

PLASHTRAN: An Expert Consultant on Two-Dimensional Finite Element Modeling Techniques

Jonathan Cagan

Expert Systems Laboratory, Mechanical Engineering Department, University of California, Berkeley, CA 94720

Victor Genberg

Federal Systems Division, Eastman Kodak Company, Rochester, NY 14650

Abstract. An expert consultant and teaching aid has been developed to aid users of the MSC/NASTRAN (MacNeal-Schwendler Corp, Los Angeles, CA, USA) finite element code in the modeling process with two-dimensional elements. Written in LISP and LOOPS, an object-oriented programming language, the system, known as PLASHTRAN, allows engineers to work in a natural environment to obtain modeling recommendations. The program performs efficiently, especially when iterations in design require changes in the finite element model. The easily expandable modeling framework allows the knowledge base to incorporate new information.

1 Introduction

When inexperienced engineers attempt to perform an analysis on a complicated structure with finite element methods (FEMs), they are confronted with two major obstacles. The first is the FEM code itself. Major FEM products contain a wealth of capabilities, but this often comes at the expense of having to choose among numerous options on potentially confusing "bulk data" cards. The engineer is immediately presented with the difficulty of knowing which cards to employ to model his or her structure and what information to enter into the different fields of the cards. Fenves [1] states that "it may take a user a year or more to learn to effectively and efficiently use [FEM programs]" and Sriram et al. [2] mention that in a single engineer's career it is unlikely that a major FEM program will be totally exercised.

Once the inexperienced engineer achieves an understanding of the program's capabilities and applications, the new user finds him or herself in a

different dilemma. Answers from an FEM analysis are only as good as the model itself, and an FEM model is only as good as the engineer's understanding of the physics of the problem. Often a young engineer is told to analyze a structure, for example, a plate or a shell, without having a fundamental knowledge of the underlying mechanics of how the structure may behave. This invites failure in the modeling of the structure and the interpretation of the results.

These problems are not original to this discussion. There exist several tools to allow users to create FEM models quickly and easily. The most popular technique is that of solid modeling [3] to perform pre- and postprocessing of the data. Widely employed in industry, PDA/PATRAN [4] performs this pre- and postprocessing for model generation. However, the problems mentioned earlier are still an inherent part of this modeling process. The engineer must still know how to select an appropriate mesh and element type for the structure, although research in automatic mesh generation [5] may help relieve part of this difficulty. Further, the engineer must know which material, property, and load cards to choose.

The solution to these problems can be assisted by artificial intelligence (AI) technology. There have been numerous recent publications that discuss the technique of expert systems. The books by Harmon and King [6], Hayes-Roth et al. [7], and Rich [8] provide an understanding of these tools. Zumsteg and Flagg [9] give a brief overview of knowledge-based systems (KBS) in regard to computer-aided design (CAD). They coin an appropriate acronym to meld CAD with KBS called KAD (knowledge-based analysis and design). Since there have been many discussions of present AI techniques, this paper will assume a basic understanding of these concepts.

In 1979, Bennett and Engelmores [10] attempted to solve this problem by utilizing knowledge-based techniques in the well-known SACON (Structural Analysis CONSULTANT) system. SACON utilized the EMYCIN production rule-based shell of the MYCIN program [11] as a front end to the MARC [12] FEM package. Unfortunately this effort was halted before a robust completion. It showed, however, that the EMYCIN system could be employed to create a rule-based knowledge system to model some analysis expertise. Masse [13] took a different approach to the solution of FEM problems utilizing KBS techniques. He created a system that actually generates a finite element code for different situations, but, he did not address the modeling problems as discussed previously.

This paper discusses a prototype expert consultant that helps users of the MSC/NASTRAN [14] FEM package both to understand how to model the structure with NASTRAN and to understand better the mechanics of their structure. In particular, this prototype, called PLASHTRAN (PLates And SHells sTRuctural ANalysis), considers only structures modeled with two-dimensional elements. This choice reflects the fact that structures of these elements are the most difficult to model of the three-dimensional types.

