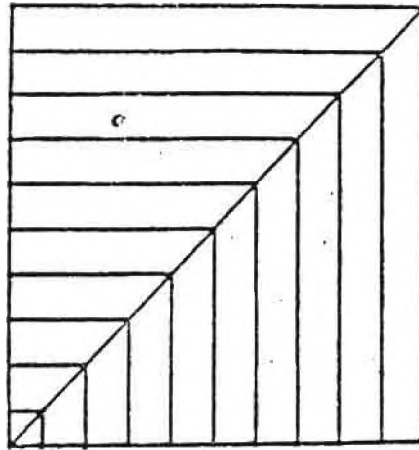


Figure 21. Plot of the function relating r and r^* with uniform spacing of the resulting nodes.

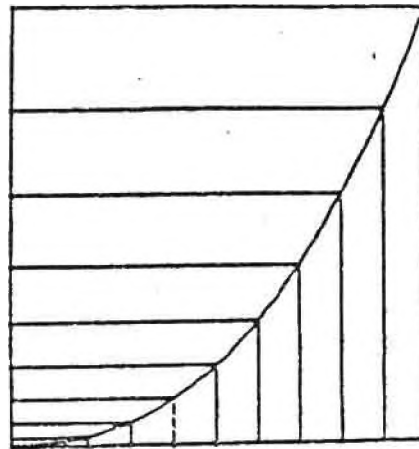


Figure 22. Plot of the function relating r and r^* with nodal spacing varying from small to large.

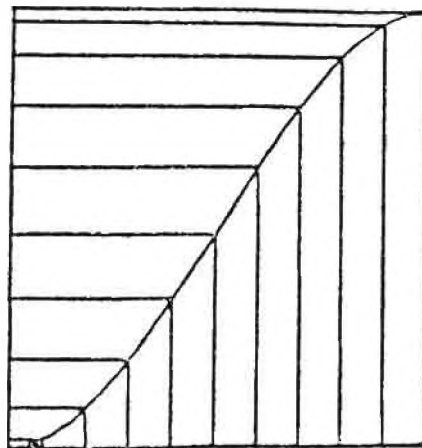


Figure 23. Plot of the function relating r and r^* with nodal spacing varying small, large and then small.

$$r = P_3^H F(r^*) = r^{*3}(F'(0)-2+F'(1)) + r^{*2}(-2F'(0)+3-F'(1)) + r^*F'(0). \quad (5.6)$$

For Figure 21 the $F'(0)=F'(1)=1$ and we have a constant (uniform) variation. If the derivatives are reduced from one towards zero the the nodal spacing is also reduced toward zero, Figure 23. The alternative is to increase the derivative beyond one and this increases the nodal spacing, Figure 23. Using such a scheme will allow a wide range of variation of the nodal coordinates in our body. The system implemented requires that the derivatives be specified for each direction r , s , and t at the corners, and no derivative specified is taken to mean 1.0, or uniform variation. Examples of this are presented in Chapter 6.

Generating Elements

Once the nodal coordinates are specified the required element lists may be generated. The stiffness matrix resulting from our model will be a sparse symmetric matrix. The number of zero and non-zero elements of the matrix is constant but their location within the matrix is a function of the node numbering. Being a sparse symmetric matrix if the non-zero elements can be kept close to the diagonal then only the banded portion of the matrix needs to be stored [9]. To store the whole matrix of order N we need N^2 locations. Taking advantage of the symmetry reduces this to $N^2/2$. If we have a banded matrix we will only need $N*B$ locations where B is the semibandwidth. To demonstrate this let us look at a two dimensional truss system, Figure 24. The semibandwidth for a truss system is given by

$$B = 2 + 2*ABS(MAXDIF) \quad (5.7)$$

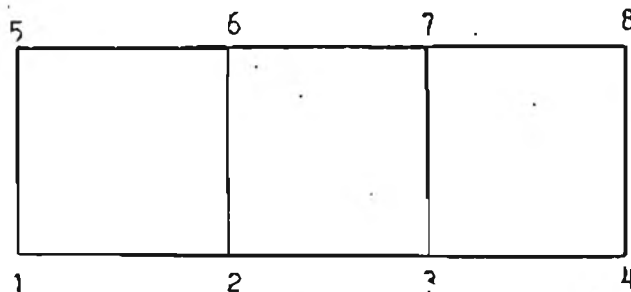


Figure 24. Two dimensional mesh numbered largest first.

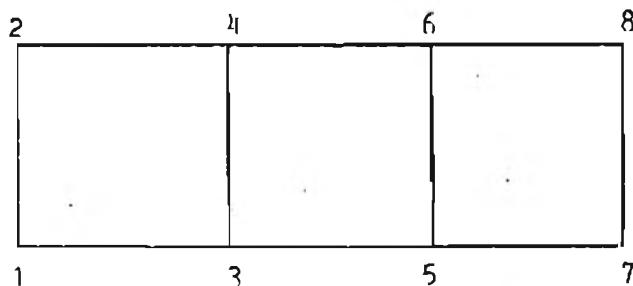


Figure 25. Two dimensional mesh numbered smallest first.

in which $MAXDIF$ is the maximum difference of node numbers defining an element [35]. For Figure 24 B is equal to twelve. If instead of numbering the nodes in the longest direction first we number in the smallest direction first then B is reduced to eight (Figure 25). Equations of a similar nature exist for models made up of two- and three-dimensional continuum elements. The general rule is: number nodes first in the smallest direction, then in the next smallest and finally in the largest direction. For many problems there may be no obvious way to number the nodes to reduce the bandwidth. In these

cases other programs have been developed to minimize the bandwidth by renumbering the nodes [32,39,40].

Checking for Geometric Merit and Improvement

First the previous schemes which have been developed and used will be considered. These are mostly for two-dimensional meshes but a few have their counterparts for the three-dimensional model. In Chapter 4 the idea of checking the stiffness of an element was developed. This is expanded upon and the implementation of the testing is discussed along with an alternative. Finally the actual movement of the nodes to minimize the stiffness is discussed.

Previous Schemes

There have been many schemes to check the geometry of a mesh before analysis and possibly adjust the nodes of the mesh when they are found to be bad. One of the most commonly used methods takes Laplace's equation for two (or three) dimensions and solves it by difference approximations. From the resulting equations, we get that a node's position should be the average of its four or six neighboring position values. Because the equation was solved by a difference scheme this only works for regular meshes and further it only works well for convex regions. To use the Laplacian scheme most implementations specify all nodal positions on the boundaries and then allow the interior positions to be calculated iteratively. There is no guarantee that the final positions will be within the specified boundaries, when they are outside it is known as overspill [63].

To correct some of the faults of the Laplacian scheme, A. E. Winslow used implicit differentiation to solve Laplace's

equation [98]. His scheme is called the "equipotential" method and works for many non-convex regions but it still requires a regular mesh. L. E. Herrman has another alternative, he has taken a linear combination of the isoparametric interpolant for bivariates and the differenced Laplacian to come up with a mesh generator [60]. The constants applied to each term are under the control of the user. Again this works well for convex regions.

O. C. Zienkiewicz has stated that since numerical integration is used for most element stiffness calculations, the determinant of the Jacobian transformation ($|J|$) should exist and be positive throughout the model [99]. R. Taylor in his FEAP (Finite Element Analysis Program) series of codes has a checking routine which calculates J at each node point of each element. If any are found to be zero or negative the program halts after checking the mesh. No provision has been made to try and fix the bad element once it has been found. In a recent paper by B. Mujerkij a measure of the element's warp called the Distortion Index was presented [70]. This value is zero for elements whose opposite sides are parallel and non-zero when they are not parrallel. This could also be used as a check for geomtric merit.

Implemented Schemes

In Chapter 4 the idea of a geometric merit criterion based upon the eigenvalues of an element stiffness matrix was developed. This has been implemented in the system. The procedure has been to generate the nodal positions, then check the mesh by calculating the trace of each stiffness matrix. Summing all of the traces we arrive at an overall measure for the stiffness of the model. To improve the model each node

is considered in turn and a location which would minimize the stiffness of the elements in which the node is used is found. Additionally the node's parametric values are checked to assure that they are not outside of the boundaries, so no overspill may ever occur. Since the high level model is kept around the boundary nodes are not considered as permanently fixed at their original location. During optimization they may be moved about the boundary.

The actual moving of the nodes to minimize the stiffness locally has been implemented using a Sequential Simplex Search method [11]. This method was used, as opposed to a conjugate gradient method, because analytical expressions for the derivatives of the trace of the stiffness matrix were too complicated to calculate. The simplex method needs no derivative data and is simple to program which made it very attractive. The basic idea is that for a function of N variables, $N+1$ sample points are chosen and the function we are minimizing is calculated at each point. For two dimensions the three sample points are taken to be the vertices of an equilateral triangle. For three dimensions a tetrahedron is used. The function of two variables to be minimized could be thought of as a surface. This surface is approximated by a plane (the equilateral triangle). The "most uphill" (largest value of the function) point of the triangle is found and then reflected about the other two points. It is hoped that this is in a downward direction. The process is then repeated for the new triangle just generated. There are several restrictions which must be met. The first is that no return may be made to a previously calculated point. Second if the best vertex (minimum value) remains the same for some predetermined number M of iterations, then the size

of the triangle is reduced and the process is repeated. This is the halting criterion, when the triangle size is smaller than the error tolerance, we stop. Some research has been done on what optimum value of M should be. The results have been expressed in the following equation

$$M = 1.65N + 0.05N^2 \quad (5.8)$$

where N is the number of dimensions [88]. The simplex scheme implemented does the optimization in the parametric space. When a boundary node is being considered, the optimization scheme is reduced by a dimension. This assures that the node will not be moved off the boundary. Because the optimization is done in the parametric space the overspill problem is eliminated by restricting the possible parametric values to always fall within the prescribed range. This is different from all other known schemes for modifying the node coordinates after they are generated. The other schemes work with nodes in the real space and they must keep the boundary nodes fixed.

After implementing the simplex scheme to minimize the trace of the element stiffness, it was found that the calculation of the element stiffness was relatively time consuming. A less expensive calculation was needed. If we examine the components of the stiffness calculation (4.19) we see a number of values multiplied by the determinant of the Jacobian matrix. We know that if we have a parallelogram in two dimensions, (see Figure 26) then the area of the parallelogram is given by

$$\text{AREA} = \partial\alpha \times \partial\beta = \begin{vmatrix} \frac{\partial\alpha}{\partial x} & \frac{\partial\alpha}{\partial y} \\ \frac{\partial\beta}{\partial x} & \frac{\partial\beta}{\partial y} \end{vmatrix} = |\partial\alpha| \cdot |\partial\beta| \cdot \sin e = |J| \quad (5.9)$$

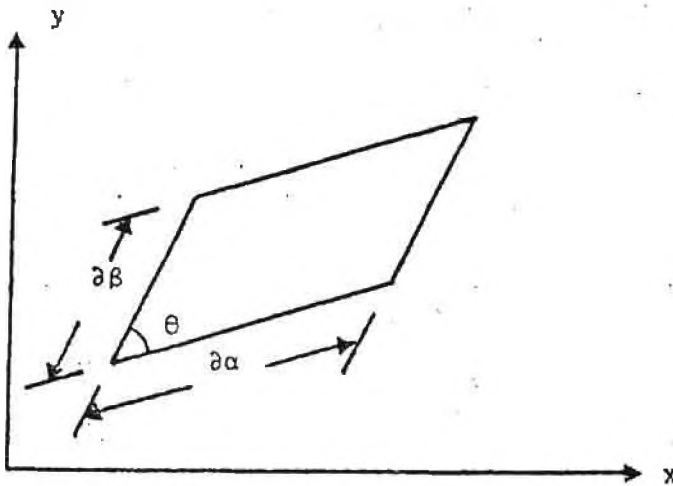


Figure 26. Parrallelogram.

Therefore the stiffness is a function of the angle.

In Chapter 4 the variation of the trace of the stiffness was demonstrated as a node was moved in space, (Figures 16 and 17). We repeated the calculation adding the calculation of the sine of the angle between the r and s axes at the Gauss point where the stiffness was calculated. The stiffness trace and the sine of the angle are both plotted in Figure 27. It clearly shows that the sine has a maximum at the same position in space of the varying node, as the trace has a minimum. The calculation of the sine is very easy and faster than the stiffness calculation so it also was implemented and used. The next chapter will demonstrate the results of using both schemes along with examples of the other features described.

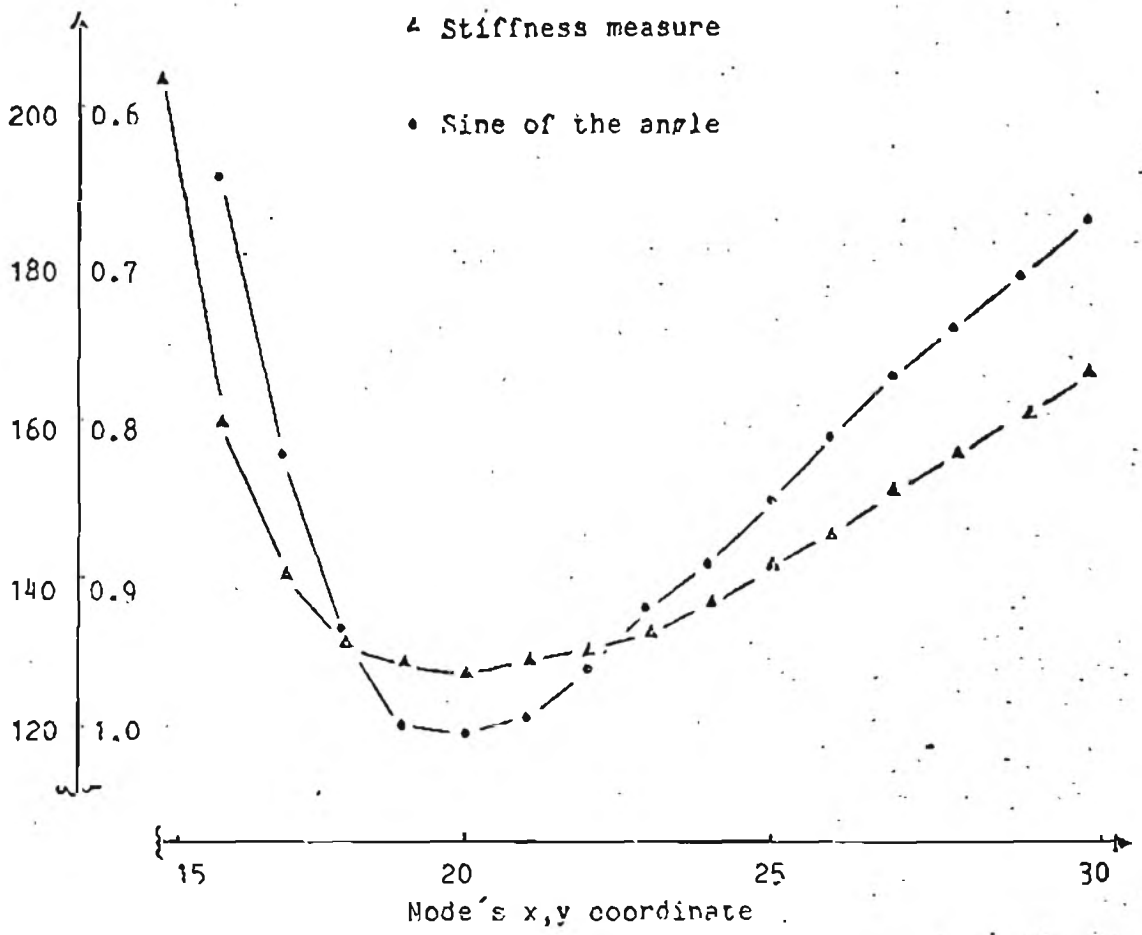


Figure 27. Trace of the stiffness matrix plotted with the sine of the angle.

CHAPTER 6

RESULTS AND CONCLUSIONS

Introduction

Two systems were developed as a result of this research. The first was a two-dimensional system, and the second was a two- and three-dimensional system. The results of the two-dimensional system are presented using two different models. The first of the two-dimensional models has boundaries defined by four cubic Bezier curves. The user specifies the polygon defining each of the curves as input to the high level model. The second model is after one used by R. E. Jones of Sandia Laboratory, in his documentation of QMESH [63]. This model is used to generate distorted elements to test the optimization schemes. The three-dimensional models used to test the second system were the skull-brain model (Figure 4) and the epoxy matrix from the crossed fiber model (Figure 3). These were chosen because the models have been used for actual analysis and had been generated by hand. The conclusions are then presented followed by a section suggesting directions for future work.

