View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by NASA Technical Reports Server

N89-22942

USING PATRAN AND SUPERTAB AS PRE- AND POSTPROCESSORS TO COSMIC/NASTRAN

Robert R. Lipman

David Taylor Research Center
Numerical Structural Mechanics Branch (Code 1844)
Bethesda, Maryland 20084-5000

SUMMARY

Patran and Supertab are interactive computer graphics pre- and postprocessors that can be used to generate NASTRAN bulk data decks and to visualize results from a NASTRAN analysis. Both of the programs are in use at the Numerical Structural Mechanics Branch of the David Taylor Research Center (DTRC). This paper will discuss various aspects of Patran and Supertab including: geometry modeling, finite element mesh generation, bulk data deck creation, results translation and visualization, and the user interface. Some advantages and disadvantages of both programs will be pointed out.

INTRODUCTION

Interactive computer graphics is an integral part of finite element mesh generation and analysis results visualization. Gone are the days of typing GRID cards on a keypunch machine and pouring over endless pages of stress and displacement output. NASTRAN has plotting capabilities in a batch or interactive mode. However, in either mode, visual feedback while creating a finite element mesh is not possible and the results visualization capabilities are limited.

Presently, there are many finite element pre- and postprocessors that run on PC's, workstations, and mainframe computers. The pre- and postprocessors allow the user to interactively define geometry, approximate that geometry with a finite element mesh, apply loads and boundary conditions, create input for a finite element analysis program, and visualize results from the analysis. The programs provide a powerful, efficient, fast, and invaluable tool to an engineer to improve productivity.

Patran (ref. 1) and Supertab (refs. 2-5) are two of the more popular and widely used finite element pre- and postprocessors. Patran is a product developed and marketed by PDA Engineering of Costa Mesa, California. Supertab is a product developed and marketed by Structural Dynamics Research Corporation (SDRC) of Milford, Ohio. Both of these programs have interfaces to NASTRAN and can be used to generate finite element models and visualize analysis results. The scope of this paper covers the usage of Patran and Supertab as related only to COSMIC/NASTRAN (ref. 6) and not any other finite element analysis programs.

Several items about this paper should be noted. Both Patran and Supertab have a wide variety of features; I have not used, nor am I familiar with all of them. However, I have had extensive experience with Patran for the last 4 years and Supertab over the last year to generate finite element models of missile launchers, periscope masts and windows, propeller blades, and other Naval structures. In the Numerical Structural Mechanics Branch there are also several other experienced Patran and Supertab users. Any opinions expressed are my own and are not necessarily those of DTRC, the Navy, or the Department of Defense.

C-2

PATRAN OVERVIEW

The standard Patran software package consists of several integrated modules to generate geometry models consisting of curves, surfaces, and solids (P/Solid module); develop a finite element mesh consisting of nodes, elements, loads, boundary conditions, and material and physical properties (P/Fem); and visualize the geometry model, the finite element model (P/Image), and the analysis results (P/Post and P/Plot). Patran also has several optional modules that perform finite element analysis, mechanical dynamics, composite analysis, and thermal analysis. Interfaces to finite element analysis programs and IGES are also optional modules. Several utility programs are also provided, including a set of Fortran subroutines to access a Patran database directly without entering Patran.

Patran runs on many of the standard workstations, mainframe computers, and graphics devices. In the Numerical Structural Mechanics Branch, Patran version 2.2 is run on a network of Apollo workstations and Patran version 2.3 is run on a VAX with a Tektronix terminal. The Branch uses an interface to COSMIC/NASTRAN, but does not have any of the optional analysis modules.

Patran is a leased product. The lease fee is paid yearly and is determined by the number of concurrent users and the desired modules, interfaces, and graphics device drivers. The fee includes hotline support and software upgrades. Software upgrades are not released for all computers, modules, or interfaces at the same time. Presently, the Apollo version is one level behind the current VAX version.