The approach to the problem solution in PLASHTRAN differs from that of SACON's. EMYCIN, a goal-driven, backward-chaining shell, limits SACON to a rigid framework of search from the user input to the required situation that would implement it. PLASHTRAN is written in an object-oriented programming language called LOOPS [15] which is embedded within an Interlisp-D environment. LOOPS allows for a data-driven, forward-chaining program that incorporates the necessary flexibility found in a design environment. The system takes information about a situation and develops a problem solution. This approach would eventually encourage research into the creative process of design.

Further, PLASHTRAN follows the logic of the expert, instead of the expert following the logic of the shell. This additional advantage of the data-driven design allows the system to flow in a natural environment for engineers.

Although not yet as broad as SACON, PLASHTRAN is a new type of aid to NASTRAN users. It performs efficiently, in a user friendly, easily expandable format. Its design allows for modeling changes within the typical design-analysis iterative loop.

2 Problem Goals

Although it is an application of AI techniques, the system PLASHTRAN should identify areas for future research that are needed to develop mechanical engineering usage of AI. These issues include interfacing the analysis tool with the design process, a proper knowledge representation for problems in structures, a user interface for data acquisition and explanations, and a system architecture to execute the program efficiently and allow for the expansion of its size and application depth.

In the development of PLASHTRAN, the system should demonstrate a knowledge base of modeling techniques employed in variable situations. In addition, it should recreate the methods of modeling a structure so that assumptions can occur to allow the system to be fast and efficient during application. If a user must answer every question in the system, the process would become a tedious task. As PLASHTRAN is a prototype, initial investigation of these issues is initiated within the project.

An expert system allows an inexperienced person to perform a task with a level of knowledge of an expert. It is, however, still important that engineers understand the environment in which they work. PLASHTRAN is meant to be a source of information for an experienced user of NASTRAN, but, more importantly, the program appears as a teaching aid to new users of the FEM package. Therefore, an explanation facility to expose the reasoning behind the system's decisions becomes both necessary and important.

As previously discussed, NASTRAN contains more capabilities than any one engineer will ever need. It seems unlikely, then, that any system would initially include information on all modeling situations. As important and frequent patterns occur, they must be easily added to the system, requiring a modular and expandable knowledge base and logic control structure within the program.

An important goal of any engineering AI program should be that of flexibility. As a design changes, variations may have to appear in the finite element model. It would seem inappropriate to require the user of PLASHTRAN to repeat the entire session to make a small change in the structure's description. The system must allow the user to vary only the pertinent information. Fenves [1] describes this process as the "fifth dimension" or the dynamic versus static modeling process. Design, though, is never static in process and flexibility should really be the first goal of PLASHTRAN.

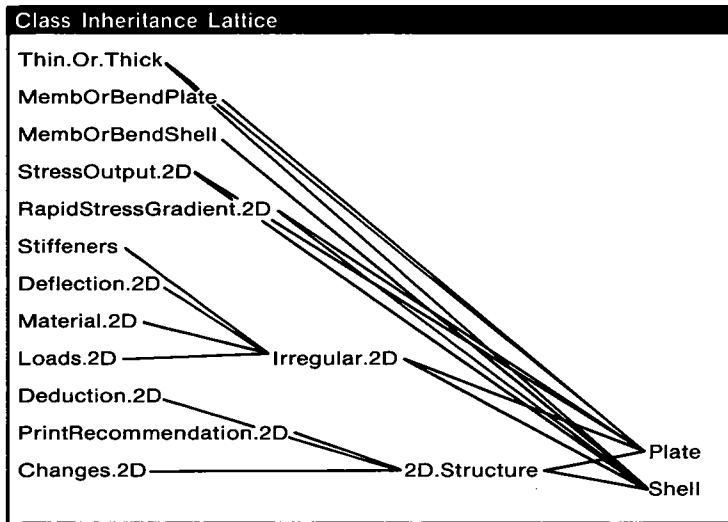


Fig. 1. Class inheritance lattice found in PLASHTRAN.