Two-Dimensional Examples

Bezier Curve Bounded Region

The curves used to define the region were shown in Figure 7. These were the input to the Boolean sum interpolant (3.12) that was used as the high level model. Figure 28 shows the discretization of

the model into 35 elements. Figure 29 shows the logical or parameter space for the model. The parametric space is shown so the user may see the distribution of nodes in the parametric directions. After optimizing the model this parametric space plot will help show where the nodal coordinates have been modified. The number of nodes and hence elements which may be generated is arbitrary. Figure 30 shows the high level model discretized into 77 elements. Both this and the model in Figure 28 were generated in a uniform manner.

Non-uniform element distribution is specified by supplying the program with the derivative information at the parametric corners. These derivatives are then used in the cubic equation presented in the previous chapter, equation (5.7), to generate the nodal positions. Figure 31 demonstrates this option using the 77 element discretization of the model. In the R direction the derivatives on $S=0$ were specified to be 3.0 and 0.1, on $S=1$ they were 2.0 and 2.0. The S direction derivatives were, along $R=0$, specified as 1.0 and 1.5, and the same along $R=1$. The distribution of the nodes in the parameter space is shown in Figure 32. We can see that very complex distributions of elements may be specified using these eight derivative values.

The 35 element model was then checked for geometric merit and the nodal coordinates adjusted to improve the model. Recall from Chapter 4 that a measure of the stiffness of an element could be calculated by taking the trace of the element stiffness matrix. A measure of the model's stiffness is obtained by taking the trace of the global stiffness matrix. For the model shown in Figure 26, the stiffness value was calculated as 182.5. If optimization is performed considering each node in turn and using the sequential simplex

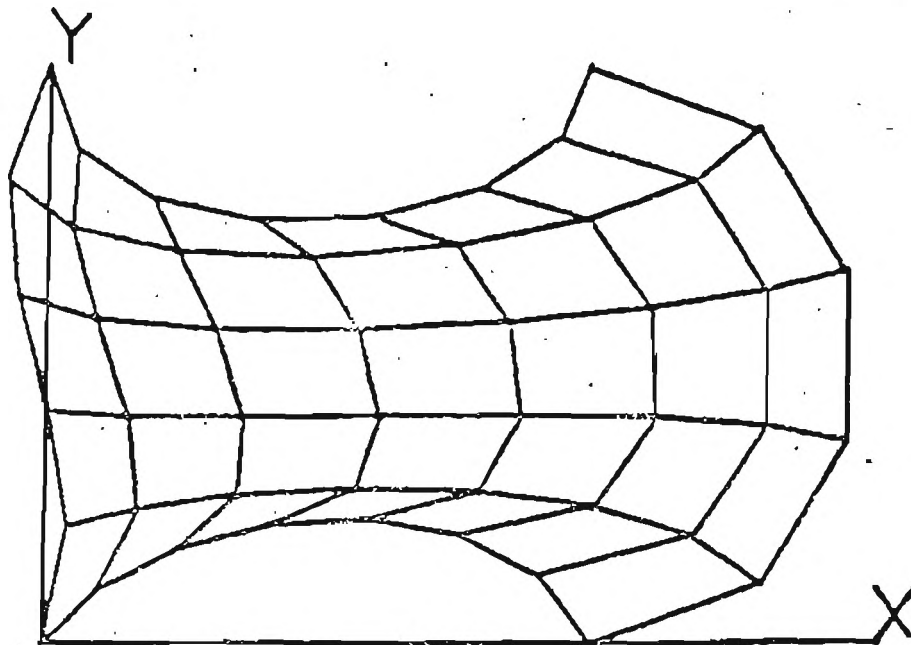


Figure 28. Bezier curve bounded region discretized into 35 elements.

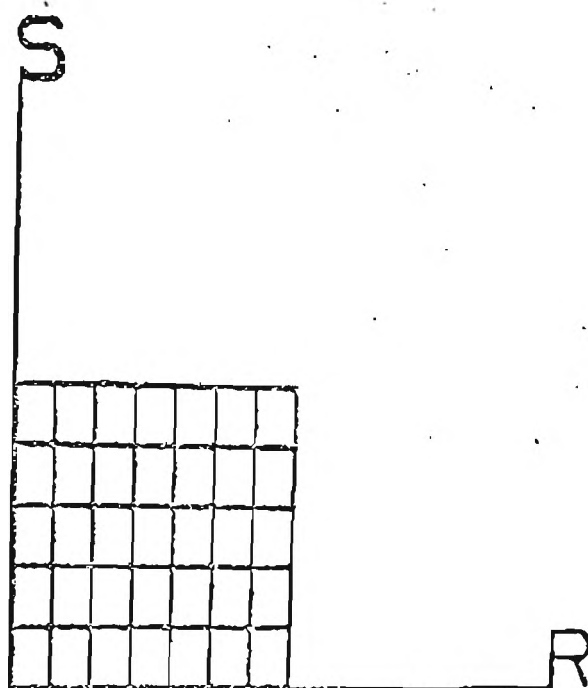


Figure 29. Parameter space of the model.

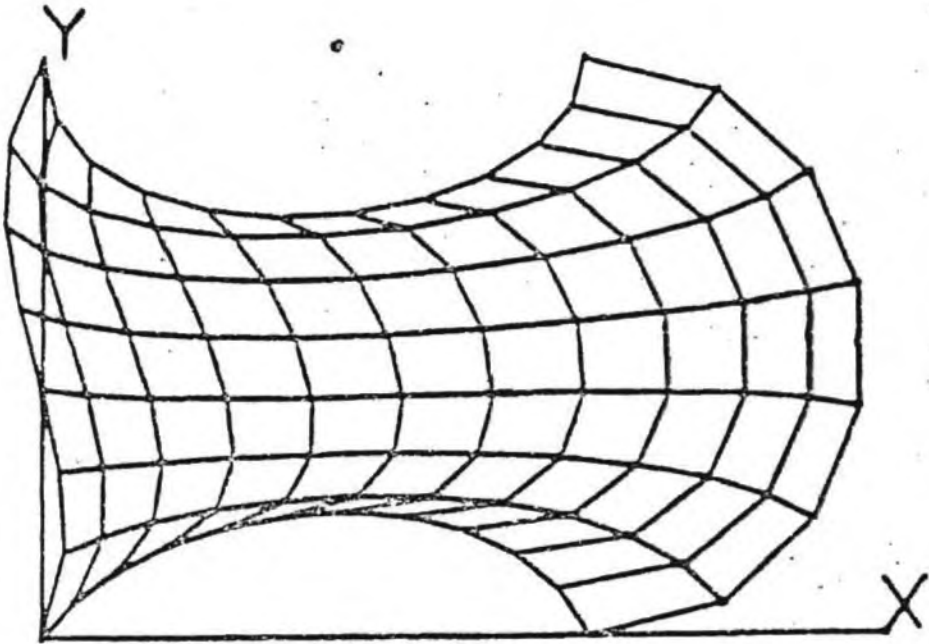


Figure 30. Region discretized into 77 elements.

algorithm to find the location of the node to minimize the trace, the value calculated is 130.5. This was after 15 iterations. Figure 33 displays the optimized mesh and Figure 34 shows the parameter space after optimization.

The process of optimizing the model using the trace of the stiffness matrix takes a relatively large amount of time. As developed in Chapter 5 an alternative to using the trace is to try to maximize the angle between the r and s axes in each element. Figure 35 shows the model after it has been optimized using the angle calculation. The value of the stiffness trace for this optimized model is 135.4

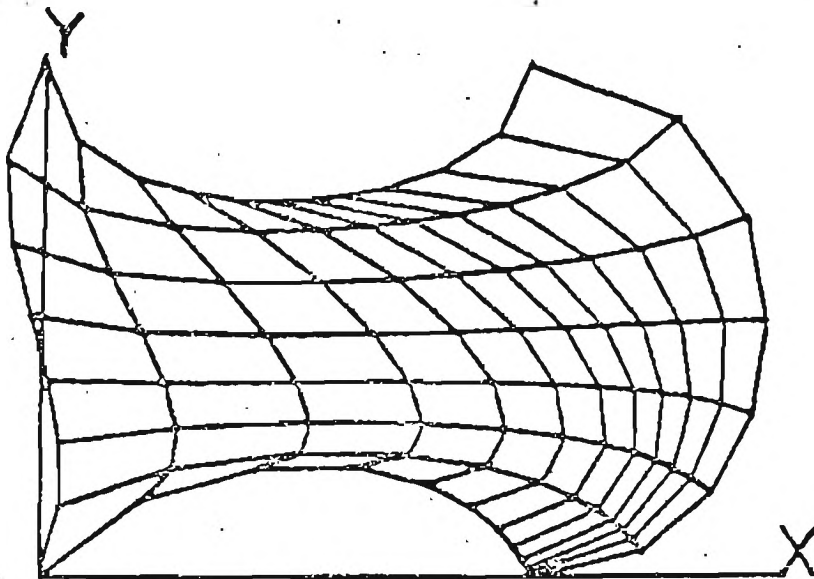


Figure 31. Non-uniform distribution of nodes.

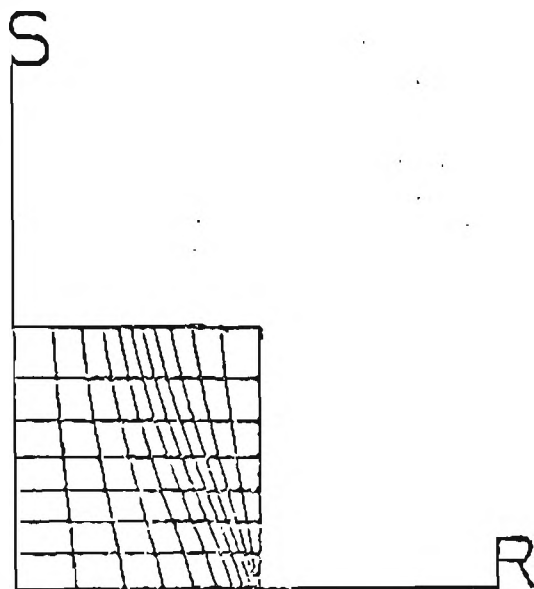


Figure 32. Parameter space non-uniform distribution of nodes.

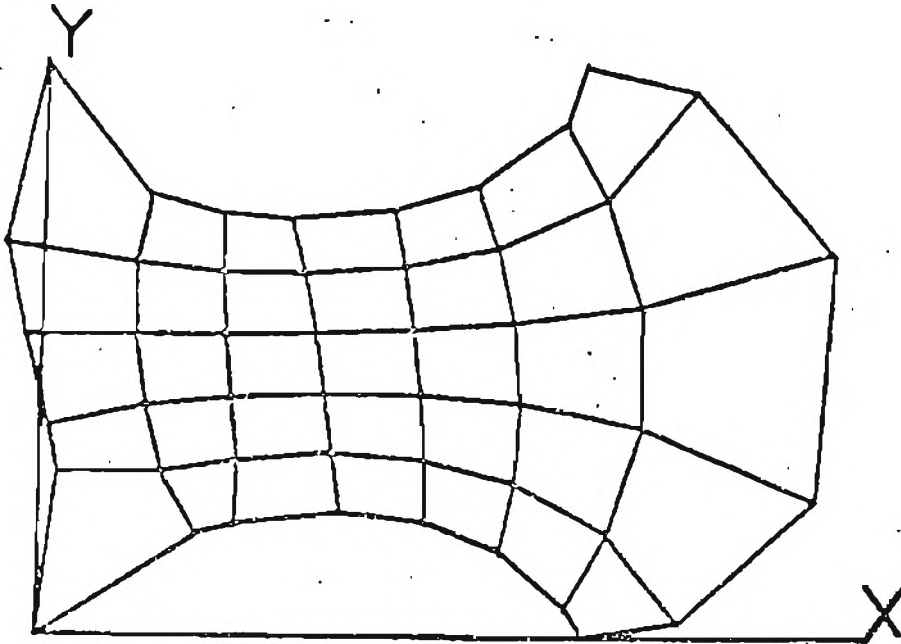


Figure 33. Optimized mesh for the 35 element model.

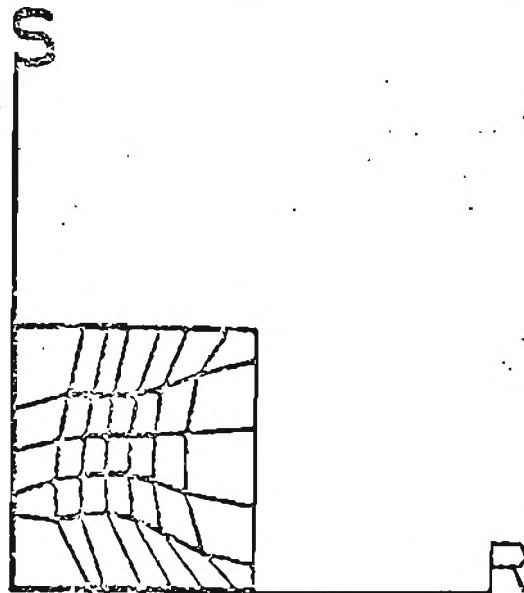


Figure 34. Parameter space for the optimized model.

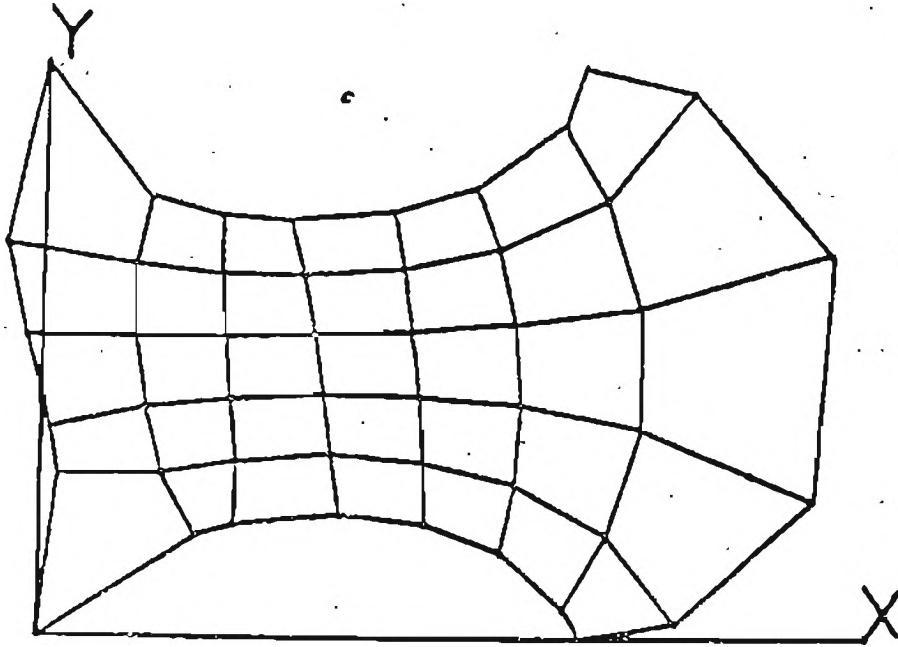


Figure 35. Optimized model using the angle calculation.

