NASA-CR-174,892

CR-174892

ats 15822

NASA

National Aeronautics and Space Administration

BURNER LINER THERMAL/STRUCTURAL LOAD MODELING

By
R. Maffeo

General Electric Company
Aircraft Engine Business Group
Cincinnati, Ohio 45215

Prepared For
National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

NASA-Lewis Research Center
Contract NAS3-23272

1. Report No. CR-174892	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Burner Liner Thermal/Structural Load Modeling		5. Report Date October 1984
		6. Performing Organization Code
7. Author(s) R. Maffeo		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address General Electric Company Aircraft Engine Business Group Cincinnati, OH 45215		11. Contract or Grant No. NAS3-23272
		13. Type of Report and Period Covered Final Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, OH 44135		14. Sponsoring Agency Code

15. Supplementary Notes
Project Manager, Dr. R.L. Thompson
Structures and Mechanical Technologies Division
NASA Lewis Research Center, Cleveland, OH 44135

16. Abstract

The software package developed under this contract was called TRansfer Analysis Code to Interface Thermal/Structural Problems (TRANCITS). The TRANCITS code satisfies all of the objectives for transferring thermal data between heat transfer and structural models of combustor liners; in addition, it can be used as a generic thermal translator between heat transfer and stress models of any component, regardless of the geometry. TRANCITS can accurately and efficiently convert the temperature distributions predicted by the heat transfer programs to those required by the stress codes. It can be used for both linear and nonlinear structural codes and can produce nodal temperatures, elemental centroid temperatures, or elemental Gauss point temperatures. The thermal output of both the MARC and SINDA heat transfer codes can be interfaced directly with TRANCITS and it will automatically produce stress model codes formatted for NASTRAN and MARC. In addition to these codes, any thermal program and structural program can be interfaced by using the neutral input and output forms supported by TRANCITS.

In summary, the TRANCITS code can be used to interface temperature data be tween thermal and structural analytical models. The use of this transfer module allows the heat transfer analyst to select the thermal mesh density and thermal analysis code best suited to solve the thermal problem and gives the same freedoms to the stress analyst, without the efficiency penalties associated with common meshes and the accuracy penalties associated with the manual transfer of thermal data.

KEY WORDS

Interface, Heat Transfer Analysis, Structural Analysis, Finite Element, Iso-parametric, Finite Difference, Transfer Module

17. Key Words (Suggested by Author(s)) Heat Transfer Stress Analysis Computer-Aided Engineering	18. Distribution Statement

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	203	

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-C-168 (Rev. 10-75)

N86-21932#

TABLE OF CONTENTS

Section		Page
1.0	SUMMARY	1
2.0	INTRODUCTION	2
3.0	DESCRIPTION OF TASKS	5
	3.1 Task I - Current Status of Thermal Load Modeling	5
	3.2 Task II - Assessment of Thermal Load Modeling	5
	3.3 Task III - Development of Thermal Load Transfer Module	6
	3.4 Task IV - Verification and Documentation of Automated Automated Thermal Transfer Load Module	6
4.0	TASK I	8
	4.1 Thermal Analysis Programs	8
	4.2 Structural Analysis Codes	9
5.0	TASK II	11
	5.1 2D Titan Evaluation	11
	5.2 Merlin Evaluation	12
	5.3 Lomap Evaluation	12
	5.4 Evaluation of Transferring Criteria	13
6.0	DEVELOPMENT AND DOCUMENTATION OF THE TRANSFER MODULE	17
	6.1 General Flow of Code	17
	6.2 Input and Output	19
	6.3 Finite Difference Consideration	21
	6.4 3D Search and Weighting Coefficient Routines	24
	6.5 Exterior Point Surfacing Routine	31
7.0	VERIFICATION AND DOCUMENTATION OF THE TRANSFER MODULE	34
	7.1 Verification of the Transfer Module	34
	7.2 Documentation of the Transfer	39
8.0	CONCLUSIONS	51
APPENDIX A		53
APPENDIX B		66
APPENDIX C		75
APPENDIX D		180
REFERENCES		203

1.0 SUMMARY

The objective of this program was to develop a thermal load transfer module that would be capable of transferring thermal data from a three dimensional analytical model of a combustor liner to a corresponding three dimensional stress analysis model. The initial goal of the transfer module was the ability to handle different mesh densities for the heat transfer analysis and the stress analysis and the capability to utilize both finite difference and finite element heat transfer codes. The extended goals of the transfer module were to develop a general purpose tool that would effectively allow for the coupling of heat transfer analysis and stress analysis while maintaining the flexibility of independent analysis codes.