SUPERTAB OVERVIEW

Supertab is one product of the I-DEAS (Integrated Design Engineering Analysis Software) family of software. The different families of software are: solid modeling (Geomod), engineering analysis (Supertab), system dynamics (Systan), drafting (Geodraw), and test data analysis (Tdas). Within each family there are different modules. The modules within Supertab are: pre/post processing, model solution, optimization, data loaders, and frame analysis. Most modules contain several tasks. Some of the tasks within the Supertab pre/postprocessing module are: geometry definition, mesh generation, model checking, and postprocessing. The data loader module of Supertab contains translators for all supported finite element analysis codes. I-DEAS software can be configured to contain only the required families and for some families, only the required modules. As part of the standard I-DEAS software package, a relational database management system (Pearl), an IGES translator, and several utility programs are provided, including software to integrate site-supplied software into I-DEAS as its own module.

Supertab also runs on many of the standard workstations, mainframe computers, and graphics devices. In the Numerical Structural Mechanics Branch, the I-DEAS product being used is called Supertab Plus version 4.0 running on a network of Apollo workstations. Supertab Plus consists of the pre/postprocessing and data loader modules of Supertab and the object modeling module of Geomod. In terms of geometry definition, the pre/postprocessing module has basic geometric modeling capablities, while the object modeling module has very powerful solid modeling capabilities.

Supertab is a licensed product. The user pays a one-time fee depending on the number of concurrent users and the desired software products. An optional yearly maintenance fee provides hotline support and software upgrades. As with Patran, software upgrades are not released for all computers, modules, or interfaces at the same time.

PATRAN GEOMETRY MODELING

Before a user creates a finite element mesh, the geometry model must first be generated. In Patran, the geometry model used to create a finite element model consists of points in space (Patran GRID entities), curves (LINE), surfaces (PATCH), and solids (HYPERPATCH). There are many ways to create these entities. For example, rotating a line about an axis will create a patch. A hyperpatch can be created from the linear interpolation of the region between two patches. The intersection between two patches creates a line. However, two hyperpatches cannot be intersected to create a third hyperpatch. Patches and hyperpatches could be reconstructed from the resulting lines of intersection between the individual patch faces of the original hyperpatches.

The mathematical formulation of the geometric entities is a parametric cubic. This representation has limitations; a line can go exactly through, at most, four grids; fitting a line through more than four grids will result in a least squares approximation. The approximation may be sufficient, or more than one line could be generated through the grids. The same problem occurs with generating a B-spline line. Given n grids, n-1 parametric cubic lines will be generated that represent a B-spline for those grids. Usually, the user would rather have one line through n grids. Having n-1 lines makes more entities to manipulate and keep track of.

Patches are always four-sided entities and hyperpatches are always six-sided entities. However, degenerate three-sided patches and degenerate five-sided hyperpatches are allowed. The sides of a patch are always single parametric cubic lines. A composite curve consisting of several lines defining the sides of a patch is not allowed. Therefore, the geometry model for a three-dimensional (3-D) object model will be divided into some combination of the geometric entities. The individual lines, patches, and hyperpatches defining the geometry model will be used to create the finite element model. Just as a finite element mesh normally is not discontinuous, the pattern of lines, patches, and hyperpatches should also not be discontinuous. Given the restrictions on the number of sides for patches (4 or 3) and hyperpatches (6 or 5), the desired finite element mesh, and the continuity of the geometric entities, an excessive number of geometric entities may be required to model some objects; and other objects will be almost impossible to model. This is more apparent when trying to divide an object into hyperpatches.

Patran has another solid modeling capability. This feature involves using boolean operations on solid primitives. The solid primitives available are bricks, cones, cylinders, elbows, spheres, and tori. A solid primitive can also be created from any collection of patches provided they form a closed surface. The boolean operations are difference, intersection, and union. The user can create a solid cube with a hole through the center by using a brick, a cylinder, and the difference operation. However, the primitives cannot be used directly to create a finite element mesh. First, the primitives have to be converted into geometric entities. For the cube with a hole, the desired geometric entity would be a set of hyperpatches. However, the resulting geometric entities are patches defining the original cube without a hole, patches defining the original cylinder, and the lines defining the intersection of the cube and cylinder. The hyperpatches can be constructed from those lines and patches. Using Patran primitives to generate geometric entities for finite element models is not very useful. Patran primitives are more useful in generating conceptual solid models of objects.

