Lewis/Grant
IN-61
60409
P-143

UNIFICATION OF COLOR POSTPROCESSING TECHNIQUES FOR THREE-DIMENSIONAL COMPUTATIONAL MECHANICS

by

Bruce Charles Bailey

(NASA-CR-180214) UNIFICATION OF COLOR POSTPROCESSING TECHNIQUES FOR 3-DIMENSIONAL COMPUTATIONAL MECHANICS (Cornell Univ.) 143 p CSCL 09B

N87-18998

Unclas

G3/61 43651

Report to NASA Lewis Research Center

Grant Number NAG 3-395

Program of Computer Graphics

Report Number 85-1

Cornell University

Ithaca, New York 14853

Research Supervised by

John F. Abel, Donald P. Greenberg, and Anthony R. Ingraffea

August 1985

UNIFICATION OF COLOR POSTPROCESSING TECHNIQUES FOR THREE-DIMENSIONAL COMPUTATIONAL MECHANICS

by

Bruce Charles Bailey

Report to NASA Lewis Research Center

Grant Number NAG 3-395

Program of Computer Graphics

Report Number 85-1

Cornell University

Ithaca, New York 14853

Research Supervised by

John F. Abel, Donald P. Greenberg, and Anthony R. Ingraffea

August 1985

ABSTRACT

To facilitate the understanding of complex three-dimensional numerical models, advanced interactive color postprocessing techniques are introduced. These techniques are sufficiently flexible so that postprocessing difficulties arising from model size, geometric complexity, response variation, and analysis type can be adequately overcome. They have proven themselves through use in contemporaneous research efforts, and an exemplary user interface is provided to allow their easy use.

Finite element, finite difference, and boundary element models may be evaluated with the prototype postprocessor. Elements may be removed from parent models to be studied as independent "subobjects." Discontinuous responses may be contoured including responses which become singular, and nonlinear color scales may be input by the user for the enhancement of the contouring operation. Hit testing can be performed to extract precise geometric, response, mesh, or material information from the database. In addition, stress intensity factors may be "contoured" along the crack front of a fracture model. Stepwise analyses can be studied, and the user can recontour responses repeatedly, as if he were "paging" through the response sets. As a system, these tools allow effective interpretation of complex analysis results.

ACKNOWLEDGEMENTS

This report is essentially a reproduction of the thesis submitted by the author in partial fulfillment for the Degree of Master of Science. This research on postprocessing has been sponsored by the National Aeronautics and Space Administration Lewis Research Center under Grant NAG 3-395 entitled "Interactive Color Display of Three-Dimensional Engineering Analysis Relults." The faculty investigators for the project are John F. Abel, Donald P. Greenberg, and Anthony R. Ingraffea. The assistance of Jerome F. Hajjar and Paul Wawrzynek is also gratefully acknowledged.

CONTENTS

LIST OF FIGURES

A portion of the figures in this thesis are reductions of computer-generated images. A few of these figures have been reduced to the degree that small lettering is not legible. In each of these cases, the small lettering contains no pertinent information and is included only to illustrate the nature of the computer graphics displays.

CHAPTER 1

INTRODUCTION

Interactive postprocessing is becoming widely accepted as a significant component of computational engineering mechanics. As computer hardware and software costs decrease, an increasing number of universities, government laboratories, and private industries are discovering that numerical modeling methods provide great insight into both research and design problems. Countless analysis programs have been developed in research and commercial environments and are used for a large variety of applications. Although knowledge and use of these numerical techniques has existed and become widespread over the last 20 years, efficient methods for the interpretation of computed results have not existed until recently. Before the advent of interactive computer graphics, preprocessing (the preparation of data for the analysis program) and postprocessing (the sifting and evaluation of the computed response parameters) were found to be the most cumbersome and tedious parts of the modeling process. Computer graphics now allows preprocessing to be performed more easily and has provided a means for interactive postprocessing, an essential tool for the effective interpretation of complex analysis results [1-2].

This thesis describes the second year of development of three-dimensional color postprocessing techniques. The interactive postprocessing implemented to date is currently being used for the evaluation of finite difference, finite element, and boundary integral equation results. It has been useful for many types of analysis ranging from mechanical engineering problems, such as the study of seal-flow behavior, to structural engineering problems involving stress and fracture analysis. Emphasis during the second year of research has been on the development of expanded capabilities for viewing and evaluating behavioral parameters, both to improve the original efforts and to allow postprocessing of a larger variety of analysis problems. Problems incurred during the development and use of these capabilities are discussed, and it is shown how the solutions to these problems lead to powerful adaptations and exciting possibilities for future postprocessing.

1.1 Background

With any postprocessing system, a variety of tools must be provided for the user which enable evaluation of the behavioral parameters. Interpretation of these behavioral parameters is referred to in this thesis as "response viewing," and this process is certainly the most important function of a postprocessor. If an analysis is performed and the postprocessor does not provide the means for adequate viewing of responses, then the postprocessor is useless. For analyses of various size, complexity, and type, different tools are sometimes necessary to interpret the response data.

Postprocessors provide several general functions for the interpretation of analysis data. Together, these allow the analyst to obtain an overall picture of the behavior of the system quickly, without having to perform a tedious search through tabulated numerical output. The postprocessor may represent the response on the surface of the model in the form of color contours. Thus, the analyst can quickly obtain a qualitative appreciation of the analysis results by observing where contours fall close together (regions of high response gradients) or by merely looking for areas where the responses are high or low. By displaying the responses, boundary conditions, and loads in a graphical fashion, the postprocessor also gives vital insights by which to judge the validity of the analysis and perhaps also its accuracy. Once an overall view of the analysis behavior is obtained, detailed quantitative information for areas of interest must be available for evaluation. All numerical information in the postprocessor database is therefore accessible. Finally, a postprocessor provides feedback into the preprocessing stage to facilitate further cycles through the design loop. This information may then be used for purposes such as mesh refinement and redesign.

The postprocessing system upon which this research is based is the result of one year of development. It relies on two fundamental assumptions. First, the geometric description of the model must be in a polygonal format with the response parameters represented at the vertices of these polygons. Second, at most only six-dimensional displays are possible; these are three spatial dimensions, two behavioral parameters, and time [3]. Several translator programs were created to transform data into a format which fits the first assumption. One of these translators is used to transform the data into the desired polygonal format, and another computes the intersections of these polygons to create a representational outline of the model. The remaining translators are each specific to particular analysis programs and are used to read numerical results and then reorganize and smooth the data so that they are represented at the polygon vertices. Once the analysis data are organized in a form which is digestible by the postprocessor, evaluation of the analysis results can take place.

The prototype system has been developed using a VAX 11/780 super-minicomputer which provides a time-sharing environment for interactive program execution. The color raster display is the System 3400 monitor and 12-bit frame buffer produced by the Lexidata Corporation. The display device is a separate unit from the computer, and the VAX system is able to drive a number of such devices simultaneously.

After the first year of development, the postprocessor allowed evaluation of single-load-case, single-load-step analyses for problems of homogeneous material properties. Both finite difference and finite element problems were studied with the system. Several features were implemented which facilitated interpretation of the analysis data. The model could be rotated into any desired position and sliced open with an arbitrarily defined cutting plane. If the shading of the model appeared inadequate due to poorly illuminated surfaces, the "light source" could be interactively moved until satisfactory illumination was obtained. Once the desired view, illumination, and section were obtained, it was possible to display scalar, vector, and second-rank tensor quantities in the form of color contours painted on the surface of the model, where each color corresponded to a different range of response. One of several color scales could be chosen for use in the contouring process. Any individual contour could be changed to a contrasting color to facilitate association of the contour with its particular response range. Finally, the mesh could be superimposed over the contoured image in either a distorted or an undistorted form, where the distorted mesh showed simultaneously both the magnitude and direction of a vector sum of response, such as displacement or velocity. Almost all these functions occurred in an interactive mode, where the minor operations specified by the user were performed almost instantaneously, and major operations were performed in a few seconds. With these fundamental tools, it was possible to interpret the results for some types of numerical analysis.

1.2 Objectives

Analysis problems today are becoming increasingly complex. Fluid flow problems with moving or time-varying boundary conditions occur in the study of seals. Propagation of cracks through solids involves continuously changing topology of the model. Such detailed, dynamic models make it virtually impossible for the engineer to interpret numerical output without the aid of postprocessing.

Some problems involve nonlinear solutions where the results from successive load or time steps must be observed and evaluated. It may be desirable to display several views simultaneously, perhaps the same response contoured on different views of the model where each view corresponds to a different load step, time step, or load case. The engineer may wish to perform a linear analysis which includes several load cases, and then superimpose the responses so that their combined effect can be viewed.

A model can contain discontinuities in response due to an interface between material types. When fracture mechanisms are modeled, response singularities occur at crack tips. All these advanced models present particularly challenging response viewing problems for an interactive postprocessor, but other difficulties are also apparent.

Since preprocessing of such advanced models can be complicated, the possibility of errors or inadequacies in the original problem description and discretization must be considered. With complex three-dimensional problems, the engineer may not know what the results should look like before the analysis, and therefore it may be difficult for him to tell whether the analysis is accurate simply by looking at responses. Hence, the postprocessor must allow easy access to all of the original input data to enable verification of their accuracy. This ability to retrieve the original input data for the analysis is called "attribute viewing," and is an important postprocessing function.

Thus, the main objective in this second year of development of postprocessing is to expand the present fundamental capabilities to allow efficient evaluation of numerical data for complex analysis problems. To realize this objective, enhancement of current response and attribute viewing capabilities is necessary. While this objective is paramount, the specific principles outlined during the first year of development are observed as well. These are reiterated in short here:

(1) Generality of application. A postprocessor should be able to accept problems from a wide range of engineering fields.

(2) Flexible image manipulation. The user must have complete control over the viewing of objects, domains, or systems analyzed.

(3) Interactive graphical capability. A postprocessor should be as rapidly responsive as possible, and its operation and control should be natural and "user friendly."

(4) Transportability of software. An attempt to ease transportability to other display devices should be made.

1.3 Scope

Countless minor refinements have been implemented in the current system which are far too numerous to list. Virtually all these refinements increase either the interactivity or the usability of the postprocessor. Focus in this thesis is placed on algorithms which have a major impact on response viewing, although if these algorithms are useful to functions other than response viewing, they are discussed with respect to these other operations as well.

After completion of the second year of research, the postprocessor can be used to evaluate results from finite difference, finite element, or boundary element (boundary integral equation) analyses. The database capacity has been increased so that large problems may be viewed; recently, problems with up to 5,000 nodes, 800 elements, and 15,000 degrees of freedom have been evaluated on the system. The analyst can "hit-test" (point to the three-dimensional image of the model) to extract most types of geometric, attribute, mesh, or response information from the database. Selective listed hardcopy of the geometric and response information can be obtained if desired. The analyst can store any current image on the screen for rapid retrieval at a later time.

Also at present, discontinuous responses due to multiple material types or geometric discontinuities may be contoured, including responses which become singular. The variation of response ranges in the color scale used for contouring may be manipulated by the user, so that cutoff values or a nonlinear variation may be selected; for example, an upper cutoff limit may be specified, above which all responses will be contoured with a single color. The variety of response parameters which may be viewed has been increased so that traction, stress intensity, strain energy density, effective stress, effective strain, effective error stress, and effective error strain may now be viewed. Results from a stepwise nonlinear analysis or a linear analysis with multiple load cases may be displayed, and a single response may be recontoured repeatedly as if the analyst were "paging" through the response sets. For a linear analysis with multiple load cases, different load cases may be superimposed. Finally, it is possible to isolate separate regions of a complex model by extracting groups of elements that are of interest. These regions are called "subobjects" and are treated as independent entities which can be inspected in detail.

Postprocessing has often been thought of as the last step in the preprocessing-analysis-postprocessing sequence. However, in a larger sense, postprocessing should not only provide the analyst with insight into a complicated analysis but should also provide direct feedback into the preprocessing phase. All of the above mentioned response viewing capabilities facilitate the interpretation of numerical data, but two of these improvements may also eventually provide useful information to a preprocessor. It may be possible, for example, to use the computation of error quantities for interactive mesh optimization. Subobjects may provide a means for refined analysis, where the response data associated with the subobject are used as input in an analysis which "zooms in" on the region which it defines. In the future, postprocessing should no longer be thought of as the last step in the analysis sequence, but instead as one vertex in the preprocessing-analysis-postprocessing-redesign loop. Future research performed in the area of postprocessing should focus on closing this loop.

1.4 Organization of this Thesis

The evolution of postprocessing techniques during the second year of development is discussed in the second chapter of this thesis. The instrumental role of the research environment in which development took place is described because parallel utilization of the evolving postprocessor revealed limitations in it. It is shown how these limitations led to necessary and powerful adaptations for response viewing. Additional features implemented are also outlined. The third chapter shows how the use of existing interactive graphics techniques and additional response viewing tools combine to create a versatile, robust response viewing capacity. The organization of the response viewing page is explained and is found to provide an effective interface between the analyst and the postprocessor. A short scenario is also given which helps to describe the response viewing capabilities in greater detail.

In Chapter 4, the major algorithms implemented during the second year of development are described in detail. The fifth chapter uses several example analysis problems along with diagrams and photographs to illustrate how the present implementation is used to obtain a rapid understanding of a complex numerical analysis. It is also shown how some of the existing capabilities are already being used to provide feedback into the preprocessing stage (to "close the loop"). Chapter 6 summarizes this research to date, and suggests an expanded role for interactive postprocessing. The newly developed postprocessing techniques are shown to fit into such a revised scheme, and speculations upon the future of numerical modeling under such a scheme are offered.

Finally, two appendices present future considerations. Appendix A describes a hardware change which is being considered for the prototype postprocessing system. Reasons for consideration of this change are discussed, and the resulting adaptations which will be necessary are explained. Appendix B speculates on postprocessing in a supercomputing environment. Timing statistics are presented for a few of the more computationally intensive postprocessing algorithms in an attempt to find future "bottlenecks."

CHAPTER 2

EVOLUTION OF A THREE-DIMENSIONAL POSTPROCESSOR

Although the original intention was to follow the suggestions for research proposed after completion of the first year of work, certain unforeseen circumstances and problems catalyzed developments of new postprocessing techniques. When the system was utilized in parallel computational mechanics research and discovered to be restrictive in certain respects, adaptations were made to eliminate the restrictions. Some of the techniques found to be most useful in interactive postprocessing were developed in this manner.

Instrumental to the evolution of the postprocessing system has been the engineering research environment in which development took place. The prototype system has been used to interpret numerical results for engineering problems since the beginning of its second year of development. Another contemporaneous research effort throughout this time has been an attempt to understand fracture mechanisms within steel girders subjected to cyclic loading. Linear elastic analyses were performed on large (4,000 nodes, 500 elements) finite element models, whose results were continually being evaluated with the postprocessor. Observations and suggestions made during this evaluation process were considered carefully. When difficulties were discovered which inhibited or prevented adequate interpretation of analysis results, solutions to these problems were typically given a high priority. This continual use served as a sort of "proof testing," and many useful features resulted from discussions with the researchers using the system. Thus, the postprocessor has evolved as deficiencies detected led to adaptations to the system. This chapter discusses the limitations which were discovered during development and then presents the adaptations which were made in response to these deficiencies. Finally, features which are not a direct result of the evolution process are presented. All these implemented features are described in greater detail in Chapters 3 and 4.

2.1 Postprocessing Limitations

Limitations encountered in the development of postprocessing techniques fall into four categories: insufficient memory capacity, inadequacies in contouring, response over-smoothing, and slow response due to excessive database size.

The first limitation discovered was a limited memory capacity in the prototype system. Some of the first analysis problems evaluated were so large that allocation of enough memory for all of the numerical data was difficult. When a model is sectioned (cut into subassemblies), an undetermined but significant amount of data is generated and inserted into the database. Also, the amount of data continues to increase if successive cuts are made. For this reason it is impossible to predetermine the amount of necessary storage capacity, and the database must therefore be oversized. Clearly, interactive postprocessing requires a great deal of memory if the voluminous response data from a large analysis is to be kept available for fast and easy access by the user.

Serious problems with the existing contouring scheme became apparent when the results of large analyses were contoured. Large problems may have wide ranges in response values, where the highest responses are concentrated in regions near the applied loads and the lower responses are spread throughout the remaining model. When a limited number of color contours is used, the contours tend to "bunch" in the high gradient regions, thus leaving the response in the remainder of the model to be described with only one or two contour colors, as shown in Figure 2.1. If the areas of interest in the model are not in the vicinity of loading, it may thus be impossible to interpret the response information by studying the contoured image. Even if the analyst is interested in the region where most of the contours exist, the region may be too small to see. The hardware zoom does not always overcome this problem, because high magnification is necessary and the resolution in the zoomed contoured region may not be high enough to provide any useful information.

Another problem with the contouring process, more subtle than the previous restriction, is apparent when one tries to compare the results from two or more analyses. Response distributions in models with similar

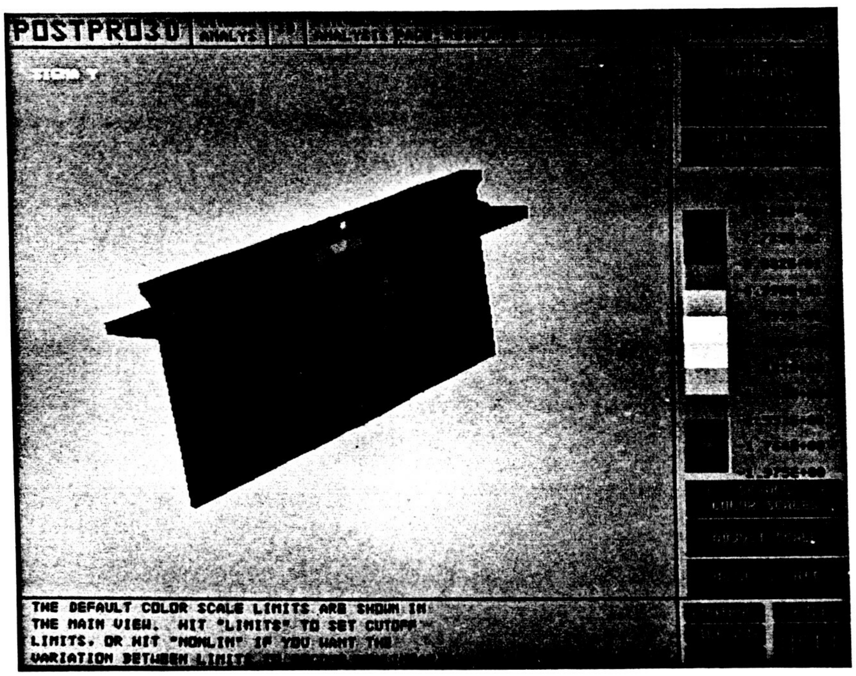

Figure 2.1 Example of poor contour distribution in a complex model

geometries may appear similar even if the response ranges are different. Because the numerical values associated with the contours are correct, this problem is a visual difficulty. Several finite element analyses were performed on models of girders with and without stiffeners. In one analysis a vertical load was applied to a model with no stiffener and in a second analysis the same load was applied directly over a stiffener. When the vertical stresses in the web are displayed for each of the loading conditions, the response distribution is a "bulb" of stress contours radiating outward from the point of load application. Figures 2.2 and 2.3 show the responses contoured on the girder web for the two respective models, but the stiffener in Figure 2.3 is not shown. Although the two figures look similar, examination of the contour numerical values shows that when the load is over the stiffener the stresses in the web are reduced. However, the distribution does not appear to change because, when each model is contoured, its database is first scanned to find the maximum and minimum values for the desired response, and the contours are distributed throughout this range. Although the ranges are different for the two cases, the spatial variations of stresses are nearly the same.

The third limitation is apparent when response smoothing is performed at geometric discontinuities, and information concerning the discontinuous responses at these locations is lost. As discussed in Section 1.1, translators read in and reorganize analysis data so that they are in a format suitable for postprocessing. In finite element stress analysis, stresses and strains are usually output at integration

ORIGINAL PAGE IS
OF POOR QUALITY

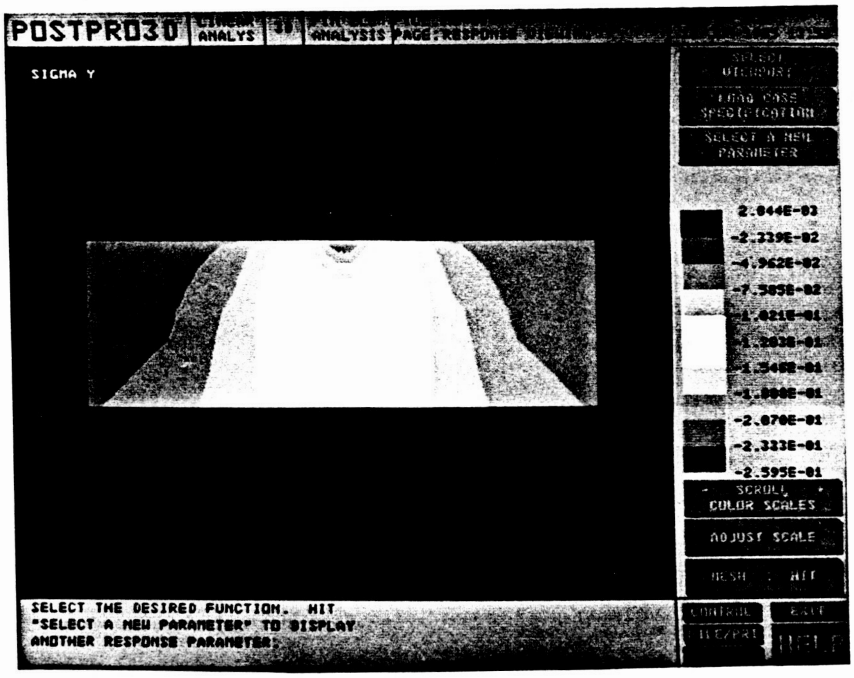

Figure 2.2 Contoured girder web with no stiffener

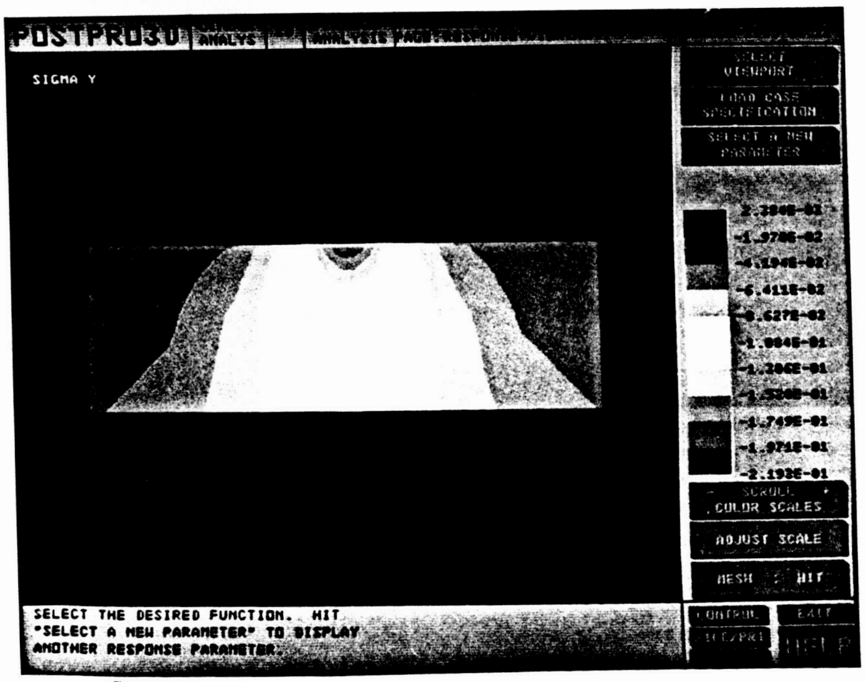

Figure 2.3 Contoured girder web with load above stiffener

(Gauss) points, and must be extrapolated out to the element nodes. The nodes then become the polygon vertices so the values associated with these nodes are critical in the representation of analysis data. However, errors in calculated stresses and strains of adjacent elements usually occur. When these values are extrapolated to the element nodes, a single node may have several different values associated with it. For this reason, the nodal values are averaged to obtain a single response value for each individual node.

