THREE-DIMENSIONAL STRUCTURAL ANALYSIS

USING INTERACTIVE GRAPHICS*

Johnny H. Biffle and Hugh A. Sumlin

Sandia Laboratories

INTRODUCTION

The design process for a structure consists of three phases: (1) the conception, from which a design drawing is produced; (2) the analysis, from which the integrity of the structure is determined; and (3) the redesign, if it is deemed necessary. When a three-dimensional analysis is needed, the time required to complete the analysis phase can be greatly reduced by the use of interactive graphics. The first area which requires a large amount of time is describing the structure geometrically to the analysis computer program when given a design drawing. When redesign is necessary, it becomes extremely important to be able to effectively redescribe the geometry of the structure for the analysis program. The second time-consuming area is the interpretation of the output from the analysis program.

When using the finite element analysis technique, the structure is subdivided into a finite number of distorted brick type elements or blocks; this results in the generation of a large amount of data which must be checked. The use of interactive graphics is an effective means by which to generate and check input data and to correctly and completely examine the large volume of output data. For a static analysis, the output data consist of the values of pertinent variables, such as stresses and displacements at each of the nodes (i.e., the corners of the elements) which are associated with the structure. For a dynamic analysis, velocities and accelerations are also output and a set of data exists for several times during the dynamic process.

7

If the structure is simple, without much curvature, batch processing is convenient and can be used effectively. However, for a complex structure, interactive graphics is a valuable tool. The analyst is provided the capability of not having to preselect the plots he wishes to see to examine the structure, as in a batch job. He can activate the job interactively and obtain the desired plotted data before leaving the interactive console. With the use of the Sandia Interactive Graphics System (SIGS), which consists of a Vector General cathode

*This work was supported by the U. S. Energy Research and Development Administration.

ray tube, a DEC PDP-9 computer, a CDC 6600 computer, and an interactive graphics computer software system, both input and output can be effectively and efficiently processed. The hardware and software systems are described in more detail in the appendix.

THREE-DIMENSIONAL INPUT PROCESSING

To describe the geometry for the analysis computer program, a mesh generation program is executed. Better analysis can be performed by using mesh generators because the structure can be divided into more elements than is practical by hand. MESH3D is a three-dimensional code (Ref. 1) which generates a mesh for which the input and logic is simple and straightforward, so it was selected as a basis for the interactive program.

In both the batch code and the interactive code, the process of generating a mesh consists first of selecting a rectangular set of elements which is to become a section of the three-dimensional mesh. The block of elements has I-elements in the Ix direction, J-elements in the Iy direction, and K-elements in the Iz direction (Figure 1). Twenty key points on the block are given x, y, z coordinate values; then the block is isoparametrically mapped into x, y, z space with the key point x, y, z coordinates controlling the mapping process (Figure 1). The structure is composed of a number of these building blocks. To examine the resulting generated structure, the three-dimensional representation must be plotted from several viewing directions. By examining the mesh, the analyst must decide if the geometry is defined correctly and if the size and distribution of the elements is what is needed for an accurate analysis.

With interactive graphics, the analyst can interact with the mesh generation program to edit data, plot when it is desirable to examine the mesh during execution, and re-execute after editing. Because the generator is modular in design, if he desires the analyst can plot after each operation or series of operations which generate a portion of the mesh. If there are errors in the mesh, the generating data can be displayed, corrected, and re-executed by interacting with the computer through the use of a light pen and a keyboard. Since all input errors in the initial data set can be corrected during the interactive session, the time consumed by resubmitting batch jobs to correct data is eliminated.

The ease with which the mesh generation code lends itself for use with interactive graphics is inherent in the concept of constructing the structure by assembling a number of building blocks. The operation for each portion of the structure and the block of elements which make up the portion can be completed before other parts of the mesh need to be generated for the rest of the structure.

The key words, DEFINE, MAP, MERGE, JOIN, COPY, ERASE, and UPDATE, are used to describe each operation involved in building the structure. For example, the divisions along the sides of a block are defined and then the block is mapped into x, y, z space. Another block can be defined and mapped; then the two blocks can be merged to form a new block. A list, such as DEFINE, MAP, MERGE, COPY, MERGE, etc., in the order the analyst wants the execution to follow is displayed on the screen. The analyst can then execute down the data, examining the results by plotting, change the data of a key word or add and/or delete key words and their associated data as is necessary. When the interactive session is over, the output consists of a set of data which can be used to execute the batch version of the mesh generator. If changes are necessary at a later time, the data can be used as input to restart the interactive program.

The surface of the structure is plotted on the cathode ray tube for examination. Rotation of the structure about its geometric center is accomplished by turning knobs on the control dials at the console, and the results are viewed instantaneously. Each knob provides the capability of rotating the body about an axis which is embedded in the body. After a desired view is obtained, as in Figure 2, a request can be made for a picture without hidden lines; the result is as shown in Figure 3. Either node or element numbers can be added to a hidden line drawing (Figure 4).

An example of building a structure is depicted in Figures 5 through 8. Figure 5 is a hidden line drawing of block 1, and Figure 3 is a view of block 2. A new block 1 is created by merging blocks 1 and 2 to obtain the block shown in Figure 6. A new block 2 is then created and merged with block 1 to obtain the structure shown in Figure 7. A rotated view of the block of element in Figure 7 is shown in Figure 8.

As has been mentioned, the design of the interactive graphics code is built around the batch version of the code. The program is overlayed and dynamically dimensioned, which means that only variables which need to be used are in core at the time of use. The data are kept in extended core storage (ECS) in the batch code but are put on a disk file with a data manager for the interactive code. However, all the interfaces with the data manager are in the same location as they were in the batch code, so conversion to ECS can be done efficiently when it becomes available to the graphics system.

A schematic of the organization of the PDP-9 and CDC 6600 function is shown in Figure 9. The commands EXECUTE, EDIT, PLOT, RESTART, ADD, DELETE, or TERMINATE are initiated at the PDP-9 machine by the use of a light pen. Control is then transferred to the CDC 6600 where the overlay CONTROL calls the appropriate overlays, such as DEFINE or BLDAR, to accomplish the requested task. When the task is completed or there is an error, control is passed to the PDP-9 for further action, with the exception being TERMINATE for the end of the interactive session. Errors, when recognized, are returned through the overlay BURP to the PDP-9 and displayed on the CRT.

When the interactive job is initiated, the CDC 6600 program starts with overlay PROSTRT to initialize the location and size of arrays which are needed in blank common. Then control is passed to the overlay CONTROL where overlay BLD reads an initial set of data from a disk and stores the data on a random disk file through the use of a data manager. All data throughout execution are carried on a disk file, and only data which are needed for an operation are made available in core through the use of the data manager. Next, CONTROL calls overlay BILDISP which builds the main menu display with the key words of the initial data set and the PDP-9 commands. The main menu is sent to the PDP-9, and the interaction begins.