Second Example

The second example was defined by a series of straight lines and circular arcs. As was the case with the Bezier curve bounded region, these boundary curves were used in the Boolean sum interpolant (3.12) to form the high level model. Figure 36 shows these curves. The first discretization of the model was performed using points 1, 2, 3, and 4 as the parametric corners. Five nodes were specified in the R direction and nine nodes in the S direction. The resulting model contains 32 elements and is shown in Figure 37. The initial parameter space is displayed in Figure 38. Again the model could be discretized into any number of elements and Figure 39 shows a model derived using 96 nodes [8 x 12]. If the engineer thought that the mid-region of

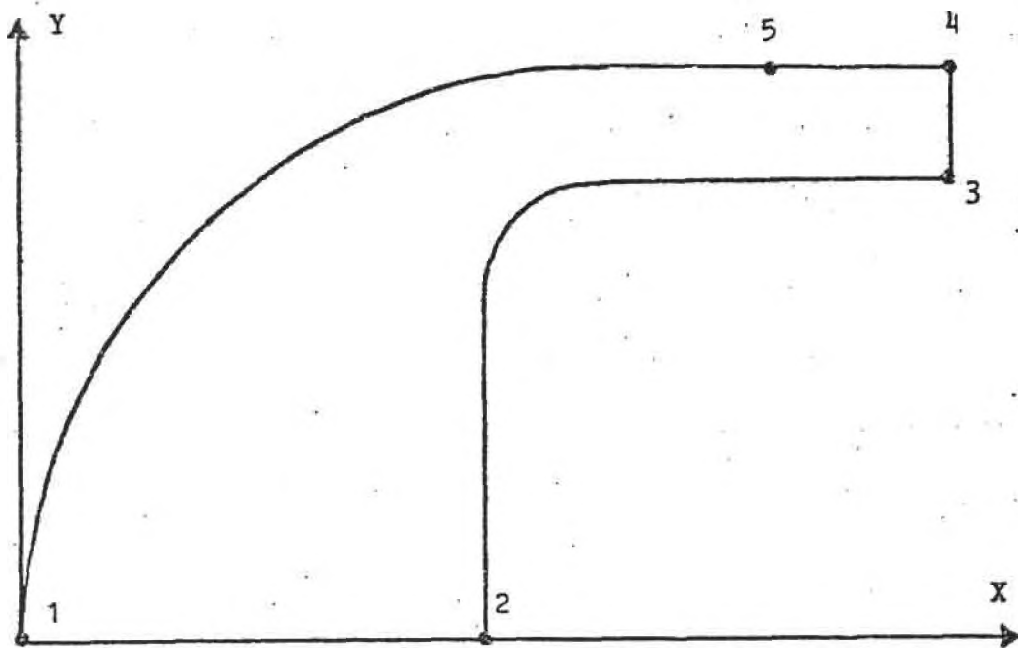


Figure 36. Boundary curves for the second example.

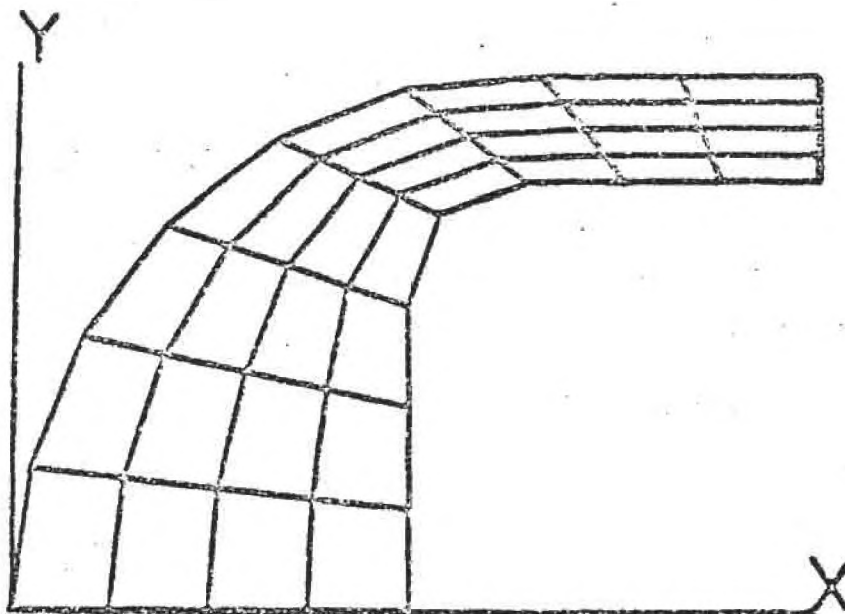


Figure 37. Second example discretized into 32 elements.

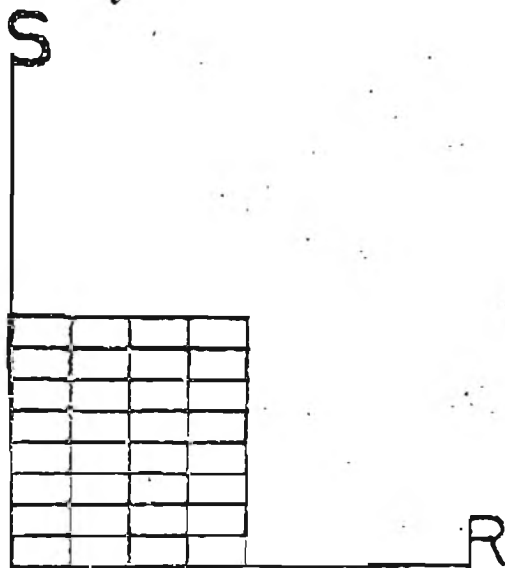


Figure 38. Parameter space for the second example.

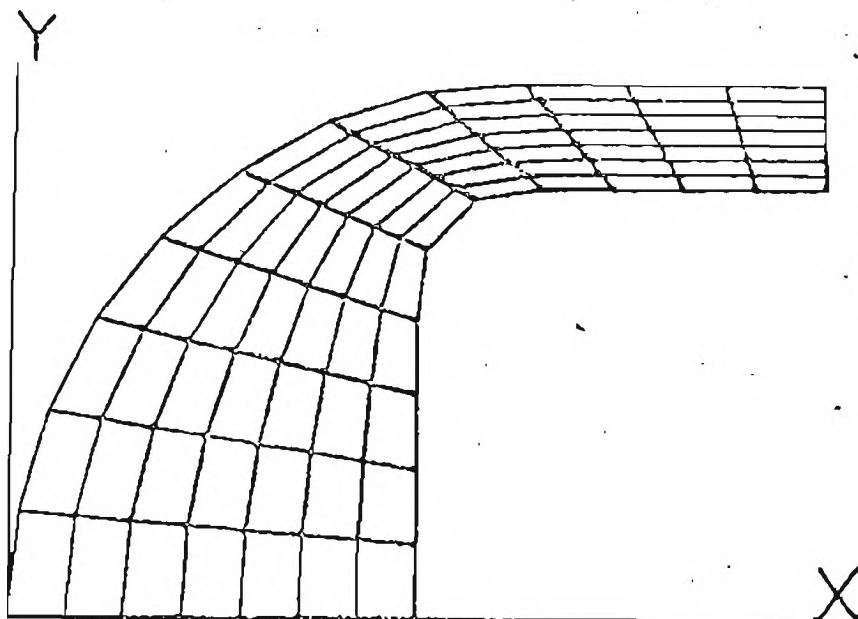


Figure 39. Second example defined by 96 nodes.

the body would have a higher stress gradient, after loading, then a non-uniform generation scheme could be used. The total number of elements will not be changed only the distribution within the body. The 32 element model was regenerated having smaller elements in the mid region and the result is shown in Figure 40. To obtain this distribution all four derivatives specified in the R direction were 1.0. In the S direction they were given as 2.0. The parameter space for the non-uniform nodal spacing model is displayed in Figure 41.

The optimization of the model was performed next. Figure 42 shows the uniform 32 element mesh after four iterations. The parameter space is seen in Figure 43. For this optimized mesh the stiffness measure value was calculated to be 179.9 down from 196.1 for the initial configuration, Figure 37. The graduated mesh may also be optimized. The initial value of the stiffness measure was 202.3, after four iteration of the optimization, (Figures 44 and 45) it was 184.8. The method used to optimize the model will result in the same final nodal configuration regardless of whether the starting configuration was specified uniformly or non-uniformly. Only when the optimization is halted prematurely will the initial configuration have any influence on the nodal coordinates.

If instead of using points 1,2,3, and 4 on Figure 36 as the parameteric corners, points 1,2,3, and 5 are used, then the initial mesh looks like Figure 46. The parameter space of this model is identical to Figure 38. This is a bad parameterization because it has produced several very distorted elements in the upper region. The distance between points 1 and 2 is about the same as the distance between points 3 and 5. If parameterization of the model is performed

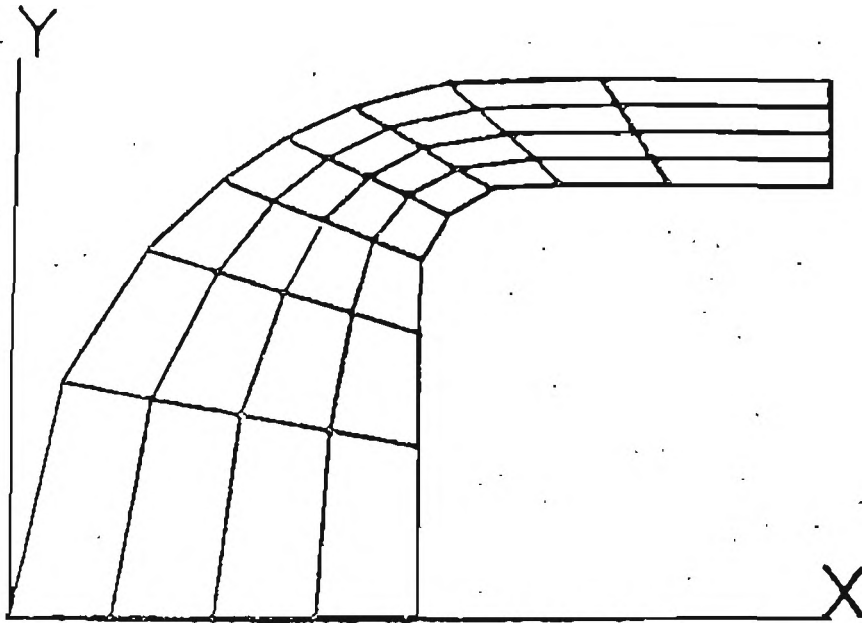


Figure 40. Non-uniform elements for the second example.

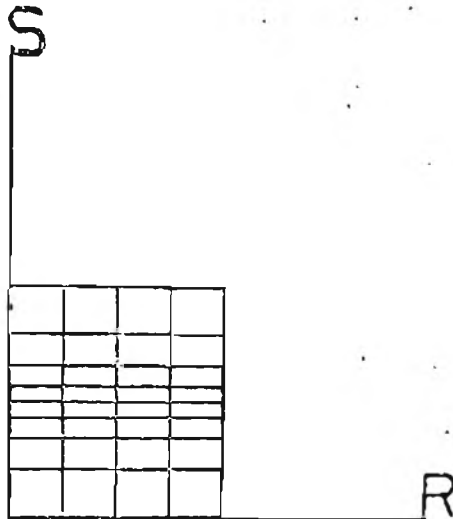


Figure 41. Non-uniform model's parametric space.

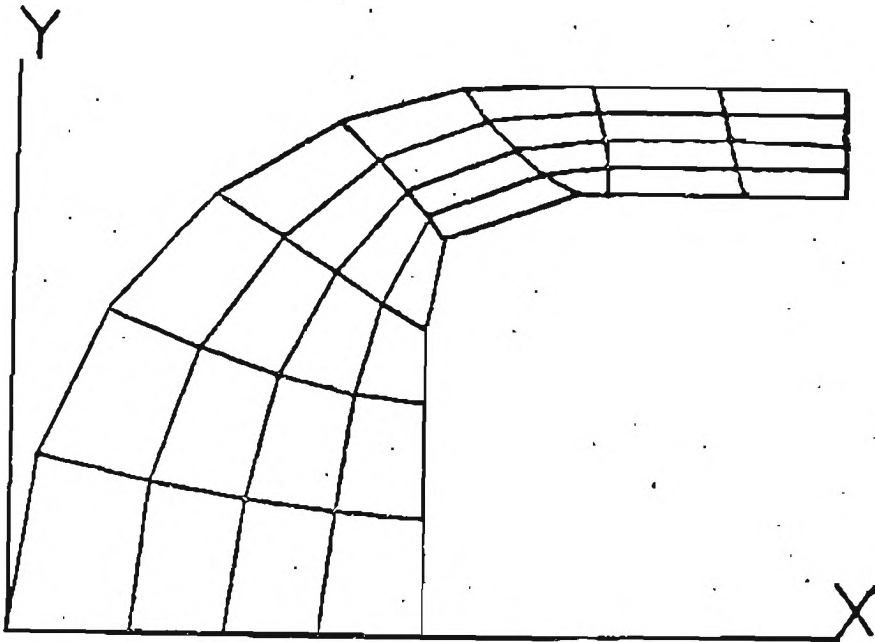


Figure 42. Optimized uniform mesh.

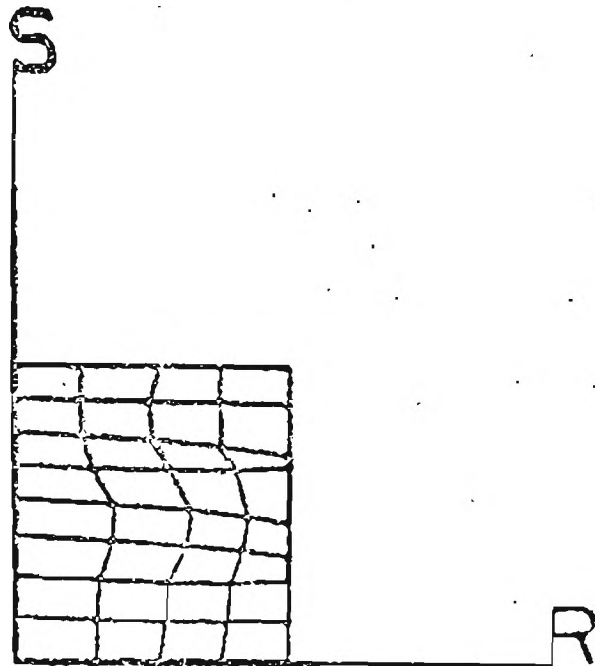


Figure 43. Optimized uniform mesh parametric space.

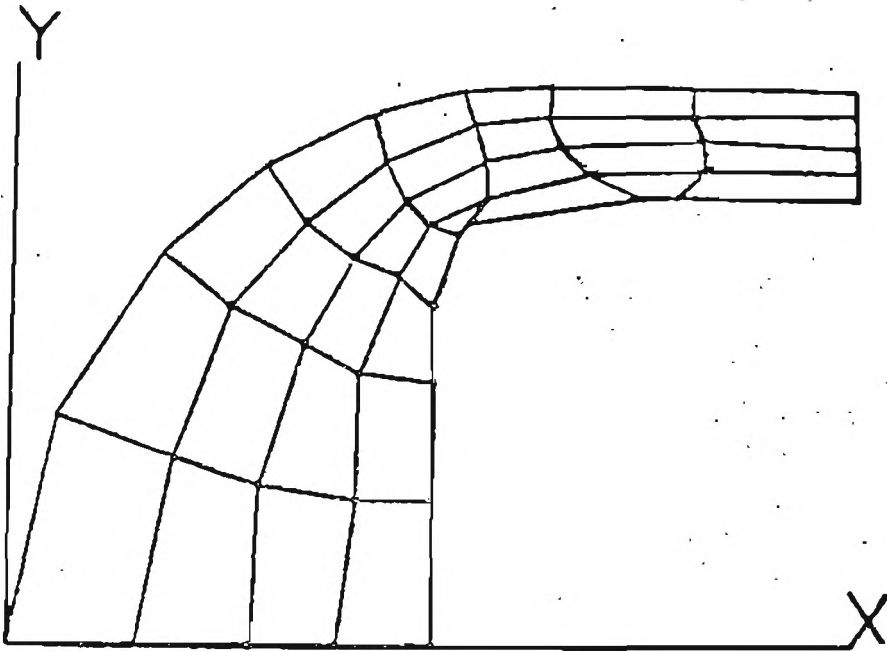


Figure 44. Optimized non-uniform mesh.