Because numerical modeling procedures typically discretize bodies and evaluate responses at sampling points within, processes such as smoothing and nodal averaging are usually necessary if any reasonable display of the response distribution is desired. Some cases have been found, however, for which nodal averaging may introduce significant errors into the responses that are used for postprocessing. When nodes are located at geometric discontinuities, e.g., re-entrant corners, averaging can lead to glaringly incorrect stress distributions. As an illustration, consider the T-shaped model depicted in Figure 2.4. Suppose an axial load is applied directly above the stem and the horizontal projections remain unloaded. It is clear that if the nodal stresses at the geometric discontinuity are averaged, interpolated vertical stresses in the element that joins the horizontal and vertical projections will be too low, while the interpolated vertical stresses in the adjoining, horizontally projecting elements will be too high. If this model is then contoured, a smooth variation in vertical stress will be displayed across the junction, where a discontinuity in response

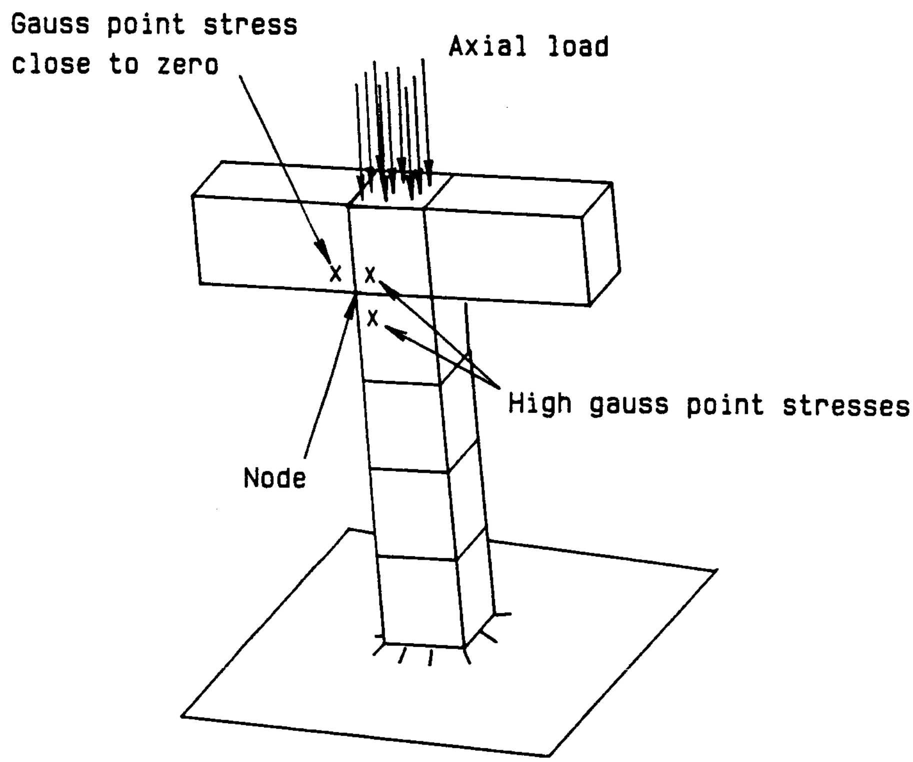

Figure 2.4 T-Shaped model illustrating errors introduced by nodal averaging

should occur. (The model used here as an example is coarse, and an extremely refined analysis would eliminate much of this problem.)

Finally, postprocessing can cease to be interactive when large analysis models require the processing of large amounts of data. Increasing complexity in an analysis model requires that increasing amounts of data be managed in the postprocessor database. For a large model with a complicated geometry, large sets of coordinates must be transformed each time the model is rotated to a new orientation, and the number of polygons that must be rendered may be enormous. As the database becomes unwieldy, complicated operations such as sectioning are no longer interactive at all, and the effectiveness of the postprocessor is reduced. This loss of interactivity presents a major difficulty, as interactive postprocessing is particularly indispensable in the evaluation of the very problems which are the cause of the loss.

2.2 Postprocessing Adaptations

Each of the major limitations discussed in the previous section was addressed in the process of continued development of postprocessing techniques and capabilities.

The restriction of database size was the first problem tackled. The computing environment in which the postprocessing techniques are being developed is a virtual memory environment. Because there are more virtual than core memory addresses, enough virtual memory can be allocated so that large problems are now accepted by the postprocessor.

Extra memory is allocated in addition to the memory necessary for the storage of responses and geometric information, so that the database can be expanded when sectioning occurs. The use of virtual memory for the large postprocessor database has proved to be flexible, and the size limit for an analysis has yet to be reached.

The difficulties encountered when contouring a complex model led to an unexpected but powerful and useful addition to the system. Since the contours in a large model often are crowded into regions of high gradients and these regions are not always of interest, a method was developed to isolate regions of elements that are of interest. This technique is called "element extraction," and the isolated regions of elements are referred to as "subobjects." To evaluate the behavior of a large model, it is often useful to study regions independently of one another, and the creation of various subobjects from the "parent" object can be advantageous. Also, because analysts are sometimes accustomed to thinking about problems as collections of parts, subobjects are a natural way of picking apart and understanding the analysis results.

To create a subobject the user can isolate a particular region of elements by interactively moving a set of six surfaces until the region of interest is "boxed." Elements which have centroids either within or outside the region defined by the enclosing surfaces may be chosen to become parts of the subobject, while all other elements are rejected. Alternatively, subobjects may be isolated by selecting all elements of a designated material type. Once a subobject is requested and defined, the user is prompted for a name which is assigned to files for the subobject.

The postprocessor then re-activates the boundary and outline extractor algorithms to compute the subobject's polygonal and outline data. Next, all geometric and response data associated with the subobject are extracted from the current database and used to create the necessary geometry and response files so that the subobject can be treated as a separate entity. At this point, the subobject extraction is complete, and the subobject files can be read and treated as if they were ordinary translated analysis results.

The postprocessor treats subobjects in the same manner as it treats the parent, or top-level, model. All the original postprocessing functions can be performed on a subobject, subobjects can be made from subobjects, and so on. However, when a postprocessing function is used that refers to node or element numbers in the subobject, the numbers displayed are those from the original analysis. The number of nodes and elements in the original analysis or remaining in the subobject may also be displayed. When the subobject is contoured, the database is scanned as usual to find the maximum and minimum response values, but the limiting values found are only for the elements in the subobject. Therefore, the full range of contours is available for displaying the response information of the subobject alone. By creating subobjects, the user can effectively force contours into an area of interest by extracting the elements for that area. Of course, there is a limit to the increase of information which can be obtained by use of this technique due to the limited number of sampled points per element in the original analysis. For example, if a single element is extracted, all the contours will be used on that element but most of the displayed information may be meaningless.

Although faster than sectioning, the creation of subobjects is not always interactive (the creation of a large subobject containing several hundred elements, one thousand nodes, and a few load cases takes about three minutes on the VAX 11/780). Nevertheless, many advantages are gained through the use of subobjects which outweigh the time necessary for their creation. Subobjects contain information from the translated, original analysis results. Time consuming operations such as response smoothing, which occur during translation, are not repeated when the subobject is created; instead, the information corresponding to the subobject is taken directly from the parent database. Once a subobject has been created, the user can keep the information for use at later times so the actual element extraction is performed only once.

If subobjects are used to eliminate single layers of elements in the parent models, these layers of elements can effectively be "peeled" away to expose interior layers. If elements of selected material types are extracted from a problem containing several materials, the responses on material interfaces can then be inspected. Behavior of some models can be more easily understood by using subobjects to break them up into separate parts, and studying the parts independently of one another. In a complex model, far more information can be obtained by the contouring of subobjects than can be obtained by contouring the parent, and the contoured image of the subobject is larger because it is scaled to fill the screen. When the postprocessor is used to evaluate behavior of individual subobjects, the database contains less information than it would have with the parent model, and interactivity is therefore increased dramatically for all postprocessing operations, such as sectioning.

Finally subobjects and the information associated with them provide a possible means of feedback into the preprocessing stage. The geometry and response information of a subobject can be used as input into a more refined analysis. Forms of this technique are currently being employed [4]. No direct connection has yet been established between subobjects and preprocessors, but the following example is cited to illustrate the possibility of such an application.

In the previously mentioned fracture studies, a boundary element analysis program was available for use in modeling actual crack propagation, but was not economical for the analysis of problems with such high surface-to-volume ratios as the girders. Therefore, large scale finite element analyses were first performed on models of the steel girders. Subobjects were created in regions of high major principal stress, and hardcopies of the subobject geometry and response information were obtained. This provided geometry and response information for use in the creation of boundary element models. The subobject geometry information was used for manual construction of the boundary element geometry, and the response information was converted into the actual input for these fracture analyses. This technique allowed the use of the boundary element method on a reduced geometry, for which it was more economical. Thus, by starting with the large model and focusing on a specific region for subsequent analyses, a "zooming" of analysis is possible. In this sense the technique has unified the two numerical techniques in this research environment.

Resolution of the stress bulb problem led to a simple but useful technique. When the user is allowed to input cutoff values for the color scale, all response values above the top limit or below the bottom limit are contoured with the top and bottom contour colors. Cutoff values can be held constant by the user for different load cases, so the interior range of the scale remains fixed and contouring is forced to occur consistently between problems. In Figure 2.5, the results shown previously in Figure 2.2 are shown again. Cutoff values are specified for the girder web so that the interior range of contour values in Figures 2.3 and 2.5 is the same. The stress bulb for the unstiffened web is enlarged, and a better visual comparison can be made. Specification of cutoff values is useful for other types of problems as well. The girder subassembly of Figure 2.1 is recontoured and shown again in Figure 2.6, with a lower cutoff range specified to force more contours into the region of the web. All responses below the cutoff value are contoured with the bottom color, and the remaining contours give a better indication of the behavior in the girder web.

To contour discontinuous responses across a material interface, all nodes along the interface are duplicated. In this way, multiple response values can be stored at single locations. This may be visualized as the creation of zero thickness elements between elements of different material types, which prevents the nodal averaging process from occurring

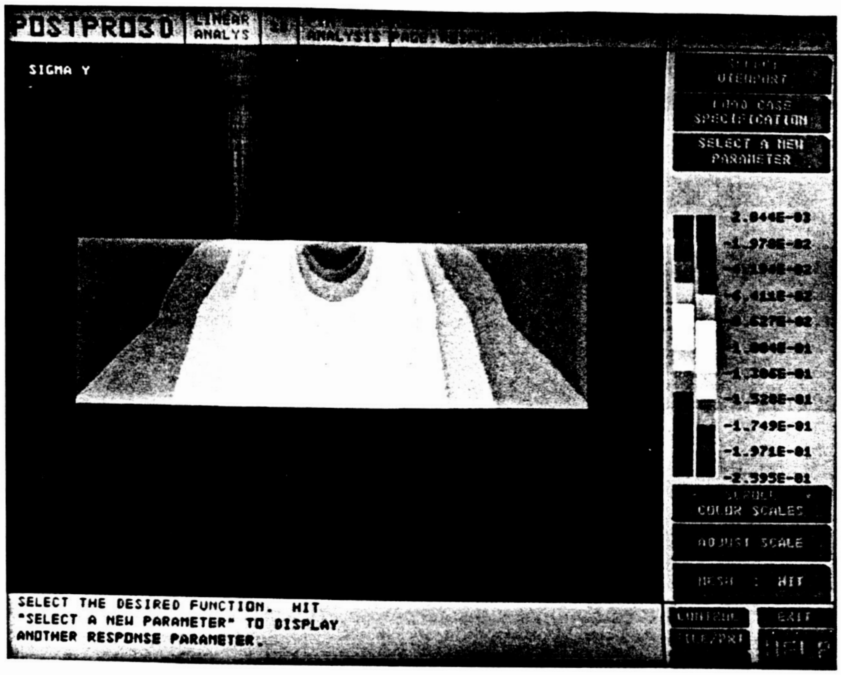

Figure 2.5 Contoured girder web with no stiffener and cutoff value specified (cf. Fig. 2.2)

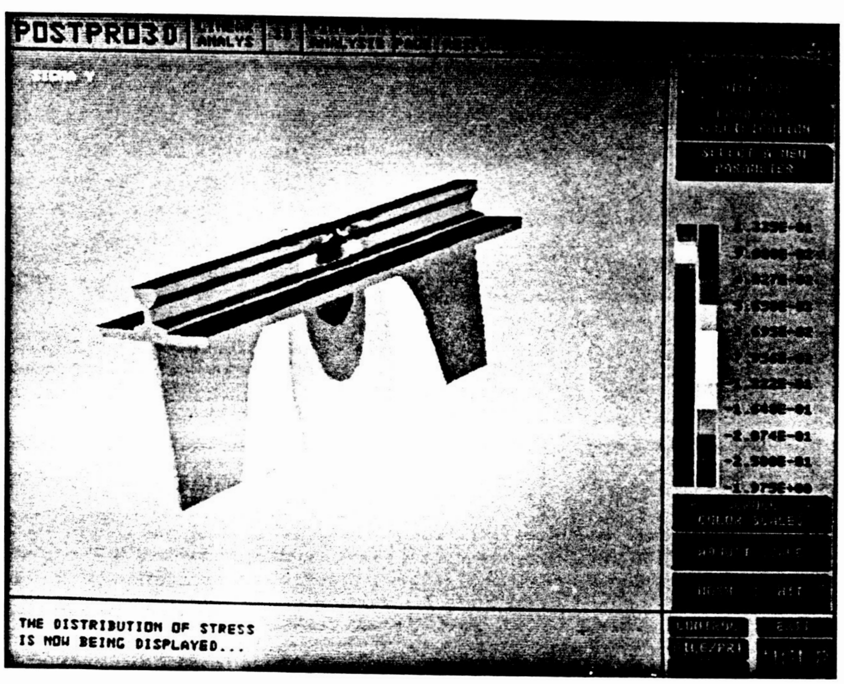

Figure 2.6 Contoured model with lower cutoff value specified (cf. Fig. 2.1)

between the elements. The technique serendipitously proved to be a solution to the problem of averaging at geometric discontinuities. With reference again to the T-shaped model of Figure 2.4, the contouring problem depicted could be avoided by assigning one material type to the entire vertical section of the model, and a separate material type to the horizontal projections. Note that the material properties associated with these two types could be identical. This permits an approach to preprocessing in which different attributes are assigned to different sections of the model so that averaging can be avoided at the junctions that create geometric discontinuities. For example, in the T-shaped model, the attribute assigned to the vertical section might be called "web," and the attribute assigned to the horizontal sections might be called "flange," but the numerical values of the properties for the two materials would be the same.

The size of the database was excessive because it was originally designed to store the response information for all load cases simultaneously. With a linear analysis involving several load cases, it was intended that pointers would be maintained so that desired information in the database could be located for any load case. Maintaining such a large database reduces interactivity prohibitively. Therefore, to avoid processing more response data than absolutely necessary, the response section of the database was redesigned so that only a single "response set" would be contained in it at any one time. (A response set refers to all scalar, vector, or second-rank tensor quantities for all nodes in the model. In a nonlinear or dynamic analysis, the response set contains all response information for a single load or time step. For a linear analysis, the response set contains the information for a single load case.) When analysis results are translated for postprocessing, all response information is organized by the translator into a form directly acceptable by the postprocessor database. These response sets are stored sequentially in a file, and because the information is in the exact format required by the database, the sets can be "swapped" into and out of the database. The swapping procedure is sufficiently rapid to make it possible to page through the different response sets. Only desired information is stored, so the size of the database is kept to a minimum.

2.3 Additional Postprocessing Implementations

Other refinements have been implemented which enhance the system's overall capabilities. Although these may not have the impact of the adaptations mentioned in the previous section, they have improved the system by increasing interactivity, adding flexibility, and allowing a greater number of analysis types to be handled.

An improvement was introduced which allows rotations of models with complex geometries to be performed more quickly. Instead of multiplying all coordinates in the database by the transformation matrix necessary for an increment of rotation, only the coordinates of nodes included in the representational outline are updated. When the outline information is read in by the postprocessor, a list is assembled which contains all of the node numbers included in the outline of the model. This list is retained in the database so that whenever rotations are performed, the nodal coordinates of the outline itself are the only coordinates which are transformed for each increment of rotation. The increase in the speed of rotations is noticable, especially for large models with relatively simple outlines.

A new printout menu page allows the user to obtain selective hardcopy of any analytical, geometric, or response information that is in the database at the time the hardcopy is requested. For example, the type of analysis can be output along with the nodal coordinates or element connectivities. Any or all analysis information that is contained in the database is available upon request.

The printout page is activated in several cases. One of the permanent menu switches in the lower right portion of the menu area is labeled FILE/PRI. At any time during the postprocessing session the user can select this menu item, and the printout page is entered directly. He can then select the information to be printed and have the printout created. The user may then return to the previous menu page directly. Also, whenever the user hits the EXIT or SELECT NEW PROBLEM buttons, the printout page is automatically displayed so that hardcopy can be obtained if desired before exit or re-initialization of the database, respectively.

Below the FILE/PRI button in the permanent menu area is a new button labeled SNAP. When this feature is used, a file is created which saves all pixel information on the screen at that particular time. This file may then be read in by a utility program which re-displays the image. Unfortunately, the files are not run-length encoded and are therefore very large. Required memory for the storage of an excessive number of these files prevents their liberal use; however, they have proven to be useful for such short-term purposes as preparation for photographic recording. Thus, at any time during the postprocessing session, the user may save the current picture for fast retrieval at a later time. Approximately five seconds is required for the image to be recreated.

A single algorithm has been developed which is useful for many different postprocessing applications. By pointing to a specific location of interest on the three-dimensional image of the model, precise geometric, response, mesh, or material information can be extracted directly from the database and displayed. This function is known as "hit-testing" and is used on the attribute, coordinate system, and response viewing pages for direct display of data. In the attribute viewing page, the user may point to individual elements or nodes so that their numbers are displayed. When hit-testing is used for the evaluation of material properties, element material characteristics are displayed below the main view; in this case, the exterior surface of the chosen element is highlighted as well. In the coordinate system page, when the user points to the model, the coordinates of the node closest to the cursor are displayed in a cartesian, cylindrical, or spherical coordinate system. Finally, in the response viewing page, hit-testing is used to extract behavioral information for the currently contoured response (see also Sections 3.1 and 4.1).

A new translator has been developed which reorganizes output data from a boundary integral equation analysis program into the format accepted by the postprocessor. In these boundary element analyses, sampling information is recovered for only the surface of the model. Responses for the interior of the model are not computed because they are seldom of interest and because their recovery is computationally intensive. This modeling method is currently being used for fracture propagation analysis, where the element discretization is easier to produce and modify than with finite elements [5]. In preparation for postprocessing, because the surface information is described by the boundary elements themselves, the elements can be used as the polygonal representation of the model and a boundary extraction does not need to be performed.

Infinite plane sectioning is not implemented for boundary element models simply because no interior results are available from the boundary element analysis program. However, element extraction is permitted, and the element extraction process is less computationally intensive for boundary elements than it is for solid elements because only the outline extraction algorithm needs to be activated when creating a boundary element subobject. Note that if not all elements in a particular domain are included in a subobject, the subobject appears hollow because of the lack of an interior discretization and of internal information.

Responses which are organized by the boundary element translator are stress intensity factors, displacements, and tractions. Normally, traction components which are nonzero on one surface of a boundary element model will be discontinuous at the edges of that surface. For this reason, no averaging is performed and node numbers are duplicated at such edges. Discontinuities in tractions are thus preserved when contouring takes place, as described in Section 2.2.

A significantly different contouring approach is used for for the representation of stress intensity factors along a crack front (see also Sections 3.1 and 4.4). Only the outline of the body is displayed so that the interior is visible, and the selected stress intensity factor variation is shown in the form of color variations along the curved line representing the crack front. For an irregular and arbitrary crack, the crack front is seen as a "contoured" line wandering through the body in three-dimensional space (see Figure 3.1).

When modeling the propagation of a crack with the boundary element method, one usually employs two domains for the representation of the body. One domain is always located on one side of the crack surface, while the other domain lies on the other side. This allows the formulation of equations for the domains coupled only at the interface between the two domains. The postprocessor can be used to "scroll" between the two domains, thus revealing the elements on either side of the crack, since each domain is entirely enclosed by boundary elements. If both domains are included in the database for a boundary element analysis (i.e., a subobject has not been created to isolate a single domain), this technique is necessary when contouring responses which are near the crack front to prevent the response information from being hidden between the two domains.

The adaptations implemented as solutions to the discovered limitations of the system have been integrated with other newly implemented features. The postprocessor now boasts a powerful set of tools useful in the evaluation of numerical data from several types of analysis. These tools complement one another and allow the analyst to approach the interpretation of data in a number of ways. Typically, many of these tools may not be necessary for a particular problem. In certain cases, however, some are invaluable and can be applied naturally and with ease. The next chapter describes how the tools complement one another, and how their easy use provides a powerful approach to response viewing.

CHAPTER 3

RESPONSE VIEWING

Response viewing is the central function of postprocessing. If response viewing capabilities are to be adequate, tools must be available which allow fast, interactive interpretation of behavioral parameters for problems of all sizes and geometric configurations. These tools should be presented and organized in a logical and easy to use fashion. Several such tools have been developed which together make up an interactive system for the display of meaningful response information for a large variety of analysis problems. This chapter describes these response viewing tools, and how they have been integrated into a powerful, interactive response viewing system.

3.1 Response Viewing Capabilities

This section elaborates on the use and flexibility of response viewing tools. Color contouring is the main technique used for response viewing, and is enhanced by access to quantitative information, display of discontinuities, representation of crack fronts, color scale manipulation, and zooming by creating subobjects. Other tools facilitate the interpretation of multiple response sets. Sets can either be "scrolled" or selected directly, and color scale limits are chosen to be consistent for a particular analysis. Finally, additional response parameters are provided for interpretation of behavior and evaluation of analysis quality.

Typically, after reading in the translated results from an analysis problem and specifying a view of the model, the user will enter the response viewing page. The most commonly used technique for the display of response information is that of color contouring. Once contouring of the image has been performed, many options are available to the user, and his choice of actions can follow any course that he feels is necessary.

To maintain three-dimensional perception, shades of color contours are varied according to the orientation of the surface with respect to the direction of illumination. If the association of a contour color with its corresponding numerical value is not obvious because the contour changes shading as it "goes around corners" of the surface, interactive contrasting (by which the selected contour color is darkened) eases this association [3].

If more precise numerical information is desired, a new hit-testing function may be activated by selecting the HIT command. Now when the cursor is used to point to the contoured image of the model, instead of contrasting the selected contour, the node nearest to the cursor is highlighted and that node's global number and response value are extracted from the database and displayed in the message box below the main view. Displaying the mesh is sometimes helpful when hit-testing, because the nodal locations for which the responses are stored become obvious.

If the analysis model contains multiple materials, contours may be discontinuous across a material interface, and hit-testing will quantify the discontinuity. The user may point to either side of a node on the interface to display the nodal response associated with the material on the selected side of the interface. The response values shown below the main view differ for opposite sides of the discontinuity, but the node numbers shown are the same. The node duplication process described in Section 2.2 is thus transparent to the user and remains an artifice useful for internal management of discontinuous response data. A similar situation occurs when responses are discontinuous because of geometric discontinuities. Finally, traction components which are contoured from a boundary element analysis are usually discontinuous at corners and may be discontinuous at any interelement boundary; and hit-testing on either side of such a corner or interelement boundary also reflects these discontinuities.

Boundary element analysis is sometimes used to model crack propagation through solids. If the user wishes to display the distribution of stress intensity along the crack front, the variation is displayed as a three-dimensional "contoured" line, which corresponds to the crack front as it exists in three-dimensional space (Section 2.3). Because the crack front is usually hidden in the interior of the solid, only the outline of the model is displayed so that the crack front is revealed. Figure 3.1 shows how hit-testing can be used to display the value of a selected stress intensity factor along a crack front in a boundary element model.

For some large analysis problems, if the entire model is displayed, the contour distribution may reveal only that the majority of response activity is occuring in a high gradient region. In this case most of the model will be contoured with only one or two of the contour colors, as discussed in Section 2.2. At this point the user can hit ADJUST SCALE to input cutoff limits for the contours. By contrasting the bunched contours in the high gradient region or by hit-testing, one can identify appropriate cutoff values for input. When the user re-contours the image, all contour colors (except for the colors beyond the cutoff limits) will be forced into the region where previously only a few contours existed.