The main menu, with which the analyst works to activate the PDP-9 commands, consists of the commands and, as mentioned in the discussion of the key words, a key word list which represents the data necessary to build the mesh. If EXECUTE is hit with the light pen, the key word down to and through which the analysis is to be executed is hit. Control returns to the CDC 6600; and when the execution is completed or an error is detected, the main menu is displayed on the CRT and control is passed back to the PDP-9. Error messages are written on the display using the overlay BURP in the CDC 6600 program.

If EDIT is hit, the key word of the data the analyst wishes to edit is hit. Control is passed to the CDC 6600, and it creates an interactive display containing the data of the key word. Control then passes to the PDP-9, and the data are then displayed and corrected by the use of the light pen and keyboard before control is passed back to the 6600 to obtain the main menu.

If PLOT is hit, the analyst is asked what block he wishes plotted and then the CDC 6600 creates a display file for the plot. When control is returned to the PDP-9, the plot is displayed and can then be rotated. When the analyst is through examining the plot, the main menu is again displayed.

The command RESTART is used when the analyst wishes to reallocate blank common. The structure of blank common is such that the analyst must define in advance the number of nodes and elements in each block 1 through M where M is the total number of active blocks of geometry at any time. The overlay PROSTRT is used to accomplish the task, as in the beginning of the execution of the program.

If for some reason the analyst wishes to add a complete set of data including its key word, he hits the PDP-9 command ADD. A menu of key words is displayed from which to choose. With the light pen he hits a key word in the data set on the display just above where he wishes the data to be placed. The CDC 6600 then returns a display for the key word with a set of blank data which the analyst fills in before the data are inserted in the data base.

To delete a complete set of data and its key word, the command DELETE and the key word which represents the data are hit, causing the data to be deleted from the data manager file at the CDC 6600.

TERMINATE is used to end the interactive session. The corrected set of data is taken off the data manager file by overlay WRTOUT and written in a coded format on a disk. The set of data can then be used either to restart another interactive session or as input to the batch version of the mesh generator. A list of coordinates and node numbers which defines the structure located in block 1 is written to the OUTPUT file by the overlay OUTER, and the CDC 6600 execution is terminated.

If for any reason there is an abnormal termination of the job through an arithmetic error at any time during the interactive session, a recovery is made and overlay WRTOUT is called to create a set of input as it stood at the time of the error. The recovery was added to prevent loss of corrections the analyst had made to his mesh up to the time of the error. A new CDC 6600 job can immediately be activated and the analyst can continue to construct the mesh.

THREE-DIMENSIONAL OUTPUT PROCESSING

As with the three-dimensional mesh generation there exists a batch code (PLTZ, Ref. 2) which is capable of performing all the needed tasks for output processing of three-dimensional data from a finite element code.

To display output data on a plane which cuts the structure, the plane is defined and then the following options are available.

Option 1 -- A plot of the intersection of the mesh with the cutting plane, showing both interior and exterior mesh lines.

Option 2 -- A plot of the intersection of the exterior boundary of the mesh with the cutting plane.

Option 3 -- A contour plot of the values of a variable (stress, strain, temperature, etc.) within the body on the cutting plane. The program accepts values of a variable at each node to use as data for contour plotting. The exterior boundary as in option 2 is also plotted.

Option 5 -- A plot containing option 2 with the magnified displacements superimposed.

For example, the three-dimensional object shown in Figure 10 can be cut with a plane to produce a deformed shape (Figure 11); or option 4 could be used to produce a contour plot of a variable (Figure 12). It is also possible to define x and y limits so that blowups of a plot can be obtained (Figure 13). In the interactive mode, all this is accomplished by merely editing a small amount of card cata for the batch code or the data which appear on the screen with the light pen and keyboard for the interactive program. After editing the data in the interactive mode, the plot is drawn in approximately 5 to 10 seconds. While sitting at the console, the analyst can view as many plots as he deems necessary to understand the output.

The CDC 6600 computer code is organized in the same manner as the batch version is organized. There is a main program which instead of reading card data retrieves a set of data from the PDP-9 to execute to produce a display file. The file is then displayed on the CRT along with the control data which was used to generate the plot. If the analyst wishes another plot, he changes only the data necessary to obtain the next plot and then signals the PDP-9 with the light pen to activate the CDC 6600 program. The cycle is repeated until all the plots that are needed have been generated.

APPLICATIONS

To date, several problems have been analyzed which have utilized the capabilities of the graphics computer programs. In the area of electronic component design, the stresses and displacements associated with the epoxy in which components are encapsulated are important under various environments. The isometric view of the mesh for a particular analysis and output processing plots are shown in Figures 10 through 13. The encapsulant experienced cracks during a gun firing environment, which spun the assembly about the Z-axis. The analyst determined the stresses and displacements with and without a supportive ring on the circumference of the encapsulant.

Figure 14 shows the three-dimensional mesh which was used in a static analysis of a component design. The structure is a container for an electronic component, and the load consisted of a 10,000 g inertial loading and a 7.58×10^6 Pa pressure loading.

CLOSURE

With the interactive computer programs, the analyst can effectively perform three-dimensional input and output processing. Future work will be done in the area of output processing to remove the restriction of specifying a cutting plane by using visualization of a plane cutting and structure. With a color CRT, output processing will be performed to view deformed shape isometric plots and to view the contours of variables on the surface by using intensities of different colors.

APPENDIX

INTERACTIVE GRAPHICS SYSTEM HARDWARE AND SOFTWARE

The SIGS hardware, which is shown schematically in Figure 15 and pictorially in Figure 16, consists of a CDC 6600 computer, a DEC PDP-9 computer, and a Vector General 3D2 display. The CDC 6600 is used to generate all the Vector General display commands using Sandia-coded FORTRAN callable display generation routines. The 24K memory, DEC PDP-9 computer services the display interrupts and provides the necessary "bookkeeping" for the 6600-generated display file. Each of the Vector General displays is equipped with an alpha-numeric keyboard, a light pen, a character generator, display hardware subroutining, a 21-inch high-speed cathode ray tube, picture label scaling, intensity modulation, control dials, data tablet, and an interface to the PDP-9 through a direct memory access multiplexer.

CDC's Interactive Graphics System (IGS) provides the software interface to CDC's SCOPE operating system. At the remote terminals, the PDP-9 computer interfaces with the IGS through a Sandia-coded executive device handler (DPB). DPB handles all code translation, format conversion, communication synchronization, input-output file transmission, data transmission, and display file transmission.