SUPERTAB GEOMETRY MODELING

Usually, before a finite element mesh is created in Supertab, two-dimensional (2-D) and 3-D regions, called mesh-areas and mesh-volumes, have to be defined. Mesh-areas and mesh-volumes are generated from curves and surfaces. There are two methods for creating curves and surfaces in Supertab. The first method is to generate a geometry model with the object modeling module of

Geomod and to transfer the curves and surfaces to Supertab. The second method is to use the geometry definition task in the pre/postprocessing module of Supertab.

The object modeling module of Geomod creates solid geometry models using 3-D primitives or 2-D cross-sections. The primitives availables are the same as in Patran. User-defined primitives can be generated by extruding or revolving cross-sections or by building an object from a set of cross-sections. The primitives can be cut, joined, or intersected using boolean operations. The curves and surfaces associated with the resulting primitives can be transferred to Supertab to be used in generating a finite element mesh.

The mathematical formulation of the curves and surfaces is a nonrational uniform B-spline (NURB). This representation allows one curve to be fit to any number of points. However, only planar outlines or cross-sections are allowed. Therefore, in Geomod and Supertab, a user-defined NURB must always lie in a plane. This is a limitation for some geometry models. For example, propeller blades are usually defined in terms of radial cross-sections. A radial cross-section is not allowed in Geomod or Supertab.

The geometry definition task of the pre/processing module of Supertab provides another alternative to create curves and surfaces that can be used to generate mesh-areas and mesh-volumes. This task is similar to object modeling in Geomod; however, only 2-D cross-sections can be created. Having only this geometry creation capability in Supertab is sufficient for many geometry models, making Geomod unnecessary.

PATRAN FINITE ELEMENT MODEL GENERATION

Nodes and Elements

In Patran, a finite element mesh is generated on the lines, patches, and hyperpatches that define the geometry model. For example, plate elements are generated on a patch. Two methods are available to create nodes and elements.

The first method is a two step process. In the first step, the GFEG command is used to create nodes on a line, patch, or hyperpatch. For a patch, the user specifies the number of nodes on each of two adjacent sides. This will create a mapped mesh of nodes from one side of the patch to the opposite side. The GFEG command allows for biasing the nodes and some limited mesh transitioning. The second step uses the CFEG command which specifies the type of element (for example: BAR, QUAD, HEX), the number of nodes per element, and a configuration code. The configuration codes are used to differentiate between different element types with the same number of nodes, such as CQUAD2 and CQUAD4. The number of elements generated depends on the pattern and number of nodes created with the GFEG command.

The second, and more powerful, method for generating a finite element mesh was implemented in the latest version of Patran. However, the method applies only to creating nodes and quadrilateral or triangular elements on patches. The MESH command is used to specify the type of element, the number of nodes per element, the configuration code, and the number of elements along all four sides of the patch or the approximate element edge length. Each side can have an arbitrary number of elements. This allows for whatever mesh transitioning or element size a user requires. A mesh smoothing command can be used to modify the resulting pattern of nodes and elements created by the MESH command.

The GFEG, CFEG, and MESH commands can create nodes and elements on more than one geometric entity at a time. For example, a geometry model consisting only of patches, might require that the GFEG command be repeated for each set of patches with the same pattern of nodes, and the

CFEG command used only once to generate the same type of elements on all the patches.

Material and Physical Properties, Loads, and Boundary Conditions

In Patran, material and physical properties, loads, and boundary conditions are applied to the finite element mesh with the PMAT, PFEG, and DFEG commands. Standard material models are available such as isotropic, orthotropic, and anisotropic. The physical properties specified are the same as would be required on a NASTRAN property card. Loads and boundary conditions can be applied to the nodes and elements associated with a geometric entity, a specific node or element, or all nodes or elements lying in a specified plane. Multi-point constraints (MPC's) can also be generated. Loads, boundary conditions, and physical properties can be defined by an algebraic function by using data entities or the FIELD command.

Equivalencing, Optimization, and Model Checking

When generating nodes on adjacent patches, nodes will be generated along the common boundary associated with each patch. Equivalencing eliminates one of the coincident nodes between adjacent geometric entities and readjusts the element connectivity. Optimization performs nodal resequencing based on based on bandwidth or wavefront. This capability is the same as the resequencing procedure in NASTRAN. The model checking capability checks the aspect ratio, warp, skew, taper, normals, and duplication of 2-D elements. If an element does not pass the check, the element can be split into two elements. No element checking is available for 3-D elements.