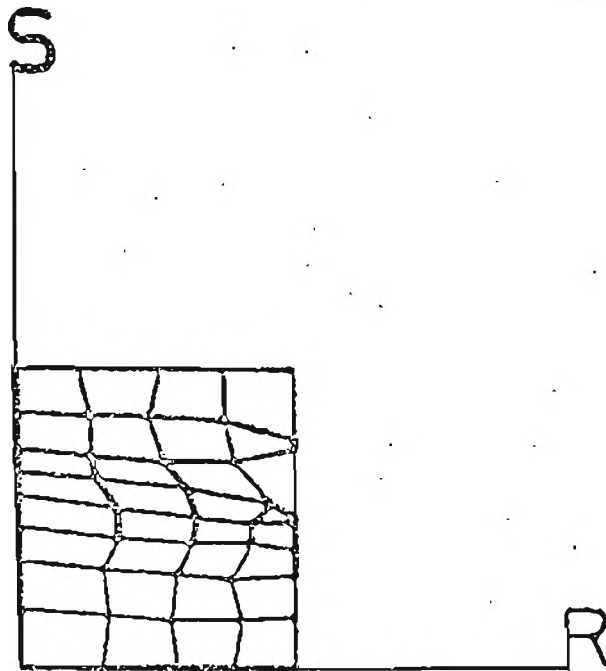


Figure 45. Parametric space for the optimized non-uniform mesh.

using arc length considerations then point 5 of Figure 36 may be made a parametric corner. This is one of the problems with automatic generation of models by programs which allow for no user interaction. The program may force a parametric corner to be in a location that does not correspond to a physical corner. Although, the user may also make the same type of mistake and specify a parametric corner in the wrong place. When working with two dimensions a parametric corner in the wrong place is easily found by looking at a plot of the model. Finding a parametric corner located at the wrong place in a three dimensional model may be more difficult to detect and to correct. Optimization of this mesh is possible and the initial value of the stiffness measure was 414.4. Figures 47 and 48 show the optimized mesh which has a stiffness measure value of 170.

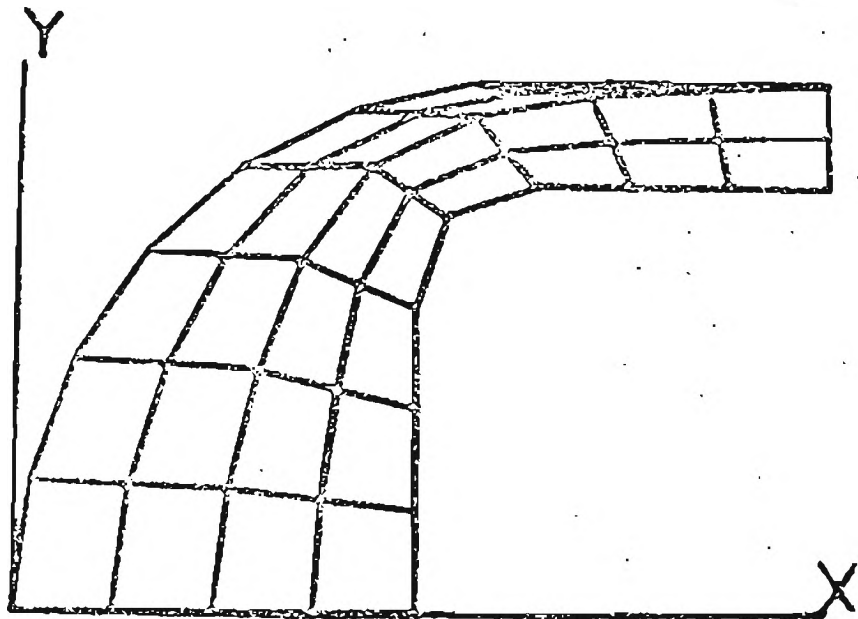


Figure 46. Mesh after parameterization by arc length.

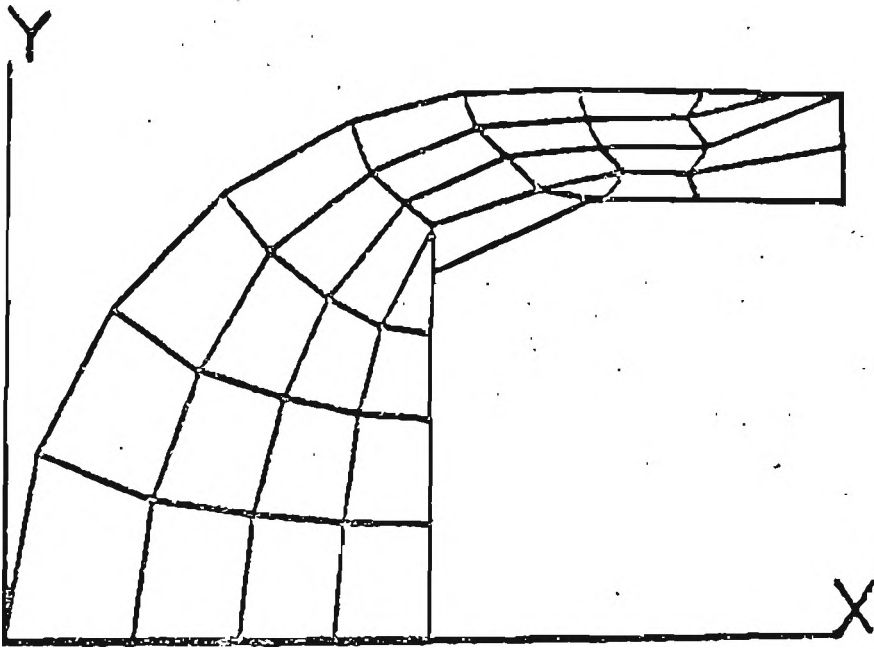


Figure 47. Optimized mesh of the model parameterized by arc length.

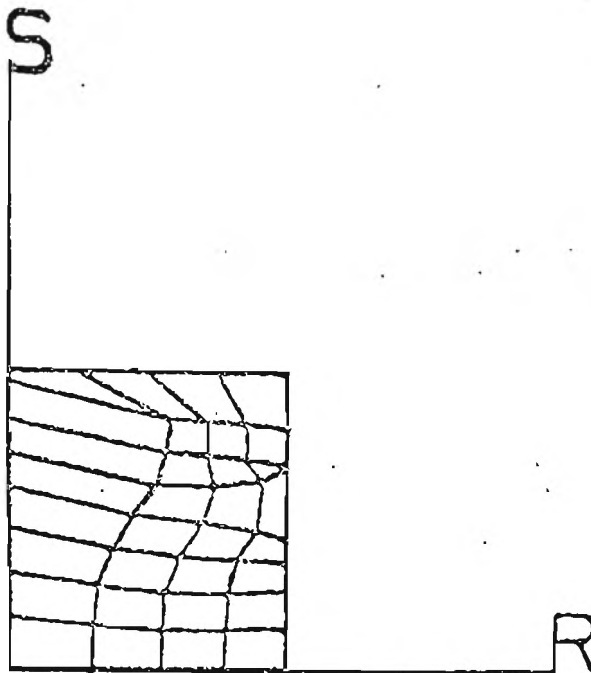


Figure 48. The parametric space for the optimized arc length model.

Actual analysis of this body has been done and the potential energies calculated for the models used. The body was loaded along the edge 3-4 and the nodes restrained between 1 and 2 of Figure 36. As the number of elements is increased the potential energy goes toward a minimum. Results of the analysis performed at the University of California Los Alamos Scientific Laboratory using the NON-SAP finite element program and discretizing the model into 1500 elements shows the potential energy to be -19.06. This will be considered the minimum. The potential energies were calculated for the various models at the University of Utah using SAPIV, and are summarized in Table 1.

Table 1.

Values of the Stiffness Measure and the Potential Energy
for the Second Example

| Model shown in Figure | Stiffness Measure | Potential Energy |
|-----------------------|-------------------|------------------|
| 37 | 196.1 | -14.08 |
| 40 | 202.3 | |
| 42 | 179.9 | -15.86 |
| 44 | 184.8 | |
| 46 | 414.4 | -3.48 |
| 47 | 170.0 | -7.36 |

The table clearly demonstrates that the answers produced by a given model can be improved if the optimization is performed. In his documentation of QMESH [63], R.E. Jones has used the same model and

done some optimization of the nodal coordinates. He uses the equipotential method to adjust the nodal positions of the model shown in Figure 46. The model optimized using the equipotential method has been analyzed and the potential energy was calculated to be -5.63. He then does some restructuring, that is removing some of the elements and adjusting the nodes, and the potential energy goes to -13.07. When compared to the equipotential method the optimization scheme used here produced a better mesh. Restructuring of the models was not attempted so no comparison can be made directly, but if the model is parameterized in a reasonable way, Figure 37 versus Figure 46, it may not be needed.

Three-Dimensional Examples

Skull-Brain Model

The skull-brain model was originally generated from a set of surface points whose coordinates were manually measured on an actual skull [87]. These points were used as input to a special model generator which had been written specifically for the skull-brain data. Whenever a different number of nodes defining the model for analysis was wanted, the procedure was to remeasure the coordinates. This process took from two to four weeks. The surface data points have been taken and collected into six sets, one for each surface of a parametric cube. Each set was then used as the control net for an approximating Bezier surface. Having a definition for each surface the trivariate Boolean sum interpolant was used as the basis for the high level model. The six surfaces defining the skull's exterior are shown in Figure 49. Each surface is displayed using one hundred points from

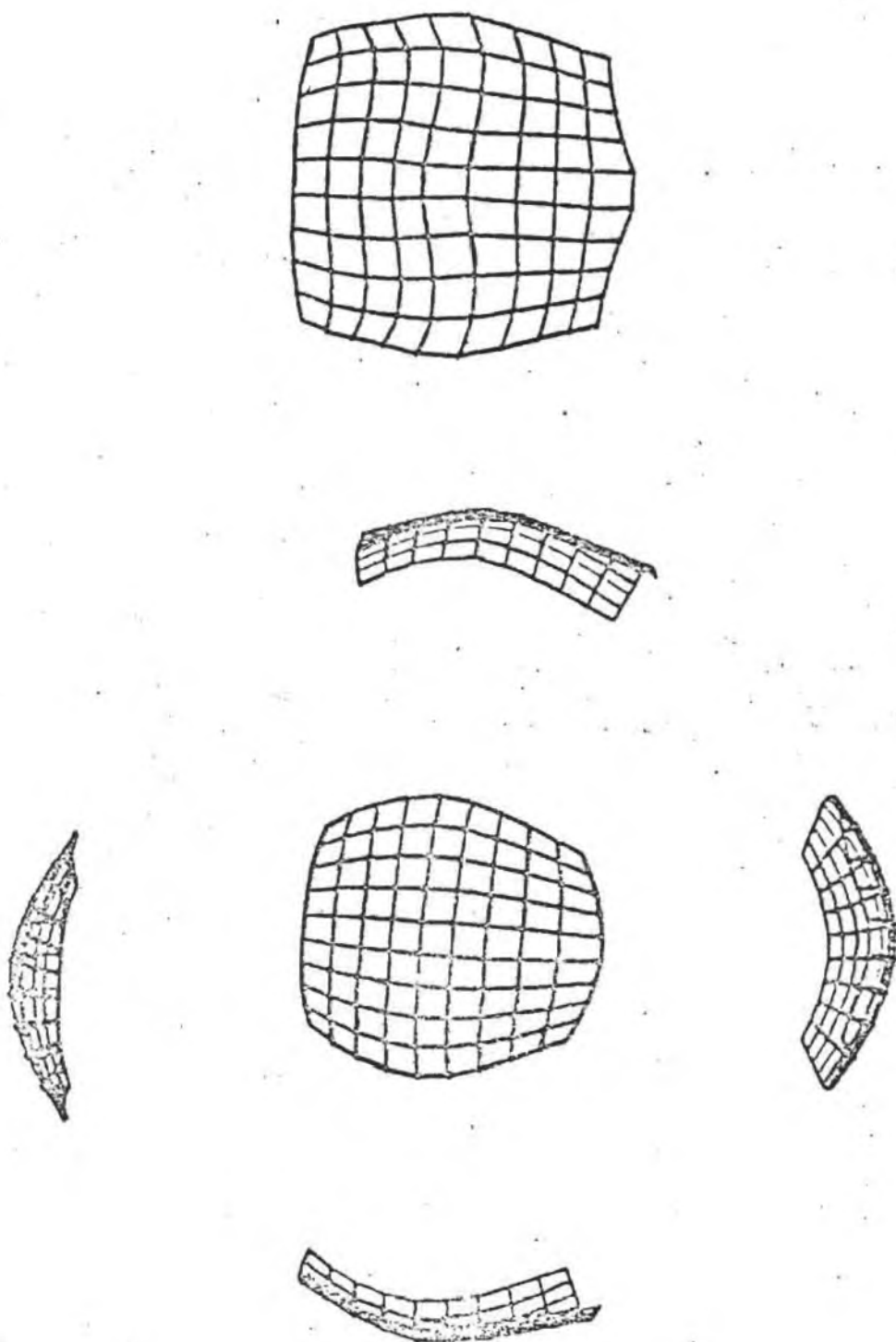


Figure 49. Six surfaces defining the skull's exterior.

the surface. The boundary curves of the surfaces were shown in Figure 14. A model for analysis was generated using six nodes in each parametric direction and is shown in Figure 50. Changing the number of nodes for a model in any of the parametric direction is easily done and a new model produced. Figure 51 shows three views of the skull-brain high level model discretized using eight nodes in the r direction, seven nodes in the s direction and six nodes in the t direction. The removal of the hidden portions are possible but not available on the present computer for use in real time. An alternative to aid in viewing the model is to display any one layer of bricks. The boundary curves are also drawn to provide a reference. Figure 52 shows several views of different brick layers.

Optimization of the model was performed and the results for the [6 x 6 x 6] model, optimized for four iterations, are shown in Figures 53 and 54. In Figure 53 we show the first layer in the t parametric direction. This is the bottom of the skull. The top image is the layer of bricks before the optimization was performed. The middle image shows the layer of bricks after optimization. We can see that the bricks are more orthogonal than they were originally. This is particularly noticeable around the boundaries. We see that the nodes on the surface have been adjusted so the faces of the hexahedrons are perpendicular to the boundary surface. The bottom image is the parametric space for the optimized layer. In Figure 54 we show the same sequence, but this time for the middle layer in the s parametric direction. This corresponds to the midsagittal plane of the brain.

No actual finite element analysis was performed on any of the three-dimensional models but the trace of the global stiffness matrix

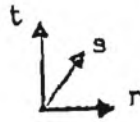
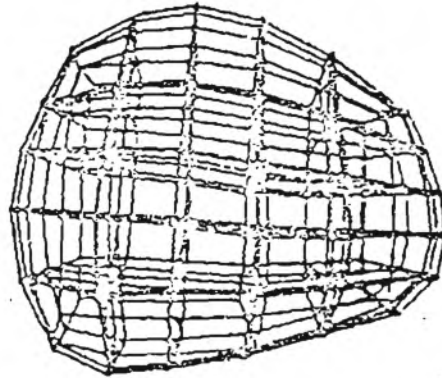
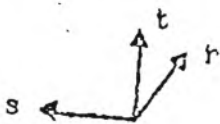
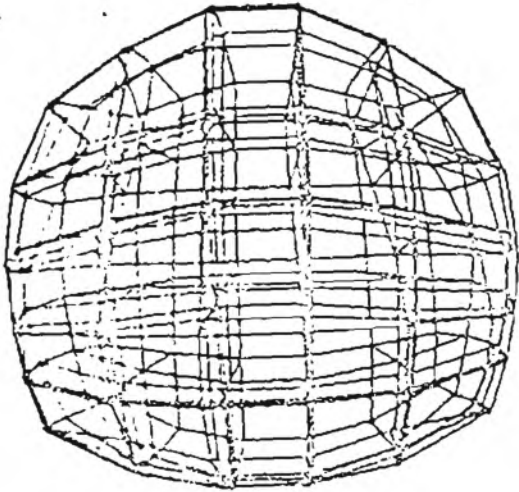
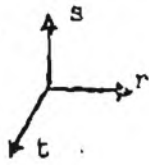
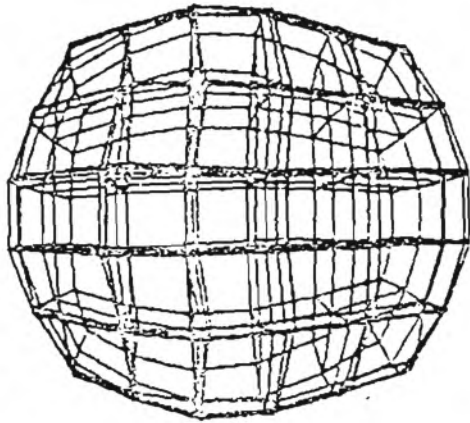


Figure 50. Skull-brain model [6 x 6 x 6].