A Sandia-code device handler (VGI) services all Vector General display interrupts, starts and stops the display, and allows the display file to be modified and manipulated. The Sandia-coded display manipulative routines are PDP-9 FORTRAN callable routines which permit the user to interact with or change all or part of the display using the light pen and/or keyboard.

REFERENCES

1. Hutula, D. N., and Zeiler, S. M., "MESH3D: A three-Dimensional Finite Element Mesh Generator Program for Eight-Node Isoparametric Elements," Bettis Atomic Power Laboratory, Pittsburgh, Pennsylvania, AEC Research and Development Report WAPD-TM-1079, March 1973.

2. Hutula, D. N., and Wiancko, B. E., "MATUS: A Three-Dimensional Finite Element Program for Small-Strain Elastic Analysis," Bettis Atomic Power Laboratory, Pittsburgh, Pennsylvania, AEC Research and Development Report WAPD-TM-1081, March 1973.

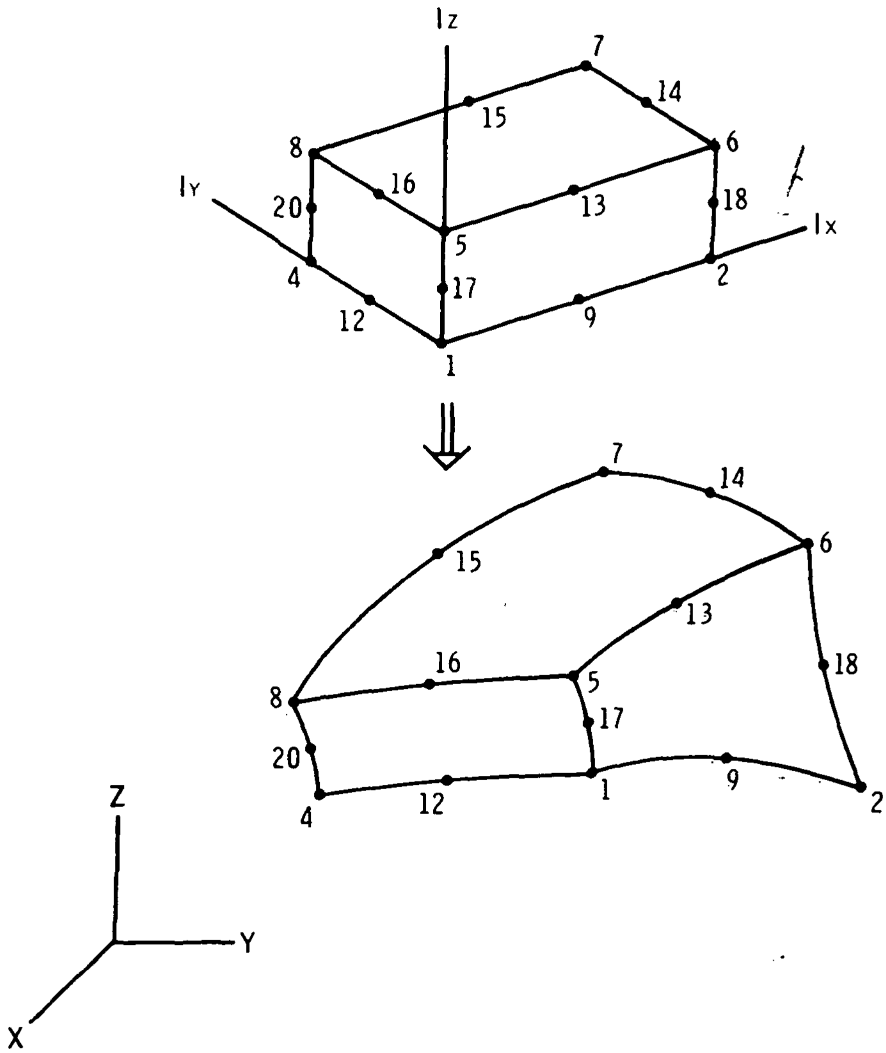

Figure 1. Mapping an integer block of elements into a curvilinear block of elements in x, y, z space.

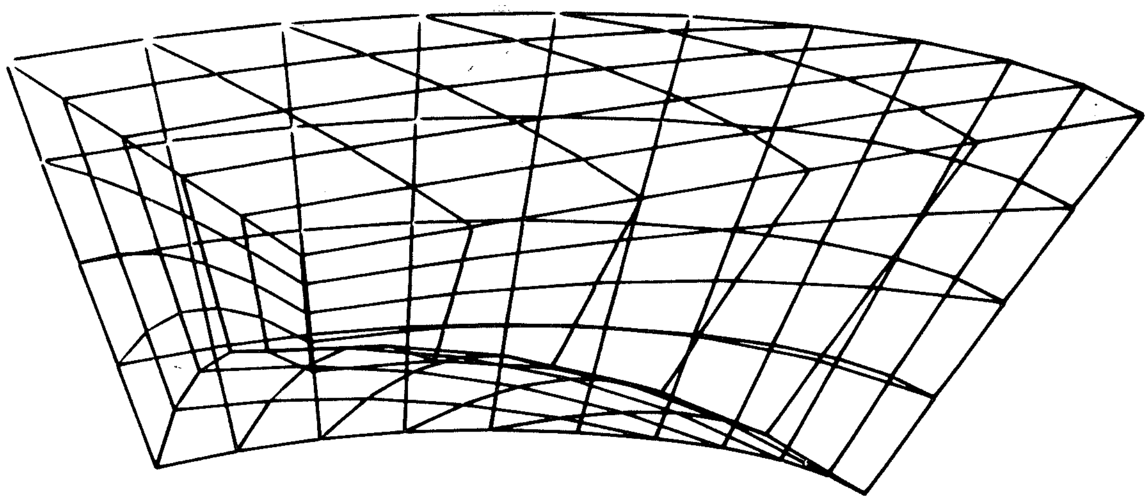

Figure 2. Rotated view of three-dimensional mesh.