SUPERTAB FINITE ELEMENT MODEL GENERATION

Nodes and Elements

Supertab usually generates a finite element mesh on 2-D (mesh-areas) and 3-D (mesh-volumes) regions defined by curves and surfaces. The curves and surfaces come from the object modeling module of Geomod or the geometry definition task in the pre/postprocessing module of Supertab. A mesh-area is defined by a closed region of curves and a mesh-volume is defined by a closed volume of mesh-areas. A mesh-area can be defined by any number of curves and does not have to be planar. This is a very powerful tool to model any arbitrary 2-D region with only one mesh-area. The same is true for mesh-volumes. For a 3-D model only one mesh-volume, made up of multiple mesh-areas, is required.

Three methods for generating nodes and elements are available. The first method is mapped meshing, similar to the GFEG and CFEG commands in Patran. The number of elements along two adjacent "sides" of a mesh-area are specified. Because mesh-areas can have any arbitrary shape, mapped meshing is more appropriate for mesh-areas that are four "sided". Each "side" of a mesh-area, used for mapped-meshing, can be composed of any number of curves. The user specifies the number of elements along each curve of the two adjacent "sides" of the mesh-area. Biasing of the mesh is allowed. The element is specified by element type (rod, beam, plate, membrane, solid, etc.), element order (linear, parabolic, cubic), and element topology (triangle, quadrilateral, wedge, hexahedron, etc.). This specification does not distinguish between different NASTRAN elements that have the same element type, order, and topology, such as CQUAD2 and CQUAD4 elements.

Free mesh generation is the second method for generating nodes and elements. This capability can be used for mesh-areas and mesh-volumes and is similar to the Patran MESH command. To use free meshing, a global element size for a mesh-area or mesh-volume and the element type is specified.

Local element sizes and the number of elements per curve can also be specified. Free meshing is a very powerful capability, but care should be taken in creating mesh-areas and mesh-volumes and in specifying element sizes so that the resulting mesh is acceptable.

The third method for finite element mesh generation does not require any geometry model. Nodes can be created by entering or digitizing XYZ coordinates and copying, reflecting, or generating nodes between existing nodes. Rectangular, cylindrical, or spherical coordinate systems can be used. Elements can be created by picking the nodes for an element and copying, reflecting, extruding, or revolving existing elements. This method for finite element mesh generation can be used with nodes and elements created with either of the other two methods.

Material and Physical Properties, Loads, and Boundary Conditions

Material and physical properties are generated by creating tables of values for these properties. Only the properties that Supertab allows are permitted in the tables, which may not be sufficient to define all NASTRAN material and property cards. Both types of properties are associated with an element when the element is created. Loads and boundary conditions are applied to individual nodes and elements, nodes associated with elements, or nodes and elements on a geometric entity (curve, mesh-area, or mesh-volume). Load values can be defined by an algebraic function.

Equivalencing, Optimization, and Model Checking

Nodal equivalencing and resequencing in Supertab is similar to that in Patran. Nodes can also be resequenced by sweeping along a coordinate axis and sorting nodes based on nodal coordinates. Element checking is available for 2-D and 3-D elements. The adaptive meshing task in Supertab can be used to refine the finite element mesh based on element checking criteria.

BULK DATA DECK CREATION AND ANALYSIS RESULTS TRANSLATION

Patran

The ultimate goal of any finite element pre- and postprocessor is to create input for an analysis program and to tranlate results for the postprocessor. The programs used by Patran to accomplish this are PATCOS (PATran to COSmic/nastran translator) and COSPAT (COSmic/nastran to PATran translator). PATCOS and COSPAT (refs. 7-9) are developed and supported by PDA Engineering. Although PDA is currently updating the translators, the current versions of PATCOS and COSPAT have several bugs and have not been updated to include many bulk data cards that are new or which were missing from previous versions. Fortunately, when the Numerical Structural Mechanics Branch originally obtained COSPAT and PATCOS, PDA supplied the Fortran source code, which allowed us to bring COSPAT and PATCOS up-to-date by implementing many bug fixes, additions, and enhancements.