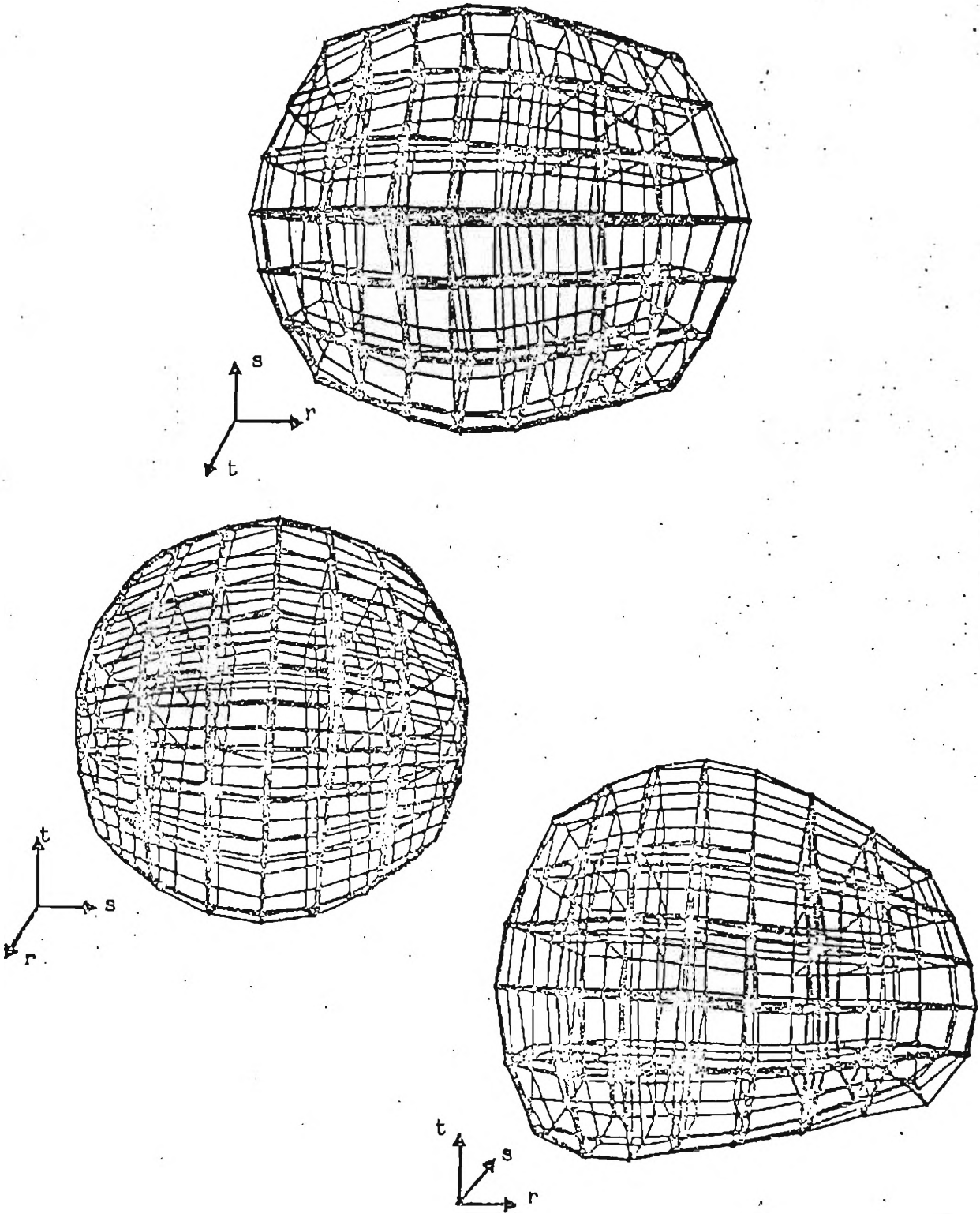


Figure 51. Skull-brain model [8 x 7 x 6].

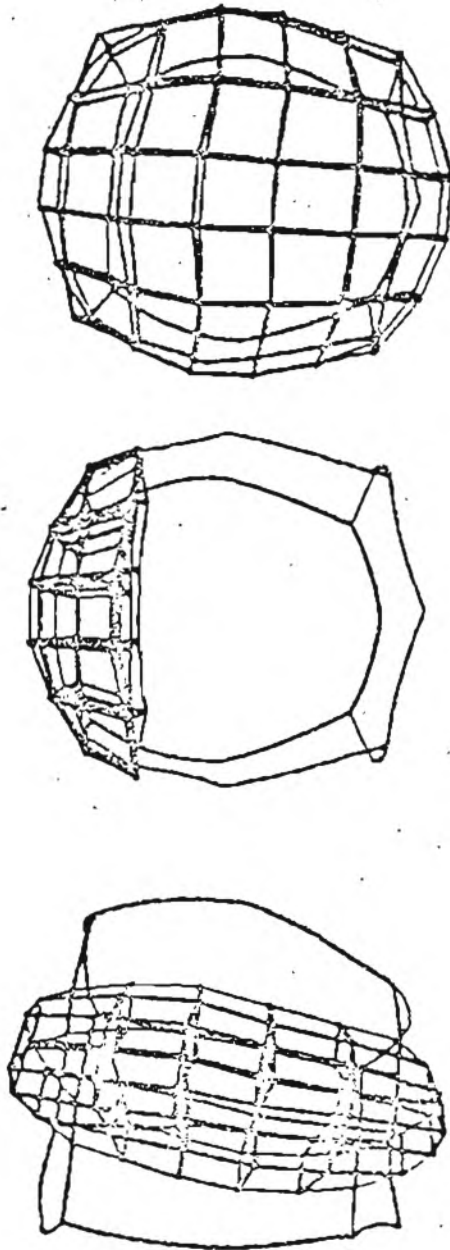


Figure 52. Several layers of bricks from the skull-brain model.

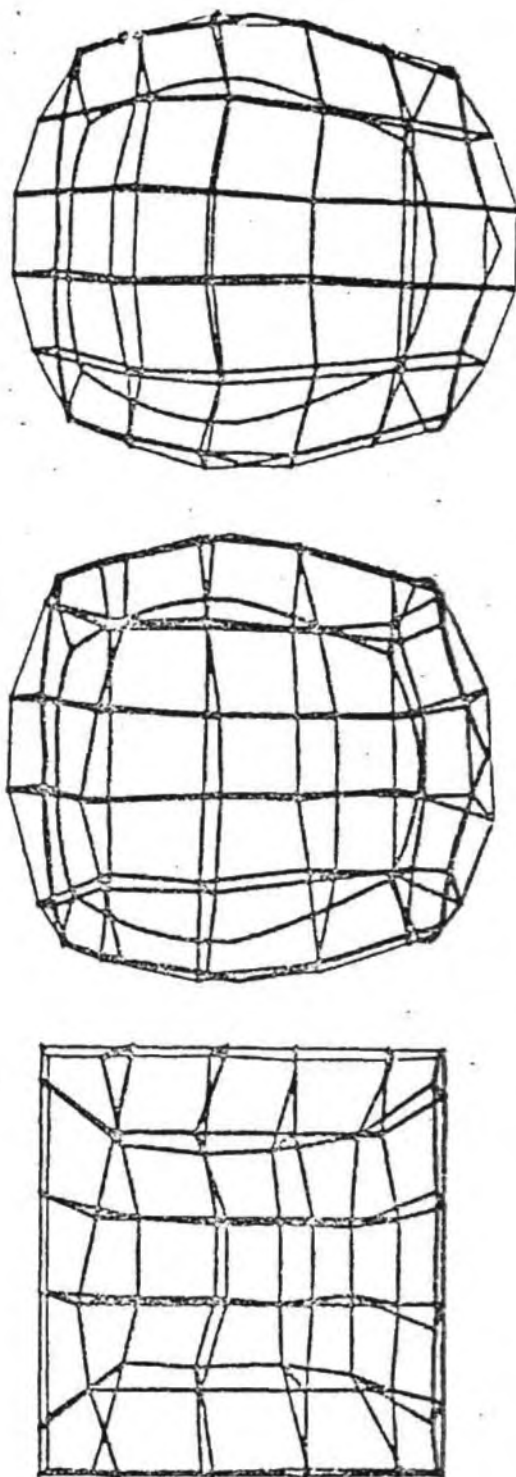


Figure 53. Optimized skull-brain model [6 x 6 x 6] bottom layer.

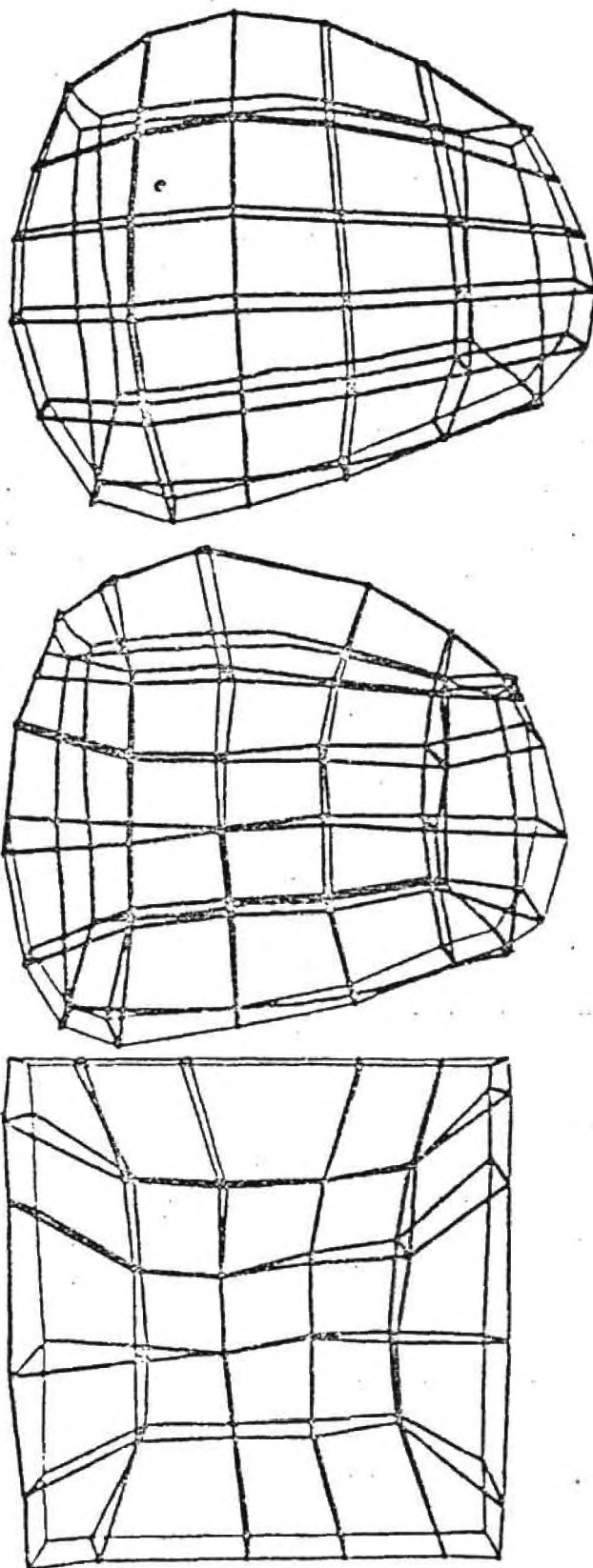


Figure 54. Optimized midsagittal layer.

Table 2.

Values of the Stiffness Measure for the Skull-Brain Model

| Iteration number | Stiffness Measure | Percent of Original |
|------------------|-------------------|---------------------|
| 0 | 427.8 | 100.0 |
| 1 | 407.7 | 95.3 |
| 4 | 383.8 | 89.7 |
| 10 | 381.4 | 89.2 |

was calculated and it is summarized in Table 2.

Epoxy Matrix Model

The model of the epoxy matrix around the crossed fibers was generated by using the boundary curves as input to equation (3.24). These twelve curves are shown in Figure 55. If 196 brick elements are wanted, seven in the r direction, seven in the s direction, and four in the t direction, then the resulting model for analysis would look like that in Figure 56. To understand the model better selected slices are shown in Figure 57.

Optimization of the epoxy matrix model was performed upon the [7 x 7 x 7] discretization. The trace of the stiffness matrix was also calculated and is summarized in Table 3.

This model has two planes of symmetry. The first is about the plane of $r=0.5$ and the second about the plane of $s=0.5$. When the optimization is performed, the resulting model should be symmetric about these planes. Figure 58 shows the center slice in the parametric

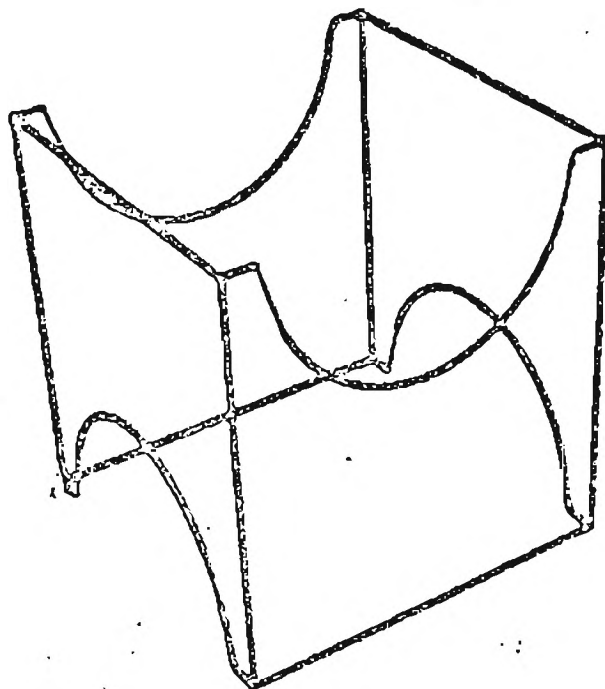


Figure 55. Twelve curves defining the epoxy matrix model.

s direction. Figure 58a shows of the original unoptimized slice. Figure 58 b and c show the slice after three iterations, the euclidean space is plotted on the left and the parametric space is plotted on the right. The elements have again more orthogonal faces, particularly around the region where the fiber goes, the semi-circular area. Figure 58 d and e show the same slice after five iterations. Comparing it to the slice after three iterations we can see very little difference, but the parametric plots indicate changes in the lower region. A very small difference is expected because the trace values for these two models vary by 1.6%.

In Figure 59 we see the first layer in the parametric t direction. The top images are the original and optimized (after five iterations) slice, viewed straight on. The lower images are the same

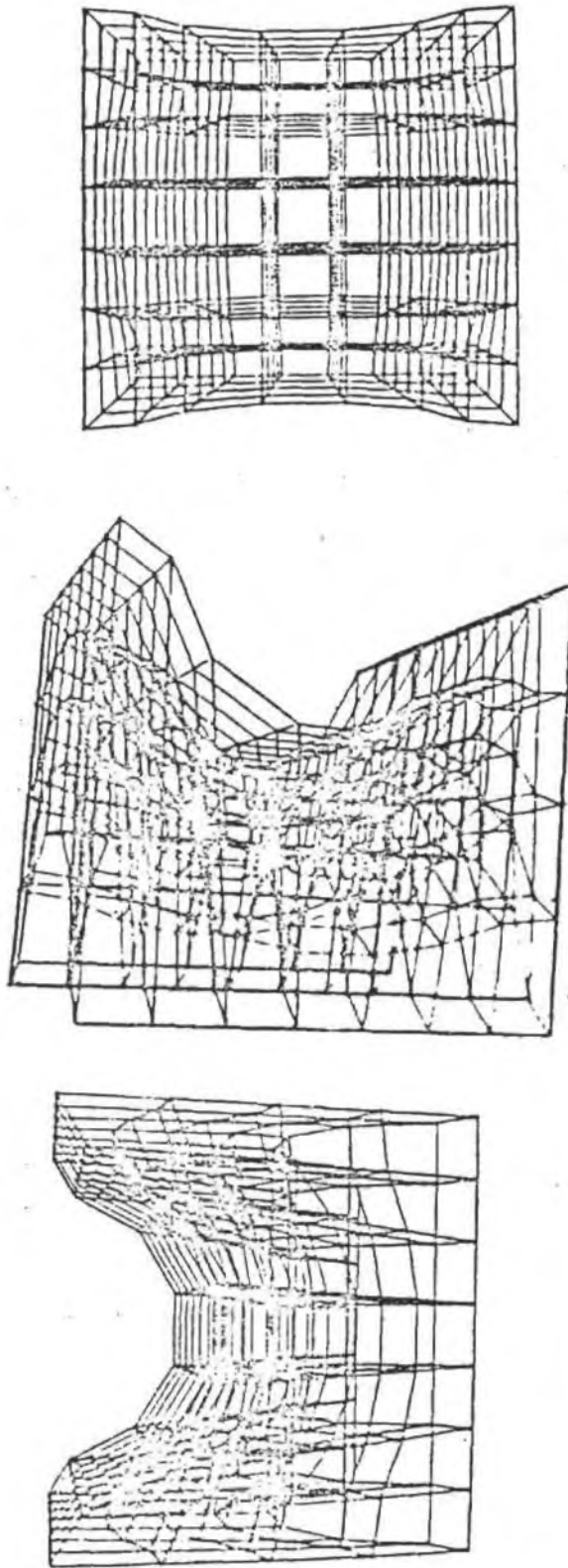


Figure 56. Three views of the epoxy matrix model.