Finally, this project investigates the application of object-oriented programming in mechanical engineering structural concerns. Engineers tend to group information together. When one analyzes a shell problem, important features such as radius, thickness, material properties, and decay distance are important quantities that must be known in order to solve the problem. A frame-based system would permit an efficient way to store that knowledge, thus compiling congruous information. In a different application of design, Mittal et al. [16] have employed LOOPS, the same frame-based system as the one used in PLASHTRAN, in the PRIDE project to design office machine parts.

3 Object-Oriented Programming

Stefik and Bobrow [17] give a thorough discussion of the concepts of object-oriented programming with emphasis on the LOOPS language. The terminology utilized in the following sections is briefly described here. Information packets, called "classes," are frames that contain variables and "methods" (similar to functions). One "instance," or type of replication, of a class varies from another with different values for its "instance variables" versus its "class variables" that remain the same in all instances of the class. In LOOPS, variables can be converted into "active values." An active value allows a process to initiate whenever a value for the variable is asserted. This assertion is performed by employing "put functions."

A class inherits information from the "super" classes, or those found previously (diagrammatically) in the inheritance lattice. Some classes exist that as an entity in themselves are meaningless, but their information influences various subclasses. These classes are known as "mixins."

4 Problem Implementation

PLASHTRAN is implemented in LOOPS and Interlisp-D. The latter environment presents a robust user interface, allowing for a menu-driven, near fool-proof design. Forward chaining directs the user through the structure description while deducing information about the circumstances; backward chaining implements the explanation facilities.

The knowledge base and inference mechanism are coupled together, yet each can be accessed easily. The knowledge is stored in classes with instance variables. Mixins allow for clean program logic control by separating each area of the program into separate classes. The class methods are then called when that information is required.

Figure 1 shows the inheritance lattice that, when viewed from right to left, shows which classes inherit information from other classes. **2D. Structure** contains generic information about plates and shells, whereas **Plate** and **Shell** are classes that differentiate between the two.

The other classes shown in Fig. 1 are mixins. **Thin.Or.Thick** contains an algorithm to discriminate between thin and thick shells (or plates). If the

ratio of the overall shell radius (or smallest plate dimension) as compared to its thickness is greater than 10, then PLASHTRAN assumes the structure to be thin. If the same ratio is less than 10 but greater than 7, then the structure is modeled as a thick shell (plate). If the ratio is less than 7, then the structure should not be modeled with two-dimensional elements. Thus the scheme possesses a basic knowledge of the mechanics of structures. **MembrOrBendPlate** includes information on the type of stiffness found in the elements of the plate, with similar information for a shell found in **MembrOrBendPlate**. **StressOutput.2D** queries the user as to whether stress output will be requested. **RapidStressGradient.2D** contains information as to whether the stress gradients change quickly.

Stiffeners, Deflection.2D, and so on contain information on some of the irregularities in the structure. If the structural reinforcements are composite, sandwich, corrugated, smeared, or discrete stiffeners, the **Stiffeners** class asserts the appropriate information. For example, a corrugated shell requires the MAT2 card for the material card and the PSHELL card for the property card. In addition, the solution must include transverse shear effects, even if PLASHTRAN previously determined the shell to be thin and thus not require those consequences. If discrete loading, linear deflection and material properties, and no structural reinforcements exist, then **Irregular.2C** allows default values to be asserted without the need of entering each of these modules of the program.

Deduction.2D contains instance variables deduced from other variables but never specifically asserted by the user. For example, PLASHTRAN infers the order of the elements (higher or lower) and the solution sequence (linear or nonlinear) from the information queried from the user. **PrintRecommendation.2D** prints the recommended solution. At present, the backward-chaining mechanism for the explanation facility is incorporated within this class. In the future, this function should be separated; the print facility should only access variables and not make decisions as to what caused the values of those variables. **Changes.2D** encompasses the design-analysis iterative loop.