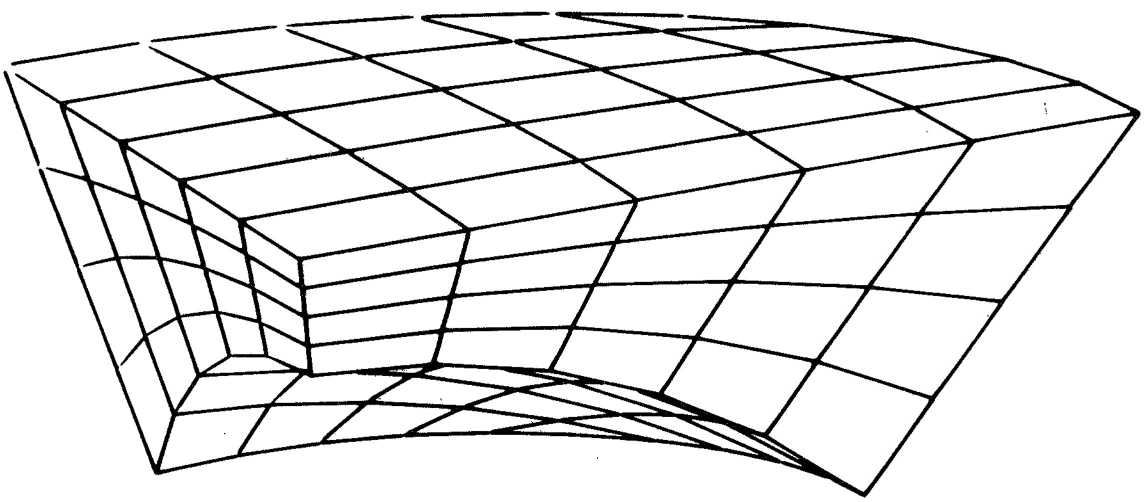

Figure 3. Hidden line view of three-dimensional mesh.

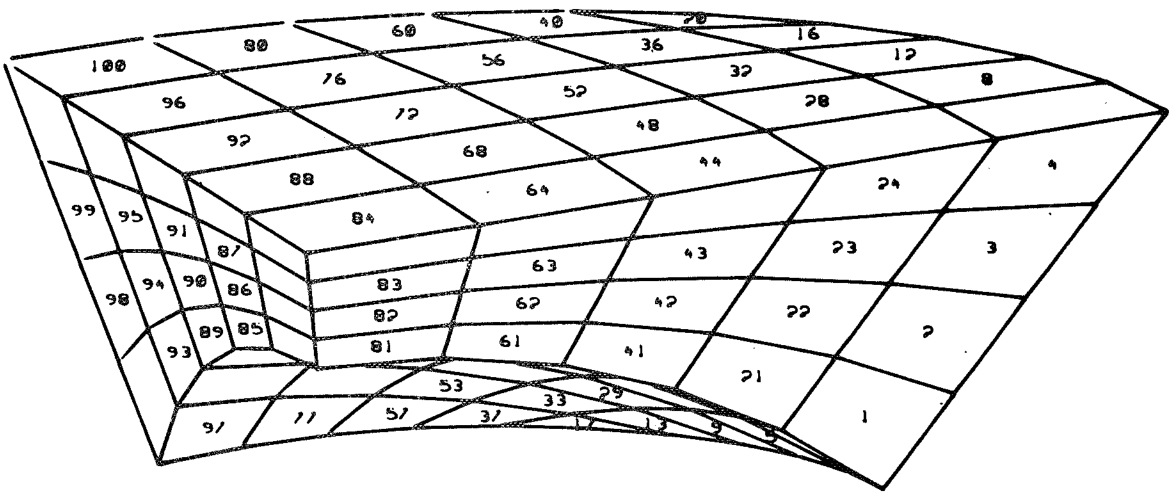

Figure 4. Hidden line view of mesh with element numbers.

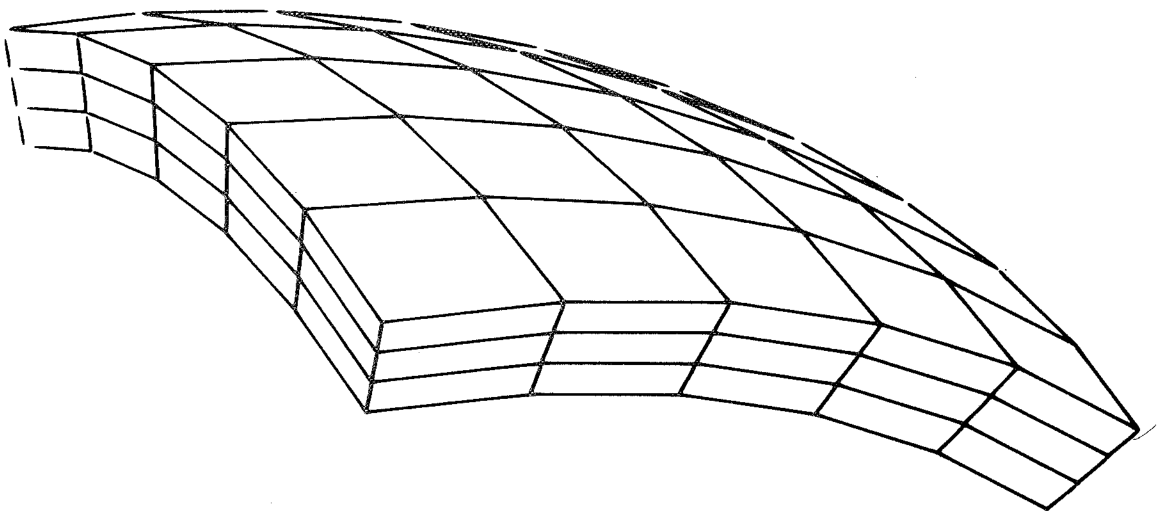

Figure 5. View of block 1 mesh.

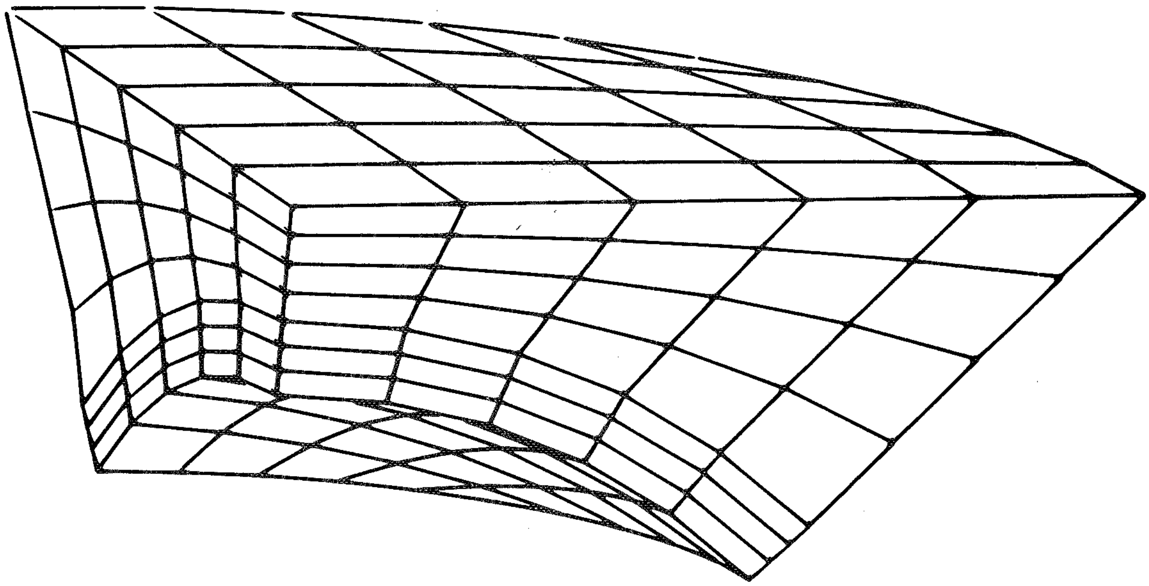

Figure 6. View of block 1 and block 2 merged.