Table 3.

Values of the Stiffness Measure for the Epoxy Matrix Model

| Iteration number | Stiffness Measure | Percent of Original |
|------------------|-------------------|---------------------|
| 0 | 176.3 | 100.0 |
| 1 | 172.4 | 97.8 |
| 3 | 165.5 | 93.8 |
| 5 | 162.6 | 92.2 |

slices rotated. The major adjustment of the nodes have been along the top and bottom edges.

Conclusions

The engineering design process has been presented and the design analysis loop part of the process has been discussed in some detail. This loop has many problems which are related to the way in which models are designed and analyzed. A higher level of abstraction for the modeling is needed to solve many of these problems. Two sample systems have been implemented with a high level model representation for planar regions and volumes. These systems use the high level model in two different ways. The first was for viewing and interacting, while the second was to generate models suitable for analysis. The geometry of the high level model has been expressed in a trivariate parametric form. The trivariate expressions were developed by extending the bivariate parametric patch expressions for surface

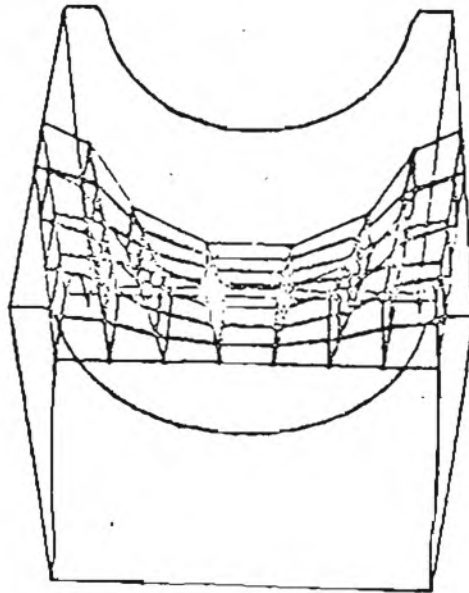
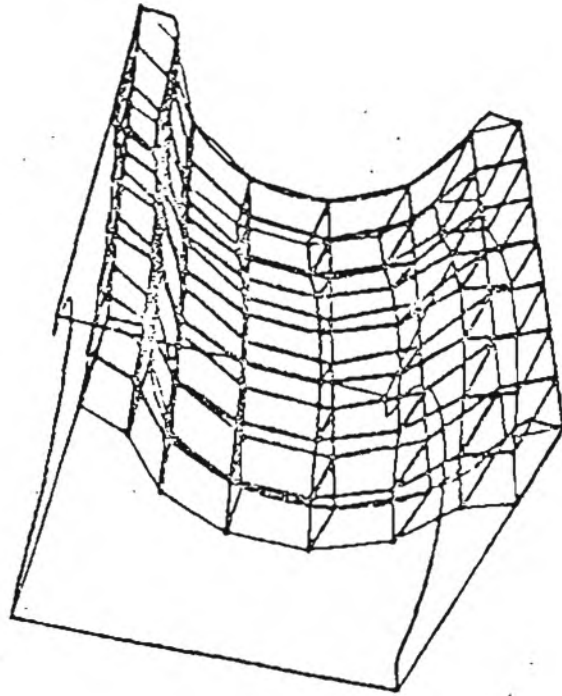
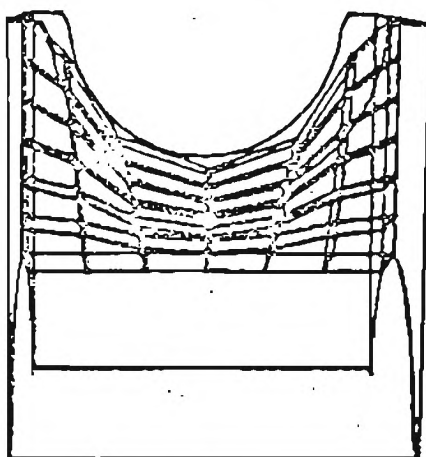
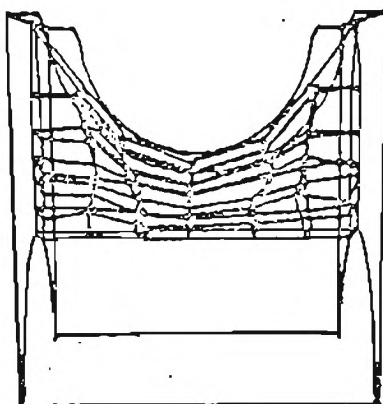


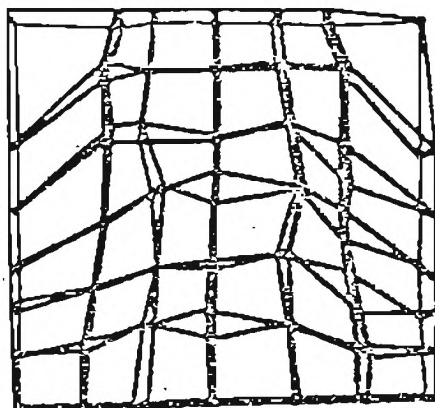
Figure 57. Selected slices of the epoxy matrix model.



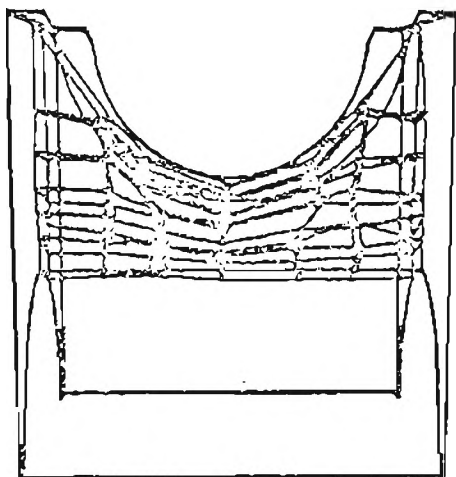
a



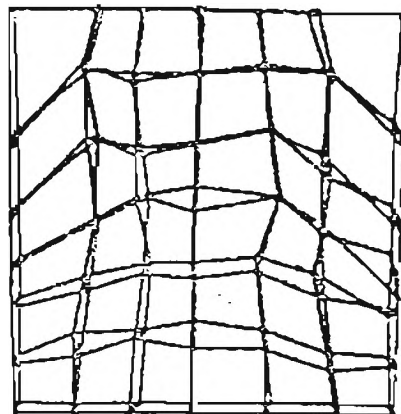
b



c



d



e

Figure 58. Optimized epoxy matrix model.

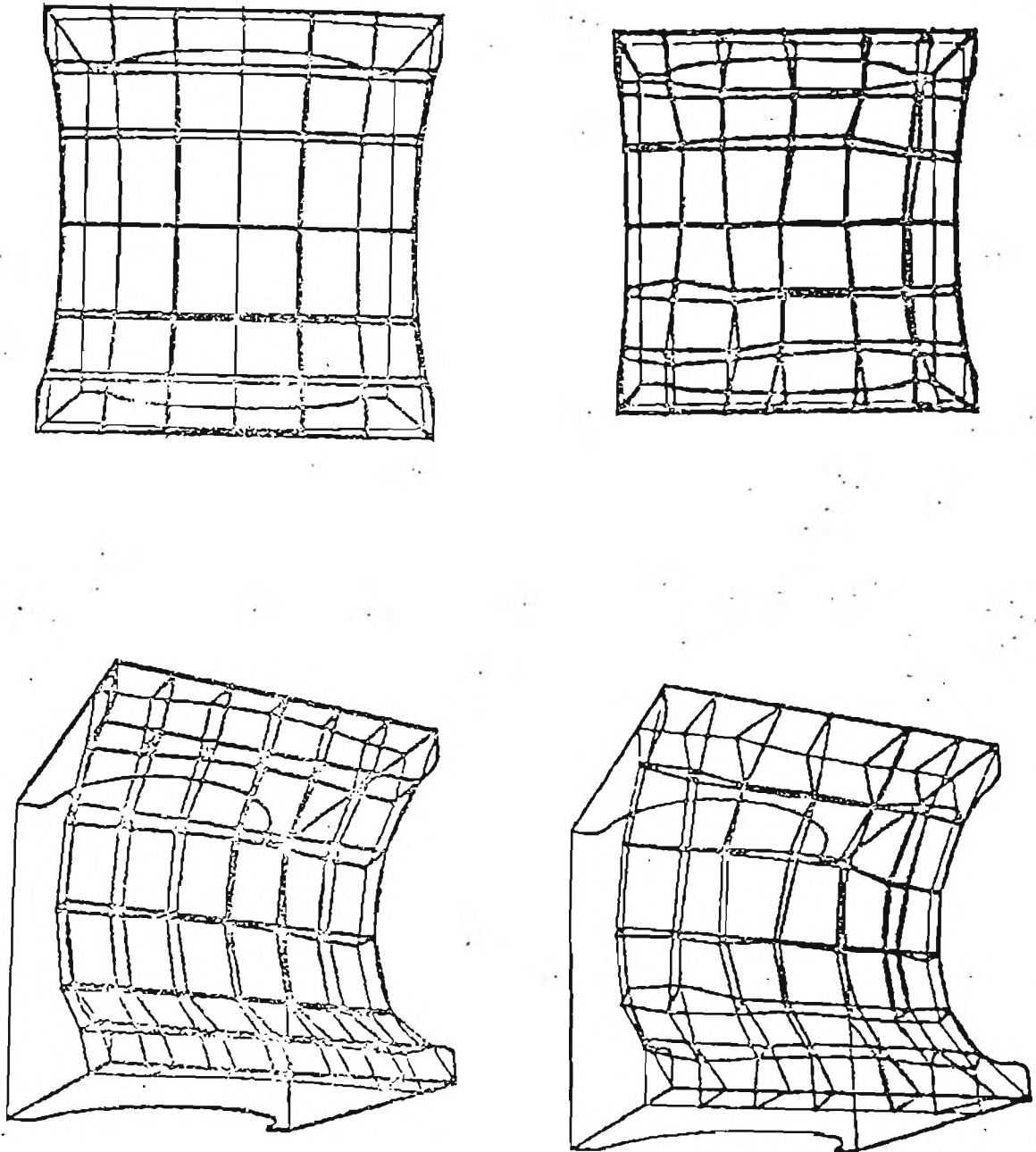


Figure 59. Optimized first parametric t layer of the epoxy matrix model.

representation.

The formulation of one specific analysis method has been investigated to provide information about model requirements. These model requirements or geometric merit criterion are then used when the models for analysis are generated from the high level model to make better analysis models. The actual implementation uses an iterative method to optimize on the geometric merit criteria in the analysis model. This iterative method is general enough to allow a framework for testing various optimization criteria. These systems as implemented are different from any other analysis model generation schemes in that the optimization of the model is performed in the parameter space. The optimization routine uses the high level model when boundary nodes are being adjusted to restrict the possible nodal positions to only those upon the boundary. This makes sure that the boundaries are not violated by trying to position a node outside of them. Every other known analysis model generator considers the boundary nodes fixed.

One high level model has been taken and used to generate several different analysis models, and then these were analyzed. The results have shown that the initial parametrization has a great deal of influence on the accuracy of the answers. We have also seen that the optimization of the model does improve the results regardless of the parameterization. The cost of the geometric optimization is believed to be very low when compared to an analysis run. However exact figures are not available because three different computers have been used in the process.

Future Work

A system was proposed in Chapter 2 which is based upon the idea of a higher level of abstraction for the modeling process. It is called the high level model and all of the information about the physical object that is being designed should be with that model. From this high level model all other models would be derived. A system has been implemented which uses the high level model's geometric information to produce various graphical views and models suitable for finite element analysis. To implement the complete system as proposed the high level model needs to be imbedded within a data base. Many more routines are needed to generate, modify and manipulate the data stored.

For the geometric information routines are needed which allow interactive design and modification of the building blocks for the volumes (i.e. points, space curves and surfaces). Also the structure or connections between various volumes need to be specified and stored. For example consider the model of the skull-brain, if another model of the neck and upper trunk were added, the system should allow the user to work with either model, separately or together. When they are used together the system needs to know at what point and with what orientation the two should be connected.

After analysis viewing of the calculated results is needed. Also some functions such as maximum principle stress or von-Mises stress are derived from from the results. The system needs a capability to produce plots of the results as well as calculating the derived quantities and plotting them.

The initial configuration of the nodal coordinates may be

specified in a uniform or nonuniform manner. When the optimization is applied the resulting model for analysis will have the same final nodal configuration regardless of the scheme initially used. More research is needed on the optimization scheme so that the initial nodal spacing will have some influence on the final configuration.

These are a few of the additional features needed to realize more fully the proposed system.

REFERENCES

1. Adamek, J.R., An Automatic Mesh Generator using Two- and Three-Dimensional Isoparametric Elements. Masters Thesis Naval Postgraduate School, Monterey, California, June 1973. (Available NTIS AD-764481)
2. Akyuz, F.A., Natural Coordinate Systems, An Automatic Input Data Generation Scheme for a Finite Element Method. Nuclear Engineering and Design, Vol. 11 (1970) pp.195-207.
3. Aral, K., Data Generation for a 3-D Finite Element System. ASME Paper No. 74-PVP-23, March 1974.
4. Armit, A.P., Curve and Surface Design using Multipatch and Multiobject Design Systems. Computer Aided Design, Vol. 3, No. 4, (Summer 1971), pp. 3-12.
5. Ball, A.A., CONSURF, Introduction of the Conic Lofting Tile. Computer Aided Design, Vol. 6, No. 4 (Oct. 1974), pp. 243-249.
6. Barnhill, R.E., Blending Function Interpolation: A Survey and some New Results. University of Dundee, Scotland, Department of Mathematics, Numerical Analysis Reports 9, July 1975.
7. Barnhill, R.E., and Gregory, J.A., Blending Function Interpolation to Boundary Data on Triangles. Brunel University, Department of Mathematics, TR/14, Aug. 1972.
8. Barnhill, R.E., and Riesenfeld, R.F., (eds.) Computer Aided Geometric Design. Academic Press, New York, 1975.
9. Bathe, K.J., and Wilson, E.L. Numerical Methods in Finite Element Analysis. Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
10. Beatty, J.N., An Interactive Graphics System for Mechanical Drafting and Design. In Proceedings of the CURE Symposium, University of California, Lawrence Livermore Laboratory, ERDA-741001, pp. 436-441.
11. Beveridge, G.S.G., and Schechter, R.S., Optimization: Theory and Practice. McGraw-Hill, New York, 1970.