Active values (AVs) initiate the inference mechanism. Most instance variables in PLASHTRAN are AVs. When their values are asserted, "put" functions deduce the implications of the given data on the modeling technique. To initiate the algorithm about the thickness of the elements, the system prompts the user for the overall

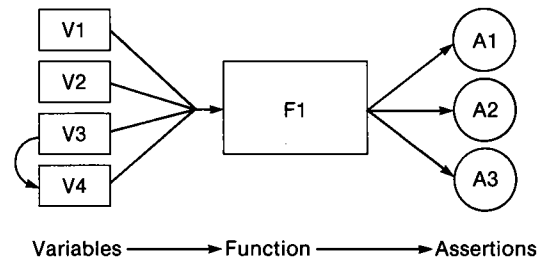


Fig. 2. A function may utilize different variables to deduce various assertions.

dimensions of the structure. As the value of the thickness is asserted with a put function, comparison between the thickness and the next smallest overall dimension takes place to determine if the shell is a thin, thick, or solid, three-dimensional structure. If the program asserts the instance variable **TypeOf2D** as solid, the user receives a message explaining that the structure should be modeled with solid elements and that PLASHTRAN does not consider that information. By asserting **TypeOf2D** as thick, the system generates the implications such as the need to include transverse shear effects in the NASTRAN model.

The inference mechanism, then, is dynamically initiated as the knowledge base modifies. This creates the opportunity to allow for the design-analysis iterative loop; as soon as the design changes, a new recommendation is available. In the preceding example, if the dimensions change, they are rechecked by the put function and implications are again deduced. In PLASHTRAN, any piece of data may be modified by the user.

The case of the element being thin or thick appears simple, only affected by the dimensions. Most other inferred values enter in a more highly influenced manner; three or more variables may affect the inferred variable. Figure 2 shows a map of this situation. Function F1 reads variables V1 through V4 and contains heuristics that can lead to assertions A1, A2, or A3, depending on the variable values. Variable V3 influences variable V4. Each time one of the variables modifies its binding, the function must check through the rule base to determine the correct assertion. For a problem of moderate breadth and depth, each as found here, this approach is still efficient and leads to immediate response for the modeling deduction. However, if many variables were highly coupled together, a different inference approach would need investigation.

An example of this more complicated influence

relationship can be found by determining what order (higher or lower) the elements should be. The stress gradient, the type of structure (plate or shell), and the solution sequence (linear or nonlinear) affect this assertion. If the solution sequence is nonlinear, then the elements must be lower; otherwise if either the structure is a shell or has a rapid stress gradient, then higher-order elements are preferred. If neither of these cases exists, then lower-order elements should be employed. In addition, the type of structural reinforcements and the linearity of the deflection and material influence the solution sequence. Thus choosing a type of structural reinforcement affects the solution sequence and further the order of the elements, along with other effects that are not mentioned.

5 A Session With PLASHTRAN

This section describes a typical session with PLASHTRAN. Figure 3 shows many of the windows and menus utilized to input data into the system. Initially PLASHTRAN requests the user's name (Fig. 3a) and what type of structure is being modeled (Fig. 3b), influencing the order of the elements. If the user selects "DontKnow," then an explanation of a plate and shell appears without further introductory information about the system (Fig. 3c). This type of explanation facility occurs throughout the program where the user may have questions on what the system assumes. In this example a shell is being modeled. (A shell would also be assumed if the user selected "StillDontKnow" in Fig. 3c, since that is the most general structure.) The system also requests a title for the structure [dome (selected responses appear in parentheses)].

Next, the program requests the overall dimensions of the shell (Fig. 3d). As discussed earlier, at this time the system deduces the thickness of the shell (thin). This assertion also affects the property card information. Next, PLASHTRAN requests the type of stiffness (Fig. 3e, membrane and bending) that further contributes to the property card.

The user decides if there exist any irregularities in the structure. If none exists, then the program asserts default values for each category (Fig. 3f). Otherwise, the user selects the areas that contain the irregularities (StructuralReinforcements and Loads). Each of these areas then requests further information (Fig. 3g, corrugated; Fig. 3h, distributed). The former assertion influences the property card, material card, and solution sequence; the lat-

ter affects the load card. Note that if PLASHTRAN requires more detailed input, submenus appear.

In this example, deflection and material are assumed linear, affecting the solution sequence and the order of the elements. The user then inputs the stress gradient (rapid), contributing to the order of the elements, and the stress output (requested).