Input of cutoff values is particularly useful when a response is contoured which becomes high or singular at certain locations, as is the case with the crack propagation models and with models involving stress concentrations. In both cases, the response components are so high in some regions that nearly all contours crowd into these regions unless cutoff values are specified. If the user desires a nonlinear variation between color and response, he may select ADJUST SCALE and then manipulate the scale variation with a potentiometer. Combinations of cutoff values and nonlinear variations may be input as well. Once he specifies the desired variation, the user must hit DISPLAY CONTOURS again

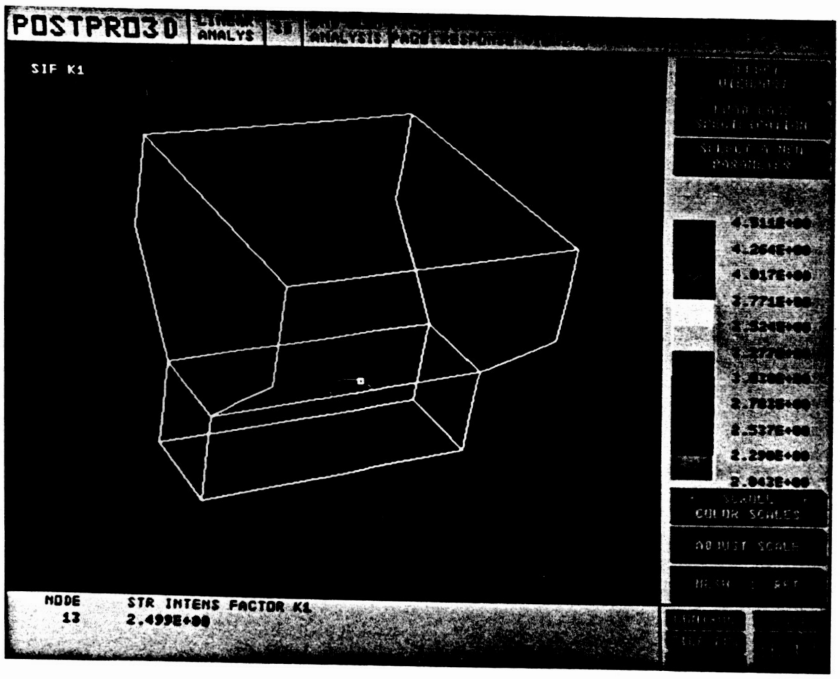

Figure 3.1 Hit testing along a crack front to retrieve stress intensity factors

to re-contour the image. Involved preparatory operations which are performed whenever a response is contoured for the first time are not performed again for re-contouring. Instead, the computed preliminary information is saved unless and until a new response is chosen and contoured. The re-contouring operation is therefore quicker than the original contouring.

Figures 3.2 through 3.5 show the input of a cutoff value and the manipulation of the potentiometer for a finite element model which has a stress concentration. Figure 3.2 shows the contour distribution of sigma x contoured with the default, linear color scale. In Figure 3.3, the user has input an upper cutoff value of stress, and in Figure 3.4 the potentiometer has been used to specify a nonlinear variation between stress increments and color. The resulting contoured image is given in Figure 3.5. After any rescaling, e.g., in Figure 3.5, two adjacent color scales are displayed in the transient menu area on the right of the screen. The left scale shows the proportion of the entire stress range which is given by each contour color and graphically depicts the cutoff and/or nonlinear relationship. The color scale on the right is evenly divided and relates the specific contour colors with their numerical ranges. The actual color scales can be changed as well by hitting the SCROLL SCALES button to specify one of six different color schemes.

If in a large problem the user is interested in a high gradient region, these contours may be difficult to see simply because the region is so small in comparison to the remainder of the model. The region can be enlarged with the hardware zoom, or, if this is still not

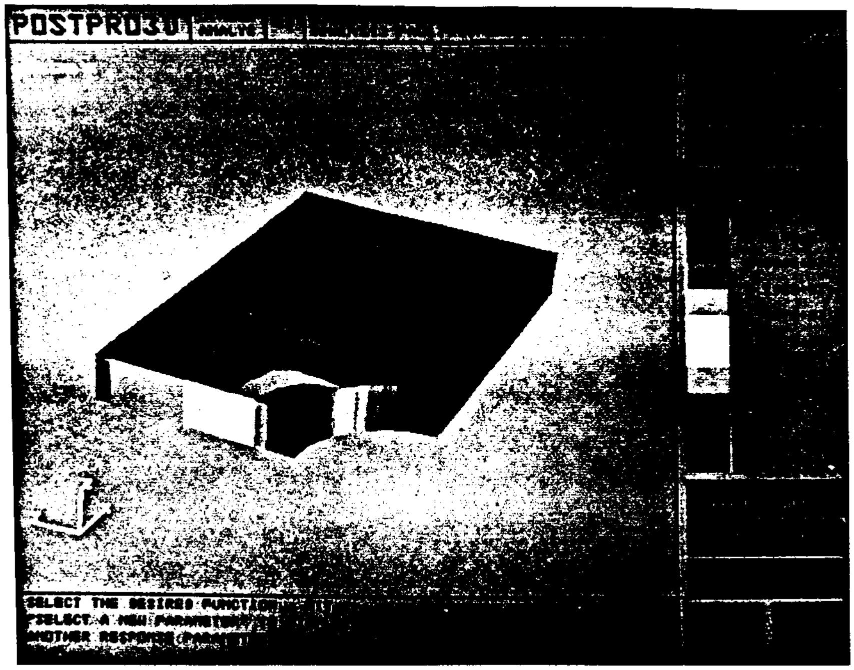

Figure 3.2 Linear contour distribution of sigma x

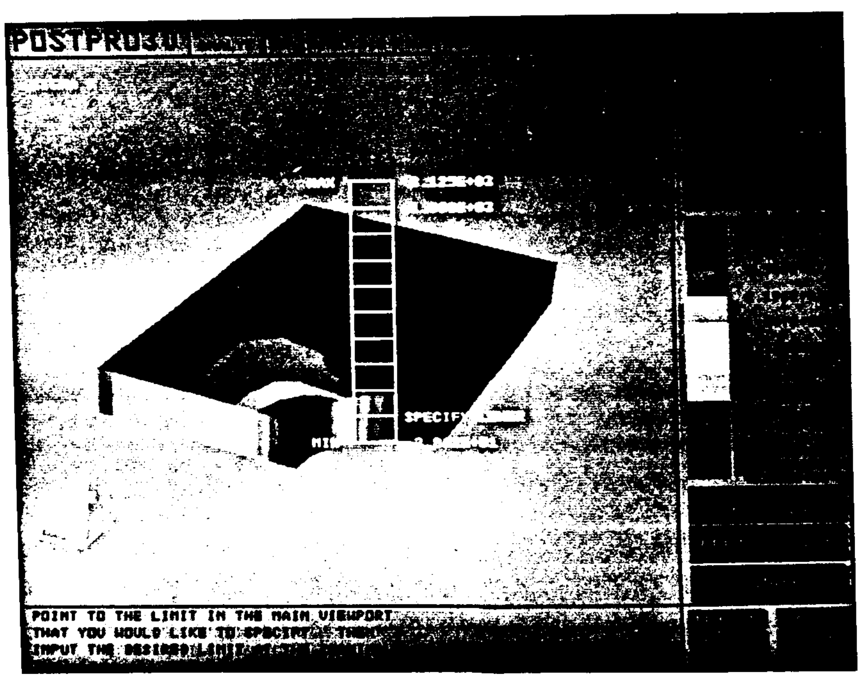

Figure 3.3 Input of an upper cutoff limit for contouring

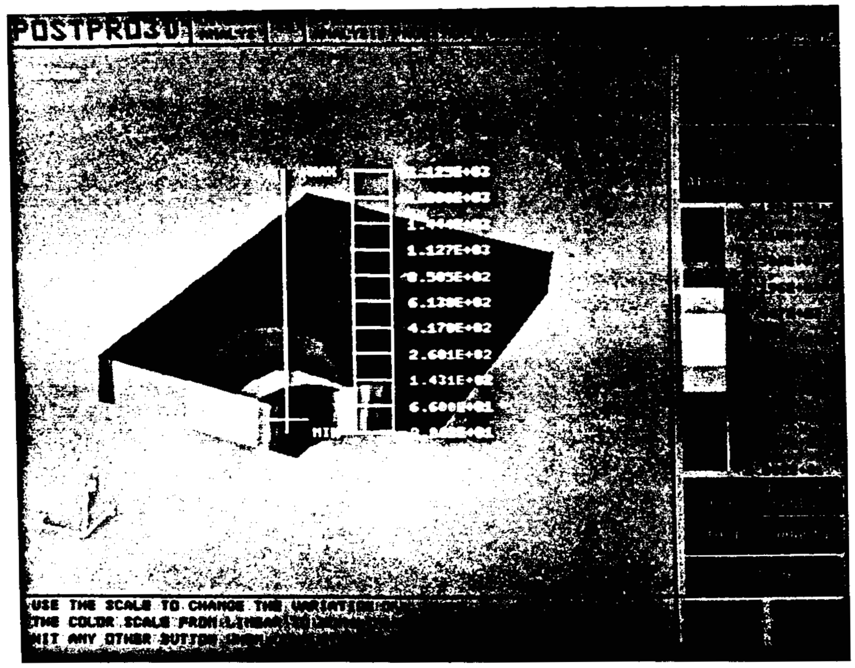

Figure 3.4 Use of the potentiometer for describing a nonlinear variation between color and sigma x

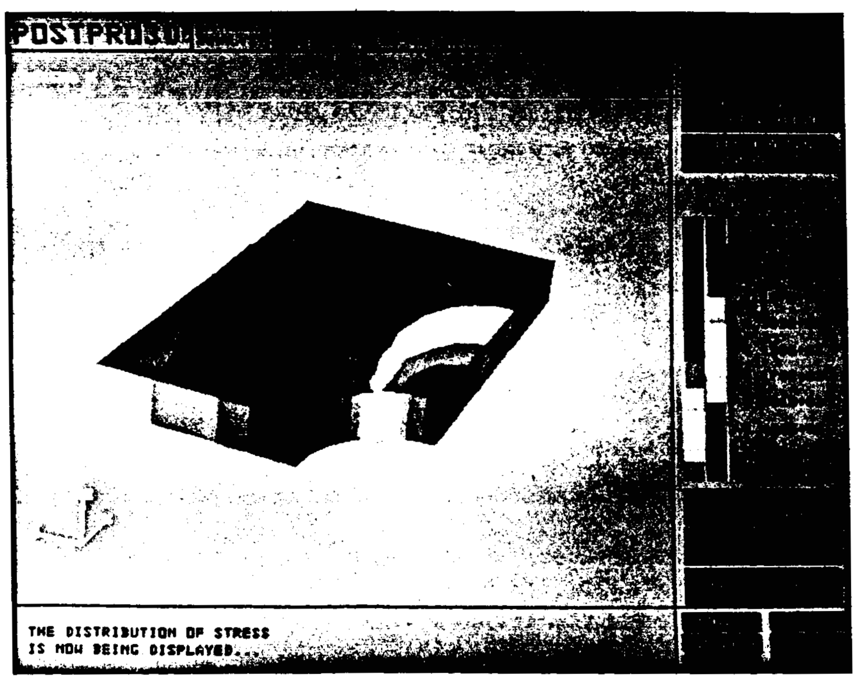

Figure 3.5 Resulting contoured image for the previously input color scale

satisfactory, a subobject can be created. This latter method provides improved contour resolution and makes a small region of interest larger and easier to see.

For a stepwise nonlinear analysis or a linear analysis with multiple load cases, more than one response set is available for interpretation. Once a response is contoured, INCREMENT RESPONSE SET can be hit, which swaps either the previous or the next set of response data into the database (depending on whether the user hits the "-" or the "+" side of the button), and re-contours the same response parameter with the new data. By hitting INCREMENT RESPONSE SET + repeatedly, the user can page forward through the different response sets to see how a particular parameter varies. Figures 3.6 and 3.7 show two successively contoured images of a material nonlinear finite element analysis. The performed analysis is of a welded tension connection, a problem which is more completely described in Section 5.2. The response parameter contoured is the effective stress. The high end of the response scale is contrasted to make it easier to see how yielding changes with increments of load.

If the analysis contains many response sets and a particular set is needed, the load case page can be entered and the desired response set can be selected directly. For a linear analysis, several load cases can be linearly combined or superimposed in this page as well. Once the desired response information is loaded into the database, the response viewing page can be entered again. A simple cantilever has been modeled with three-dimensional finite elements for two load cases, a vertical and a horizontal tip load, and a linear analysis performed. The two load

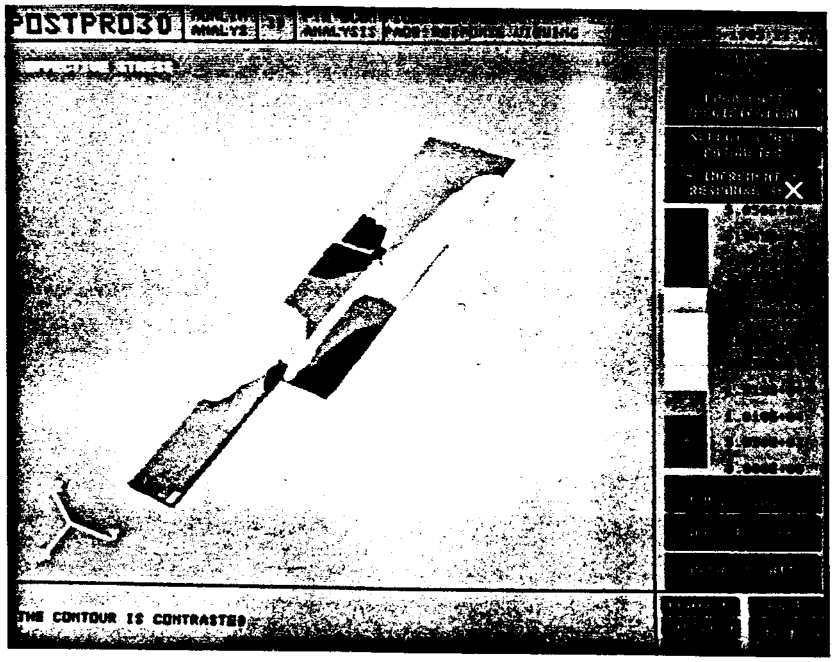

Figure 3.6 Contour distribution of effective stress in the welded tension connection for load step 7

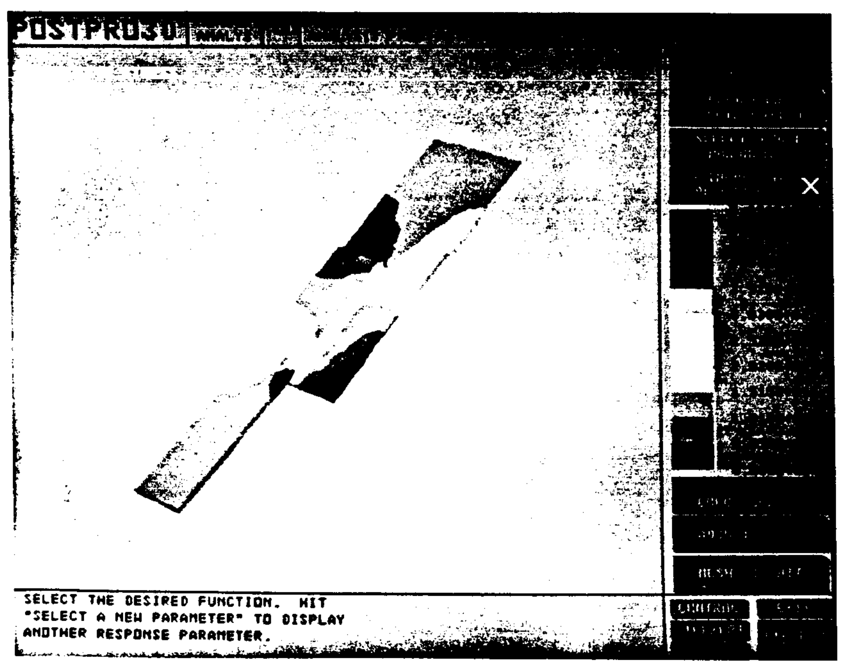

Figure 3.7 Contour distribution of effective stress in previous model for load step 8

cases have been combined and the resulting displaced mesh is shown superimposed over the longitudinal stress, sigma z, in Figure 3.8.

It is significant that default color scale limits vary depending on the type of analysis. For a nonlinear or dynamic analysis the default contour limits are the maximum and minimum response values for the entire analysis. This consistency is necessary if the user wishes to observe how contours spread through the model as the load or time varies. If the user wishes to see a contour distribution with limits that are specific to a single load or time step, these can be selected on the load case page. With a multiple load case linear analysis the default limits are local to each load case, but can be switched to limits which are global to all of the response sets. However, once load cases are superimposed or linearly combined, the color scale limits are always local to the superimposed set and may not be changed.

Several new response parameters have been added to enhance the interpretation of finite element stress analysis. Effective (von Mises) stress and strain, and strain energy density may be selected and contoured. Also, scalar measures referred to as "effective error stress" and "effective error strain" can be used to evaluate analysis accuracy. These "effective error" quantities are computed during translation before averaging occurs. By computing the maximum differences in the stress or strain components being averaged, combining them in the effective stress formula, and normalizing them by the averaged effective stress, a scalar measure for the overall accuracy of the analysis is obtained. In Figure 3.9 the displaced shape of a finite element model of a dam is shown. All

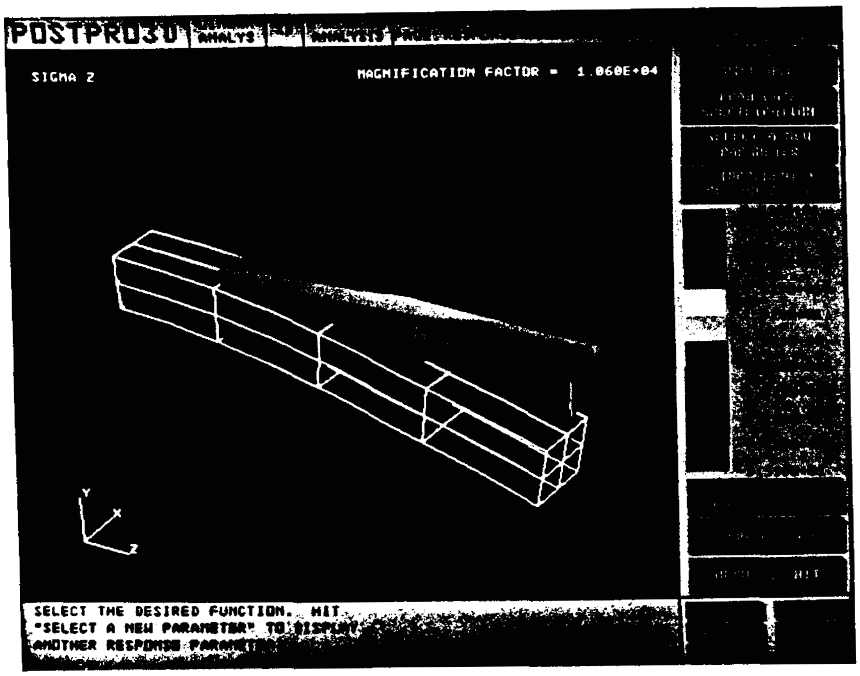

Figure 3.8 Displaced mesh superimposed over the bending stress sigma z for two combined load cases

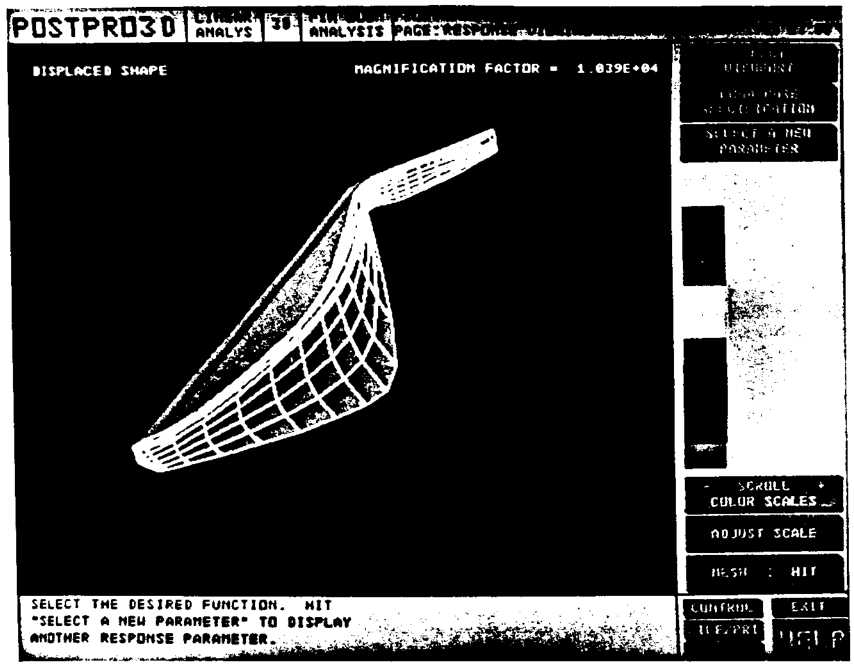

Figure 3.9 Displaced mesh of a finite element model of a dam

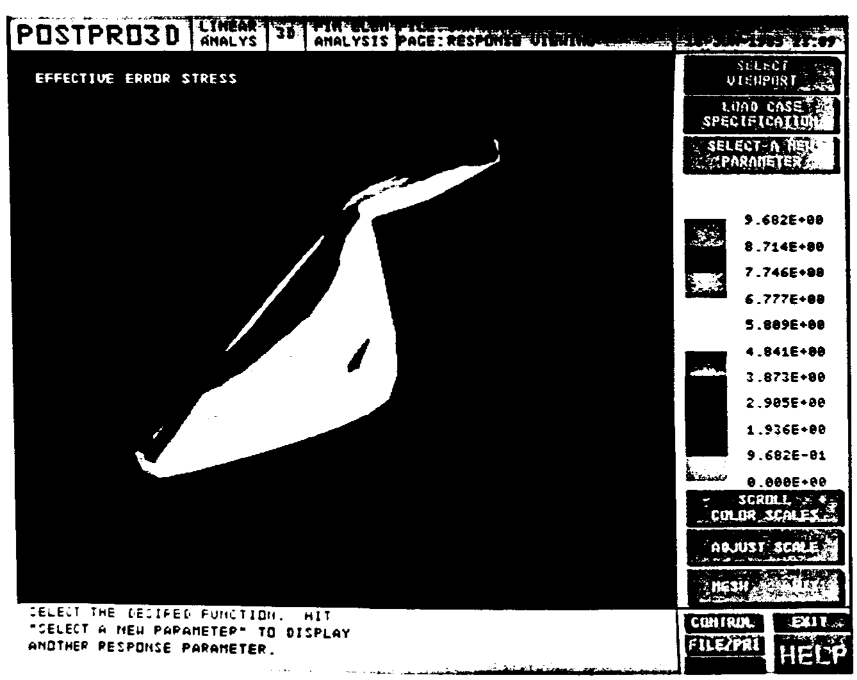

Figure 3.10 Contour distribution of effective error stress for previous dam model

elements used in the model are linear-displacement eight-noded bricks which, in relatively coarse meshes, are known to model bending stresses poorly. The top of the dam is seen to be subjected to transverse bending, and in Figure 3.10, it is evident that the "effective error stress" is highest in this region. Unfortunately, difficulties encountered in interpreting these error quantities have made it clear that a different estimator of discretization error would be more useful; for example, the nodal errors could be used to compute an energy quantity. This more common approach to error evaluation opens up the possibility of using the error quantity as feedback to the preprocessor, where perhaps a mesh optimization technique could be utilized [6].

3.2 User Interface Design

In the design of the response viewing page, special attention has been paid to the rapport between the user and the computer. Consideration of human factors [7-8] is necessary if effective communication between man and machine is to be maintained. Certainly, the reason for developing an interactive postprocessor is to reduce the burden placed on the analyst by eliminating the difficult and tedious tasks of data evaluation. It follows that the use of the postprocessor itself should not be difficult or tedious. A well designed user interface allows the engineer to follow any of a number of paths when approaching a problem and provides completely flexible operation of the tools used along any of these paths.