12. Bezier, P., Mathematical and Practical Possibilities of UNISURF. In Computer Aided Geometric Design, Barnhill, R.E., and Riesenfeld, R.F., (eds.) Academic Press, New York, 1974, pp. 127-152.
13. Birkhoff, G., Cavendish, J.C., and Gordon, W.J., Multivariate Approximation by Locally Blended Univariate Interpolants. In Proceedings of the National Academy of Science, U.S.A. Vol. 71, No. 9, (Sept. 1974), pp. 3423-3425.
14. Boisserie, J.M., Generation of Two- and Three-Dimensional Finite Elements. International Journal for Numerical Methods in Engineering, Vol. 3, (1971), pp. 327-347.
15. Bond, T.J., Swannell, J.H., Henshell, R.D., and Warburton, G.B., A Comparison of Some Curved Two-Dimensional Finite Elements. Journal of Strain Analysis, Vol. 8, No. 3, (1973), pp. 182-190.
16. Bousquet, R.D., Yates, D.N., Sable, W.W., and Vinson, T.J., The Development of Computer Graphics for Large Scale Finite Element Codes. In Proceedings of the Third Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, Oct. 1971, pp. 437-447.
17. Braid, I.C., Designing with Volumes, 2nd Edition, Cantab Press. Cambridge, England, 1974.
18. Braid, I.C., The Synthesis of Solids Bounded by Many Faces. Communications of ACM, Vol. 18, No. 4, (April 1974), pp. 209-216.
19. Braid, I.C., Six Systems for Shape Design and Representation. University of Cambridge, Computer-Aided Design Group Document No. 87, May 1975.
20. Brewer, J.A. III, Computer Graphics and Computer-Aided Design. Engineering Computer Graphics Colloquium, United States Army Engineers, Waterways Experiment Station, Vicksburg, Miss., Jan. 1975, pp. 57-84.
21. Brown, E.E., Computer Graphics as a Tool for Engineering Analysis. In Proceedings of the CUBE Symposium, University of California, Lawrence Livermore Laboratory, ERDA-741001, pp. 36-44.
22. Brown, R.E., MOVIE.LASL Version 1.0 User's Manual. University of California, Los Alamos Scientific Laboratory, LA-NUREG-6532-M, Dec. 1976.
23. Carroll, W.E., Inclusive Criteria for Optimum Grid Generation in Discrete Analysis Technique. Computers and Structures, Vol. 6 (1976), pp. 333-337.

24. Carroll, W.E., and Barker, R.M., A Theorem for Optimum Finite Element Idealizations. International Journal of Solids and Structures, Vol. 9, (1973), pp. 883-895.
25. Catmul, E., A Subdivision Algorithm for Computer Display of Curved Surfaces. University of Utah, Department of Computer Science, UTEC-CSc-74-133, Dec. 1974.
26. Catmul, E., and Rom, R., A Class of Local Interpolating Splines. In Computer Aided Geometric Design, Barnhill, R.E., and Riesenfeld, R.F., (eds.), Academic Press, New York, 1974, pp. 317-326.
27. Christiansen, H.N., Display of Kinematic and Elastic Systems. In Proceedings of the Third Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, Oct. 1971, pp. 532-547.
28. Christiansen, H.N., Applications of Continuous Tone Computer Generated Images in Structural Mechanics. In Structural Mechanics Computer Programs Pilkey et al (eds.), Academic Press, New York, 1974, pp. 731-747.
29. Clark, J.H., Three-Dimensional Design of Free-Form B-Spline Surfaces. University of Utah, Department of Computer Science, UTEC-CSc-74-120, Sept. 1974.
30. Clough, R.W., Comparison of Three Dimensional Finite Elements. In Proceedings of the Symposium on Applications of Finite Elements in Civil Engineering. Rowan, W.H., and Hackett, R.M. (eds.), ASCE (1969), pp. 1-26.
31. Coles, L.S., An Experiment in Robot Tool Using. Technical Note 41, Artificial Intelligence Center, Stanford Research Insititue, Menlo Park, California, 1970.
32. Collins, R.J., Bandwidth Reduction by Automatic Renumbering. International Journal for Numerical Methods in Engineering, Vol. 6, (1973) pp. 345-356.
33. Conte, S.D., and de Boor, C., Elementary Numerical Analysis 2nd Edition, McGraw-Hill, New York, 1972.
34. Control Data, 6400/6600 Computer Systems APT Reference Maunual Version 1.0, Control Data Corp. Palo Alto, California, Oct. 1966.
35. Cook, R.D., Concepts and Applications of Finite Element Analysis. John Wiley and Sons, New York, 1974.
36. Cook, W.A., Body Oriented (Natural) Coordinates for Generating Three-Dimensional Meshes. International Journal for Numerical Methods in Engineering, Vol. 8, (1974), pp. 27-43.

37. Coons, S.A., Surfaces for Computer Aided Design of Space Forms, M.I.T. Project MAC TR-41, (June 1967).
38. Cowper, G.R., Variational Procedures and Convergence of Finite-Element Methods. In Numerical and Computer Methods in Structural Mechanics, Fenves, et al (eds.) Academic Press, New York, 1973, pp. 1-12.
39. Crane, H.L., Gibbs, N.E., Poole, W.G. Jr., and Stockmeyer, P.K., Matrix Bandwidth and Profile Reduction. Institute for Computer Applications in Science and Engineering, Hampton, Virginia, Report No. 75-9, April 1975.
40. Cuthill, E., and McKee, J., Reducing the Bandwidth of Sparse Symmetric Matrices. In Proceedings of the 24th National Conference of ACM, (1969) pp. 157-172.
41. Davis, P.J., Interpolation and Approximation. Dover Books, New York, 1975.
42. de Boor, C., and Lynch, R.E., On Splines and their Minimum Properties. Journal of Mathematics and Mechanics, Vol. 15, No. 6, (1966), pp. 953-969.
43. Desai, C.S., and Abel, J.F., Introduction to the Finite Element Method. Van Nostrand Reinhold Co. New York, 1972.
44. Dube, R.P., Local Schemes for Computer Aided Geometric Design. Ph.D. Dissertation, University of Utah, Department of Mathematics, June, 1975.
45. Eastman, C., Lividini, J., and Stoker, D., A Database for Designing Large Physical Systems. In Proceedings of the National Computer Conference ACM, 1975, pp. 603-607.
46. Ergatoudis, I., Irons, B.M., and Zienkiewicz, O.C., Curved Isoparametric, "Quadrilateral" Elements for Finite Element Analysis. International Journal for Solids and Structures, Vol. 4, (1968), pp. 31-42.
47. Federic, C.O., Wong, Y.C., and Edge, F.W., Two-Dimensional Automatic Mesh Generation for Structural Analysis. International Journal for Numerical Methods in Engineering, Vol. 2, (1970), pp. 133-144.
48. Fenves, S.J., Perrone, N., Robinson, A.R., and Schnobrich, W.C., (eds.) Numerical and Computer Methods in Structural Mechanics, Academic Press, New York, 1973.
49. Ferguson, J., Multivariate Curve Interpolation. Journal of the ACM, Vol. 11, No. 2, (April 1964), pp. 221-228.

50. Forrest, A.R., Computational Geometry. In Proceedings of the Royal Society of London, A. 321, (1971), pp. 187-195.
51. Forrest, A.R., On Coons and Other Methods. Computer Graphics and Image Processing, Vol. 1, (1972), pp. 341-359.
52. Forrest, A.R., Mathematical Principles for Curve and Surface Representation. In proceedings of Curved Surfaces in Engineering, held at Churchill College, Cambridge, England, (March 1972), pp. 5-13.
53. Fried, I., Accuracy of Complex Finite Elements. Journal of the American Institute of Aeronautics and Astronautics, Vol. 10, No. 3, (1972), pp. 347-349.
54. Fung, Y.C., Foundations of Solid Mechanics. Prentice-Hall, Englewood Cliffs, New Jersey, 1965.
55. Gill, J.I., Computer-Aided Design of Shell Surfaces using Finite Element Method. Ph.D. dissertation, University of Cambridge, 1972.
56. Gordon, W.J., Distributive Lattices and the Approximation of Multivariate Functions. In Approximation with Special Emphasis on Spline Functions, Schoenber, I.J., (ed), Academic Press, New York, 1969, pp. 223-277.
57. Gordon, W.J., Blending-Function Methods of Bivariate and Multivariate Interpolation and Approximation. SIAM Journal of Numerical Analysis, Vol. 8, No. 1, (March 1971) pp. 158-177.
58. Gordon, W.J., and Hall, C.A., Transfinite Element Methods: Blending-Function Interpolation over Arbitrary Curved Domains. Numerische Mathematik, Vol. 21, (1973), pp. 109-129.
59. Henshell, R.D., Walters, D., and Warburton, G.R., A New Family of Curvilinear Plate Bending Elements for Vibration and Stability. Journal of Sound and Vibration, Vol. 20, No. 3, (1972), pp. 381-397.
60. Herrman, L.R., Laplacian-Isoparametric Grid Generation Scheme. ASCE Journal of the Engineering Mechanics Division, Vol. EM5, (Oct. 1976), pp. 749-755.
61. Hutula, D.N., and Zeiler, S.M., MESH3D: A Three-Dimensional Finite Element Mesh Generator Program for Eight Node Isoparametric Elements. Westinghouse Electric Corp., Bettis Atomic Power Laboratory, APD-TM-1079, March 1973.
62. Irons, B.M., Engineering Applications of Numerical Integration in Stiffness Methods. Journal, American Institute of Aeronautics and Astronautics, Vol. 4, No. 11, (1966), pp. 2035-2037.

63. Jones, R.E., OMESH: A Self-Organizing Mesh Generation Program. Sandia Laboratories, Albuquerque, New Mexico, Report SLA-73-1088, July 1974.
64. Kamel, H.A., and Shanta, P.J., A Solid Mesh Generator and Result Display Package. ASME Paper 74-PVP-34, March 1974.
65. Larder, R.A., The Application of Three-Dimensional Finite Element Analysis to the Micromechanics of Fibrous Composite Material. In Proceedings of the CUFE Symposium, University of California, Lawrence Livermore Laboratory, ERDA-741001, pp. 109-113.
66. McCleary, L.E., and Smith, R.R., GENCON- A Contour Plotting Program for Arbitrary Regions. Naval Undersea Center, San Diego, California.
67. McNeice, G.M., and Hunnisett, S.F., Mixed-Displacement Finite Element Analysis with Particular Application using Plane Stress Triangles. Journal of Strain Analysis, Vol. 7, No. 4, 1972, pp. 243-252.
68. McNeice, G.M., and Marcel, P.V., Optimization of Finite Element Grids Based on Minimum Potential Energy. ASME Journal of Engineering for Industry, (February 1973), pp. 186-190.
69. Mikhlin, S.G., Variational Methods in Mathematical Physics. Pergamon Press, Oxford, (1964).
70. Mukherji, B., High Speed Integration for Eight Node Bricks. presented at the SAPIV user's conference, Summer 1976.
71. Nering, E.D., Linear Algebra and Matrix Theory. John Wiley and Sons, New York, 1963.
72. Newell, M.E., Man Machine Communication in Three Dimensions. In Computer Aided Geometric Design, Barnhill, R.E., and Riesenfeld, R.F., (eds.) Academic Press, New York, 1974.
73. Newell, M.E., The Utilization of Procedure Models in Digital Image Synthesis. University of Utah, Department of Computer Science UTEC-CSc-76-218.
74. Newell, M.E., Newell, R.G., and Sancha, T.L., A Solution to the Hidden Surface Problem. In Proceedings of the ACM National Conference, 1974, pp. 443-448.
75. Newman, W.M., and Sproull, R.F., Principles of Interactive Computer Graphics. McGraw-Hill, New York, 1973.

76. Oliveira, E.R.A., Optimization of Finite Element Solutions. In Proceedings of the Third Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, Oct. 1971, pp. 6133-672. •
77. Peters, G.J., Interactive Computer Graphics Applications of the Parametric Bi-Cubic Surface to Engineering Design Problems. In Computer Aided Geometric Design, Barnhill, R.E., and Riesenfeld, R.F., (eds.) Academic Press, New York, 1974, pp. 259-302.
78. Pilkey, W.D., Saczalski, K., and Schaefffer, H.G., (eds.) Structural Mechanics Computer Programs, Surveys, Assessments, and Availability. Univeristy Press of Virginia, Charlottesville, Virginia, 1974.
79. Prenter, P.M., Splines and Variational Methods. John Wiley and Sons, New York, 1973.
80. Ramstad, H., Convergence and Numerical Accuracy with Special Reference to Plate Bending. In Finite Element Methods in Stress Analysis, Holand, I., and Bell, K., (eds.), Tapir Press, Norway, 1969, pp. 179-211.
81. Riesenfeld, R.F., Applications of B-Spline Approximation to Geometric Problems of Computer Aided Design. University of Utah Department of Computer Science, UTEC-CSc-73-126, March 1973.
82. Riesenfeld, R.F., Aspects of Modelling in Computer Aided Geometric Design. In Proceedings of the National Computer Conference, Vol. 44, AFIPS Press, Montvale, New Jersey, 1975, pp. 597-602.
83. Rogers, D.F., and Adams, J.A., Mathematical Elements for Computer Graphics, McGraw-Hill, New York, 1976.
84. Roney, B.D., and Stone, B.J., An Application of Computer Graphics to Finite Element Mesh Generation. In Proceedings of a Conference on Stress and Strain in Engineering. Brisbane, Australia, 1973, pp. 156-163.
85. Sabin, M.A., An Existing System in the Aircraft Industry. The British Aircraft Corporation Numerical Master Geometry System. In Proceedings of the Royal Society of London, A. 321, (1971), pp. 197-205.
86. Schumaker, L.L., Fitting Surfaces to Scattered Data. In Approximation Theory II, Lorentz, G.G., Chui, C.K., and Schumaker, L.L., (eds.) Academic Press, New York, (1976), pp. 203-268.

87. Shugar, T.A., Transient Structural Response of the Linear Skull Brain System. In Proceedings of the 19th STAP Car Crash Conference, Society of Automotive Engineers, Warrendale, Penn. (1975) pp.581-614.
88. Spendley, W., Hext, G.R., and Himsworth, F.R., Sequential Applications of Simplex Designs in Optimization and Evolutionary Operations. *Technometrics*, Vol. 4, (1962), pp. 441-461.
89. Stricklin, J.S., Ho, W.S., Richardson, E.Q., and Haisler, W.E., On Isoparametric versus Linear Strain Triangular Elements. Texas A and M University, Aerospace Engineering Department, College Station, Texas. (to appear)
90. Sutherland, I.E., Sproull, R.F., and Schumacker, R.A., A Characterization of Ten Hidden-Surface Algorithms. *ACM Computing Surveys*, Vol. 6, No. 1, (March 1974), pp. 1-55.
91. Timoshenko, S.P., and Goodier, J.N., Theory of Elasticity, 3rd Edition, McGraw-Hill, New York, 1970.
92. Tocher, J.L., and Felippa, C.A., Computer Graphics Applied to Production Structural Analysis. In Proceedings of the Symposium of the International Union of Theoretical and Applied Mechanics, University of Liege, Belgium, August 1970, pp. 521-545.
93. Turcke, D.J., On Optimum Finite Element Grid Configurations. *Journal, American Institute of Aeronautics and Astronautics*, Vol. 14, No. 2, (February 1976), pp. 264-265.
94. Turcke, D.J., and McNeice, G.M., Guidelines for Selecting Finite Element Grids based upon an Optimization Study. *Computers and Structures*, Vol. 4, (1974), pp. 499-519.
95. Turner, M.J., Clough, R.W., Martin, H.C., and Topp, L.J., Stiffness and Deflection Analysis of Complex Structures. *Journal of Aeronautical Sciences*, Vol. 23, No. 9, (1956) pp. 805-823.
96. Voelcker, H., Organization of Information in Discrete Part Manufacturing. University of Rochester, Production Automation Project Technical Paper.
97. Watkins, G.S., A Real-Time Visible Surface Algorithm. University of Utah, Department of Computer Science, UTEC-CSc-70-101, June 1970
98. Winslow, A.M., 'Equipotential' Zoning of Two-Dimensional Meshes. University of California, Lawrence Livermore Laboratory, UCRL-7312, 1963.
99. Zienkiewicz, O.C., The Finite Element Method, McGraw-Hill, New York, 1973.

APPENDIX

The purpose of this appendix is to show by substitution the interpolation properties of the three trivariate interpolants derived in Chapter 3. These operators are combinations of the three projectors whose equations are repeated below.

$$P_1 F(r,s,t) = (1-t)F(r,s,0) + tF(r,s,1)$$

$$P_2 F(r,s,t) = (1-s)F(r,0,t) + sF(r,1,t)$$

$$P_3 F(r,s,t) = (1-r)F(0,s,t) + rF(1,s,t)$$

The equation of the tensor product is

$$\begin{aligned} (P_1 P_2 P_3) F(r,s,t) &= (1-r)(1-s)(1-t)F(0,0,0) + (1-r)(1-s)tF(0,0,1) \\ &+ (1-r)s(1-t)F(0,1,0) + (1-r)stF(0,1,1) \\ &+ r(1-s)(1-t)F(1,0,0) + r(1-s)tF(1,0,1) \\ &+ rs(1-t)F(1,1,0) + rstF(1,1,1). \end{aligned}$$

The interpolation properties will be shown by substitution for one corner point, one curve, and one surface.