A recommendation (Fig. 4a) appears with the best approach to model the dome with two-dimensional elements. Figure 4b shows a summary of the input entered by the user or inferred by the system. The window in Fig. 4b also shows a menu that can be utilized to modify the input data; that is, the design-analysis iterative loop.

Explanations appear in many situations so the user can follow the basic logic of how the system deduces the given solution to the problem. Warnings are given as appropriate to prevent the user from falling into a modeling trap and obtaining poor results with NASTRAN.

At this time, the user utilizes the menu in Fig. 4b to change the dimensions of the dome to a radius of 80 and a thickness of 10, the structural reinforcements to sandwich construction, and the deflection to nonlinear. After selecting "DONE," an updated recommendation appears (Fig. 5).

6 Future Work

At present, the areas of information covered by PLASHTRAN include property and material cards, structural reinforcements, various loading situations, nonlinearity, and the types of elements to employ best. Mesh sizing is not included. Boundary conditions will be the next area of implementation. In addition, certain products frequently analyzed by engineers at Kodak with NASTRAN are being implemented with default frames; much of the needed information for that particular structure will be included without a user interface.

The explanation facility should also be modified. Only explanations that are important to experienced NASTRAN users are generated in the output. For a new user, every recommendation should have access to an explanation. As a source of information for all users, though, explanations on everything would be cumbersome to those that have interest in only certain comments. In addition, questions arise as to how to explain the system's reasoning properly.

Two solutions exist to solve the explanation facility problem. Both assume that no explanation

PLASHTRAN (PLates And SHElls sTRuctural ANALysis)

Welcome to PLASHTRAN, the NASTRAN design aid for two-dimensional structures.
What is your first name?

: > Jon

Thank you Jon.
You will be asked a series of questions about your structure.
This system will deduce as much as it can from your responses, and thus ask as few questions as possible.
It will then give you guidance as to the proper approach to take to model your structure properly.

REMEMBER! When selecting items from the menus, use the LEFT mouse button.

Happy modelling, Jon!

EXPERT: Vic Genberg
KNOWLEDGE ENGINEER: Jonathan Cagan

PUSH LEFT mouse button to continue.

copyright 1986, Eastman Kodak Company

(a)

WHAT IS YOUR STRUCTURE

PLATE
SHELL
DontKnow

(b)

Plate or Shell Information

PLATES and SHELLS are bodies in which one dimension is much smaller than the other two.
In particular, a PLATE is considered FLAT, while a SHELL has CURVATURE.

In Nastran, one imbeds a 3-dimensional structure in 2-dimensional elements.
Each subelement must be considered separately.
In PLASHTRAN we consider only each zone.
For example, if analyzing a stiffened cylinder with a flat end-cap, the cylinder must be discussed separately from the end-cap here.

NOW, WHAT TYPE OF STRUCTURE ARE YOU MODELLING?

Plate
Shell
StillDontKnow

(c)

Enter overall dimensions of your dome

radius :> 80
thickness :> 7

(d)

Fig. 3. a-h Windows and menus utilized for input during a session with PLASHTRAN. See Section 5 for explanations of windows and menus.

WHAT TYPE OF STIFFNESS DO YOU HAVE?
 MembraneOnly
 BOTH BENDING AND MEMBRANE

(e)

WHAT TYPE OF IRREGULARITIES? ... (SELECT DONE when finished)
 StructuralReinforcements
 NonlinearDeflections
 NonlinearMaterials
 Loads
 DONE

(f)

WHAT TYPE OF STRUCTURAL REINFORCEMENTS ARE THERE?
 LayeredComposites
 Stiffeners
 SandwichConstruction
 Corrugated
 None

WHAT TYPE OF STIFFENERS?
 Smearable
 Discrete
 DontKnow
 None

(g)

WHAT TYPE OF LOADS ARE THERE?
 Discrete
 Distributed
 Thermal

WHAT TYPE OF THERMAL LOADS?
 TemperatureThruElements
 TemperatureAtGrids

HOW IS TEMP THRU ELEMENT KNOWN?
 KnowTemperatureDistribution
 KnowThermalGradient

(h)

Fig. 3 (con't). Legend on page 204.

originally appears in the output. One solution suggests that each recommendation resides within an individual window; when the mouse toggles inside a window, an explanation on how that answer was concluded would appear. A text editor implements the output window shown in Fig. 4a. The second suggestion allows for certain text to be accessible by the mouse; when the user selects the words with the mouse, the explanation appears.