The software package developed under this contract is called Transfer Analysis Code to Interface Thermal/Structural Problems or TRANCITS. The TRANCITS code satisfies all of the objectives for transferring thermal data between heat transfer and structural models of combustor liners and, in addition, can be used as a generic thermal translator between heat transfer and stress models of any components regardless of the geometry. TRANCITS can accurately and efficiently convert the temperature distributions predicted by the heat transfer programs to those required by the stress codes. It can be used for both linear and non-linear structural codes and can produce nodal temperatures, elemental centroid temperatures or elemental gauss point temperatures. The thermal output of both the MARC and SINDA heat transfer codes can be interfaced directly with TRANCITS and it will automatically produce stress model temperatures formatted for NASTRAN and MARC. In addition to these codes, any thermal program and structural program can be interfaced by utilizing the neutral input and output forms supported by TRANCITS.

In summary, the TRANCITS code can be used to interface temperature data between thermal and structural analytical models. The use of this transfer module allows the heat transfer analyst to select the thermal mesh density and thermal analysis code that is best suited to solve the thermal problem and gives the same freedoms to the stress analyst without the efficiency penalties associated with common meshes and the accuracy penalties associated with the manual transfer of thermal data.

2.0 INTRODUCTION

The overall objective of this program was to develop a thermal data transfer computer program module for the Burner Liner Thermal-Structural Load Modeling Program. This was accomplished by (1) reviewing existing methodologies for thermal data transfer and selecting three heat transfer codes for application in this program, (2) evaluating the selected codes to develop criteria for developing a computer program module to transfer thermal data from the heat transfer codes to selected stress analysis codes, (3) developing the automated thermal load transfer module, and (4) verifying and documenting the module.

In aircraft turbine engine hot section components, combustor liners, and turbine blades and vanes, cyclic thermal stresses are the most important damage mechanism. Consequently, accurate and reliable prediction of thermal loads is essential to improving durability. To achieve this goal, a considerable effort over the past 20 years has been devoted to the acquisition of engine temperature test data, as well as the development of accurate, reliable, and efficient computer codes for the prediction of steady-state and transient temperatures and for the calculation of elastic and inelastic cyclic stresses and strains in hot section components. There is a need for continued development of these codes, because the availability of more accurate analysis techniques for complex configurations has enabled engine designers to use more sophisticated designs to achieve higher cycle efficiency and reduce weight.

It has become apparent in recent years that there is a serious problem of interfacing the output temperatures and temperature gradients from the heat transfer codes with the input to the stress analysis codes. In part, this is a penalty for success. When computers were slower and computer memory smaller, the size of problems that could be analyzed, in terms of heat transfer nodes and stress analysis finite elements, was limited. Manual transfer of the output temperatures from the heat transfer codes to the stress analysis input, with manual interpolation to accommodate the mismatch in the nodal meshes of the two programs, was not unduly burdensome. With the growth in computer capacity and speed and the development of input preprocessors and output postprocessors, the analysis of components using hundreds and even thousands of nodes in the heat transfer and stress models has become economical and routine. This has exacerbated the interfacing problem to where the engineering effort required is comparable to that required for the remainder of the analysis. Furthermore, a considerable amount of approximation has been introduced in an effort to accelerate the process. This tends to introduce errors into the temperature data which negates the improved accuracy in the temperature distribution achieved through use of a finer mesh.