To create a bulk data deck, a Patran neutral file must first be created. The neutral file is an ASCII file containing all geometric and finite element model information and is generated by Patran. PATCOS reads the neutral file and generates a bulk data deck. If a new type of bulk data card is required, then PATCOS has to be modified. It is also possible to generate elements in Patran that are not supported by NASTRAN or PATCOS, such as a 15-noded wedge. The user has to be aware of the capabilities of PATCOS when generating a finite element model.

COSPAT serves two functions. The first is to read in a NASTRAN bulk data deck and generate a Patran neutral file. If COSPAT does not recognize a particular NASTRAN card type, then nothing is written to the neutral file. The resulting neutral file can be read into Patran. Once in Patran, the finite element model can be displayed and used for postprocessing.

The other function of COSPAT is to translate NASTRAN results into a format that can be read into Patran. COSPAT reads in displacement and stress data blocks that are written to a NASTRAN UT1 file with an OUTPUT2 statement. COSPAT generates up to three different Patran results files. One file contains nodal translations and rotations. The second file contains element centroidal stresses. The third file is generated only if NASTRAN computes nodal stresses (CIHEXi, CQUAD2, CTRIA2, etc.). Any of these three files can be used in Patran to visualize analysis results. With a user-written postprocessor, any type of data can be written in the Patran results files format so the data can be visualized with Patran.

Supertab

To generate a bulk data deck in Supertab, the finite element model must first be written to an I-DEAS Pearl database. The Pearl database is read by a program that generates the bulk data deck. Using the Pearl database to create the bulk data deck is a time-consuming procedure. A more efficient way to generate a bulk data deck might be to create it directly from the model file or from an I-DEAS universal file. A universal file is an ASCII file containing the geometry model, finite element model, analysis results, and viewing parameters. A universal file is similar to a Patran neutral file. Because Supertab cannot differentiate between a CQUAD1, CQUAD2, and CQUAD4 element, all linear quadrilateral thin shell elements will be translated to a CQUAD2 element. This problem also affects other element types. The user has to edit the bulk data deck to change elements to the desired element type. The source code for the program which generates the bulk data deck is not available.

To read in results from a NASTRAN analysis the data loader module of Supertab is used. Similar to PATRAN, the NASTRAN data loader reads data blocks that are written to a NASTRAN UT1 file with an OUTPUT2 statement. In addition to displacement and stress data blocks, data blocks for strains, forces, strain energy, and eight others which define the finite element model are required: CSTM, GPL, GPDT, EPT, MPT, GEOM2, GEOM3, and GEOM4. CSTM is generated only when a coordinate system definition card is included in the bulk data deck. If the default coordinate system is being used, then a dummy coordinate card must be included in the bulk data deck to force the generation of the data block CSTM. Although the user might not be interested in a particular type of output (for example, strain energy), the data block for that type of output is still required.

The NASTRAN data loader creates an I-DEAS universal file which can be read into Supertab. The analysis results can then be used for postprocessing. The data loader cannot read a bulk data deck to create a universal file of the finite element model. The source code for the data loaders is available from SDRC.

VISUALIZATION

Both Patran and Supertab have similar capabilities for visualizing the geometry model, finite element model, and analysis results. There are an infinite number of ways to display either type of model. The user has control over: viewing angles, which parts of the model are to be displayed, the color assigned to different entities (curves, mesh-areas, patches, element types, etc.), how to draw an entity (shrink elements, a circle or dot for a node, etc.), entity labels, display options, etc. The display option can be wireframe, hidden line, continuous tone (Supertab) or fill-hide (Patran), or shaded image. For graphics terminals and workstations with hardware 3-D rotations and shading, the user should be able to dynamically rotate a model drawn with any display option. Patran cannot display a shaded image of a finite element model. Supertab has an advanced display capability known as ray tracing. Ray traced images can have shadows, reflections, and transparency. However, ray tracing, as implemented in Supertab, is extremely computationally intensive (several days for one image on an Apollo DN580-T).

To visualize analysis results, several types of display options are available. They include: deformed geometry, animation of modal vibrations, contour plots, color fringe plots, vector plots, fill-hide plots, beam shear and moment diagrams, and XY graphs. The appropriate types of display options can be dynamically rotated. Various attributes of these types of displays can be set by the user. Patran can assign colors to elements based on analysis results or a value such as element or material ID. Currently, COSPAT does not generate Patran beam results files that can be used for beam shear and moment diagrams in Patran.