The functions provided on the response viewing menu page are organized in a top-down fashion, so that the most common sequence of operations can be achieved by their sequential use. A common sequence of operations might be LOAD CASE SPECIFICATION, SELECT A NEW PARAMETER, DISPLAY CONTOURS, INCREMENT RESPONSE SET, SCROLL SCALES, ADJUST SCALE, MESH, and finally HIT. Thus the menu buttons are organized in this order. However, the user may choose to follow a different sequence and is not limited to the one implied by the top-down order. The first button on the response viewing page is SELECT VIEWPORT, although viewport selection (multiple views) will not be implemented until a higher resolution display device is obtained (see Appendix A).

For an analysis involving multiple response sets the user may start by hitting the LOAD CASE SPECIFICATION button. The functions on the load case page allow selection of a specific response set, superposition of sets, and the choice of global or local color scale limits. To facilitate ease of use, the response viewing and load case pages have been designed to allow direct access of either page from the other. Once the user has selected the desired response set and it is loaded into the database, he returns to the response viewing page, points to a response type listed in the main view, and then hits DISPLAY CONTOURS to perform the contouring operation. At this point, the DISPLAY CONTOURS button changes to INCREMENT RESPONSE SET. If this is hit, the response which is currently contoured will be contoured again, but with the next response set. This allows the user to page through the response sets by hitting this button repeatedly and frees him from having to enter the load case page each time to select the next response set.

Once a response is contoured for a particular response set, manipulation of the color scale is possible by using either the SCROLL SCALES button or the ADJUST SCALE button, both of which are located beneath the color scale. If the user is not happy with the present color scale, the color or numerical variation can be altered. The color scale can be adjusted repeatedly after a response has been specified, and can be scrolled indefinitely once a contoured image has been displayed until the user is satisfied with the color scale and numerical distribution.

Hit-testing for response values is possible on any contoured image displayed within the response viewing page. When hit-testing, it is often desirable to display the mesh, so a MESH button is located next to the HIT button. All other response viewing functions may be used while the hit-testing mode is activated except for contrasting by pointing at the contoured image. If the user decides to contrast, he may do so either by pointing to the colors on the color scale or by hitting the RET (return) button. The latter method deactivates the hit-testing function so that contrasting can again be performed by pointing to the contoured image.

The response viewing functions occur within less than a second or two with the exception of contouring, which takes from five to thirty seconds depending on the complexity of the image. (Future improvements in the hardware-assisted visible surface rendering of polygons should make even this process more rapid.) Thus the communication between the user and the postprocessor remains interactive, allowing continuity and spontaneity in the postprocessing session.

The sequence of operations outlined previously is only one of any number of sequences that can be used to interpret response data. The chosen sequence is completely flexible and allows the use of virtually any function at any time. Obviously, some illogical sequences cannot be allowed; for example, the user cannot hit-test for response values until a response has been chosen and contoured. Recovery from ill-chosen functions is easy as all buttons provided on the page are active. For instance, if the user is evaluating a fluid flow problem and selects a particular component of velocity for contouring, he may suddenly realize that he would really like to look at the pressure distribution instead. He may then push SELECT A NEW PARAMETER instead of pushing DISPLAY CONTOURS, so that pressure can be selected. Or, the user may be in the middle of an operation such as adjusting the color scale, and then suddenly decide to go back to the load case page or to select a new response parameter. It is also possible to change from hit-testing mode to new parameter selection or to load case selection, and so on. In short, the response viewing page is an exemplary user interface because it is both interactive and well organized.

3.3 Response Viewing Scenario

Perhaps the most significant user-oriented aspect of the response viewing capabilities (and overall postprocessor capabilities) in the prototype system is that use of any feature does not eliminate the use of any other. The user is never forced down a path where he suddenly discovers that necessary tools are unavailable and the responses are inadequately displayed. This flexibility has already been described with respect to the creation of subobjects (Section 2.2), but an extreme example of the flexibility of the prototype system is offered here in the form of a short scenario.

Suppose that a large, three-dimensional model of a machine part is created and consists of several material types. A material nonlinear finite element analysis is performed, and the problem is translated and input into the postprocessor.

The user first rotates the model and then sections it to reveal an interior surface. Next, he selects the last response set because he is interested in the responses for the final load step. He contours a discontinuous response component and discovers that the contour resolution is not adequate in an area of interest because of a high gradient in the vicinity of loading, which occurs away from this area.

At this point, the user decides to "zoom" in on the desired region by creating a subobject. He returns to the sectioning page, enters the element extraction page, and creates a subobject from the sectioned model. Because the element extraction process is not specific to response viewing it is performed on a separate menu page.

When the subobject files are read in, the user is effectively starting with an entirely new model which is a subset of the original problem. He exposes the interior surface of interest again via sectioning. The response viewing page is entered from the load case page after selecting the final load step, and the previous response is contoured. Now, because the high gradient region is not included in the subobject, the contour resolution is improved, and the contoured image is larger. The user then hit-tests on the sectioned surface to obtain a more quantitative appreciation of the results.

The user next decides that it would be useful to evaluate the response of the subobject by dividing it into two subobjects for separate consideration. He does this and sections one of the new subobjects again. The sectioning processes (which is the most computationally intensive function in the postprocessor) occurs quickly due the drastically reduced subobject database. He enters the response viewing page once again after selecting the final load step. The model is contoured and hit-testing is performed on the sectioned face, so that the exact magnitude of the response discontinuity can be evaluated along a material interface.

As he is hit-testing, the user realizes that steps previous to the last one must be examined to find a critical value of load. The "-" side of the INCREMENT RESPONSE SET button is used to decrement to the previous load step. This is done several more times so that the discontinuity can be observed for a few of the load steps previous to the last one. At an intermediate load step, hit-testing is performed on the sectioned face again, and alternate contour colors are contrasted for easy numerical evaluation. Throughout this process of examining various load sets for the sectioned subobject, the original sectioning remains valid because all information required for the interpolation of any response set is retained during the sectioning process. When response sets are swapped in and re-contoured, the necessary interpolation on the sectioned surface is performed automatically prior to re-contouring.

After experimenting with a few alternative color scales, the user moves from the response viewing page to the printout page where he obtains hardcopy information for the subobject nodal responses. He then returns to the response page and saves the contoured image for later retrieval from a file, by pushing the SNAP button in the permanent menu area.

The above scenario could continue, but enough has been presented to show that the system is flexible and that the possibilities for approaching the response viewing task are virtually endless. All functions that are central to response viewing are provided on the response viewing page. Other functions, such as sectioning and element extraction, which are sometimes used to enhance response viewing, are provided on separate menu pages. An analysis which is large and complex can be broken down into small pieces for individual interpretation, for which the basic response viewing tools prove to be more effective. These functions, when taken together, combine to create a powerful system which allows interactive interpretation of complex numerical analysis problems.

CHAPTER 4

ALGORITHMS OF THE POSTPROCESSING SYSTEM

In this chapter a description of the algorithms developed during the second year of research is given. Most of the routines described have been helpful in expanding response viewing capabilities of the prototype system. The first five sections in this chapter describe algorithms which are used mostly to aid in the viewing of different responses, while the last section describes the handling of multiple response sets, i.e., response information from a stepwise nonlinear or dynamic analysis or from a linear analysis with multiple load cases.

4.1 Hit-Testing

A commonly used and powerful feature of the prototype postprocessor is the ability to point to an area of interest on the shaded or contoured image to extract information from the analysis database directly. This technique is referred to as "hit-testing." In the attribute viewing page, hit-testing is used to display element and node numbers, as well as element material properties. In the coordinate system page, the user can point to the model and the node number and coordinates for the closest node are displayed. In the response viewing page, the user can "hit" the contoured image of the model, and the nodal value of the contoured response closest to the cursor is displayed in the lower portion of the screen.

Typically, upon entrance to a menu page where the hit testing feature is provided, computations are performed which increase the speed of the function. The exception to this is on the response viewing page, where these computations are performed when the user selects HIT. In all cases, the nodal coordinates of the model are transformed into screen coordinate space. The remaining computations are performed in this space for two reasons. First, when the user points to the main viewport, the firmware returns the screen coordinates of the hit pixel. Second, the screen coordinate space is an integer space, so computations are performed quickly.

After the transformation, the list of boundary polygons used in the description of the model is analyzed. The normal vector for each polygon is retrieved from the database. By computing the dot product of this vector and a vector which points in the positive z screen direction, polygons which face away from the user can be detected and rejected. This test is commonly referred to as "culling," and is performed since it is impossible to select, or even to see, any polygons which face away from the user. However, the culling operation is not performed for models consisting of shell elements or for boundary element subobjects. For these models, polygons which face away from the user are displayed and can be pointed to, so they must be considered. Next, the nodal screen coordinates of a "surrounding cube" are computed and stored for each remaining polygon. That is, the minimum and maximum x, y, and z screen coordinates are computed. For a large problem with, say, 500 three-dimensional elements and 4000 nodes these first few operations may take up to about 5 seconds, and the user therefore experiences some delay before admittance to the desired page, or until hit-testing can be performed for response retrieval. However, this small amount of preprocessing greatly increases the speed of the remainder of the algorithm. Once the surrounding cube information has been obtained the user can point to the model.

When the user then hits the object, the actual search for the selected polygon begins. The firmware is first used to retrieve the x, y, and z screen coordinates of the hit pixel. If the selected pixel is part of the background or menu the remainder of this algorithm is not executed, and the user must select again. If the selected pixel is not part of the background or menu, the pixel coordinates are tested against the surrounding cube data to see if the hit coordinates fall inside any of the cubes. If any coordinate is found to fall outside one of the cube limits, the cube is immediately rejected without checking the remaining limits.

Next, an "interior" test is performed on each polygon passing the cube test. The polygon is first divided into triangles to ensure that computations are performed on planar, convex polygons. The x and y screen coordinates of the hit point are then substituted into the two-dimensional line equations for each edge of a triangle. If the signs of all substitutions are the same, the point is "inside" every edge and is thus interior to the polygon. If any triangle from a given polygon passes the interior test then the remaining triangles associated with that polygon are not tested. A list of all polygons which pass both the interior and cube tests is created. If only a single polygon passes these tests then it is the hit polygon and the final test is skipped.

If more that one polygon passes the interior test then one must lie in front of the others. To find the desired polygon (the polygon closest to the screen) the hit coordinates are substituted into the plane equation for each of the remaining polygons. The plane equation which is most closely satisfied is the visible one, i.e., is closest to the screen, and therefore belongs to the hit polygon. Once the hit polygon is known, it is a trivial matter to extract the desired node or element information from the database.

Hit-testing can be performed on sectioned surfaces of models composed of solid elements. When a model is sectioned, new polygons are created and stored with the original polygons so that separate lists of boundary polygons exist for each subassembly. All new polygons are triangles, and are stored in the database in the same format as the original polygons. New node numbers are created for all polygons on the sectioned surfaces, and these new nodes are assigned numbers which are higher than the original number of nodes in the model. When hit-testing is performed on surfaces, displayed node numbers are shown as these new values.

4.2 Response Discontinuities

The method used for handling discontinuous responses was introduced in Section 2.2 and is discussed in greater detail here. The technique which allows the contouring of discontinuities is that of node duplication. This is a process by which artificial nodes are created at element interfaces. In duplicating nodes between two connected elements, new node numbers are assigned to each node at the element interface so that node numbers in each element are no longer common to the other. Thus, separate responses can be stored for nodes on each side of a discontinuity.

Coordinates which are identical to the original nodal coordinates are assigned to the new nodes, and the three-dimensional element connectivities and boundary polygon vertices are updated consistently with the new numbers. The program then "sees" only the renumbered state of the model and treats the information in the same manner as for a problem which has not been renumbered. However, a node "name" map is maintained in the database which returns a name identical to the original node number for any node in a problem where node duplication has occurred. When operations are performed which display the node numbers on the screen, the node name is used so that only the original numbers are shown (unless the node is an entirely new one created with the sectioning process as described in the previous section).

When contouring is performed, all polygons and the responses associated with their nodes (vertices) are sent to the contouring algorithm. Because this algorithm treats each polygon separately, when the adjacent polygons are contoured, the discontinuous responses that are sent with the polygons appear at their intersection as discontinuous contours.

Node duplication has been used for the display of discontinuous responses in two cases. The first case is that of multiple materials, where material interfaces necessitate this type of handling. The second is with boundary element models, where geometric or surface-traction discontinuities are responsible for discontinuities in response. Treatment of these cases is similar in that node duplication is used for both; however, differences do exist. Material interfaces between solid elements allow discontinuities to occur in the interior of the model as well as on the exterior, whereas with boundary element models, only the exterior of the model is of concern. The details of these two cases are discussed next.

4.2.1 Inhomogeneous Materials

For problems composed of three-dimensional solid elements and multiple materials, a method has been developed to perform the renumbering operation at the material interfaces. This process occurs in the translation stage between the analysis and the postprocessor.

Note that although boundary element models can be created which contain multiple materials, response information in the interior is not a concern as mentioned in Section 2.3. Multiple- and single-material boundary element models are therefore treated identically, and are discussed in the following section. This section, then, applies to other analysis types.

To understand the node duplication procedure, a short description of the boundary extraction algorithm, which is part of the translation stage, is necessary. When creating the polygonal representation of the model, the boundary extractor compares element faces so that interior ones can be found and a list of exterior ones can be created. If two element faces are found which match between two elements, they are marked as interior. Once all elements have been tested, the remaining, unmarked faces are labeled as exterior, and are then sent to the postprocessor to be used for the polygonal description of the model.

Modifications have been made to the boundary extractor so that it is now used to detect material interfaces between elements. By sending element material information, nodal coordinates, and element connectivities to the boundary extractor, it is possible to perform an additional test during this part of the translation. When a match is found during the actual face to face comparisons, the material types of the adjacent elements are compared to see if they are the same. If the elements are not of the same material, the connecting nodes are sent to a routine which duplicates them. Because the database can only accept a sequential set of node numbers, the new numbers used in the renumbering process must be sequential and without gaps. This is ensured by preventing any node associated with a particular element from being assigned more than one new number. Original numbers for duplicated nodes are saved and used to create the node name map.

In the renumbering procedure it is necessary to force each original node number to remain in its original location. When the next stage of translation, outline extraction, is performed, the renumbered boundary polygon nodes are mapped back to their original state, and polygon intersections are computed. Nodes which are found to correspond to these intersections (outline nodes) are the original nodes, and are thus guaranteed to be in the same position on the renumbered model. This allows the outline to remain "glued" to the outside of the model.

Once all interior faces have been located, the node duplication is complete. The renumbered polygons and connectivities are then output along with the node name map, coordinates for the new nodes, and normal vectors corresponding to the boundary polygons.

A similar algorithm has been coded so that surface elements (membrane, plate, or shell elements) which lie along material interfaces may be renumbered. The procedure is almost identical to that used for solid elements, except that because all the flat elements lie on the exterior of the model, no boundary extraction needs to be performed. For this reason, the outline extractor is used to detect material interfaces between elements (polygons), and the renumbering procedure occurs in a way similar to that for solid elements. The material interfaces occur at the element edges in this case, and the nodes along these edges are renumbered accordingly. Once all the outlines have been found, the list of boundary polygons is revised and output along with the map array and the list of nodal coordinates corresponding to the renumbered nodes. This renumbering scheme has not yet been used on surface elements.

4.2.2 Boundary Elements

Response discontinuities for boundary element models are handled using node duplication. Traction components which are non-zero on surfaces of boundary element models are often discontinuous at the edges of these surfaces or at interelement boundaries within the surface. In addition, when boundary element intersections occur on continuous surfaces, the errors in tractions between the elements are often negligible. For these reasons it is undesirable and unnecessary to average nodal responses anywhere on a boundary element model, and a renumbering approach is therefore adopted. Because no averaging occurs, homogeneous and multiple material boundary element models do not require different treatment. Thus, responses which are discontinuous for any reasons are preserved and contoured as described earlier in this section.

The renumbering procedure used for boundary elements occurs in the translation stage and is straightforward. As with the solid elements, this technique allows the postprocessor to treat the renumbered model exactly as it would any other model.

Two arrays are maintained to perform the renumbering. The first is a flag array which keeps a record of all nodes encountered previously. The second array is the node name map. After the boundary element translator reads in the geometric description and the behavioral parameters of the analysis, the nodes are renumbered on an element-by-element basis. When this procedure is complete, elements no longer share nodes with each other. This is consistent with the boundary element analysis, which outputs separate nodal responses for each element. (In particular, one must admit the possibility of "jumps" in traction at interelement boundaries.) The first time a node is encountered it is left unchanged, but a flag is set for the node indicating that it has been observed once. The next time that particular node is discovered, its flag indicates that it lies on another element as well, and it is then given a new number. The node name map is updated with the original number and a running count of the newly assigned nodes is kept as in the previous section. Once all the elements have been renumbered, the boundary polygons can be output with their respective normal vectors.

When activated, the outline extractor uses the node name map to convert the boundary polygons back to their original state, and then computes the polygon intersections. Because the original nodes are left untouched the first time they are encountered, outline nodes remain in their original positions for both the original and renumbered polygons. Thus, the outline is guaranteed to remain attached to the renumbered model.

4.3 Element Extraction

Element extraction was introduced in Section 2.2 and found to be one of the most powerful new techniques implemented. By interactively isolating groups of elements either with sets of enclosing surfaces or by material type, the user creates subobjects which can be studied independently from the parent model. In this section the algorithm which performs the subobject creation is described for three-dimensional solid elements. A similar procedure occurs for boundary element models and models with flat elements, the major difference being that a boundary extraction for such subobjects is unnecessary.

If the user wishes to isolate the desired elements with enclosing surfaces, he first selects the coordinate system in which he would like to perform the extraction. Cartesian (x, y, z) and cylindrical (r, theta, z) systems are available, and implementation of a spherical system will not be difficult. The centroids of all elements in the model are then computed in terms of the selected coordinate system upon entrance to the enclosure page. This computation is performed here to reduce the amount of computation necessary when the subobject is actually created. The user then activates any one of six surfaces corresponding to a minimum or maximum value of x, y, or z (or r, theta, or z). Once a surface is activated, a grid representing it is displayed in an overlay plane in the main view. The grid can then be positioned either by using a potentiometer or by pointing to the desired position on the three-dimensional image of the object. A short list is created which contains the positions of all enclosing surfaces for later use in the determination of elements which are to be included in the subobject.

The user may also elect to extract only elements of designated material types and can do so by entering the material extraction page. Once in this page, a table of all material types included in the model is shown in the main view. The desired material types are then selected by the user and a record of these materials is obtained for the extraction process.

When the user is satisfied with the positions of all enclosing surfaces or has chosen the desired materials, the EXTRACT command is selected to activate the actual element extraction process. An extraction routine is activated to create a list of the elements which are to be included in the actual subobject. If the elements are chosen by position, the list of surface locations is compared with the coordinates of the element centroids. If an element centroid is within the surrounding surfaces and the user has specified that interior elements be accepted as parts of the subobject, the element is accepted. Similarly, if the element centroid is outside of the volume enclosed by the surfaces and the user has specified that exterior elements be accepted, the element becomes part of the subobject. If the elements are chosen by material type, the list of desired materials is compared to the list of element material properties contained in the database. Elements are then accepted or rejected accordingly.

Figure 4.1 shows the positioning of the "minimum r" surface in a cylindrical extraction of a finite element model of a shaft. All elements with centroids that fall outside of this radius will be accepted. Thus the subobject of the shaft will have a core removed from its interior as in Figure 4.2. Notice that if layers of elements are in skewed positions with respect to an enclosing surface, the exterior region of accepted elements in the final subobject may be jagged. A subobject has been created from a portion of the concrete gravity used as an example in Reference [3]. Figure 4.3 is a picture of this subobject with the mesh displayed so that element locations are obvious. An enclosing plane is shown in a skewed position with respect to some of the elements and the resulting subobject is shown in Figure 4.4. The accepted elements in the vicinity of the enclosing plane do not form a clean boundary, although the sectioning feature can be used to smooth this surface.

Several points must be mentioned here. Because subobjects can be made from subobjects, an immediate subobject parent is called the parent, whereas the original complete model is referred to as the "top-level" parent. The key to the creation of subobjects is in the reorganization of parent geometrical and response data into a form which is self contained. The database is organized such that responses are stored for nodes numbered consecutively. The first node number must therefore be unity and the last, the same as the number of nodes in the model. If the nodes are numbered in any other way, it is impossible to retrieve their respective response values from the database. Certainly, the included

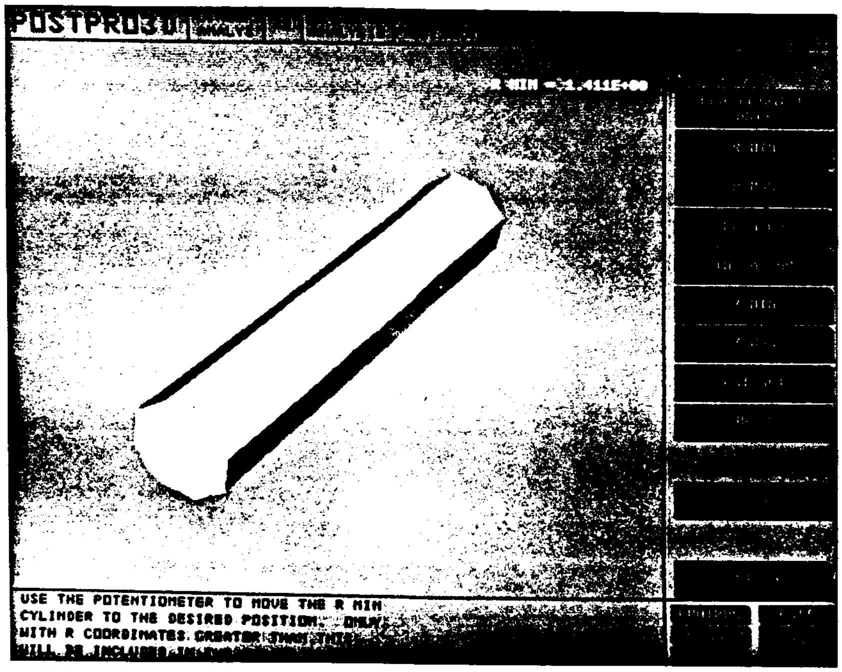

Figure 4.1 Minimum r plane used to remove core from a torsional shaft

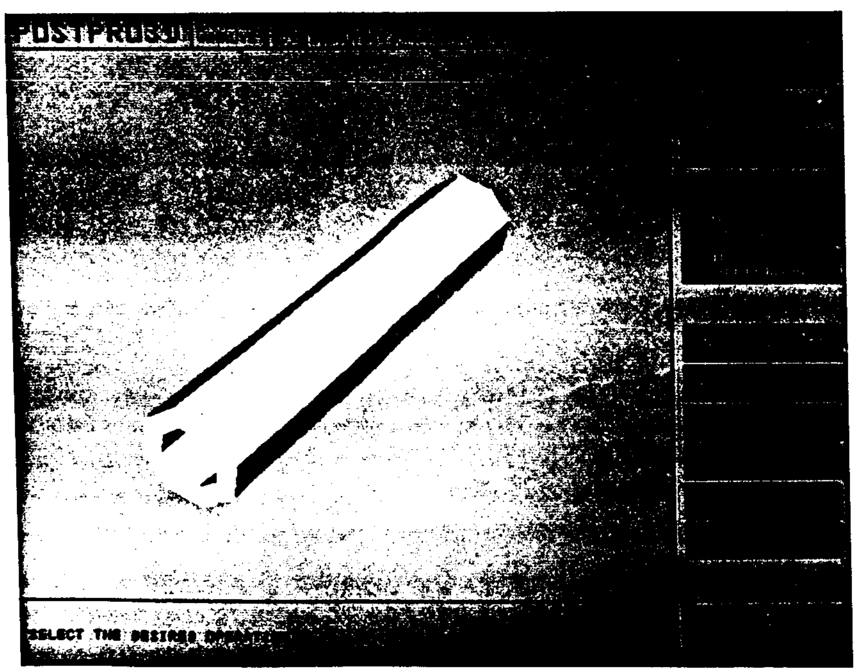

Figure 4.2 Subobject of torsional shaft showing removal of core

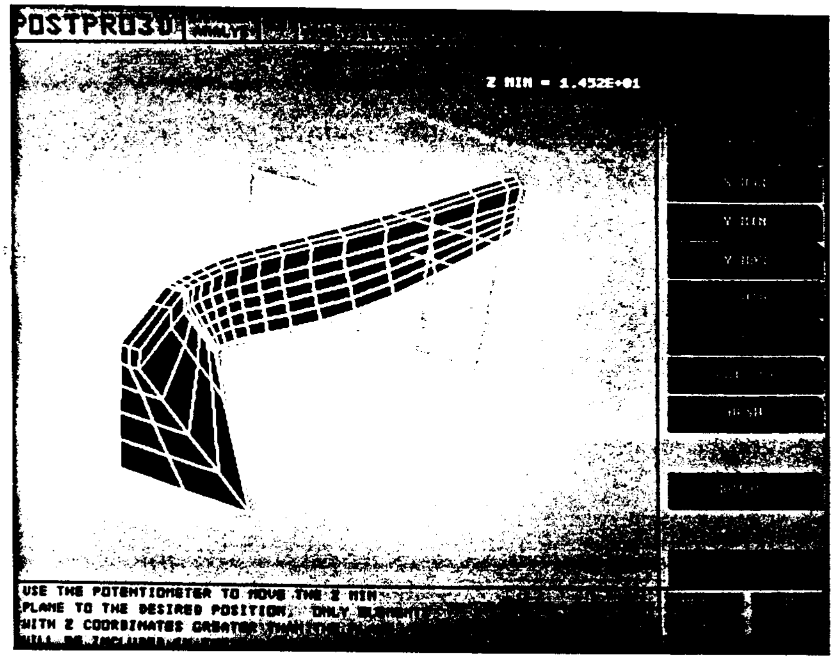

Figure 4.3 Subobject of dam shown with enclosing plane at a skewed position with respect to the elements