One solution to these interfacing problems is to force the heat transfer model and the stress model to be performed using the same analysis code and the exact same mesh densities. Although this technique may be acceptable for some applications, it has several disadvantages. In many models the areas of high stress gradient may be influenced by thermal loads but are predominantly controlled by geometrical discontinuities such as fillets and holes. A structural mesh that was fine enough to represent the gradients due to these discontinuities would be in general much denser than necessary for the thermal problem. This "overkill" might be tolerated for 2D analysis but many times the additional cost associated with these extra nodes in 3D analysis cannot be afforded. It is sometimes argued that because of the reduced degrees of freedom for a thermal problem as compared to a structural problem (typically 1 dof for thermal and 2 to 6 dofs for structural) that this additional cost is minimized. This is true for steady-state analyses, but for transient problems, the number of thermal runs far exceeds the number of structural runs usually performed.

Another example of the disadvantage of common meshes is the current trend in structural analysis toward adaptive mesh refinement. This technique will automatically refine the structural mesh in areas of high strain gradients. Unless the high strain gradient were caused by high thermal gradients there is certainly no need to refine the thermal mesh and redo the thermal analysis just because the structural mesh was made denser. All of these factors point toward the need for different thermal and structural mesh densities.

A further drawback of common heat transfer and stress codes and meshes is the widespread use of the finite difference technique to solve the thermal problem as opposed to the use of finite element codes for the structural problem. Again this suggests the need for an external process to transfer the thermal data to an independent structural analysis.

Therefore, the primary objective to this project is to develop, implement and demonstrate an interfacing module that will address the problems associated with the transfer of temperature data from thermal analyses to structural analyses.

This interfacing module must be able to handle different thermal and structural meshes and must support both finite difference and finite element heat transfer codes.

Conceptually, the overall program structure is shown in Figure 1. Direct inputs are temperature data, and the geometries associated with the thermal data and with the stress model. The particular heat transfer and stress analysis programs used affect the process inside the transfer module. Final output is the temperature data at the required locations in the stress model.