USER INTERFACE, DOCUMENTATION, AND BUGS

Patran

The Patran user interface is a mixture of a command-driven and menu-driven input system. The user interface can be used in a command line mode or on-screen menu mode. In the command line mode, the user enters whatever commands are desired. However, to do some tasks, a menu pick is required. To pick a particular menu item, the user enters the number associated with it. Some tasks can be executed by entering a command or using menu picks with the same results. Commands are also available to jump to particular menus. In the on-screen menu mode, menu items are chosen by using the cursor (controlled by a mouse, thumbwheels, etc.) to pick from a dynamic menu. Some commands still have to be entered in the on-screen menu mode. Other commands can be set by using the cursor. A less ambiguous and more structured user interface would be desirable for Patran.

Most commands in Patran can be divided into several catagories: commands for creating geometric entities (GR, LI, PA, HP), commands for creating the finite element model (GFEG, CFEG, PFEG, DFEG, etc.), and commands prefaced by SET, SHOW, or RUN. The SET and SHOW commands are used to set and show the value of almost 300 parameters. Generally, only a small subset of the parameters might have to be set. For example, labels can be turned on by entering SET,LABEL,ON. Currently, there are over 50 RUN procedures. The RUN procedures allow the user to do such things as generate hidden line plots, assign colors to elements, or compute contour line values.

The text that is entered for many of the commands, SET/SHOW parameters, and RUN procedures is not obvious. If the user did not know how to set the number of line segments plotted per parametric cubic line, entering SET,NLSPPC,10 would not be obvious. Therefore, documentation is essential. Patran documentation is divided into major tasks such as: creating geometric entities, creating a finite element model, visualizing models and results, and using SET, SHOW, and RUN commands. Generally, a description in words and graphics is given alphabetically for each command in a task. There are also functional listings of SET, SHOW, and RUN commands. For some types of commands there are conceptual descriptions of what can be done, along with related commands that might be used. The documentation is a complete reference of any capability or option in Patran. When running Patran, a user may access on-line help consisting of command descriptions.

No program of Patran's size is without bugs. PDA publishes a technical bulletin every month or two that lists a few known bugs and possible work-arounds. The release notes for a new version of Patran contain a list of bugs that have been fixed. The user does not have a list of all known bugs.

Supertab

Supertab's user interface is a tree-structured menu-driven system. The user picks a menu response which results in: (1) another menu, (2) a required alphanumeric response, or (3) a system action (for example, an element is generated). Menu items can be picked by using the cursor or by typing in the command. More than one command can be entered at a time, allowing the user to move through several menus at one time. Menu picks that require an alphanumeric response (file names, coordinate values, various parameters, etc.), generally have a default value which is used if the user hits the return key. In addition to the current menu, the user can pick from a global menu that has commands that can be executed anytime within a module.

Generally, every task in Supertab has its own main menu with many submenus. The menu system can be confusing. If the user knows what he wants to do, it is not always obvious what the command name might be or under what menu to find the command. As part of the standard documentation, menu guides are provided which list all the commands for each menu in a hierarchical form. The standard manuals do not have an explanation of all of the commands. Rather, the manuals introduce the use of Supertab conceptually and through step by step examples. This is a good method; however, the manuals are almost useless if a user is trying to determine what a specific command does. Optional reference manuals are available that give a description of each command. When running Supertab, a user may access on-line help consisting of command descriptions, a command search capability, and some overviews and methods.

As with Patran, Supertab is not without bugs. SDRC publishes a quarterly update of all known bugs and work-arounds and hints, limitations, and extended documentation for some features. Although the list of bugs is extensive, it is not complete because not all bugs are reported to SDRC.

OTHER FEATURES

Patran

When Patran is run, a session file is generated containing everything that was entered at the keyboard. This file can be used to reconstruct the model or to model objects with similar shape but different dimensions. For example, a session file that made hyperpatches defining a cylinder could be edited to change the radius and length. The session file could be rerun to create a new model with new dimensions. A session file can only be input at the beginning of Patran. If a file of commands is to be entered while already in Patran, the "read file" option under the geometry menu can be used.