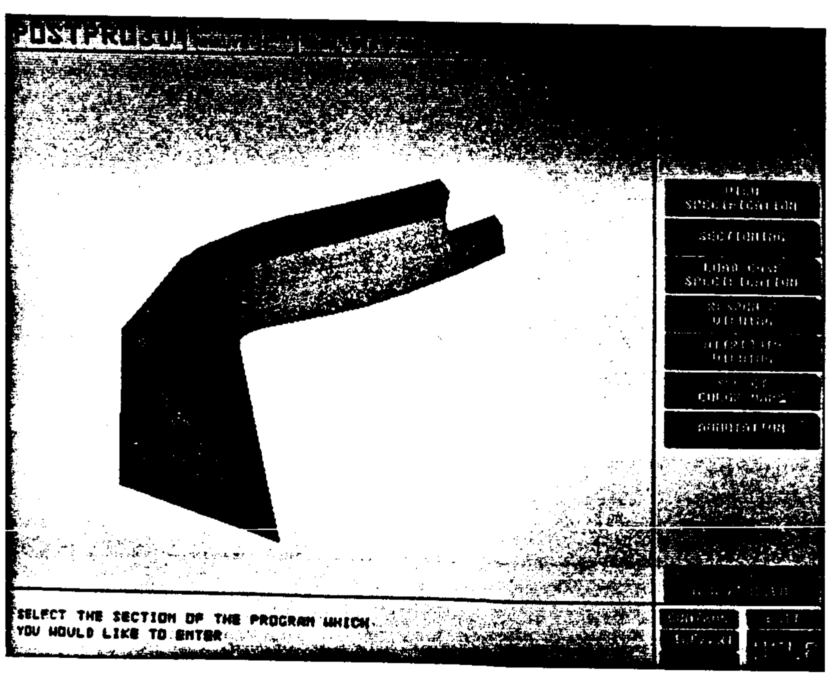

Figure 4.4 Subobject of previous subobject shows irregular region of included elements

elements do not necessarily contain nodes 1, 2, 3, and so on, up to the number of nodes in the subobject. The operation which renumbers the elements so that the nodes will fit this format is referred to as "compressing" the connectivity list. Elements must also be renumbered so that the subobject will contain elements numbered 1, 2, 3, etc. Finally, to retain a direct link with the top-level parent, it is necessary to maintain a single node name map which ties all the renumbered subobject nodes back to their original node names (numbers).

After the list of included elements has been obtained for the subobject, an algorithm is initiated which performs the bulk of the subobject creation. Four lists are constructed which are used for this process. The first list is called the "compressor map" and is used to compress the connectivities of the subobject. The second list is the "n-map" which is the node name map described previously. A map is also kept which holds the original element numbers of the subobject elements and is called the "element map." The final list is called the "parameter map" or the "p-map." The p-map contains parent node numbers for all of the subobject nodes and is used to retrieve subobject response information from the parent database.

Before a compressed connectivity list for the subobject can be created, a connectivity list is assembled for the parent, where all parent nodes which have been renumbered are mapped back to the top-level parent state. (Recall that all models which have been renumbered have such a map in their database.) This facilitates the creation of both the subobject node and compressor maps.

When a parent connectivity list with "virgin" numbers has been prepared, the compressor map is constructed. Its size is equal to the number of nodes in the original analysis. The algorithm inspects the virgin connectivities of the included elements. At the first occurance of each node number in this list, its location in the compressor map is flagged. The list of nodes is then scanned sequentially: the first flagged node encountered is given the number one, the second, the number two, and so on, thus completing the compressor map.

Next, the compressed connectivity and coordinate lists are computed. When a node number is obtained from an included element in the virgin connectivity list, its value is used to index into the compressor map. The result is the new subobject node number. A compressed connectivity list is created in this manner and coordinates corresponding to the compressed node numbers are output. At this point, the boundary and outline extractors are re-activated (for flat, non-solid elements, only the outline extractor is re-activated). The compressed geometric information is used for the creation of the subobject polygonal and outline representation, and node duplication is repeated if the subobject consists of more than one material.

When the subobject boundary and outline extraction is complete, the n-map, element map, and p-map are produced. Creation of the element map is not difficult. The first included element is assigned the number one and the first value in the element map is the true number of that element, and so on. If the parent is itself a subobject the true number is found using the element map currently in the parent database.

For single material subobjects the n-map is computed next. The n-map returns the top-level parent node number for any renumbered node. The first value in the n-map is the sequential position of the first node in the compressor map. For example, if the first flagged node in the compressor map is node five (node 5 being the lowest node number in the top level parent which is included in the subobject), the first value in the n-map will be five, as the flagged node is in the fifth position. The sequential position of the second flagged node in the compressor map is therefore the second value in the n-map. Once the entire compressor map has been scanned in this fashion, the n-map is complete.

Creation of the n-map for a multiple material subobject is difficult. In this case, it cannot be created until after the boundary extractor has computed the subobject's polygonal data, because nodes are further duplicated at the material interfaces. Duplicated, compressed nodes are thus first mapped back to their originally compressed state (using the node name map returned by the boundary extractor). The resulting node number is located in the compressor map, and its sequential position is the desired top-level parent node number.

To create the p-map, it is recognized that for any node in the subobject's final connectivity list, there is a one-to-one correspondence to the respective node on the same element in the parent list. This is the case for both single and multiple material subobject connectivities. The parent node numbers can thus be used to locate subobject response information in the database. Suppose that the first node of an element in the subobject is subobject node number 1. The corresponding element in the parent database is located and its first node is found to be number 8. The first value in the p-map is therefore eight. When extracting responses from the parent database which correspond to subobject node 1, the p-map points to node 8 of the parent, and the correct information is obtained.

4.4 Stress Intensity Factors

To enhance the study of fracture mechanisms, a technique has been developed by which stress intensity factors are contoured along a three-dimensional crack front (Section 2.3 and Figure 3.1). Because a crack typically extends from the exterior of a solid into the interior, it is desirable to expose the crack in such a way that its geometry is easily visualized. If this can be achieved, stress intensity factors may be used to help predict subsequent crack propagation.

During the translation stage, a list of the nodes which exist along the crack front is created. These nodes are organized into one-dimensional connectivities so that lines can be drawn between them. Color "contouring" is then performed on the lines in a fashion similar to that for polygons. Because only the outline of the model is displayed together with the contoured lines of stress intensity, the crack front is exposed for inspection. The maximum and minimum stress intensity factors are computed and the color scale is divided so that each color corresponds to a different range of stress intensity. End coordinates and the stress intensity factors associated with the endpoints are then sent one at a time into the following line-contouring routine. If both values of stress intensity fall within a single response (color) range, the line segment is drawn immediately with the respective color. However, if the endpoint stress intensities occur in different response ranges, intersection points between ranges are interpolated. The original line segment is then shown as two or more shorter segments, each displayed in the color associated with its particular response range.

For line connectivities which originate from edges of higher than linear order elements, the contouring procedure is more involved because both the geometry and the variation of stress intensity factors between nodes is of higher order. Essentially, the shape functions are used to divide such curved crack fronts into several segments. The shape functions allow the computation of the endpoint coordinates of each segment as well as the stress intensity factors for the respective endpoints. Once this information is obtained, each segment is contoured in the usual fashion. Curved lines are thus approximated with several straight segments.

4.5 Nonlinear Color Scales

It has been shown that manipulation of the ranges of response for given colors in the color scale can be used to improve a poorly contoured image. By specifying cutoff values of response, one contours all responses above the top limit or below the bottom limit with the top or bottom color. Thus, the remaining eight or nine contours can be forced into regions of interest, and additional information can be obtained.

Implementation of this technique is achieved simply by allowing the user to input the desired response limits at the terminal. The user pushes LIMITS to activate the limit selection process. Response limits are retrieved from the database for the contoured image or the selected response component and are displayed along with a representation of the color scale in an overlay plane (See Figure 3.4). The user then points to either TOP LIMIT or BOTTOM LIMIT and keys in the desired numerical cutoff limit at the terminal. The input limit is then compared to the total response range. If a limit is found to fall outside of this range, the value is rejected and the user is given an appropriate message. When an admissable limit has been input, it is shown in its respective position on the color scale representation.

Once the limits have been chosen, the DISPLAY CONTOURS button can be hit to contour the image with the newly adjusted color scale. Before the actual contouring process occurs, flags are checked to determine whether the top, bottom, or both limits have been input. A linear distribution of contours is then computed for the remaining contour colors. If top and bottom cutoff limits have been input, the remaining colors between these limits are given a linear distribution, whereas if only a top or bottom limit is specified, a linear distribution is assigned between this cutoff value and the original opposite-extreme response value.

As an alternative to the specification of cutoff values, it is also possible to adjust the relationship between the contour colors and response ranges so that it varies in a nonlinear fashion. By manipulating a potentiometer, the user can "drag" the contours so that more of the colors are in either the high or the low end of the response range. The variation between contour colors and response ranges is parabolic. Figures 4.5 and 4.6 show how the parabolic variation is adjusted. By moving the cursor to the upper end of the potentiometer, one moves the center point of the linear segment in Figure 4.5 diagonally up to the left, to the position in Figure 4.6. The two endpoints and mid-point describe a parabola which is used to compute the response variation for the remainder of the color scale.

4.6 Multiple Response Sets

Nonlinear analyses and linear analyses with multiple load cases produce voluminous sets of response data which can be unwieldy as discussed in Section 2.1. All response sets are not included in the active database simultaneously; instead they are kept in a file. When a single set is specified by the user, it is swapped into the database and replaces the current set. This section elaborates on the swapping procedure and provides a more detailed explanation about the handling of multiple response sets.

In the redesign of the database to allow more than one response set, three objectives were kept in mind. It is desired to keep only a single response set in the database at any one time to avoid the cumbersome manipulation of any data not specifically requested by the user. It is also necessary to maximize interactivity, so a fast method of exchanging data sets is needed. Finally, if a problem with multiple response sets is sectioned, an increase in the time necessary for computations should

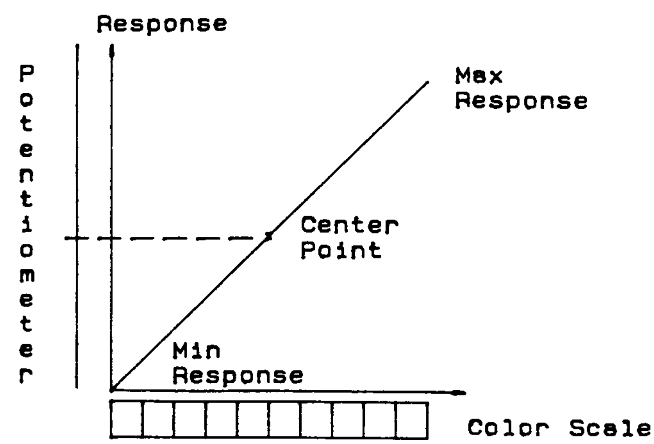

Figure 4.5 Default (linear) relation between contour colors and response

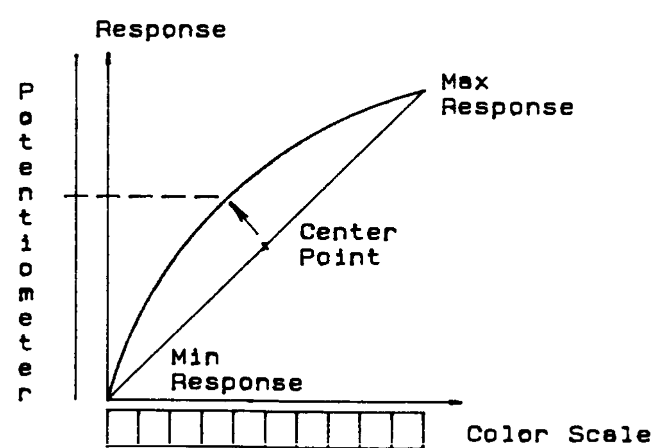

Figure 4.6 Parabolic relationship between contour colors and response created by moving center point

not increase, and the operation should not have to be repeated completely for response sets not in the database. The solution implemented satisfies all objectives. In the translation stage, the response data is organized into a form which can be accepted directly by the postprocessor database. Each load step or load case is converted into a single, independent set, and these sets are stored in a file in a sequential fashion.

Originally, responses were read by the postprocessor, organized, and placed in the database one node at a time. This method was time consuming and was therefore not suitable for the interactive exchange of response sets. The new method takes advantage of the fact that the response sets are already organized. The location of the desired set in the storage file is computed by the postprocessor and the set is read directly into the database with a fast I/O system procedure which permits the postprocessing session to continue almost without delay. An original translated response set cannot be mapped directly into the database memory (i.e., the physical file containing the response information cannot be used as part of the database) because sectioning produces additional response data which would "pollute" the virgin set. Consequently, the I/O procedure is used to effectively create a copy of the information and pollution of the original set is prevented.

For linear analyses with more than one load case, the user can superimpose two or more load cases. In the load case page, the user first selects a load case to be combined with the response set currently in the database. Then, when SUPERIMPOSE LOAD CASE is hit, the user is prompted for load factors which are used as multipliers for the two response sets. When the multiplication is complete, the two response sets are added together in the current database.

When a model containing multiple response sets is sectioned, a record of interpolation information is developed and kept as the new responses for the cut surface are computed. Each of these new nodal responses is obtained by a linear interpolation between two existing nodes. For each interpolation the two node numbers are stored along with the fraction of the distance between the nodes from one end node to the new node. After response sets are swapped for the sectioned model, the stored node numbers and interpolating fractions are used to compute the sectioned response values from the new response set. Note that the geometric information remains unchanged during the swapping process and does not need to be recomputed.

As the translators organize response data into the necessary format, the maximum and minimum values for all possible contoured responses are stored for use as color scale limits. When all of the sets have been scanned, the limits are global to the entire analysis and are stored with each of the organized response sets. These global limits are the default limits if the analysis is stepwise. If the analysis is linear with more than one load case the global limits are not default. Instead, the database is scanned by the postprocessor at the time of display so that limits which are local to the current response set can be obtained. The global limits can, however, be specified by the user. When the global limits are taken directly from the database no computation is necessary and a slight increase in speed is obtained (at the expense of increased computation during translation). Of course if the analysis contains only one response set, these pre-computed limits are always used.

CHAPTER 5

EXAMPLES

In this chapter three examples are introduced to demonstrate the potential for unification of analysis methods, the ease of behavioral interpretation, and the ability to display analytical results for complex three-dimensional geometries with the developed postprocessing techniques. The first example involves three separate analyses of a cranerail girder mentioned in Chapter 2. This example is intended mainly to show how the present implementation can be used to facilitate refinement of an existing analysis and the transition between finite element and boundary element methods. The next example is a material nonlinear model of a welded tension plate connection. Access to the sequential nonlinear response information eases understanding of the connection behavior under progressively increasing loading. The last example is the MARTA Twin Tunnels and Research Chamber. Although the geometry of this model is complex, postprocessing techniques prove to be useful in interpretation of its response.

5.1 Cranerail Girder

This example entails a sequence of analyses used to model fatigue cracking in the compression flange-web weld of a cranerail girder subjected to lateral loading at mid-span [10-11]. Figure 5.1 is a schematic representing the girder configuration. A large, coarse analysis of an entire girder is first studied to investigate the extent of localized behavior due to the lateral load. Next, a more refined analysis of a single girder panel is evaluated so that local behavior can be observed. Finally, a boundary element model created from subobject data is used to predict the direction of crack propagation through the weld. The analysis sequence is shown pictorially in Figure 5.2.

The entire girder is analyzed for linear elastic behavior with symmetry boundary conditions at the centerline. It is simply supported under its bearing stiffeners and one edge of the compression flange is restrained against lateral displacement. The material is steel with modulus of elasticity of 29,000,000 psi and Poisson's ratio of 0.3. The rail geometry is not duplicated in this coarse analysis; instead, it is modeled using a rectangular cross section with the same polar moment of inertia as the actual rail. A unit lateral load is applied to the rail at the girder centerline, between two of the web stiffeners. A finite element analysis is performed on the girder, which is modeled with 256 twenty-noded brick elements consisting of 1820 nodes.

Figure 5.3 shows the exaggerated displaced mesh of the girder. Torsion is introduced to the compression flange and bending of the web is