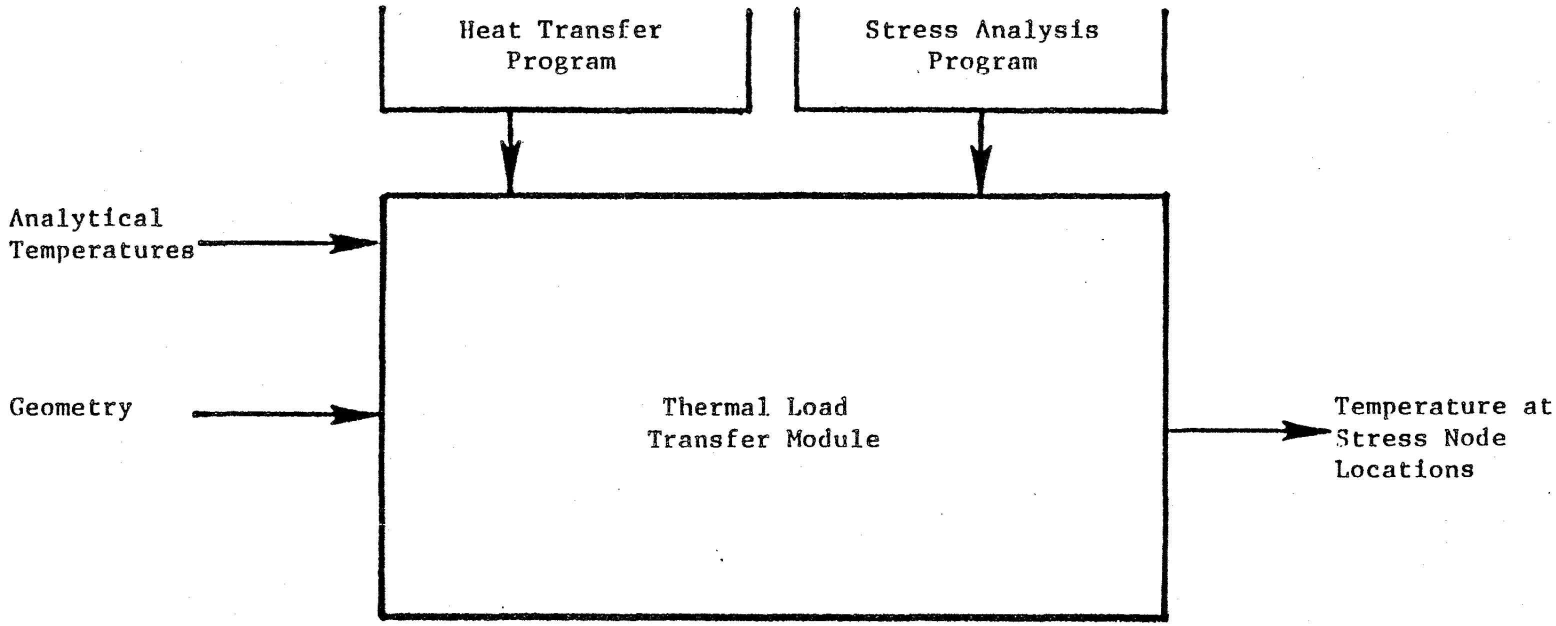


Figure 1. Overall Program Structure.

3.0 DESCRIPTION OF TASKS

This project was addressed by completing four tasks. These tasks include a review of the current status of thermal load modeling, an assessment of the thermal load modeling procedures, development of a thermal load transfer module and verification and documentation of the transfer module.

The four tasks are described below.

3.1 TASK I - CURRENT STATUS OF THERMAL LOAD MODELING

General Electric conducted a review of the current methods used to transfer thermal data from analytical sources to structural analysis programs in order to compute thermal stress and strain. The review included both in-house capabilities and papers in the open literature. The feasibility of the available methods were assessed and incorporated into the thermal load transfer module when applicable. In addition to reviewing the existing transfer techniques, General Electric also conducted a review of existing numerical temperature prediction codes including both in-house programs and those available in the open literature. This review was conducted with particular attention paid to the existing thermal load transfer methodologies and also included an examination of the temperature output format for each code reviewed. Based upon the results of this review, three computer codes were identified from among those available to the public to be used in the verification of the thermal load transfer module which was developed in Task III of this contract.

A review was also made of structural analysis computer programs as the first step in the process of selecting those codes to be used in this program. Both in-house computer programs and those programs defined in the open literature were reviewed. The objective of this review was to produce an evaluation of each code's input-output formulation, particularly with respect to temperature information. For the greatest public utility, primary emphasis was placed on those codes which are both in the open literature and actively supported by the commercial computer houses.

3.2 TASK II - ASSESSMENT OF THERMAL LOAD MODELING

In Task II, the Aircraft Engine Business Group outlined the thermal data transfer procedures for each of the methodologies approved in Task I and identified, discussed, and assessed their limitations and deficiencies in order to clearly define and evaluate the problem areas associated with thermal load modeling. Particular emphasis was placed on the questions of compatibility, ease of implementation, and thermal data format in the discussion of the current deficiencies and limitations involved in the transfer of thermal data from heat transfer analysis to the various structural analysis codes.

Once the limitations and deficiencies were identified in this task, a consistent set of evaluation criteria was developed. These criteria included factors that address the accuracy and the efficiency of the transfer procedure.

3.3 TASK III - DEVELOPMENT OF THERMAL LOAD TRANSFER MODULE

Based on the results of Tasks I and II, General Electric developed a thermal data transfer module that efficiently and accurately interfaced thermal data into structural analysis codes. The thermal data may come from both finite difference and finite element heat transfer analysis programs. The transfer module is capable of interfacing 3-D steady-state and transient heat transfer data into linear and nonlinear structural analysis programs.

State-of-the-art features in this module allow different mesh distribution for the heat transfer model, and the structural analysis model and slight differences in part geometry due to differences in component tolerances used in the models. Three-dimensional translation and rotation routines were incorporated into the module to account for possible mismatches between the origin of the structural model and the thermal model coordinate systems.

As these techniques were developed, they were encoded into a modular software package that efficiently manages the thermal data. The temperature data from the selected heat transfer programs is read directly by the interfacing module and automatically converted into a usable form for the selected structural analysis codes. The output of the interfacing module is completely compatible with the requirements for all of the structural codes selected. In addition, the flexible format of the output allows it to be easily implemented into other structural codes.