Macros can be defined which create a user-defined text string that will represent several commands. A file (OPTION.SET) is executed everytime a new Patran database is opened. This file can be used to configure the user's working environment and to define macros. A replay file can be generated in Patran that contains all the graphics that appear on the terminal. This file can be replayed later with a utility program. A hardcopy file can generated and processed with another utility program that sends the graphics to a plotter (Calcomp, laser printer, etc.).

A named component is an entity that is a user-defined collection of geometric and/or finite element model entities. This provides a simple method to refer to a large number of entities of different types. New named components can be created by mirroring, rotating, scaling, or translating existing named components.

Supertab

When Supertab is run, a program file can be generated. Similar to a Patran session file, a program file contains everything that was entered at the keyboard. A program file can be input to Supertab at any time. In addition to Supertab commands, the program file can contain Fortran-like statements including arithmetic operators, mathematical functions, character strings, and variables, as well as commands to extract data from Supertab. Program files in this language can be written for any application and executed from Supertab. The program file can contain "read" and "write" statements to prompt the user for input, in the same way Supertab prompts for input.

Macros can be defined as they are defined in Patran. A user-defined program file (USERPROF.PRG) is executed everytime a new module is entered in I-DEAS. A picture file can be generated of a graphic image and replayed within Supertab or with a utility program and sent to a plotter.

Supertab has an adaptive meshing capability which, given the finite element analysis results, will refine the finite element mesh based on selected criteria. For example, a region of a mesh with high stress gradients could be refined to have a higher mesh density in that region.

CONCLUSIONS

Either Patran or Supertab can be used successfully as a pre- and postprocessor to COSMIC/NASTRAN. Each program has advantages and disadvantages. The user will have to decide which program is better depending on his finite element pre- and postprocessing needs. My personal opinion is that Supertab and Geomod are superior to Patran. However, if only simple finite element models are required, Patran might be a better program to use once the user has become familiar with Patran. Both programs need a convenient capability for fitting a surface through a specified set of points.

Patran

The solid primitive capability in Patran would be more useful if the primitives could be used directly for finite element mesh generation. The finite element model generation capabilities are simple and straightforward. The MESH command should be extended to handle generating solid elements on hyperpatches. The bulk data deck generation and results translation processes are fast and efficient provided the user has access to the source code for PATCOS and COSPAT. PDA Engineering should provide up-to-date versions of PATCOS and COSPAT so that the user does not have to become involved with the source code. The user interface leaves a lot to be desired. A new user will find it difficult to come up to speed to generate even a moderately complex finite element model.

Supertab

The solid modeling capabilities of Geomod are very powerful. However, the restriction that cross-sections be planar is a limitation. The finite element mesh generation capabilities of Supertab are also very powerful; however, not all element types can be generated. Creating an I-DEAS Pearl database slows the bulk data deck creation process. The user should not have to generate NASTRAN data blocks that define the finite element model to do postprocessing of analysis results. The user interface is very good; however, a user can get lost in the tree-structured menu-driven input system. The user interface allows a new user to come up to speed very quickly to generate complex geometry and finite element models.

REFERENCES

1. "PATRAN Plus User Manual," PDA Engineering, Costa Mesa, CA, 1987.
2. "Getting Started With I-DEAS," SDRC, Milford, OH, 1988.
3. "I-DEAS User's Guide," SDRC, Milford, OH, 1988.
4. "I-DEAS Geomod User's Guide," SDRC, Milford, OH, 1988.
5. "I-DEAS Supertab User's Guide," SDRC, Milford, OH, 1988.
6. "NASTRAN User's Manual," NASA SP-222(06), Washington, DC, 1983.
7. "PAT/COSMIC-NASTRAN Application Interface Guide," PDA Engineering, Costa Mesa, CA, 1984.
8. Bender, L.C., and M.P. Johnson, "Enhancing Your COSMIC NASTRAN Usage with PATRAN," *Sixteenth NASTRAN User's Colloquium,* NASA CP-2505, National Aeronautics and Space Administration, Washington, D.C., pp. 31-38, April 1988.
9. Libby, D.H., "COSMIC/NASTRAN - PATRAN Interface," *Thirteenth NASTRAN User's Colloquium,* NASA CP-2373, National Aeronautics and Space Administration, Washington, D.C., pp. 133-151, May 1985.