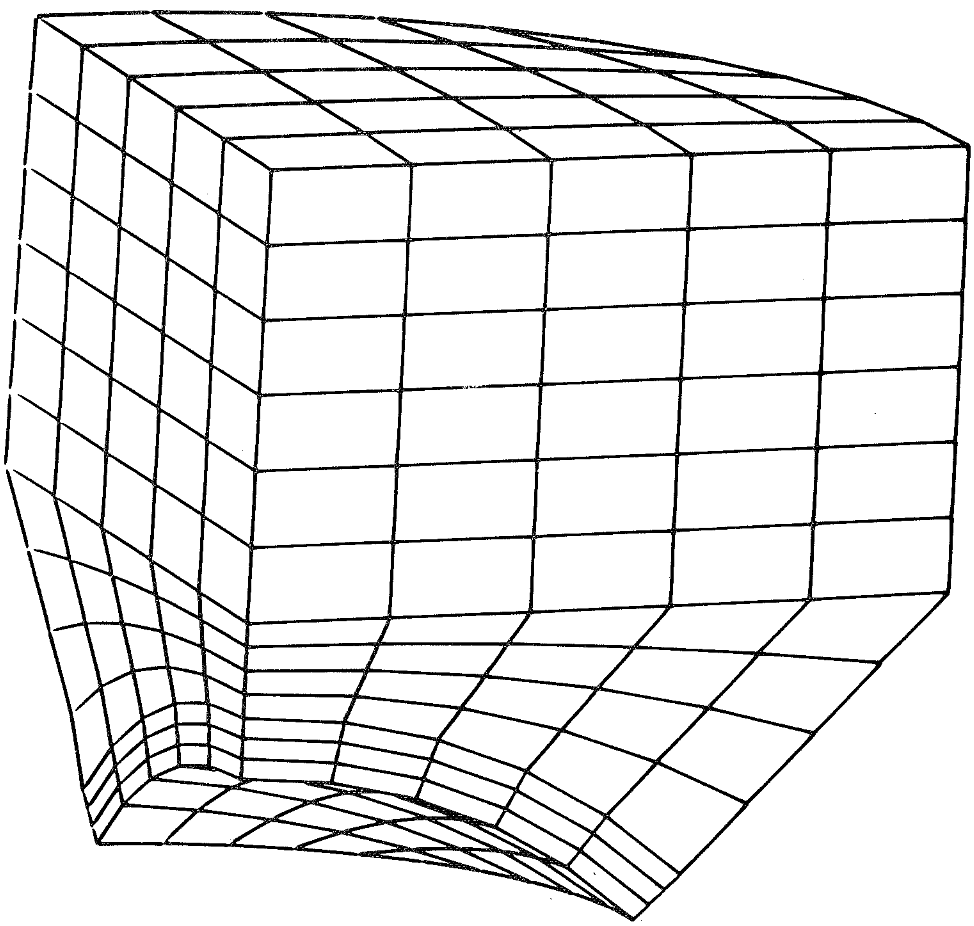

Figure 7. View of 3 blocks merged.

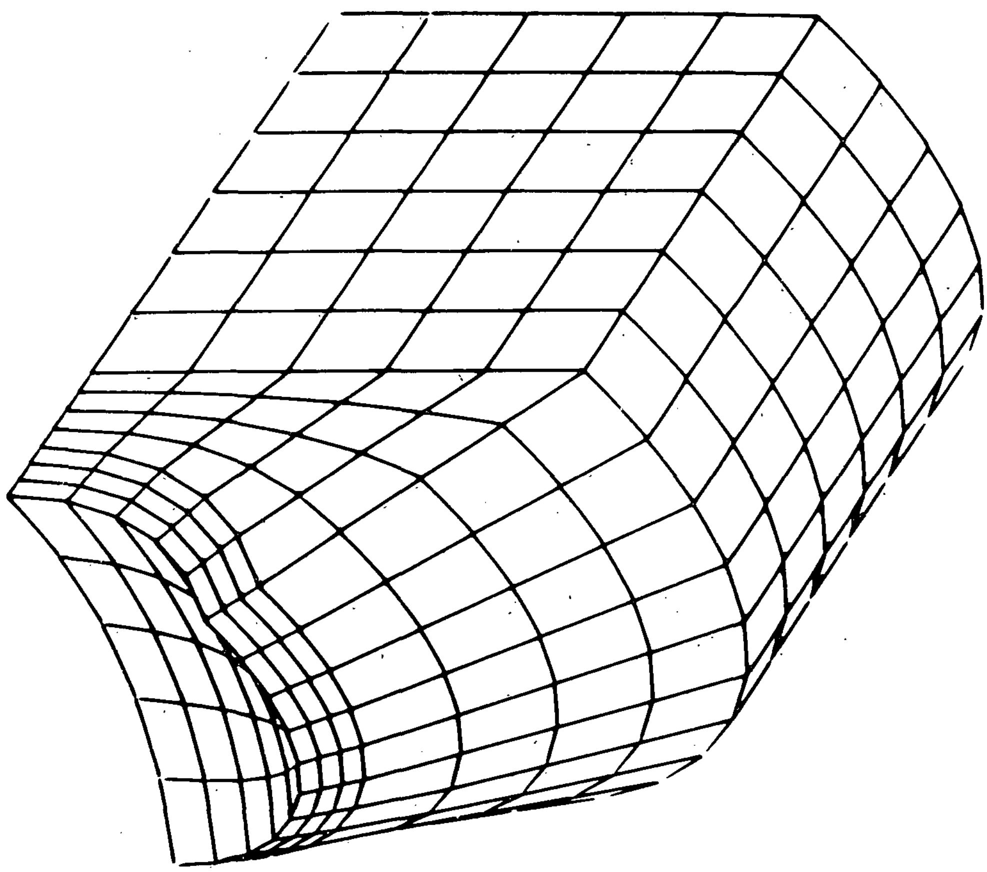

Figure 8. Rotated view of mesh.