The techniques developed for the transfer module were designed to be flexible and have as much built-in generic capability as possible. The module provides options for compatible input with different existing heat transfer and structural computer codes. In conclusion, the transfer module provides an accurate streamline method of determining burner liner temperatures and gradients at specified locations compatible with the structural analysis input requirements.

3.4 TASK IV - VERIFICATION AND DOCUMENTATION OF AUTOMATED THERMAL TRANSFER LOAD MODULE

This Task is concerned with validating the data transfer module which was developed in Task III. Several test cases were run utilizing various combinations of heat transfer and structural analysis computer codes with temperature data flowing between the programs by means of the data transfer module. One of the test cases involved a 3D model of a combustor liner. Both finite difference and finite element heat transfer codes were used to produce the thermal results and these thermal loads were then interfaced to structural analysis codes using the transfer module. Cases were run using similar and dissimilar meshes for the heat transfer and stress analysis. Stress points that lie slightly outside the heat transfer model were interfaced using the module. The accuracy of the transfer was evaluated by comparing the temperatures produced by the transfer module to those predicted by the actual heat transfer codes used.

These demonstration test cases were documented as verification of the transfer code.

The coding of the transfer module was made as hardware independent as possible. All computer system dependent routines were isolated into easy to modify subroutines. The final version of the transfer module is executable on NASA Lewis's IBM system.

As a separate document a User's Manual for the data transfer module was provided. This manual includes a precise set of operating instructions for each of the software packages developed. Included is a description of all variables needed to execute the codes.

As part of this documentation, a complete set of Fortran listings for each subroutine developed for the data transfer module along with a computer tape of this coding was delivered to NASA Lewis Research Center.

4.0 TASK I

Current status of thermal load modelling and a review of heat transfer and structural codes.

4.1 THERMAL ANALYSIS PROGRAMS

A review was conducted of the in-house and open literature concerning numerical temperature prediction codes. The primary purpose of this task was to select three codes to be used to verify the thermal load transfer module. The other criterion for the codes is that they must be available in the NASA Lewis computer system.

A comprehensive summary of the capabilities of thirty-eight different codes is contained in the excellent survey article by A.R. Noor entitled "Survey of Computer Programs for Heat Transfer Analysis". (Reference 1) We studied this summary closely, paying particular attention to any already existing transfer modules between the thermal and structural analysis modules, as well as to the output form for the temperature results.

As to the first question, none of the programs surveyed contained an automatic thermal interface. This statement is also applicable to all our in-house heat transfer programs as well. Most major finite element programs do, however, have the capability to pass thermal data directly to an identical mesh which can then be used for structural analysis. Such a capability may be extremely useful for certain applications, but for many applications it is very restrictive. It imposes mutual restrictions on the heat transfer analyst and on the structural analyst which neither of them really need to observe. It is also true that this one-to-one correspondence requires that the heat transfer program use a finite element technique. This fact is clearly not a generally desirable arrangement.

The second area we focused on was how we could access the temperatures produced by the heat transfer code. Most of the major codes surveyed also carry a file of output temperatures, which are suitable for post-processing. The form of this file is, of course, program dependent, but we believe that it can become the basic input to the thermal loads interface module. A thermal output file is not absolutely necessary since all the thermal codes print out the temperature at the heat transfer elements in some readable form. This readable form, known as "hard-copy", can be scanned and the required thermal data can be obtained.

Since all the programs considered possess both the special output file as well as the standard "hard copy" output, either form of the output could be utilized. One of the unknown questions is: Does the special output file contain all the information necessary for the thermal load interface? Frequently finite difference based programs, for example, do not carry a complete geometry definition as a part of the output. The complete geometry is essential to the interface module. It can, in most cases, be obtained from the input to the heat transfer program. Questions such as this one cannot be answered with confidence until the individual programs are actually tried.

In general, all programs were found to be roughly equivalent as far as producing temperature data usable by a structural program. Many different features are available in these codes, for example, different treatments of radiation, better plotting capabilities, etc., but the main focus of this study is on transferring temperature data. On this basis, there are nine major programs which could be used:

ANSYS
HEATING
MARC
NASTRAN
SINDA
THTD
ADINAT
SAHARA
PAFEC

Of these codes, only four are currently available on NASA Lewis computers and these are HEATING, MARC, NASTRAN, and SINDA. Further, the transfer module is demonstrated with the interface of both finite difference and finite element codes, namely the SINDA program for the finite difference program and MARC as the finite element program. Also demonstrated is the interface to NASTRAN, because of its wide popularity. For completeness, the portions of Noor's survey article which deal with these programs and others are listed in Appendix A.