7 Conclusions

PLASHTRAN performs in a user friendly, efficient format. This efficiency appears in the design-analysis iterative loop where changes in structure description lead immediately to changes in the problem solution. Also, in situations where the user takes short cuts, the system can effectively make proper assumptions. The framework of the system

a NASTRAN 2-D AID ... RECOMMENDATIONS FOR MODELLING

Jon, the dome that you have described is considered a thin Shell.

Your property card should be a PSHELL card with the first and second terms of the card filled in (both membrane and bending), and also fill in the third term since transverse shear is important in corrugated construction structural reinforcements.

You should use HIGHER order elements (like QUAD8 and TRIA6).

You should use a MAT2 material card.

Jon, you should employ solution 24 (linear solution.)

You should employ the PLOAD4 load card.

You have corrugated construction structural reinforcements. The problem is orthotropic so bending stiffnesses are different in two directions; (this is why you employ the MAT2 card for bending.)

Neutral axis offsets from grids are rigid bodies and may be used in this situation without causing erroneous answers.

To obtain the most accurate stress output, use the GPSTRESS request card in the case control. This allows for better post-processing, giving the peak outer-fiber stress. Make sure to employ a common coordinate system for the output request. In addition, you will need the OUTPUT-POST command.

REMEMBER:
Do NOT WARP elements.
Do NOT SKEW elements such that an interior angle is greater than 115 degrees.
When defining surfaces keep NORMAL constant (with the right-hand rule.)
Three-noded triangles have bad in-plane stress variation and are overly stiff in their plane. Use them with CAUTION.

PUSH LEFT mouse button to CONTINUE.

b CHANGE DESCRIPTION OF STRUCTURE

The following is a summary of the description of the dome that have been selected or inferred.

The dimensions are:
Major Radius - 80
Thickness - 7

The Shell has both membrane and bending stiffness.

The Shell has corrugated construction as its structural reinforcements.

The deflections are linear.

The material is linear.

The Shell has distributed loading.

The stress gradients are expected to be rapid.

Stress output will be requested.

Use the menu to make any changes.
Select DONE when finished.
If you wish to re-examine the structure's description, select REPRINT.

- SELECT CHANGES**
- Dimensions
 - Stiffness
 - StructuralReinforcements
 - Deflection
 - Material
 - Loads
 - StressOutput
 - RapidStressGradient
 - REPRINT
 - DONE

Fig. 4. a Typical output from a session with PLASHTRAN. b Window showing information entered or inferred by PLASHTRAN to cause the solution found in (a). Menu allows for iterative changes of information.

NASTRAN 2-D AID ... RECOMMENDATIONS FOR MODELLING

Jon, the dome that you have described is considered a thick Shell.

Your property card should be a PSHELL card with the first and second terms of the card filled in (both membrane and bending), and also fill in the third term since transverse shear is important in both thick shells and sandwich construction structural reinforcements.

You should use LOWER order elements (like QUAD4 and TRIA3), and NO higher order elements.

You should use a MAT1 material card.

Jon, you should employ solution 66 (non-linear solution.)

You should employ the PLOAD4 load card.

You have sandwich constructed structural reinforcements. Make the necessary calculations for the stiffness and transverse shear terms of the PSHELL card.

Neutral axis offsets from grids are rigid bodies. NASTRAN allows them to exist with non-linear displacements. However, erroneous answers may occur so use with CAUTION.

To obtain the most accurate stress output, use the GPSTRESS request card in the case control. This allows for better post-processing, giving the peak outer-fiber stress. Make sure to employ a common coordinate system for the output request. In addition, you will need the OUTPUT-POST command.