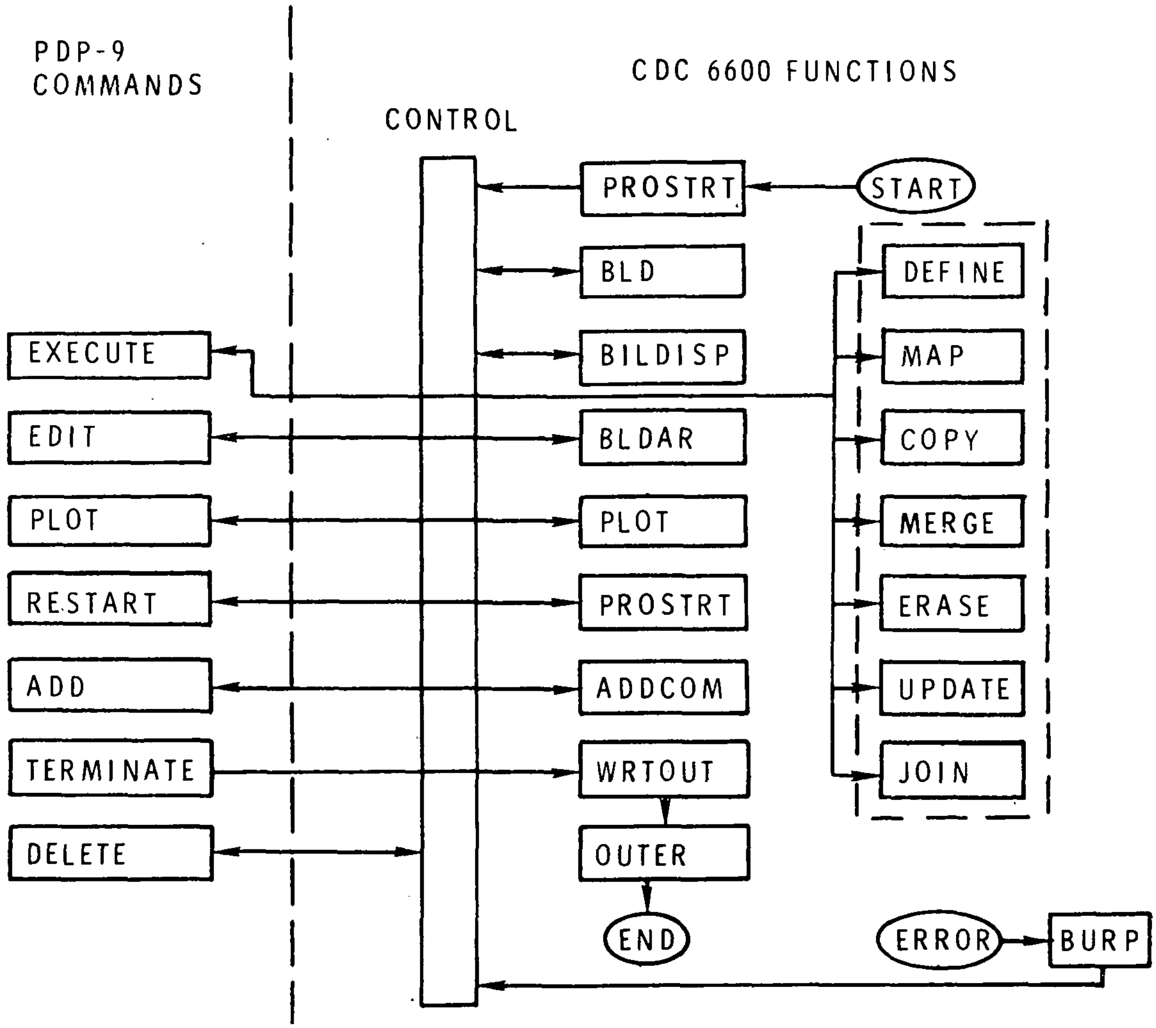

Figure 9. Interactive mesh 3-D computer code organization schematic.

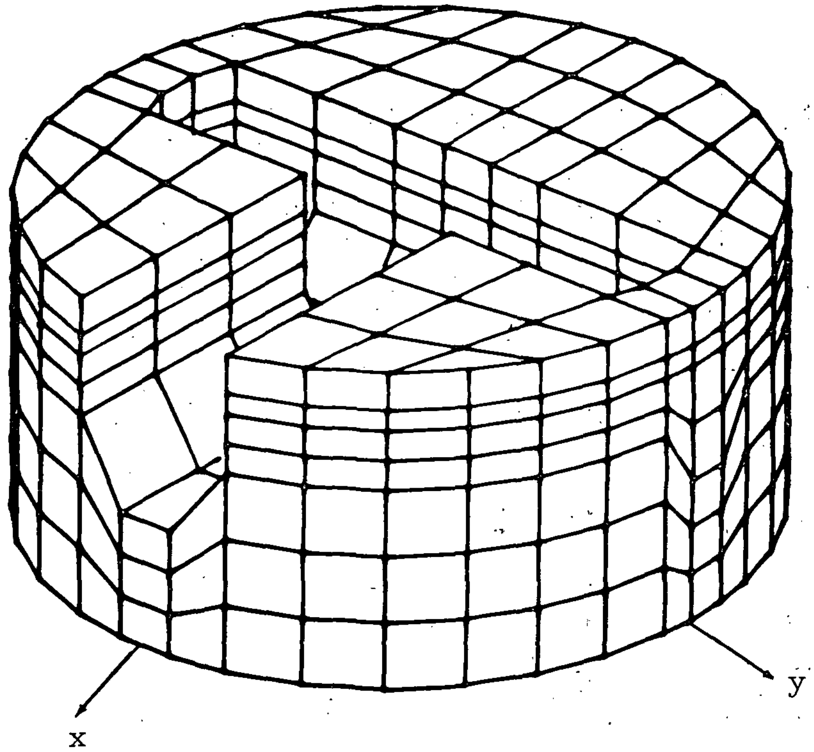

Figure 10. Three-dimensional view of encapsulant surrounding an electronic component.