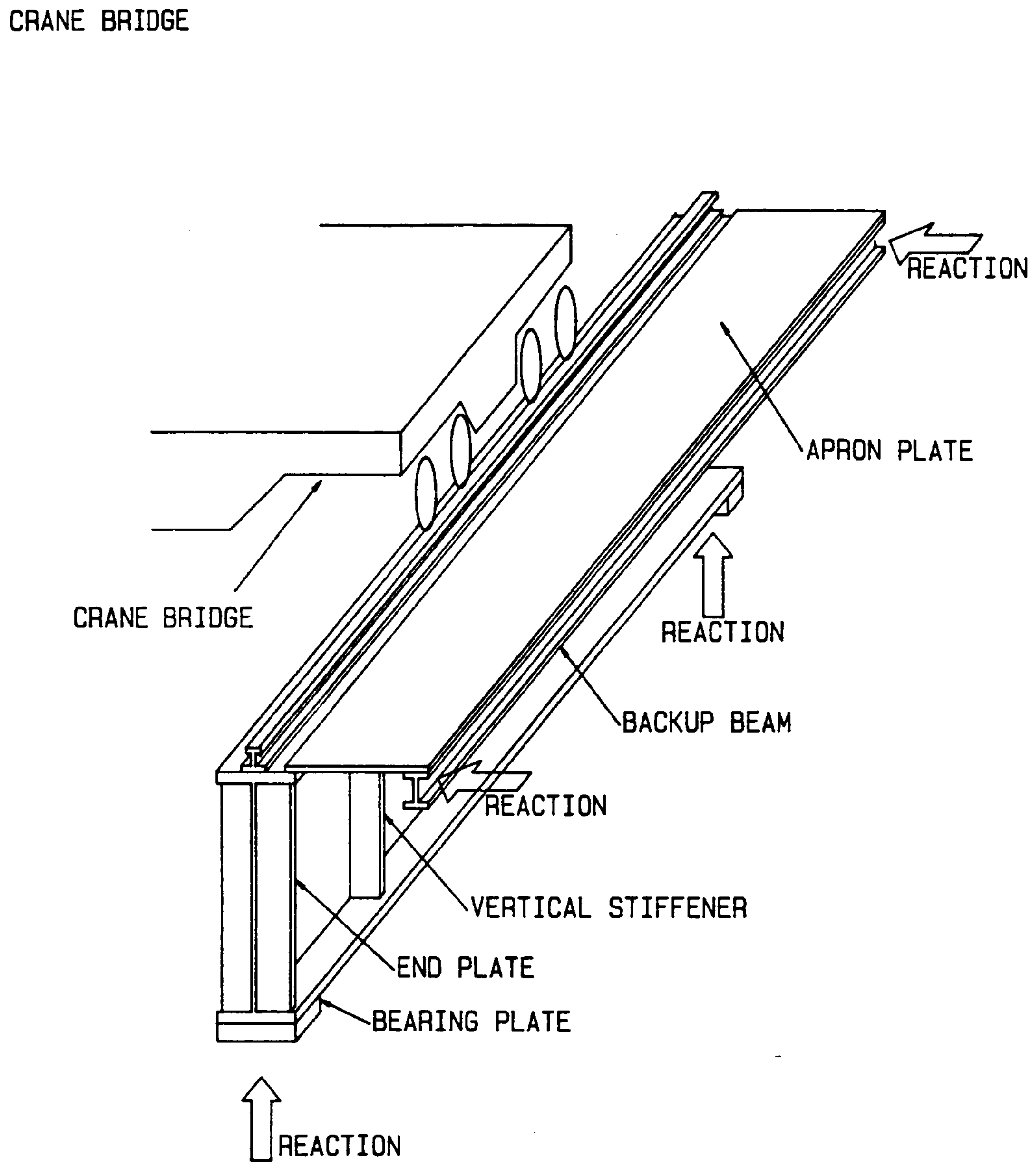

Figure 5.1 Schematic representing girder configuration

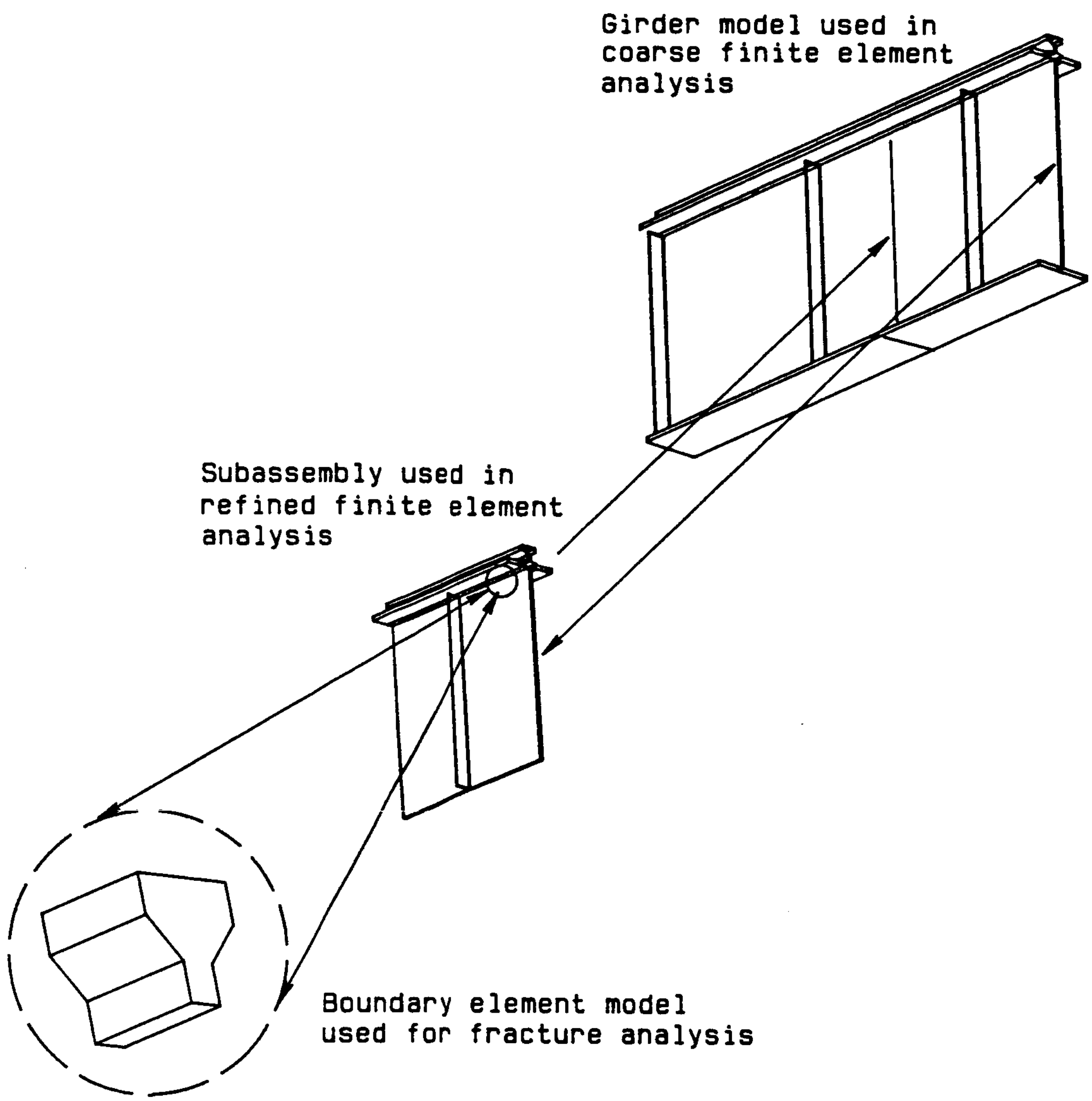

Figure 5.2 Sequence of analyses used in girder study

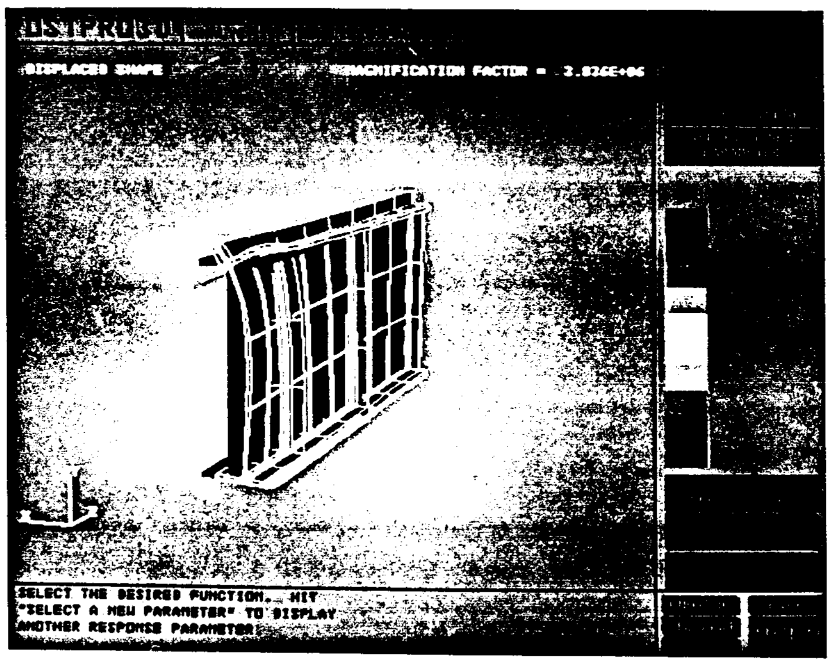

Figure 5.3 Displaced mesh of the first girder analysis

immediately apparent, but these effects can be seen to be effectively reduced by the stiffeners. Note that stresses due to lateral loading in the tension flange are minimal in comparison to those in the compression flange. As these twisting effects travel from the point of load application to the supports, they are reduced by the first set of stiffeners and die out midway between the first and second sets. These observations are crucial because it is necessary to reduce the initial girder geometry if an effective, detailed fracture analysis is to be performed. Consequently, it is necessary to explore further the behavior near the end of the girder to see if it is indeed negligible. In Figure 5.4 the distribution of vertical stress is shown. The highest stresses exist in the first set of stiffeners, which restrain the twisting flange. If the stresses in the web-flange junction are close to zero between the first and second stiffener sets, the end portion of the girder can be ignored in a subsequent, refined analysis.

Because contour resolution in the girder web is poor, a subobject of just the web is created. In Figure 5.5 the distribution of vertical stress in the web is shown again with the contour color associated with near-zero stress contrasted. The girder centerline is on the left and the stresses are highest directly below the point of load application. The response appears to have died out at the first stiffener set, leaving the remainder of the web in a low stress range. Hit testing on a node between stiffener sets one and two and at the flange-web junction has been performed, the selected node is highlighted, and the vertical stress displayed below the main view is nearly zero. Response from this point

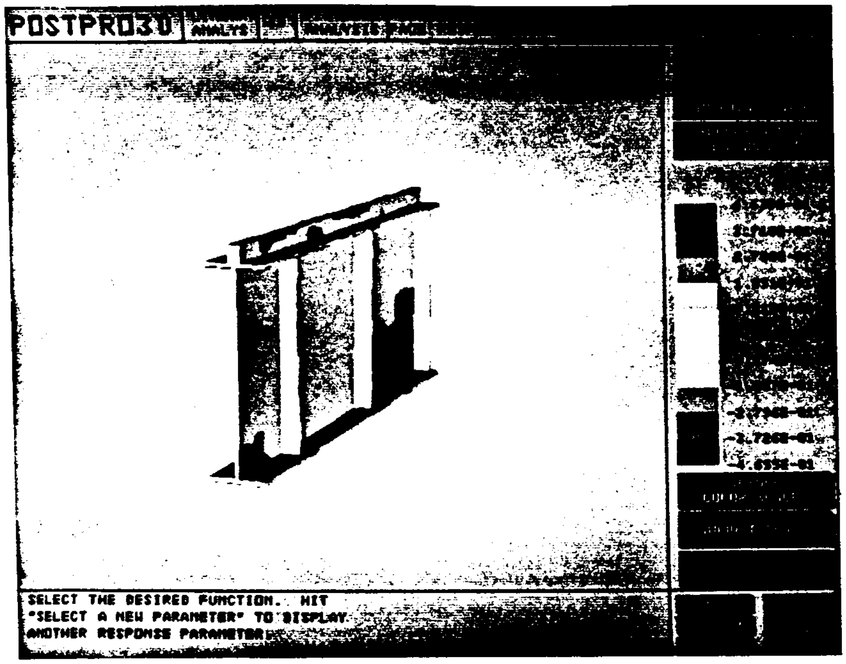

Figure 5.4 Contour distribution of sigma y for the first girder analysis

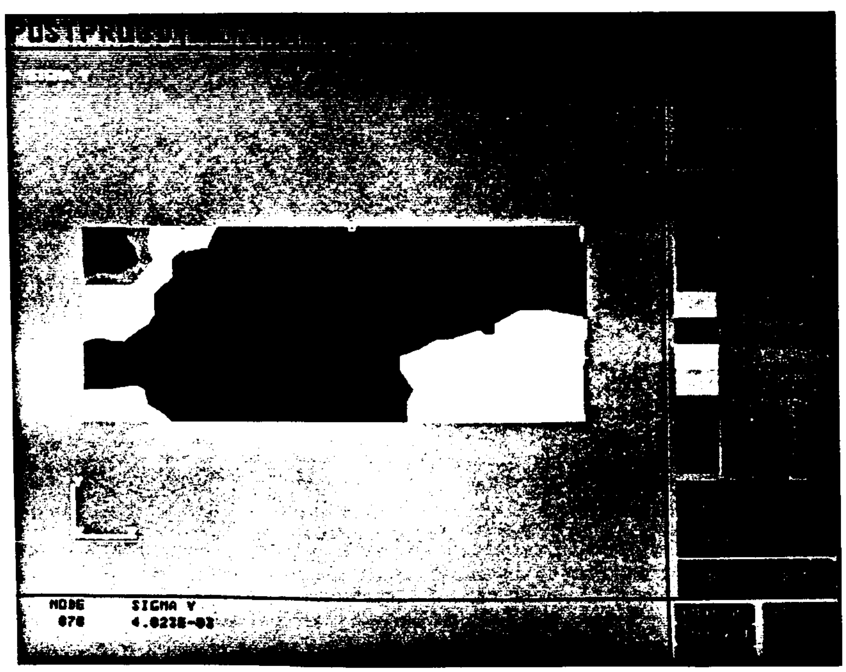

Figure 5.5 Contour distribution of sigma y for the girder web subobject of the first analysis

to the end of the girder is negligible and a reduced geometry in the next analysis therefore neglects this region.

The second analysis considers a subsection of the original girder. A single panel length from the centerline to midway between the first and second stiffener sets is modeled, and the bottom flange is ignored. Symmetry boundary conditions are employed again at the centerline, and fixity is imposed on the bottom and end of the web. Material characteristics are identical to those used in the original analysis, and linear elastic behavior is again assumed. The rail is idealized as in the first analysis and a lateral unit load is applied to the rail at the centerline. A refined finite element analysis is performed on the reduced geometry which consists of 630 twenty-noded brick elements consisting of 3835 nodes.

Because flange twist is highest beneath the point of load application, fracture of the weld at the web-flange junction is suspected. To model crack propagation in this region, it is necessary to "zoom in" on a small portion of the junction so that it can be modeled with boundary elements. A subobject of the junction from the refined analysis is shown in Figure 5.6 with the distribution of major principal stress contoured. The "hot spot" is directly beneath the load.

A small length of weld at the web-flange junction is then modeled using boundary elements as in Figure 5.7. Stresses transferred from the rail to the top of the flange are recovered from the subobject of Figure 5.6, converted to vertical tractions, and applied to the top of the

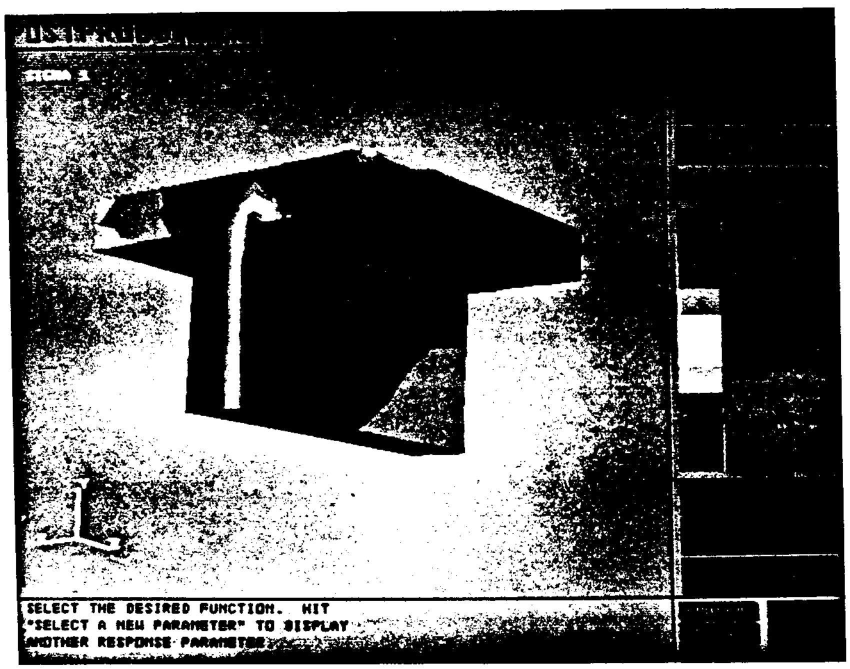

Figure 5.6 Subobject from refined analysis of web-flange junction beneath the applied load

C-2

ORIGINAL PAGE IS
OF POOR QUALITY

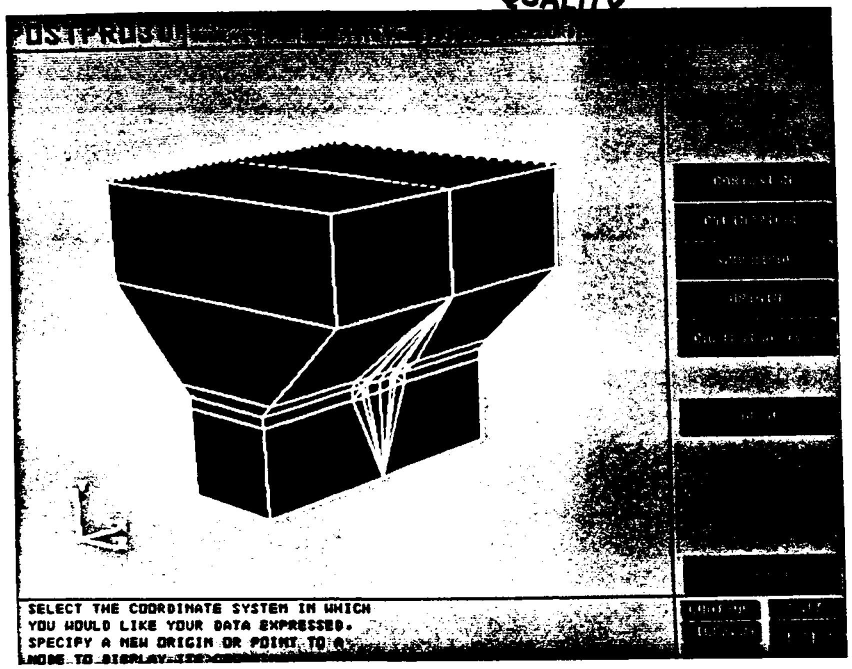

Figure 5.7 Boundary element model of weld at the web-flange junction

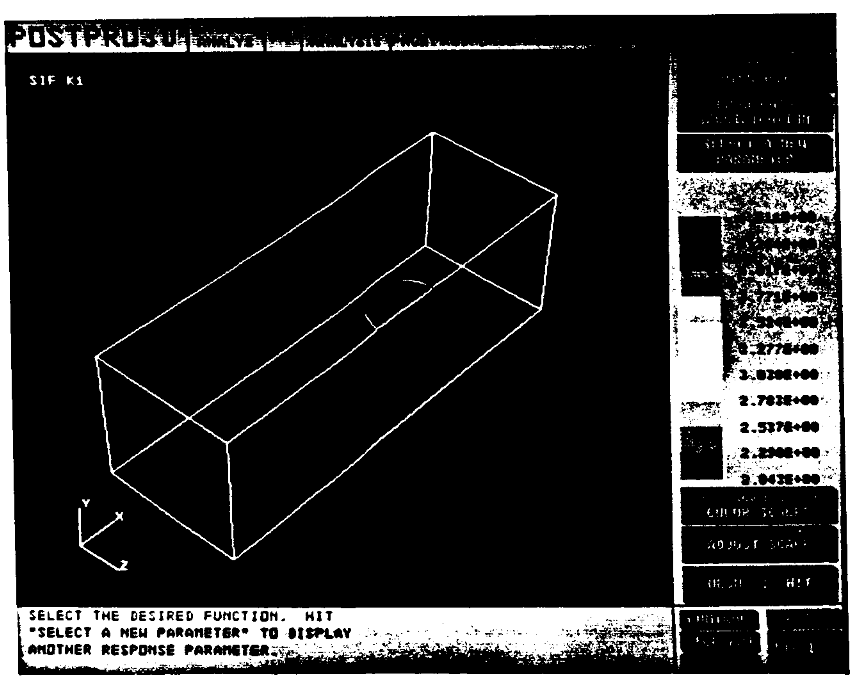

Figure 5.8 Distribution of stress intensity factor K1

boundary element model. Tractions on all other exterior surfaces except the bottom (where they are unknowns) are prespecified to be zero. Vertical fixity is imposed at its bottom, as is fixity which prevents horizontal translation. The model consists of 98 six- and eight-noded quadratic elements and 268 nodes in two domains. One domain is used for the portion of the web beneath the weld and the other is used for the weld and flange. An initial semi-circular crack is modeled between the domains. The material properties are identical to those used in the previous analyses.

It is desirable to predict the direction of subsequent crack propagation, so the variation of stress intensity factor K1 is displayed in the bottom domain in Figure 5.8. K1 is almost an order of magnitude larger than either K2 or K3, each of which may be displayed in a similar fashion as in Figure 5.8 but are not shown here. This predominance of K1 indicates that the crack will tend to propagate radially, with the largest growth near the highest values of K1 (at the crack opening). This observation is consistent with actual cranerail girders, which sometimes experience horizontal fracture along the entire web-flange weld.

Although the analysis used in this example is not rigorous because only approximate boundary conditions are used for each zoomed analysis, it serves to demonstrate the flexibility of the prototype postprocessor in aiding the understanding of complex numerical models. It also demonstrates the prototype's potential for unification of different analytical techniques. In the following example, a stepwise analysis is interpreted, and the implemented postprocessing techniques are found to make its behavior easily understandable.

5.2 Welded Tension Plates

In this example, the material nonlinear behavior of the welded steel connection analyzed by Han [12-13] is studied. Two sets of tension plates are welded to a connection plate as shown in Figure 5.9. For discussion purposes the long welds are referred to as longitudinal and the short welds are referred to as transverse. Due to symmetry only one eighth of the connection needs to be modeled. The modulus of elasticity is 29,000,000 psi and Poisson's ratio is 0.25. Yield stresses of the plates and welds are 36,000 psi and 42,000 psi, respectively, and residual stresses are neglected. The von Mises yield criterion is employed, and a strain hardening material model with a linear isotropic hardening rule is used. Twenty-one-noded brick elements are used for the plates, and 15-noded wedge elements are used for the welds. The total number of elements is 192 and the number of nodes is 1,270.

Uniform tensile stress is applied to the plate ends in eight nonlinear load steps. First, a linear elastic analysis is performed, and the results are scaled to obtain initial yielding. The eight load steps are then applied with increments equal to 0.05 of this yield load. The linear results are the first response set, giving a total of nine sets.

In Figure 5.10 the linear elastic distribution of effective stress is shown. Because the plates are connected only at the welds (which are

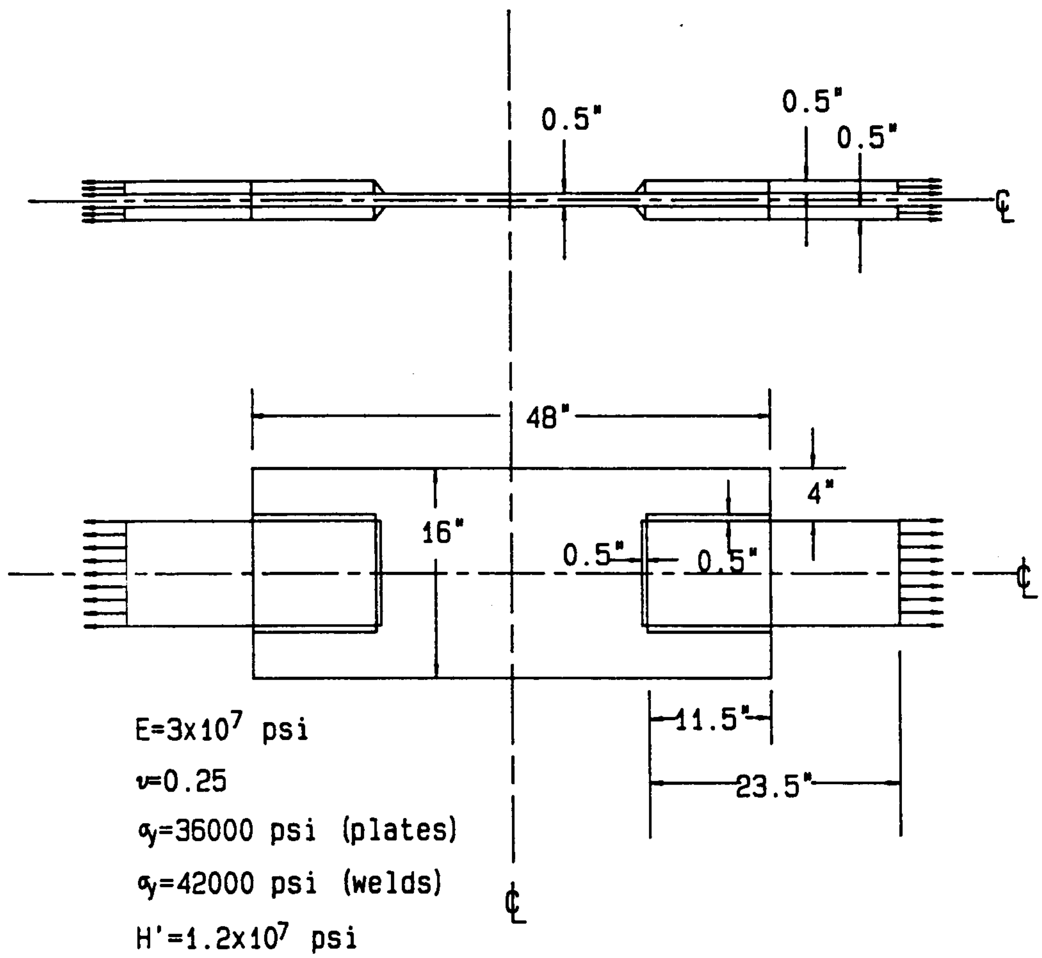

Figure 5.9 Tension plate connection modeled with material nonlinear finite elements

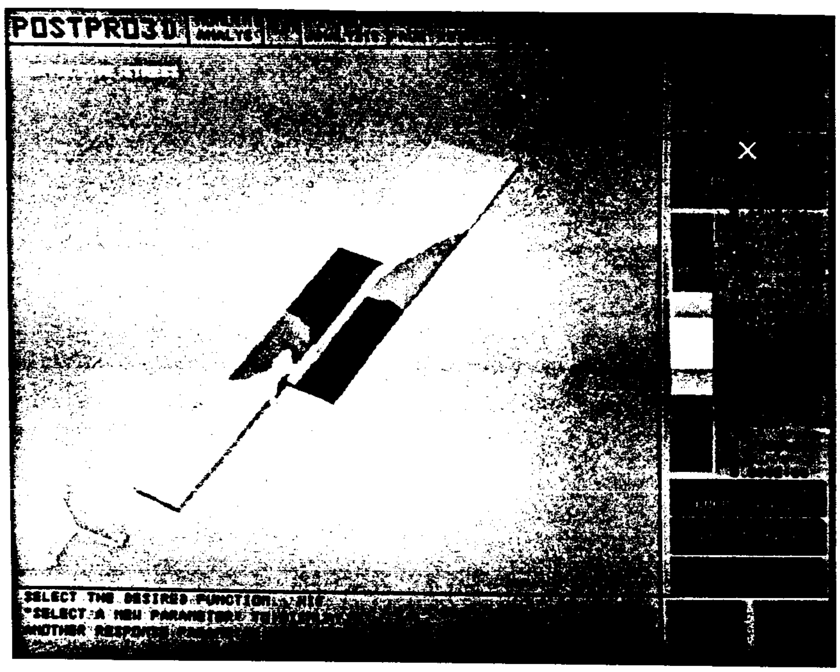

Figure 5.10 Contour distribution of effective stress for the linear elastic response set

a different material), no nodal averaging can occur between plates, or between a plate and a weld. The possibility that the magnitudes of stress at the stress concentration will be mistakenly reduced due to averaging is therefore eliminated. This is shown in Figure 5.11 where a subobject has been created of region near the stress concentration. The discontinuous distribution of the direct stress, sigma x, can be seen between the plates and the weld.

In Figures 3.6 and 3.7, the distributions of effective stress for nonlinear load steps 4 and 5 are shown. The maximum stress ranges are contrasted making the spread of yielding in the material more apparent. Yielding first occurs at the stress concentration as expected, however, yielding is not localized to this region alone. The highest stresses are transmitted through the connecting plate beyond the transverse weld, causing yielding and demonstrating this weld's higher stiffness relative to the longitudinal weld.

5.3 MARTA Twin Tunnels and Research Chamber

This final model has been discussed in several papers as an example of three-dimensional finite element analysis [14-15]. It is presented here not as an exhaustive numerical description of the tunnels, but as another example of the ease and flexibility with which the developed postprocessing techniques can be applied. Despite the complicated geometry, interactive postprocessing greatly simplifies interpretation of the behavior.