Other codes considered, but not available on NASA Lewis's computer, were ANSYS, ADINAT, SAHARA and PAFEC. ANSYS and ADINAT are both widely used finite element codes. Both of these codes have a wide range of elements that can be used for heat transfer analysis, and ANSYS is frequently used for gas turbine design problems. It would be desirable to have the transfer module deal with these heat transfer codes directly. The SAHARA program developed at Sandia laboratories is another finite difference program that could be used to validate the transfer module. The survey indicates that SAHARA, like SINDA, has an output file that can be used for postprocessing. This output file would be very useful for the transfer module. Finally the PAFEC program was considered because it uses boundary integral techniques to perform the heat transfer analysis. Here again, the ability to transfer thermal data from such a program is beyond the requirements of this contract but nevertheless will be very desirable. It is felt, however, that because of the modular construction of the transfer module, future implementation of these codes or any other common code could be easily accomplished.

4.2 STRUCTURAL ANALYSIS CODES

A survey of stress analysis programs was also conducted. By virtue of the huge number of programs available (it is certainly in the hundreds), our survey cannot be considered comprehensive, but our conclusions are consistent with the scope of this study. In particular, no structures program which was considered carries its own thermal interface transfer module. Temperature data are transferred by one of two methods.

(a) Temperatures are obtained from an exactly equivalent heat transfer analysis, utilizing the identical mesh,

or

(b) temperatures are interpolated and/or extrapolated from a different mesh to discrete points: nodes, gauss points, element centroids, etc.

Method (a) requires no interface module whatsoever, but it is not considered to be desirable for a general case.

Method (b) is ideally suited for a thermal interface module in that it is quite general. The module which was developed in Task III produces the temperature at any point in space which is consistent with a specific stress analysis. Whether this point is an actual node or gauss point makes little difference. The actual details of which points are required for temperature input to a structural model are not critical. The module will accept any selection.

Our survey of commercially available programs relied heavily on a recent review with R. Zirin of the General Electric Gas Turbine Division covering ANSYS, EASE2, MARC, NASTRAN, STARDYNE, and SUPERB.

All of these finite element structural codes have a large library of different elements. The location within these elements to which the thermal data must be input varies. Some require nodal temperatures, some need elemental centroid temperatures, others need gauss point data. They all, however, need temperatures at some discrete point in the structural model. Once again, the transfer module will produce temperatures at any location.

For the purposes of this contract, the transfer module is demonstrated on MARC and NASTRAN, the codes available at NASA Lewis. Again, for completeness, the information on MARC and NASTRAN is given in Appendix B.

The thermal load interface module developed under this study is flexible enough to drive any structural program which needs temperatures for input.

5.0 TASK II

Assessment of Thermal Load Modeling

The objective of Task II was to evaluate the thermal transferring capabilities currently available in existing heat transfer codes. In addition, an attempt was made to identify any freestanding thermal data transfer modules that might be in use by industry. Once these capabilities were evaluated, a set of criteria was compiled that included these factors and several additional criteria that were important in the development of a successful transfer module.

As stated in Section 4.1 no general capability for transferring 3D thermal data to structural codes was identified in the thermal codes surveyed. The most common mechanism for accomplishing this interfacing was through the use of similar mesh densities. One exception to this was the THTD program that did produce a thermal output file that could input temperatures directly into selected stress codes. The actual procedure of assigning the stress node temperatures, however, involved a manual correlation between the stress point and the heat transfer element chosen to be "closest" to the stress node. Not only was this time consuming, but it also did not provide for any interpolation between heat transfer elements. The proposed code would consider this shortcoming in its development.