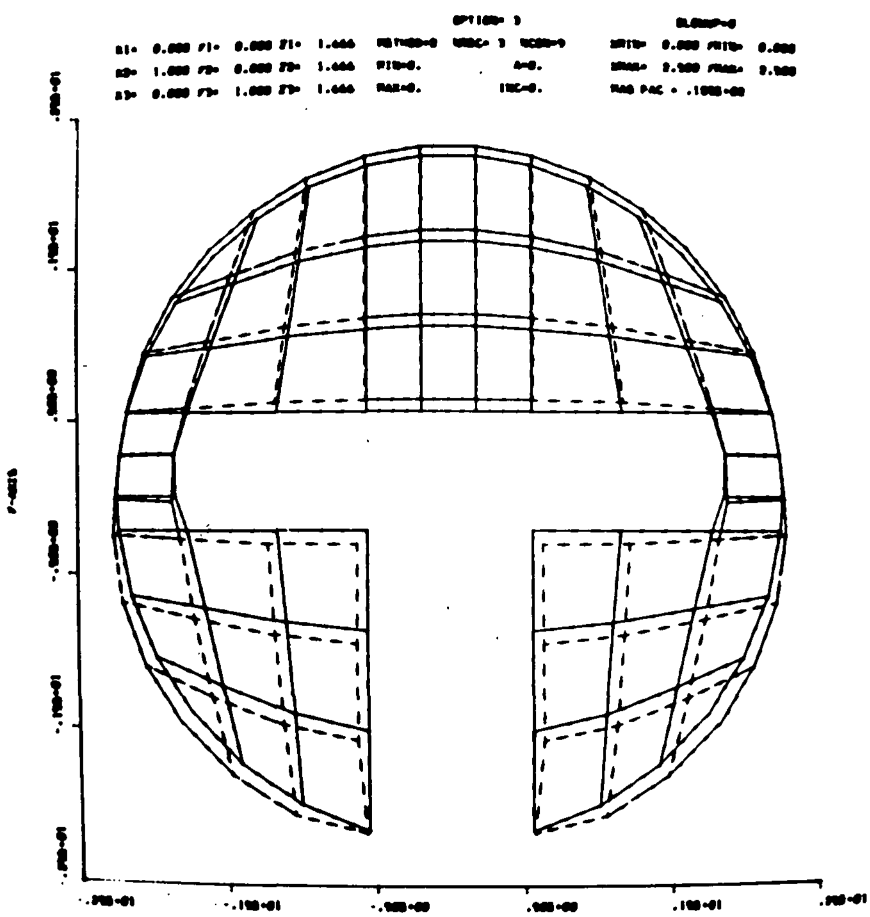

Figure 11. Deformed shape of component encapsulant in a spin environment.

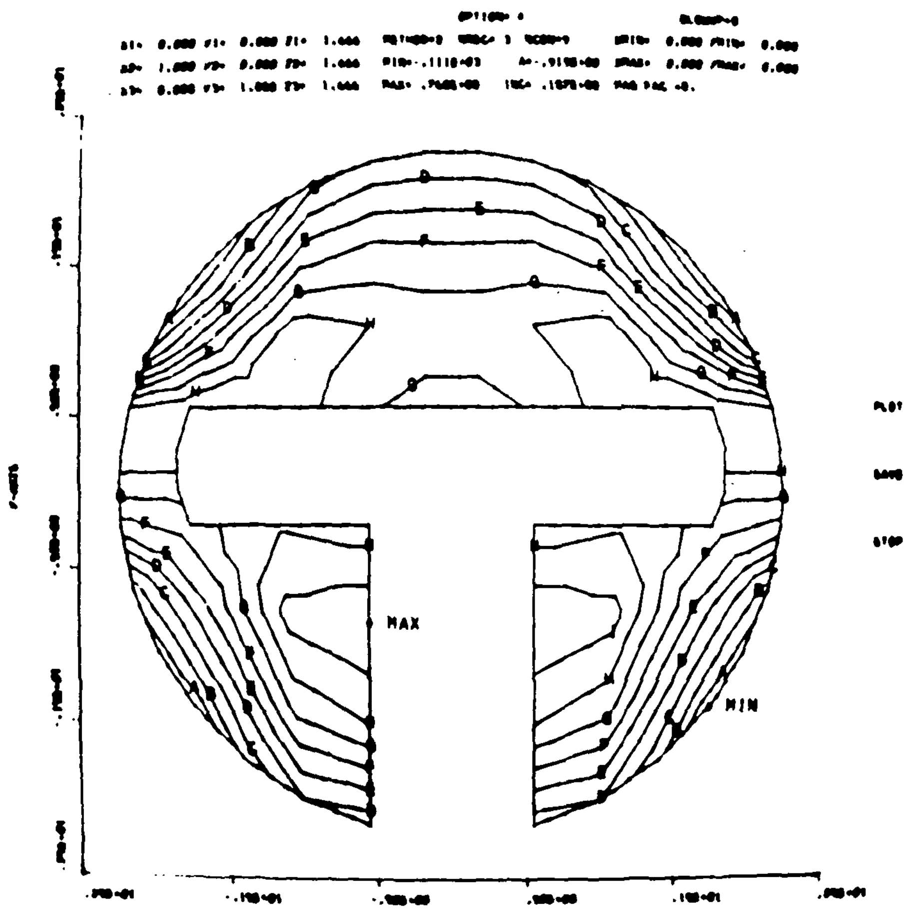

Figure 12. Contours of a stress component in a component encapsulation.

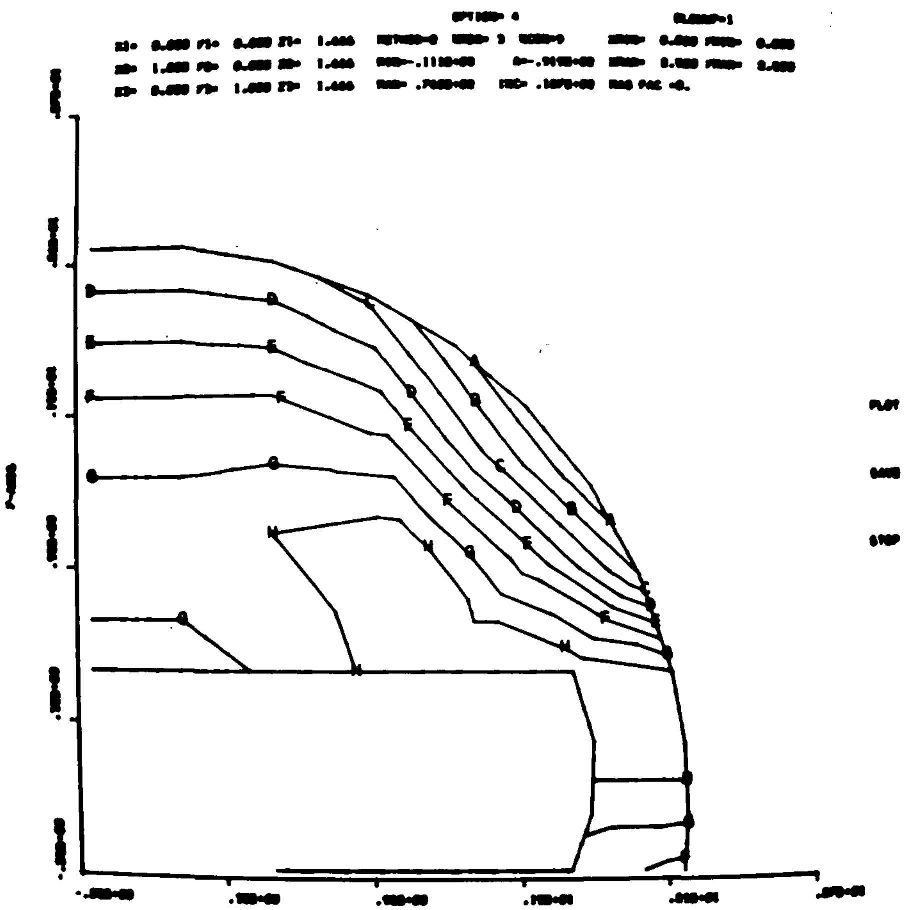

Figure 13. Blow-up of a region of a component encapsulation showing contours of a stress component.