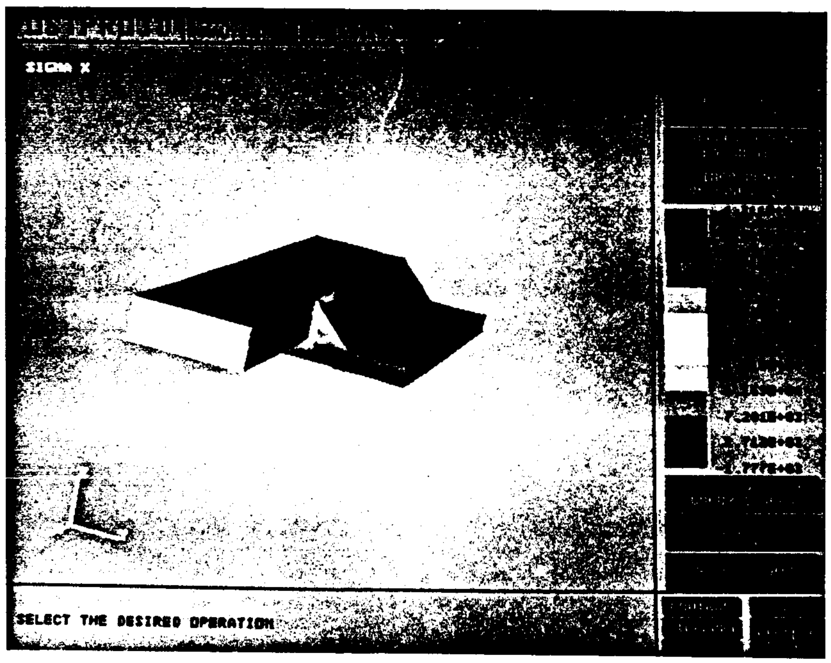

Figure 5.11 Subobject showing discontinuous distribution of direct stress sigma x between the plates and the weld

The geometry of the four tunnels of the model may be seen in Figure 5.12. The main subway cavern is the large opening, with the other three tunnels emanating from this cavern. The arched opening of one of the twin tunnels is seen in the center of the figure, with the tunnel extending to the left. One half of the research chamber is above and parallel to the twin tunnel (note that only half of this tunnel need be modelled because of the assumed symmetry condition). Finally, the transverse tunnel opens near the right side of the figure and extends in a direction perpendicular to the other three tunnels.

The three-dimensional finite element mesh used for the analysis is shown in Figure 5.13. The mesh is composed of 639 15-noded wedge elements and 20-noded brick elements for a total of 3362 nodes. Material properties are assumed homogeneous with modulus of elasticity of 720,000 ksf and Poisson's ratio of 0.17. To prevent nodal averaging at geometric discontinuities, different material types are assigned to the floor and walls, although the material properties of these are identical. Linear elastic behavior is assumed.

A symmetry plane is assumed through the center of the station cavern and research tunnel, thus nodes on this plane are fixed against translation in the x-direction. Boundary conditions on the bottom plane are fixity in the y-direction (vertical direction). Fixity in the z-direction (the direction of the main subway tunnel) is applied to the front plane. All loads are applied on the planes opposite the fixity planes such that the entire model is placed into compression. These loads represent in-situ pressure that was found to exist in the rock

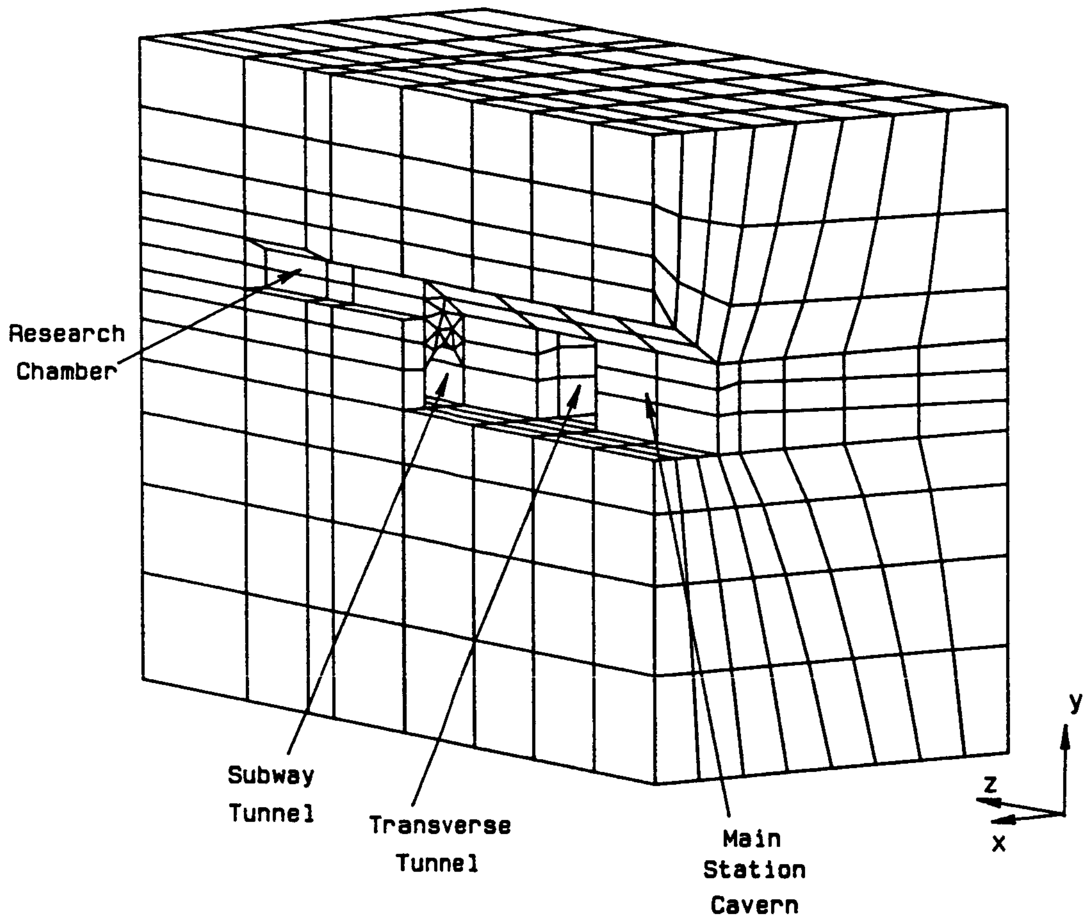

Figure 5.12 Geometry of the MARTA Twin Tunnels and Research Chamber

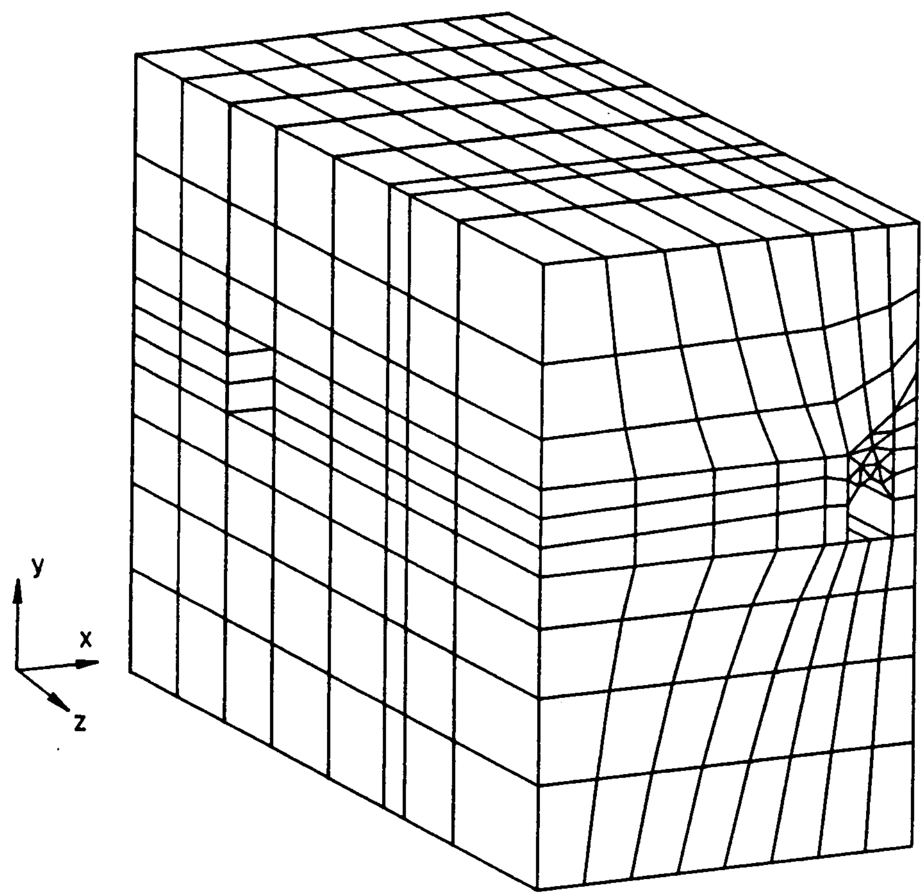

Figure 5.13 Second view of finite element mesh used in the analysis of the MARTA Twin Tunnels and Research Chamber

during testing before excavation. Uniform pressures in the (horizontal) x-direction and (vertical) y-direction are both 16.7 ksf and pressure in the (horizontal) z-direction is 83.5 ksf. Note that the largest pressure is not due to the overburden, but instead is due to very large in-situ stresses in the direction of the main subway tunnel.

In Figure 5.14 the exaggerated displaced mesh is shown. Displacements in the z-direction are large compared to those in the other orthogonal directions, due to the greater magnitude of load in this direction. A slight bowing in the walls of the transverse tunnel is evident. The displaced mesh is shown again superimposed over a subobject of the subway tunnel in Figure 5.15.

The distribution of sigma z is represented using a subobject in Figure 5.16 for the region of the transverse tunnel and main cavern. The contour distribution is discontinuous between the floor and walls of the transverse tunnel. Because the cross-sectional area of the model resisting the z stresses is reduced by the transverse tunnel, the stresses are high in the tunnel floor and ceiling, but are suddenly decreased at the tunnel walls. The walls of the transverse tunnel should be free of stress in the z-direction since they are free surfaces. However, these stresses are only close to zero due to the approximation of equilibrium which occurs in the finite element method.

Finally, the contour distribution of strain energy density is shown in Figures 5.17 and 5.18 for the previous subobject. In Figure 5.17, the strain energy on most of the free surfaces of the tunnels is close to

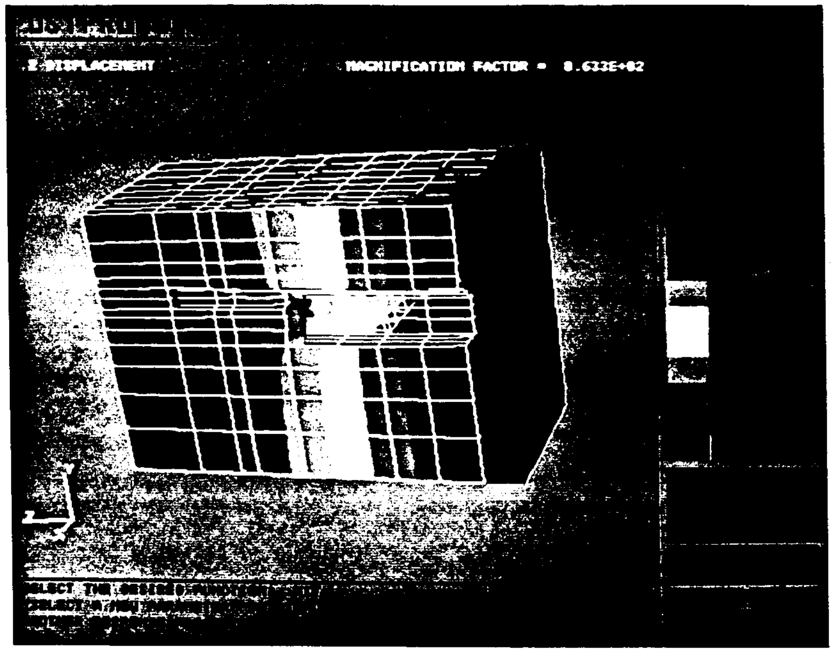

Figure 5.14 Displaced mesh of MARTA Twin Tunnels and Research Chamber

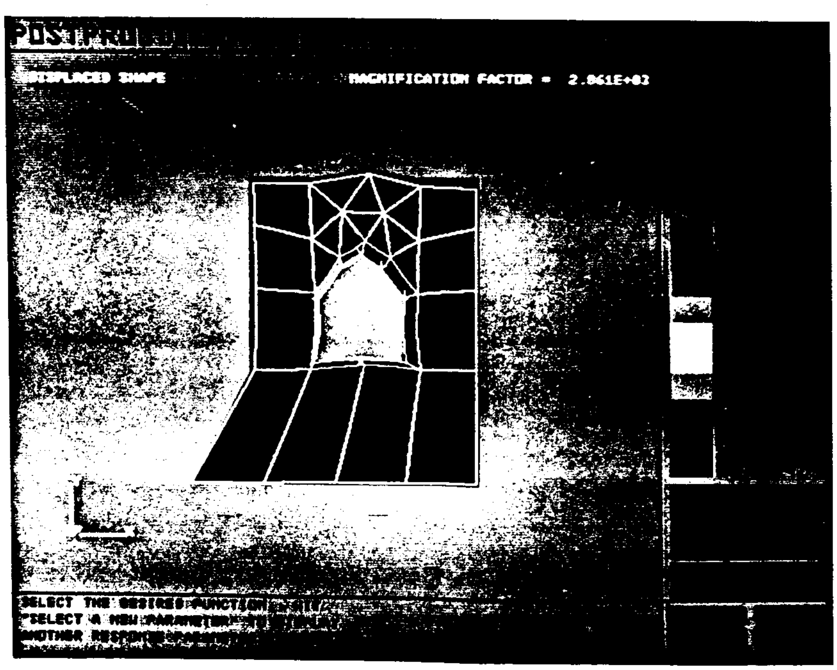

Figure 5.15 Displaced mesh superimposed over a subobject of the subway tunnel

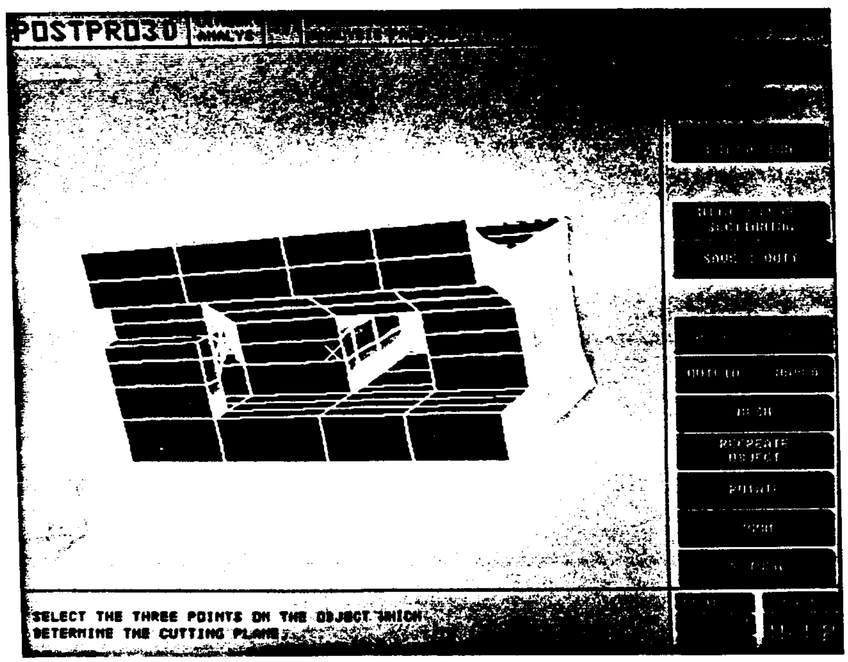

Figure 5.16 Contour distribution of sigma z shown for a subobject of the transverse tunnel and main cavern

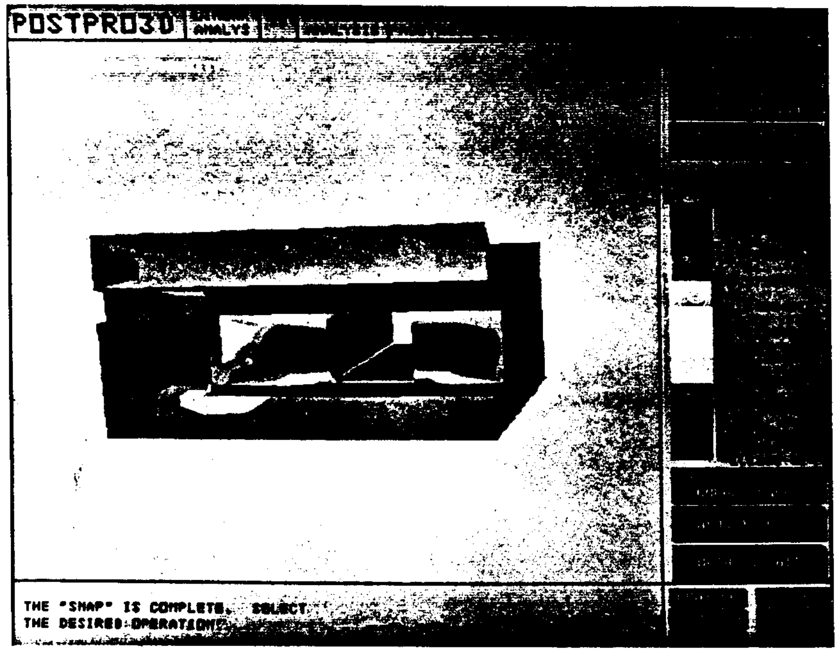

Figure 5.17 Contour distribution of strain energy density shown for the previous subobject

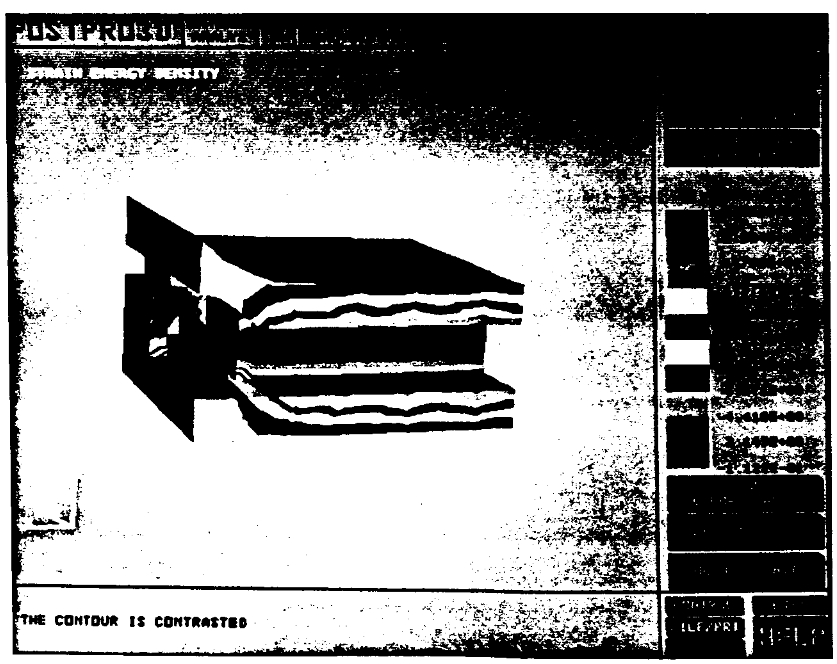

Figure 5.18 Contour distribution of strain energy density shown for the sectioned subobject

zero. The highest magnitudes occur in the ceiling and floor of the transverse tunnel, and on the corners of walls. Figure 5.18 shows the same subobject after it has been sectioned, thus exposing the strain energy distribution throughout the floor and ceiling of the transverse tunnel. The strain energy is high near the surfaces of the floor and ceiling, but decreases within short distances from these surfaces.

The previous examples show the potential of the current implementation for evaluation of various complex analytical models. Transition between analysis types can be facilitated, and behavior is easily interpreted. Most importantly, when an area of interest in a model is discovered, it can be spontaneously inspected in detail with any of a variety of tools.

CHAPTER 6

CONCLUSION

The purpose of this research is to develop techniques to facilitate response and attribute viewing in interactive postprocessing. Tools are introduced which together allow the analyst to evaluate results of geometrically complex numerical models effectively. These interactive techniques are currently being used in a prototype system for the evaluation of actual analysis problems, thus proving themselves through use.

This final chapter summarizes work during the second year of research and then speculates on the future of postprocessing. A revised look at the role of postprocessing in an analysis or design loop is suggested. Finally, suggestions for future research are presented.

6.1 Summary

Many numerical techniques employed today use complex three-dimensional models for the computation of behavioral parameters.

Methods of analysis exist for models involving time-varying boundary conditions or changing topology such as crack propagation. Nonlinear problems in which loads are applied in a stepwise or time-varying fashion are not uncommon. Both the input and the output from such analyses are so involved that interpretation of this information by non-interactive means is difficult and cumbersome, if not prohibitive. Evaluation of complicated numerical output presents a particularly challenging postprocessing problem, and it is important that postprocessing can effect not only result interpretation but also accuracy verification. Tools must therefore be available that enable the engineer or analyst to view responses and attributes easily for many types of complex analysis problems.

In the creation of postprocessing techniques and their assembly into the prototype system, attention is given to specific principles outlined during the first year of research:

(1) Generality of application. The prototype is currently being used to evaluate problems from a variety of engineering fields. Results from finite difference, finite element, and boundary element analyses have been studied.

(2) Flexible image manipulation. User control over the object being viewed is improved with the addition of features such as stress intensity factors, subobjects, color scale adjustment, and domain selection.

(3) Interactive graphical capability. Virtually all of the response viewing functions occur in an interactive mode, where the user maintains

a constant and friendly rapport with the system. Rotation speed is increased, and the database size is kept minimal to reduce data manipulation. Postprocessing of subobjects is always more interactive than for parent models. The performance of various significant stages of postprocessing image display is evaluated in Appendix B.

(4) Transportability of software. In light of probable hardware changes, an attempt has been made to decrease hardware dependence by segregating firmware routines within software. Impact of a future display hardware change is assessed in Appendix A.

As techniques were being created for interactive postprocessing, the prototype system was constantly being used for evaluation of engineering research problems. This use uncovered several limitations in the evolving system, the solutions of which led to useful and powerful postprocessing adaptations. These limitations fell into four categories: insufficient memory capacity, inadequacies in contouring, response over-smoothing, and slow response due to excessive database size.

The first adaptation is a more flexible method of memory allocation for the database to allow postprocessing of large analysis models. The next two adaptations are in response to contouring difficulties. A method which enables elements in regions of interest to be extracted from the parent is implemented. The extracted elements are called "subobjects" and can be evaluated independently from the parent model. Subobjects not only improve the resolution of contours in an area of interest, but they also allow the model to be broken into a collection of parts, provide an alternative to sectioning, and can be stored for later use. Another technique lets the user input cutoff limits for the color scale, thus forcing contours into regions where no details could otherwise be seen.

Response over-smoothing problems are now overcome through use of a node duplication technique designed for discontinuous responses at material interfaces. The final adaptation helps to prevent the database size from becoming excessively large. Multiple response sets from stepwise or multiple load case linear analyses are stored in files, with only a single response set being kept in the database at any one time.

Additional refinements are implemented in the system that are not direct results of the original restrictions. These refinements include faster rotations, selective hardcopy, image retrieval, acceptance of boundary element models, and paging through or superposition of response sets. Hit-testing can be used to obtain information directly for material types, responses, and node and element numbers. Finally, new response parameters can be displayed. These include strain energy density, stress intensity factors, effective stress, effective strain, traction, effective error stress, and effective error strain.

Because response viewing is critical to postprocessing, special attention is given to the response viewing interface in the prototype. The described postprocessing tools allow fast, interactive interpretation of behavioral parameters for problems of all sizes and geometric configurations. These tools are organized in a logical and flexible fashion, and it is the combination of these tools which make them particularly effective. Each postprocessing operation, taken individually, is useful for response interpretation, but when all these techniques are offered together, it is their synergistic effect which provides a powerful, interactive system for the interpretation of complex analysis results.

The implemented algorithms are described in detail. Example problems are presented which help to demonstrate how these new postprocessing techniques can be used to evaluate large and complex analysis models.