The investigation of external thermal load transfer modules identified three codes being used by various companies. These codes were 2D TITAN developed by General Electric (AEBG), MERLIN developed by Sandia Lab and LOMAP developed by Lockheed. The evaluation of each of these codes follows:

5.1 2D TITAN EVALUATION

The 2D TITAN transfer code was developed by GE to interface the thermal results of a heat transfer code called THTD to selected structural analysis codes. The THTD code is a finite difference formulation. The transfer code will interface temperatures into any 2D mesh including plane stress, plane strain, axisymmetric and shells of revolution. It has some heat transfer windowing capability both spatial and temporal. TITAN can interface stress points that lie slightly outside of the heat transfer model. It also has built in 2D coordinate transformations to align the thermal and structural models. The program has some internal accuracy checks used to flag areas of the thermal model where the interfacing could yield inaccurate results. Another related feature is the use of interpolating functions that are consistent with the assumption associated with the THTD heat transfer element. The actual interpolation is accomplished using the 2D planar isoparametric shape functions.

The following limitations were identified:

a. No 3D capability.

b. Only interfaces with one heat transfer code.

c. No time interpolations between thermal solutions.

d. No provision for scaling the thermal results based on variations in thermal boundary conditions.

5.2 MERLIN EVALUATION

The Sandia code (MERLIN) is very similar to the in-house GE program called TITAN. Investigation into MERLIN revealed the following limitations:

a. Restricted to 2D problems.

b. Interfaces only with finite element heat transfer codes.

c. Does not interface with finite difference heat transfer codes.

d. Only interfaces into a specific set of finite element stress codes.

e. Meshes (both heat transfer and stress) must be generated using a specific mesh generator.

f. Does not allow for coordinate transformation of meshes.

g. Does not account for stress points "slightly" outside heat transfer mesh.

h. Limits the number of elements in the heat transfer model.

i. Limits the number of nodes in stress model.

j. Uses the same temperature mapping function regardless of the type of heat transfer element.

The interfacing technique used in MERLIN is a three-step process. Step one is the mesh search. This search locates which heat transfer element contains each stress point. The second step is called the element search. This process defines the local coordinates of the stress point with respect to the coordinates of the heat transfer element. The last step computes the temperature of each stress point based on the weighting coefficients derived from the local coordinates. This entire process is almost identical to the methods used in TITAN.

5.3 LOMAP EVALUATION

The Lockheed program LOMAP has very limited capabilities as a general transfer module. It will transfer temperatures from a 3D heat transfer mesh to a 3D stress mesh, however, the method used is very simplistic. The weighting coefficients used to interpolate temperatures are Proportional to the temperatures of the four closest heat transfer nodes. For a general transfer program the four closest nodes may not be the correct nodes to use. In fact, the four geometrically closest heat transfer nodes may not even be in the same thermal region as the stress point. This technique places severe restriction on the types of meshes that can be interfaced properly.

5.4 EVALUATION OF TRANSFERRING CRITERIA

Based on the study of existing thermal transfer modules, GE internal needs, and GE previous experience with TITAN, three levels of criteria for the program development were proposed.

Level I contains the general criteria which must be satisfied for a usable product.

Level II contains specific criteria which must be satisfied to meet the requirements associated with gas turbine design problems. This list stems mainly from GE internal experience.

Level III contains criteria which are desirable but not necessary. In most cases, items in this class can be achieved through a multi-step Process. Total automation might be desirable, but we do not believe this effort is warranted at this time.

Level I: General Criteria For A Thermal Transfer Module

IA. Independent Heat Transfer and Stress Geometry Meshes

This criteria lies at the heart of our effort. Useful thermal transfer modules must address this feature. Automatically included in this is the ability to transfer from finite difference heat transfer mesh to a finite element stress analysis mesh.

IB. Accurate Transfer of Data

Simplistic approaches such as averaging the closest nodes do not always yield accurate results, and the utility of the transfer program is then questionable. This criteria was met by using all available temperature information to do the interpolation and by using different mappings to correspond with different heat transfer elements.

IC. User Friendly

Programs which do not meet this criteria tend to be used incorrectly or as a last resort. The plan was to construct the thermal transfer module to encourage the analyst to use it. Any errors encountered by the module are flagged and reported with a diagnostic message.