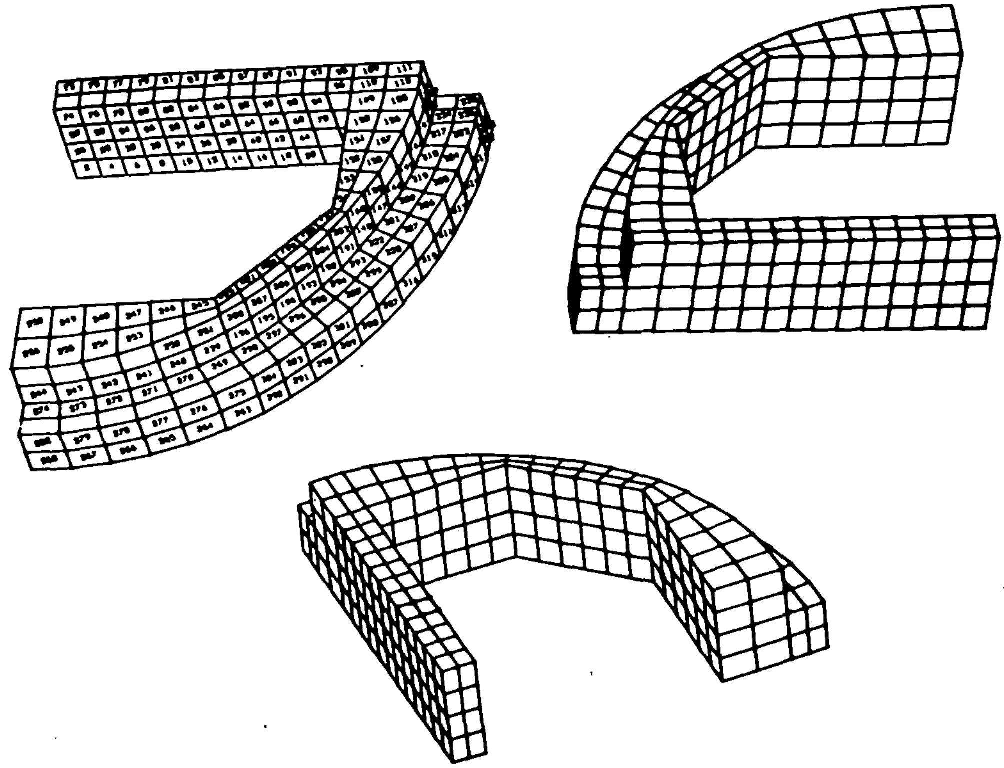

Figure 14. Three views of a mesh.

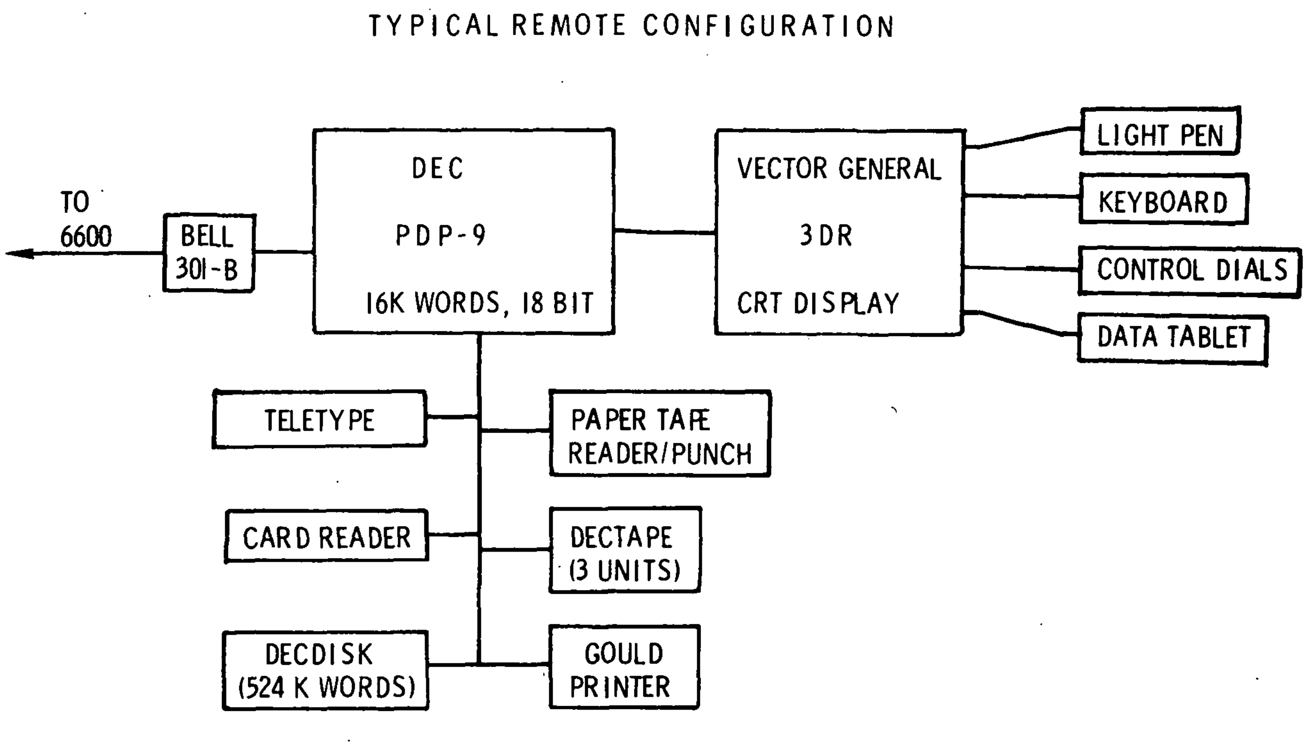

Figure 15. Schematic of interactive graphics hardware.

Figure 16. Interactive graphics hardware.