Example for corner $F(0,1,0)$

$$\begin{aligned} (P_1 P_2 P_3) F(0,1,0) &= (1-0)(1-1)(1-0)F(0,0,0) + (1-0)(1-1)(0)F(0,0,1) \\ &+ (1-0)(1)(1-0)F(0,1,0) + (1-0)(1)(0)F(0,1,1) \\ &+ (0)(1-1)(1-0)F(1,0,0) + (0)(1-1)(0)F(1,0,1) \\ &+ (0)(1)(1-0)F(1,1,0) + (0)(1)(0)F(1,1,1) \\ &= F(0,1,0) \end{aligned}$$

Example for curve $F(1,0,t)$

$$\begin{aligned}
 (P_1 P_2 P_3)F(1,0,t) &= (1-1)(1-0)(1-t)F(0,0,0) + (1-1)(1-0)tF(0,0,1) \\
 &+ (1-1)(0)(1-t)F(0,1,0) + (1-1)(0)tF(0,1,1) \\
 &+ (1)(1-0)(1-t)F(1,0,0) + (1)(1-0)tF(1,0,1) \\
 &+ (1)(0)(1-t)F(1,1,0) + (1)(0)tF(1,1,1) \\
 &= (1-t)F(0,1,0) + tF(0,1,1)
 \end{aligned}$$

Example for surface $F(1,s,t)$

$$\begin{aligned}
 (P_1 P_2 P_3)F(1,s,t) &= (1-1)(1-s)(1-t)F(0,0,0) + (1-1)(1-s)tF(0,0,1) \\
 &+ (1-1)s(1-t)F(0,1,0) + (1-1)stF(0,1,1) \\
 &+ (1)(1-s)(1-t)F(1,0,0) + (1)(1-s)tF(1,0,1) \\
 &+ (1)s(1-t)F(1,1,0) + (1)stF(1,1,1) \\
 &= (1-s)(1-t)F(1,0,0) + (1-s)tF(1,0,1) \\
 &+ s(1-t)F(1,1,0) + stF(1,1,1)
 \end{aligned}$$

The Boolean sum interpolant's equation is

$$\begin{aligned}
 (P_1 + P_2 + P_3)F(r,s,t) &= (P_1 + P_2 + P_3 - P_1 P_2 - P_2 P_3 \\
 &- P_3 P_1 + P_1 P_2 P_3) F(r,s,t).
 \end{aligned}$$

The expanded form is

$$\begin{aligned}
 (P_1 + P_2 + P_3)F(r,s,t) &= (1-t)F(r,s,0) + tF(r,s,1) \\
 &+ (1-s)F(r,0,t) + sF(r,1,t) \\
 &+ (1-r)F(0,s,t) + rF(1,s,t) \\
 &- (1-s)(1-t)F(r,0,0) - s(1-t)F(r,1,0) - (1-s)tF(r,0,1) - stF(r,1,1) \\
 &- (1-r)(1-s)F(0,0,t) - r(1-s)F(1,0,t) - (1-r)sF(0,1,t) - rsF(1,1,t) \\
 &- (1-r)(1-t)F(0,s,0) - r(1-t)F(1,s,0) - (1-r)tF(0,s,1) - rtF(1,s,1) \\
 &+ (1-r)(1-s)(1-t)F(0,0,0) + (1-r)(1-s)tF(0,0,1) + (1-r)s(1-t)F(0,1,0) \\
 &+ (1-r)stF(0,1,1) + r(1-s)(1-t)F(1,0,0) + r(1-s)tF(1,0,1)
 \end{aligned}$$

$$+ rs(1-t)F(1,1,0) + rstF(1,1,1).$$

The Boolean sum interpolant can be shown to interpolate the six surfaces, the twelve edge curves and the eight corner points. Here because of space considerations only interpolation of one corner, one curve and one surface will be shown.

Example for corner F(1,1,1)

$$\begin{aligned} (P_1 + P_2 + P_3)F(1,1,1) &= (1-1)F(1,1,0) + (1)F(1,1,1) \\ &+ (1-1)F(1,0,1) + (1)F(1,1,1) + (1-1)F(0,1,1) + (1)F(1,1,1) \\ &- (1-1)(1-1)F(1,0,0) - (1)(1-1)F(1,1,0) - (1-1)(1)F(1,0,1) \\ &- (1)(1)F(1,1,1) - (1-1)(1-1)F(0,0,1) - (1)(1-1)F(1,0,1) \\ &- (1-1)(1)F(0,1,1) - (1)(1)F(1,1,1) - (1-1)(1-1)F(0,1,0) \\ &- (1)(1-1)F(1,1,0) - (1-1)(1)F(0,1,1) - (1)(1)F(1,1,1) \\ &+ (1-1)(1-1)(1-1)F(0,0,0) + (1-1)(1-1)(1)F(0,0,1) \\ &+ (1-1)(1)(1-1)F(0,1,0) + (1-1)(1)(1)F(0,1,1) + (1)(1-1)(1-1)F(1,0,0) \\ &+ (1)(1-1)(1)F(1,0,1) + (1)(1)(1-1)F(1,1,0) + (1)(1)(1)F(1,1,1) \\ &= F(1,1,1)+F(1,1,1)+F(1,1,1)-F(1,1,1)-F(1,1,1)-F(1,1,1)+F(1,1,1) \\ &= F(1,1,1) \end{aligned}$$

Example for curve F(0,s,1)

$$\begin{aligned} (P_1 + P_2 + P_3)F(0,s,1) &= (1-1)F(0,s,0) + (1)F(0,s,1) \\ &+ (1-s)F(0,0,1) + (s)F(0,1,1) + (1-0)F(0,s,1) + (0)F(1,s,1) \\ &- (1-s)(1-1)F(0,0,0) - s(1-1)F(0,1,0) - (1-s)(1)F(0,0,1) \\ &- s(1)F(0,1,1) - (1-0)(1-s)F(0,0,1) - (0)(1-s)F(1,0,1) \\ &- (1-0)sF(0,1,1) - (0)sF(1,1,1) - (1-0)(1-1)F(0,s,0) \\ &- (0)(1-1)F(1,s,0) - (1-0)(1)F(0,s,1) - (0)(1)F(1,s,1) \\ &+ (1-0)(1-s)(1-1)F(0,0,0) + (1-0)(1-s)(1)F(0,0,1) \\ &+ (1-0)s(1-1)F(0,1,0) + (1-0)s(1)F(0,1,1) + (0)(1-s)(1-1)F(1,0,0) \end{aligned}$$

$$\begin{aligned}
& + (0)(1-s)(1)F(1,0,1) + (0)s(1-1)F(1,1,0) + (0)s(1)F(1,1,1) \\
& = F(0,s,1) + (1-s)F(0,0,1) + sF(0,1,1) + F(0,s,1) - (1-s)F(0,0,1) \\
& - sF(0,1,1) - (1-s)F(0,0,1) - sF(0,1,1) - F(0,s,1) + (1-s)F(0,0,1) \\
& = F(0,s,1)
\end{aligned}$$

Example for surface $F(0,s,t)$

$$\begin{aligned}
& (P_1 + P_2 + P_3)F(0,s,t) = (1-t)F(0,s,0) + tF(0,s,1) \\
& + (1-s)F(0,0,t) + (s)F(0,1,t) + (1-0)F(0,s,t) + (0)F(1,s,t) \\
& - (1-s)(1-t)F(0,0,0) - s(1-t)F(0,1,0) - (1-s)tF(0,0,1) - stF(0,1,1) \\
& - (1-0)(1-s)F(0,0,t) - (0)(1-s)F(1,0,t) - (1-0)sF(0,1,t) - (0)sF(1,1,t) \\
& - (1-0)(1-t)F(0,s,0) - (0)(1-t)F(1,s,0) - (1-0)tF(0,s,1) - (0)tF(1,s,1) \\
& + (1-0)(1-s)(1-t)F(0,0,0) + (1-0)(1-s)tF(0,0,1) + (1-0)s(1-t)F(0,1,0) \\
& + (1-0)stF(0,1,1) + (0)(1-s)(1-t)F(1,0,0) + (0)(1-s)tF(1,0,1) \\
& + (0)s(1-t)F(1,1,0) + (0)stF(1,1,1) \\
& = F(0,s,t)
\end{aligned}$$

The intermediate operator or Coons' "hypersurface" equation is

$$(P_1P_2 + P_2P_3 + P_3P_1)F(r,s,t) = (P_1P_2 + P_2P_3 + P_3P_1 - 2P_1P_2P_3)F(r,s,t).$$

And written out it is

$$\begin{aligned}
& (P_1P_2 + P_2P_3 + P_3P_1)F(r,s,t) = \\
& (1-s)(1-t)F(r,0,0) + s(1-t)F(r,1,0) + (1-s)tF(r,0,1) + stF(r,1,1) \\
& + (1-r)(1-s)F(0,0,t) + r(1-s)F(1,0,t) + (1-r)sF(0,1,t) + rsF(1,1,t) \\
& + (1-r)(1-t)F(0,s,0) + r(1-t)F(1,s,0) + (1-r)tF(0,s,1) + rtF(1,s,1) \\
& - 2((1-r)(1-s)(1-t)F(0,0,0) + (1-r)(1-s)tF(0,0,1) + (1-r)s(1-t)F(0,1,0) \\
& + (1-r)stF(0,1,1) + r(1-s)(1-t)F(1,0,0) + r(1-s)tF(1,0,1) \\
& + rs(1-t)F(1,1,0) + rstF(1,1,1)).
\end{aligned}$$

As before we can substitute values of r, s and t into the above

expression to see what it interpolates.

Example for corner $F(1,0,1)$

$$\begin{aligned}
 (P_1P_2 + P_2P_3 + P_3P_1)F(1,0,1) &= \\
 & (1-D)(1-1)F(1,0,0) + (D)(1-1)F(1,1,0) + (1-D)(1)F(1,0,1) \\
 & + (D)(1)F(1,1,1) + (1-1)(1-D)F(0,0,1) + (1)(1-D)F(1,0,1) \\
 & + (1-1)(D)F(0,1,1) + (1)(0)F(1,1,1) + (1-1)(1-1)F(0,0,0) \\
 & + (1)(1-1)F(1,0,0) + (1-1)(1)F(0,0,1) + (1)(1)F(1,0,1) \\
 & - 2((1-1)(1-0)(1-1)F(0,0,0) + (1-1)(1-0)(1)F(0,0,1) \\
 & + (1-1)(0)(1-1)F(0,1,0) + (1-1)(0)(1)F(0,1,1) \\
 & + (1)(1-0)(1-1)F(1,0,0) + (1)(1-0)(1)F(1,0,1) \\
 & + (1)(0)(1-1)F(1,1,0) + (1)(0)(1)F(1,1,1)) \\
 & = F(1,0,1) + F(1,0,1) + F(1,0,1) - 2F(1,0,1) \\
 & = F(1,0,1)
 \end{aligned}$$

Example for curve $F(r,0,1)$

$$\begin{aligned}
 (P_1P_2 + P_2P_3 + P_3P_1)F(r,0,1) &= \\
 & (1-0)(1-1)F(r,0,0) + (D)(1-1)F(r,1,0) + (1-0)(1)F(r,0,1) \\
 & + (D)(1)F(1,1,1) + (1-r)(1-D)F(0,0,1) + (r)(1-0)F(1,0,1) \\
 & + (1-r)(0)F(0,1,1) + (1)(0)F(1,1,1) + (1-r)(1-1)F(0,0,0) \\
 & + (r)(1-1)F(1,0,0) + (1-r)(1)F(0,0,1) + (r)(1)F(1,0,1) \\
 & - 2((1-r)(1-0)(1-1)F(0,0,0) + (1-r)(1-0)(1)F(0,0,1) \\
 & + (1-r)(0)(1-1)F(0,1,0) + (1-r)(0)(1)F(0,1,1) \\
 & + (r)(1-0)(1-1)F(1,0,0) + (r)(1-0)(1)F(1,0,1) \\
 & + (r)(0)(1-1)F(1,1,0) + (r)(0)(1)F(1,1,1)) \\
 & = F(r,0,1) + (1-r)F(0,0,1) + rF(1,0,1) + (1-r)F(0,0,1) \\
 & + rF(1,0,1) - 2((1-r)F(0,0,1) + rF(1,0,1)) \\
 & = F(r,0,1)
 \end{aligned}$$

Example for surface $F(r,s,0)$

$$\begin{aligned}
 & (P_1P_2 + P_2P_3 + P_3P_1)F(r,s,0) = \\
 & (1-s)(1-0)F(r,0,0) + s(1-0)F(r,1,0) + (1-s)(0)F(r,0,1) \\
 & + s(0)F(r,1,1) + (1-r)(1-s)F(0,0,0) + r(1-s)F(1,0,0) \\
 & + (1-r)sF(0,1,0) + rsF(1,1,0) + (1-r)(1-0)F(0,s,0) \\
 & + r(1-0)F(1,s,0) + (1-r)(0)F(0,s,1) + r(0)F(1,s,1) \\
 & - 2((1-r)(1-s)(1-0)F(0,0,0) + (1-r)(1-s)(0)F(0,0,1) \\
 & + (1-r)s(1-0)F(0,1,0) + (1-r)s(0)F(0,1,1)) \\
 & + r(1-s)(1-0)F(1,0,0) + r(1-s)(0)F(1,0,1) \\
 & + rs(1-0)F(1,0,1) + rs(0)F(1,1,1) \\
 & = (1-s)F(r,0,0) + sF(r,1,0) + (1-r)(1-s)F(0,0,0) \\
 & + r(1-s)F(1,0,0) + (1-r)sF(0,1,0) + rsF(1,1,0) \\
 & + (1-r)F(0,s,0) + rF(1,s,0) - 2((1-r)(1-s)F(0,0,0) \\
 & + (1-r)sF(0,1,0) + r(1-s)F(1,0,0) + rsF(1,1,0)) \\
 & = (1-s)F(r,0,0) + sF(r,1,0) + (1-r)F(0,s,0) + rF(1,s,0) \\
 & - (1-r)(1-s)F(0,0,0) - (1-r)sF(0,1,0) - r(1-s)F(1,0,0) - rsF(1,1,0)
 \end{aligned}$$

We see that it has interpolated the corner and curves but not the surfaces. The surface equation produced is equivalent to (3-13) for the face of the logical cube in question.

VITA

Name Bruce Eric Brown

Birthdate February 1, 1952

Birthplace Salt Lake City, Utah

High School West High School,
Salt Lake City, Utah

University Brigham Young University,
1970-1974 Provo, Utah

Degree B.S., Brigham Young University,
1974 Provo, Utah

Degree M.S., Brigham Young University,
1974 Provo, Utah

Professional Positions Visiting Staff Member, University of
California, Los Alamos Scientific
Laboratory, Los Alamos, New Mexico,
summer 1976; Research Structural
Engineer, United States Navy Civil
Engineering Laboratory, Port
Hueneme, California, summer 1975;
Mechanical Engineer, University of
California, Lawrence Livermore
Laboratory, Livermore, California,
summers 1974 and 1973.

Consulting University of California, Lawrence
Livermore Laboratory, Livermore,
California; Naval Undersea Center,
San Diego, California; United States
Army Engineers, Waterways Experiment
Station, Vicksburg, Mississippi.

Publications Brown, B.E., Computer Graphics as a
Tool for Engineering Analysis. In
Proceedings of the CIFE Symposium,
University of California, Lawrence
Livermore Laboratory, 1974.

Brown, B.E., Structural-Analysis Movie
Post-Processor User's Manual,

University of California, Lawrence
Livermore Laboratory, UCID-30097
Oct. 1975.

Christiansen, H.N., Brown, B.E., and
McCleary, L.E., A General Purpose
Computer Graphics Display System for
Finite Element Models. In 46th Shock
and Vibration Bulletin, Naval
Research Laboratory, 1975.

McCleary, L.E., Christiansen, H.N., and
Brown, B.E., MOVIE-NUC User's Manual.
Naval Undersea Center, March 1976.

Christiansen, H.N., Brown, B.E., and
McCleary, L.E., Continuous Tone Images
with Hidden Surface Removal, COMTAL
Image Processing Note 4, COMTAL Corp.
1976.

Brown, B.E., User's Manual for MOVIE.
LASL. University of California, Los
Alamos Scientific Laboratory, Dec. 1976.
LA-NUREG-6532-M.