6.2 Future Postprocessing

While advances in display hardware will make some of the developed techniques obsolete, most of the techniques will remain invaluable to postprocessing. For example, hardware improvements will soon allow the more rapid rotation of solid, shaded images, eliminating the need to rotate representational outlines, but viewing operations such as positioning the light source, sectioning the model into subassemblies, and manipulating the color scale must always be available when interpreting behavior. In addition, response viewing techniques such as the display of stress intensity factors, paging through response sets, and hit-testing will become more valuable as analysis models increase in complexity.

The prototype postprocessor has been found to be useful in interpreting results of several types of engineering analysis. Consequently, it may be considered as a unifying agent between these methods. The postprocessor is in a key position for converting data from one modeling method into a form acceptable by another. For some cases, this allows the use of modeling techniques in situations where they otherwise could not be used. For example, it may not be feasible to perform a boundary element analysis on a large model of a girder, but if a finite element model is used instead, subobjects can facilitate conversion of data into a boundary element format. The reduced subobject surface-to-volume ratio then allows economical use of the boundary element technique. Eventually, an automatic refinement of analysis will be possible where subobject boundary conditions and loads may be used as input to an analysis which focuses on the region that the subobject defines. In future generations of postprocessing a larger portion of this data conversion should be performed within the postprocessor, with the user having interactive control over the process.

Optimization of analysis quality may eventually be automated through such means as the computation of error quantities during the nodal averaging procedure. These quantities can then be contoured to show regions of the model where accuracy is poor. It would also be possible for the engineer to use this information as feedback into a preprocessor, where refinement of the mesh can take place. Engineers would then be able to evaluate analysis accuracy quickly, allowing a more direct convergence to an accurate solution. Such tools, when implemented, should create a more integrated modeling system, which would allow not only more efficient postprocessing, but more efficient analysis and design.

Although postprocessing is ideally part of an analysis or design "loop," it is commonly thought of as the "last step" in a preprocessing-analysis-postprocessing sequence. It has been known for some time that postprocessing can be useful as a means of feedback to the preprocessing stage and of converting output data into a format suitable for other analysis programs [9]. As advancements in fundamental postprocessing techniques are made, more attention should be given to integrating the design loop. The role of interactive postprocessing would then be redefined to be a vertex in a preprocessing-analysis-postprocessing-redesign loop and not the "last step" in a linear sequence.

In the future, increasingly complicated analysis models will make interactive color postprocessing more desirable and reductions in hardware costs will allow it to become commonplace. For complex models, it will be advantageous to combine various numerical modeling procedures to make the most efficient use of each. When improved supercomputing techniques are applied to solutions of these problems, it may become possible to monitor increments of a continually changing analysis as is occurs. For example, it may be possible to watch the growth of a crack through a three-dimensional body as the analysis is being performed. Continued advancement of existing postprocessing is necessary if such ambitious goals are to be realized, but future systems will certainly rely on fundamental postprocessing approaches such as the one presented in this research. Appendix B addresses the performance of some specific aspects of postprocessing relevant to the future high-function interactivity and graphical display necessary for real-time monitoring of supercomputing simulations.

6.3 Suggestions for Future Research

While the developed postprocessing techniques enable the interpretation of a wide variety of engineering problems, these represent only the second year of a proposed three-year research effort. Several techniques should be developed during the third year to complete the investigation of fundamental tools.

1) Multiple viewports. With the addition of stepwise analysis and multiple load case capabilities, the ability to display multiple views becomes more desirable. By displaying different load cases, load steps, or time steps simultaneously, comparisons would be facilitated and an appreciation of overall analysis behavior would be more easily obtained. For any analysis, different viewports could be chosen for the display of different response components. This feature opens up intriguing possibilities and has the potential to increase flexibility of all postprocessing functions dramatically. Multiple viewports should not be implemented until the prototype system is moved to a higher resolution display device. This improved resolution will be essential due to the decreased size of each view.

2) Coordinate transformations. At present all behavioral information is displayed in cartesian space. For many engineering problems, however, alternate coordinate systems are desirable. For instance, the responses of an axisymmetric model are displayed most meaningfully in a cylindrical coordinate space. Other coordinate systems, such as spherical, should be available. Simpler database coordinate transformations should be possible by admitting the specification of alternative locations for the origins of each type of coordinate system. The enclosing surfaces used for defining subobjects are currently fixed with respect to the default coordinate system. User specification of the origin position would therefore make arbitrary orientation of the surfaces possible, increasing the flexibility of element extraction.

3) Expanded attribute viewing. Additions to the attribute viewing page would improve the existing capabilities. Although it is presently possible to display node and element numbers as well as element material properties, the display of loads and boundary conditions should be implemented as well. This would make available for inspection the complete set of original input data to the analysis. Note that these expanded attribute viewing tools would be most useful on a high resolution display device. Also, the use of differently colored overlay planes would increase the usefulness of these features.

4) Expanded response viewing. Several additional response viewing techniques should be investigated. Viewing of second-rank tensor directional information, such as stress trajectories, and of normal and

tangential tensor components would improve behavioral interpretation. It may also be desirable to display contoured response components on an exaggerated deformed geometry of the model. Finally, an potentially effective response viewing enhancement would involve the simulation of a transparent model. For example, all regions in the model with response values within prespecified cutoff values would be displayed as if they were transparent, while regions of response beyond the cutoff values would be shown as if opaque. This technique could be used to make clear the distributions of various response ranges throughout a three-dimensional model.

5) Animated hardcopy. For problems involving time- or load-varying behavioral parameters, the creation of movies will be a useful postprocessing technique. This can be accomplished with the present implementation for material nonlinear finite element analyses. A video camera can be used to create an animation which, when played back, shows the response contours as they change over the surface of the model.

6) Moving boundary problems or problems with changing topology. The ability to evaluate complex, continuously changing numerical models is necessary in an effective interactive postprocessor. Clearly, added complications, such as time-varying or moving boundary conditions in the study of seal-flow behavior or changing topology in crack-propagation models, increase the desirability of interactive postprocessing. The problem of changing response sets has been addressed with the tools implemented for nonlinear analysis and multiple load cases, but the problem of continuously changing geometric definition has not. This will

certainly prove to be a challenging, but rewarding postprocessing problem.

7) Integration of the modeling system. Possibilities for closing the preprocessing-analysis-postprocessing-redesign loop have been observed throughout this thesis. Subobject responses may be used as boundary conditions for refined analyses. Also, a study should be made to see if error quantities can provide enough information to a preprocessor for the purpose of mesh optimization. Because the postprocessing stage is a logical place for the processing of certain types of analysis information in the loop, future research efforts should pursue these possibilities.

APPENDIX A

CONSIDERATIONS FOR CHANGE OF DISPLAY DEVICE

Improvements in display hardware used for the development of postprocessing techniques are anticipated in the near future. Faster color raster display devices with larger color selections than the current device will be available. This appendix discusses the effect that new display hardware will have on the prototype postprocessor. One such new device is described, and adaptations to affected algorithms are proposed which will enable their continued use.

A.1 The Proposed Display Device

The device which is being considered is a 64-bit frame buffer with a built in 16-bit z-buffer, 1024 by 1280 resolution, and 60 Hz non-interlaced refresh produced by Raster Technologies. With this relatively large depth buffer, accurate hidden surface calculations can be performed quickly in the hardware. The other 48-bit planes make up the image memory of the frame buffer, which consists of two 24-bit buffers. The double-buffer configuration allows fast generation of new images by using one buffer for refresh of the display and the other for image update. When a new image has been completely loaded into the update buffer, the two buffers are swapped. This near instantaneous display of the new image allows dynamic simulation to be more smoothly approximated.

The 24 bits of image memory in each buffer are divided into three groups of 8 bit-planes, called channels. Each channel then serves as an 8-bit lookup into either a red, green, or blue color map. Each pixel is associated with all three channels and is therefore associated with three values, each ranging from 0 to 255. With this configuration a total of over 17 million colors can be displayed on the screen simultaneously.

A.2 Use of Overlay Planes

The new hardware does not have overlay planes built into the frame buffer, but the device can be configured so that one bit-plane from each channel is sacrificed for use as an overlay. Because these three channels all contain an index to an R, G, or B intensity, this results in three overlay planes, one red, one green, and one blue. If an overlay color other than red, green, or blue is desired, two or more of the overlays may be used simultaneously.

The use of overlays in this manner should be conservative. Sacrifice of two bit planes from each channel reduces the number of possible colors for the image by a factor of 64. If smooth-shaded images are to be implemented or if the number of colors in the color scale is to be increased (effectively approaching smoothly shaded contours), it may be desirable to perform a study to find the lowest number of necessary colors for these operations. Restrictions on the use of overlays should be set accordingly.

Bits necessary for the cursor are not taken from the image memory. The cursor is instead assigned fixed color indices, resulting in no reduction of colors used for the image.

Currently, the present implementation uses a separate overlay plane for display of the reference axis. It is recommended that this be changed so that the axis is displayed in the image memory along with the shaded object. Use of an overlay for the axis allows the axis to be turned on and off, but it has been found that users always leave the axis on. When the three-dimensional image of the model is updated as it is rotated, the axis can be updated and displayed as well. Postprocessing often requires the use of two overlays simultaneously, for example, to allow the mesh to be "toggled" on and off without affecting displayed node or element numbers. Display of the axis in the image plane will thus leave the third overlay as a spare.

A.3 Changes in Present Algorithms

Given the major differences in the two hardware devices, the impact on the present software can be assessed before it is switched to the new device. The principal difference between the two devices is the number of color channels, the Rastertech having three 8-bit channels per pixel while the Lexidata has only one 12-bit channel. This difference will necessitate changes in the current software, mainly in the color handling routines. Although it was attempted in the first phase of this research to create software that was hardware independent, the prototype postprocessor was unfortunately written under the assumption that a single channel would be available to each pixel. This will necessitate changes in several of the color handling routines, although it is felt that implementing these changes will not be difficult. Of course, when moving the postprocessor to the new display device, the number of channels will be have to be parameterized to prevent similar inconveniences in subsequent hardware changes.

Four algorithms will need to be re-constructed for use with the proposed hardware configuration: Specification of colors used while contouring, contrasting of specific contours, scrolling the color scale, and moving the light source. A description of the necessary changes follows.

The most critical algorithm which must be changed is the color selection method used when drawing the contoured image. Currently, the algorithm is organized so that a base color is first computed for each polygon. Depending on the response range of the region where the polygon lies, the polygon may be contoured with one of ten base colors from a given color scale. Next, the index to that color's desired shade is calculated by using a dot product between the vector normal to the polygon surface, and the light-source vector. The different color shades are organized sequentially in a single lookup table for each of the base colors, so the lookup table entry for the desired shade can be obtained directly.

The present algorithm can be altered for the new device by loading each of the red, green, and blue lookup tables such that each intensity value equals the index value minus one, i.e., the first entry in each table contains zero intensity, the second entry contains an intensity of one, and so on up to the 256th entry with an intensity value of 255. To compute the shade of a given polygon, an intensity corresponding to an ambient level of light would first be subtracted from the three intensity values (R, G, and B) of the required base or full intensity color (stored in the current database). Next, these values would be multiplied directly by the dot product of the polygon normal vector and the light source vector, and the ambient intensity would be added back in. This eliminates the possibility of any polygon appearing black. Finally, gamma-corrections could be applied to compensate for nonlinearity between display voltages and intensities [16]. The resulting RGB values would then be used as the lookup indices into the color maps, and the polygon could then be drawn.

Contrasting is the second algorithm which will require modification under the new hardware configuration. With the present software/hardware configuration, the user can contrast (darken) an existing contour color by pointing to it either in the main viewport or on the color map shown in the transient part of the menu. Once the base color is found, lookup table intensities for all shades of the base color are re-loaded with low values thus making the original contour dark.

This method of contrasting will not be possible with the new configuration because several colors on the screen may point to the same red, green, or blue lookup table entry at any particular time. In order to perform a similar function, the entire image will have to be redrawn. Each polygon which is to be contrasted can be pointed to the correct location in the database where RGB intensities for the darkened color are stored. These values must then be multiplied by the dot product described previously, so that the proper lookup table indices are obtained. One possible alternative to redrawing the entire image might be to store a list which associates each base color with all polygons drawn from that color. This way, when a color is hit by the user, only polygons drawn in that color will have to be redrawn. However, redrawing the entire object may prove not to be a problem if the Rastertech is able to render images fast enough to simulate dynamic rotations.

Because it will not be possible to reload the color lookup tables in the same fashion as is done presently, the image will have to be completely redrawn if the user wishes to scroll the color scale. This operation will occur identically to that described previously for color selection, but with new base color scale RGB values. But, as mentioned above, this may not be a problem with the increased speed of polygon rendering.

The last algorithm impacted by the hardware change is the one which allows the light source to be interactively moved by the user. This routine will have to be changed drastically. Lookup table values cannot be changed as the light source moves, because a particular red, green, or blue intensity may be indexed by any number of polygons. Shades of polygons will therefore have to be computed "on the fly" as the light source changes position. Unfortunately, each polygon will then have to be redrawn, and the interactivity of the algorithm will be reduced. Again, it is hoped that the fast display of polygons will minimize this problem.

APPENDIX B

FUTURE CONSIDERATIONS

In anticipation of large increases of computational and graphical power, information has been compiled regarding the current performance of certain postprocessing functions in the prototype system. These timing statistics represent only the CPU time of the VAX 11/780 host computer and not the time of the display processor. However, significant performance improvements in display processors (or "graphics engines") are expected as described in Appendix A. A description of the present hardware configuration is given in Chapter 1 and of a probable next-generation display device in Appendix A.

Ideally, supercomputing and supergraphics will allow the analyst to monitor a stepwise analysis as it is occurring. Because it is expected that supercomputing will perform some analyses rapidly enough for interactive monitoring, attention is given to the operations which will be necessary to create contoured images essentially as quickly as the numerical response data are produced. The operations discussed herein include, therefore, the translation of analysis results, the input of translated files into the postprocessor, the drawing of the contoured image, and the other most time-consuming algorithms of the postprocessing process. As a hypothetical example, assume that a three-dimensional fracture analysis of a machine part is performed. The postprocessor is used to watch the propagation of a crack through the part as the analysis is occurring, i.e., several analysis steps per second.

B.1 Translation of Analysis Results

Timing tests have been performed on the finite element translator. Because boundary element translation in the present research environment is less computationally intensive than finite element translation and because the boundary element translator is less frequently used, similar tests were not performed on it.

Translation of output from a finite element analysis in the current system is performed in four steps. The first step simply organizes the geometrical information of the model into a form which is readable by the boundary and outline extractors. The boundary extraction or computation of the polygonal representation is the second step of translation. Once this information is obtained, it is written to the .BOU file. If necessary, node duplication (which preserves discontinuous responses) occurs during this step (see Chapters 2 and 4). In the third step, a representational outline of the model is computed by finding the intersections of the boundary polygons. These intersections form the outline connectivities and are written to the .OUL file. The final step of translation first arranges control and geometrical information into the .COR file. Next, the responses are smoothed and averaged, the global limits are computed, and all of the information is organized into a form directly acceptable to the postprocessor database and written to the .PAR file.

The translations of several large finite element analyses containing multiple materials were timed. Because multiple materials increase the necessary amount of computation during the boundary extraction step, this was assumed to be a "worst case." It is also assumed that the supercomputer will send analysis results to the postprocessor one analysis step at a time. Therefore, only translations of single-step analyses have been timed.

The average measured translation time is 0.104 CPU seconds/node and is a linear function of the number of nodes in the model. This breaks down into the following average amounts of CPU time: The first step of translation is the least time consuming and takes only 0.005 CPU seconds/node. Boundary extraction is the most time consuming step of the translation and takes 0.063 CPU seconds/node. Outline extraction takes 0.008 CPU seconds/node. Finally, the last step takes 0.028 CPU seconds/node.

Boundary extraction is the most time consuming part of the translation. However, this operation does not depend on the analysis; instead, it depends only on the initial geometry of the model prior to analysis plus any changes that might occur during analysis due to changing topology. Therefore, in a parallel processing environment, it will be desirable to perform the initial boundary extraction prior to analysis and to update it at the same time as the adaptive analysis is occuring. Rotations of solid images should be possible with advanced graphical display hardware, and the outline extraction will therefore be unnecessary in such an environment. The second most time consuming portion of translation is the smoothing and preparation of the response information. At present, the time necessary for this operation is a fraction of the actual analysis time. Increased computational power brought by supercomputing will make this portion of translation negligible, especially if it is combined with analysis rather than executed as a special postprocessing stage.

B.2 I/O or Transmission of Translated Files

Two different methods of data input are currently implemented in the prototype postprocessor. The first method is used for the three geometric files necessary for postprocessing (.BOU, .COR, and .OUL) and simply reads the data a single record at a time. This is a slow method of input; information was recorded to move at a maximum rate of 32 kilobytes/second. The second method is the VAX Queued I/O utility and is used for the response file (.PAR) to pipeline data from disk storage directly into postprocessor memory. This method is very fast with maximum rates of transfer recorded up to about 1.8 megabytes/second. Presumably, in the future environment, transmission from the CPU to the workstation or graphic display would need to equal or exceed this rate.

As mentioned previously, the outline information will not be necessary in the future hardware environment. Unfortunately, the recorded rate of geometric information transfer is prohibitively slow. In fracture propagation, topological information will change after each step of the simulation. It will therefore be necessary to transmit the revised geometric information to the postprocessor at rates comparable to the current I/O treatment of response information (see Chapters 2 and 4). At the present rate of 1.8 megabytes/second, 3.5 such analysis steps could be input to the postprocessor per second for a problem the size of MARTA (see Chapter 5), for which the total size of the files for geometry and response is approximately 500,000 bytes. It is also likely that information for all analysis steps will be saved for further inspection after the actual simulation. It will be essential for the future workstation to have both sufficient storage and fast enough I/O capabilities if these results are to be stored and "played back" at a later time.

B.3 Drawing the Contoured Image

The final stage for the representation of analysis results is color contouring. In the prototype postprocessor, the contouring operation consists of four steps [3]. First, coordinates are updated with the current viewing transformation. Next, the polygon normal vectors are updated with the current rotation matrix. This allows polygons which face away from the user to be culled. The polygons are triangulated in the third step. When this step is complete, response and coordinate values are associated with each vertex. In the last step, the triangles are broken into subpolygons of different colors and sent to the display processor.

Each of the above four steps has been timed for the prototype postprocessing system. Average values are given. Five percent of the CPU time during contouring is spent updating the coordinates. About 3,610 coordinates can be updated in a CPU second. Updating the normal vectors takes only 1 percent of the total time. The number of normals that can be updated in a CPU second is about 2860. Triangulation is more time consuming and typically takes about 14 percent of the total CPU time. For polygons which represent the faces of quadratic elements, approximately 190 polygons can be triangulated in a CPU second. The most time consuming operation is the last step, which requires the remaining 80 percent of the total CPU time. In this final step, 160 polygons are subdivided and sent to the display processor each CPU second. For a model the size of MARTA this entire operation takes about 24 CPU seconds.

In the future environment, maximum advantage must be taken of hardware, and software performance must be improved if contouring is to occur quickly enough to keep up with a crack propagation analysis in which each step is computed in the order of one second. Operations such as updating of the coordinates and normal vectors should not occur each step unless the model is rotated between analysis steps, or the topology of the model changes. Also, if the updating of the normal vectors is not necessary for a particular step, the efficiency of triangulation may be increased by storing a list of all polygons which must be triangulated;

this would eliminate the culling operation. Finally, it is possible that future specialized display hardware will be able to perform the polygon subdivision procedure automatically. This would prevent the final step from becoming the bottleneck, because the actual rendering of polygons once they are subdivided should take little time with the new display hardware, which should be able to render several thousand polygons per second.

B.4 Other Postprocessing Algorithms

It is evident that to achieve real-time monitoring of numerical simulation all software must be made as efficient as possible. Advances in graphical display hardware will also be instrumental in eliminating time consuming operations presently implemented only in software, such as contouring. Union of software and hardware will be necessary for postprocessing which is powerful enough to display the large amounts of supercomputing analysis output as fast as it is computed.

There are two other aspects of the prototype postprocessor, sectioning and subobject extraction, which are presently computationally so intensive that response times are not sufficiently rapid to meet interactive standards. These are now entirely implemented in software. Although precise timing studies have not been made, a typical sectioning of the MARTA example requires on the order of half an hour of CPU time, while extraction of a subobject requires about four minutes of CPU time, but depends on the size of the subobject. Sectioning is so time consuming because of the triangulation process which is necessary to allow multiple cuts and because of an inefficient process by which newly computed data are merged with the database [3]. It is obvious that, if such capabilities are to be used with real-time monitoring of analysis, dramatic improvements must be made in the software performance for these operations. As with the polygon subdivision necessary for contouring, perhaps portions or all of these computationally expensive algorithms will be implemented in future hardware.

REFERENCES

1 Knowles, N. C., "Finite Element Analysis," Computer-Aided Design, Vol. 16, No. 3, May 1984, pp. 134-139.

2 Anonymous, "New Finite Element Techniques Rely on Graphics," Computer Graphics and Applications, Vol. 5 No. 4, April 1985, p. 21.

3 Hajjar, J. F., "General-Purpose Three-Dimensional Color Postprocessing for Engineering Analysis," Thesis submitted in partial fulfillment for the degree of Master of Science, Cornell University, Ithaca, New York, November, 1984.

4 Swanson, J. A., Cameron, G. R., and Haberland, J. C., "Adapting the ANSYS Finite-Element Analysis Program to an Attached Processor," Computer, Vol. 16, No. 6, June 1983, pp. 85-91.

5 Perucchio, R., "An Integrated Boundary Element Analysis System with Interactive Computer Graphics for Three-Dimensional Linear-Elastic Fracture Mechanics," Thesis submitted in partial fulfillment for the degree of Doctor of Philosophy, Cornell University, Ithaca, New York, January 1984.

6 Babuska, I., and Miller, A., "The Post-Processing Approach in the Finite Element Method-Part 3: A Posteriori Error Estimates and Adaptive Mesh Selection," International Journal for Numerical Methods in Engineering, Vol. 20, 1984, pp. 2311-2324.

7 Abel, J. F., "Interactive Graphics in Applied Mechanics," Proceedings of the Ninth U.S. National Congress of Applied Mechanics, ASME, New York, 1982, pp. 3-27.

8 Swezey, R. W., and Davis, E. G., "A Case Study of Human Factors Guidelines in Computer Graphics," Computer Graphics and Applications, Vol. 3, No. 8, November 1983, pp. 21-30.

9 Grieger, I., and Kamel, H. A., "Application of Interactive Graphics and Other Programming Aids," State-of-the-Art Surveys on Finite Element Technology, A. K. Noor and W. D. Pilkey, editors, ASME Publication H00290, 1983, Ch. 11, pp. 341-361.

10 Wawrzynek, P. A., and Ingraffea, A. R., "The Effect of Stiffeners on the Forces Transmitted to the Web-to-Flange Junction of Crane Runway Girders," AISE/Cornell University Crane Rail Girder Project, Document 85-3, Task II, Report No. 3, January 20, 1985.

11 Ingraffea, A. R., Gerstle, W. H., Mettam, K., Wawrzynek, P., and Hellier, A. K., "Cracking of Welded Crane Runway Girders: Physical Testing and Computer Simulation," AISE Annual Convention, 1985.

12 Han, T.-Y., "Adaptive Substructuring and Interactive Graphics for Three-Dimensional Elasto-Plastic Finite Element Analysis," Thesis submitted in partial fulfillment for the degree of Doctor of Philosophy, Cornell University, Ithaca, New York, May 1984.

13 Han, T.-Y., and Abel, J. F., "Adaptive Substructuring Techniques in Elasto-Plastic Finite Element Analysis," Computers and Structures, Vol. 20, No. 1-3, 1985, pp. 181-192.

14 Schwartz, C. W., Azzouz, A. S., and Einstein, H. H., "Example Cost of 3-D FEM for Underground Openings," Journal of the Geotechnical Engineering Division, ASCE Vol. 108, September, 1982, pp. 1186-1191.

15 Han, T.-Y., Perucchio, R., Huang, Y.-P., Kulhawy, F. H., and Ingraffea, A. R., Discussion of "Example Cost of 3-D FEM for Underground Openings," Journal of the Geotechnical Engineering Division, ASCE Vol. 109, November, 1983. pp. 1496-1500.

16 Foley, J. D. and Van Dam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, Reading, Massachusetts, 1982.