REMEMBER:
 Do NOT WARP elements.
 Do NOT SKEW elements such that an interior angle is greater than 115 degrees.
 When defining surfaces keep NORMAL constant (with the right-hand rule.)
 Three-noded triangles have bad in-plane stress variation and are overly stiff in their plane. Use them with CAUTION.

PUSH LEFT mouse button to CONTINUE.

Fig. 5. After changes are made to the data in Fig. 4b, an updated recommendation is available.

permits the knowledge base to be expanded easily to allow for the addition of deeper and broader information. In addition, the system demonstrates a knowledge representation scheme that effectively organizes information for finite element and structural analysis.

As a teaching consultant, the system recommendations give basic explanations to allow the user to follow for the modeling process. Warnings keep the user from making common errors. The output also reminds new users to consider important aspects in modeling the structure.

Object-oriented programming leads to a new approach in solving mechanical engineering analysis problems. The programming format meshes

quite well with engineering situations where knowledge can be grouped into packets of information. The framework for PLASHTRAN identifies areas of future research and development, such as explanation facilities and interpretation, and the implementation of artificial intelligence techniques to mechanical engineering design.

Acknowledgments

The authors wish to thank Mike Haefner, John Solberg, Del Stevens, Gary Randolph, Robert Bryant, Richard Benson, Renato Perucchio, and Alice Agogino for their support and suggestions throughout this project.

References

1. Fenves, S.J. (1985) A framework for a knowledge-based finite element analysis assistant. In: *Applications of Knowledge-Based Systems to Engineering Analysis*. (Ed. C.L. Dym), ASME, New York, pp. 1-7
2. Sriram, D.; Maher, M.L.; Fenves, S.J. (1985) Knowledge-based expert systems in structural design. *Comput. Struct.* 20(1-3), 1-9
3. Requicha, A.A.G. (1983) Solid modeling: Current status and research directions. *IEEE Comput. Graph. Appl.* 3(7), 25-37
4. PDA Engineering (1984) PATRAN User's Guide Vol. 1. Santa Ana, CA
5. Kela, A.; Perucchio, R.; Voelcker, H.B. (1986) Toward automatic finite element analysis. *Comput. Mech. Eng.* 5(1)
6. Harmon, P.; King, D. (1985) *Expert Systems-Artificial Intelligence in Business*, New York: John Wiley & Sons
7. Hayes-Roth, F.; Waterman, D.A.; Lenat, D.B., Eds. (1983) *Building Expert Systems*. Reading MA: Addison-Wesley
8. Rich, E. (1983) *Artificial Intelligence*. New York: McGraw-Hill
9. Zumsteg, J.R.; Flaggs, D.L. (1985) Knowledge-based analysis and design systems for aerospace structures. In: *Applications of Knowledge-Based Systems to Engineering Analysis and Design* (Ed. C.L. Dym), ASME, New York, pp. 67-80
10. Bennett, J.S.; Engelmores, R.S. (1979) SACON: A knowledge-based consultant for structural analysis. In: *Proceedings of the Sixth International Conference on Artificial Intelligence*, Tokyo, pp. 47-49
11. Shortliffe, E.H. (1976) *Computer-Based Medical Consultation: MYCIN*, New York: American Elsevier
12. MARC Analysis Research Corp (1976) *MARC User/Information Manual*. Palo Alto, CA
13. Masse, P. (1984) Modelling of continuous media methodology and computer-aided design of finite element programs. *IEEE Trans. Magnet.* MAG-20(5),
14. MacNeal-Schwendler Corp. (1985) *MSC/NASTRAN User's Manual*. Vol. 1. Los Angeles, CA
15. Xerox Corp. (1983) *The Loops Manual*. Palo Alto, CA
16. Mittal, S.; Dym, C.L.; Morjaria, M. (1985) PRIDE: An expert system for the design of paper handling systems. In: *Applications of Knowledge-Based Systems to Engineering Analysis and Design* (Ed. C.L. Dym), ASME, New York, pp. 99-115
17. Stefik, M.; Bobrow, D.G. Object-oriented programming: Themes and variations. *AI Mag.* 6(4), 40-62