ID. Computationally Efficient

The program was coded to achieve an efficient flow of data. GE's past experience with TITAN has led to several improvements over the original efforts, and these have resulted in similar gains in 3D transfer problems. This criteria covers both searches to find the proper heat transfer element for a stress node as well as the single element inverse mapping functions.

IE. Flexible

The thermal transfer module was constructed such that future modifications, or even different applications, for instance, pressure or boundary condition transfer, could be accomplished without a full rewrite. This criteria stems from past experience in having to improve or draw upon old techniques which could almost, but not quite, perform the required task. The transfer module not only transfers temperatures in a state-of-the-art manner, but it will also provide a vehicle for other 3D interpolation based problems.

Level II - Specific Criteria Required For Gas Turbine Design Problem

IIA. Coordinate Transforms

Coordinate transformation that allow the heat transfer model to be aligned with the structural model.

IIB. "Out-of-Box" Provision

Provision to account for stress nodes that lie just outside the heat transfer model due to slight differences in the dimensions used in the heat transfer and stress analysis models, as a result of using different tolerances on the actual component dimensions.

IIC. Windowing

Capability to "window in" on a smaller portion of the heat transfer model.

IID. Selected Time Steps

Ability to select temperature distributions at specific time steps form a large transient thermal analysis.

Level III - Desirable But Not Essential Features

IIIA. Automatic Handling of Temperature Discontinuities

In the module these will be treated in a two-step manner. Total automation is possible, but has not been implemented.

IIIB. Scaling of Temperatures Based on Variation In Engine Power Level Settings

Scaling of the interpolated temperatures based on changes in the heat transfer boundary condition, such as engine power settings, various inlet conditions or different cooling flows requires specific knowledge of the heat transfer analysis and will not be done inside the transfer module, but could, if desired, be applied by another program to the original results from the transfer module.

IIIC. "Altered" Stress Geometry

Many times the stress analyst wants to alter the part geometry to reduce his stresses, but the deviations will not, in the judgement of the heat transfer analyst, affect the temperatures. In the past, "ad hoc" procedures to transfer temperatures to the new stress geometry have been used. This approach is not optimum and has not been included in the transfer module.

Table I compares each of the external transfer codes evaluated against some of the criteria chosen. This table illustrates areas where significant effort was placed to develop an accurate and efficient thermal transfer module.

Table 1.

Criteria	MERLIN	Codes LOMAP	GE TITAN	TRANCITS Proposed Transfer Module
Independent Meshes	Yes	Yes	Yes	Yes
2-D Capability	Yes	Yes	Yes	Yes
3-D Capability Thermal Mapping a Function of Heat Transfer Elements	No	Yes	No	Yes
Coordinate Transformations	No	No	Yes	Yes
'Out-of-Box'	No	No	Yes	Yes
Provision Isoparametric	Yes	No	Yes	Yes
Mapping Geometrical/Temporal	No	No	Yes	Yes
Windowing Capability For Time Interpolation	No	Yes	No	No

6.0 DEVELOPMENT AND DOCUMENTATION OF THE TRANSFER MODULE

The objectives in this task were to develop an efficient and accurate software package that implemented all of the features discussed in Task I and Task II. The development was divided into several subtasks, each of which addressed important program features that were critical to a successful code.

The subtasks were:

1. General Flow of Code
2. User Input
3. General Heat Transfer Input
4. Specific Heat Transfer Code Input
5. General Stress Code Input
6. General Output
7. Specific Stress Code Output
8. Finite Difference Heat Transfer Code Consideration
9. Finite Element Heat Transfer Code Consideration
10. General 3D Search Routines
11. Weighting Coefficient Routines
12. Exterior Surfacing Routines

6.1 GENERAL FLOW OF CODE

Several factors were considered in developing the general flow of the code. These factors included:

a. Program modularity

b. Program efficiency

c. Input/output requirements

d. Internal data flow

e. Flexibility/portability

f. User friendliness

The basic program flow is illustrated by Figure 6-1.

Each of the boxes in Figure 6-1 represent program modules that have been coded to minimize the system dependent features of the transfer program. In addition to this, these modules communicate with each other through an internal file structure that is hardware independent, easy to follow and easy to modify.

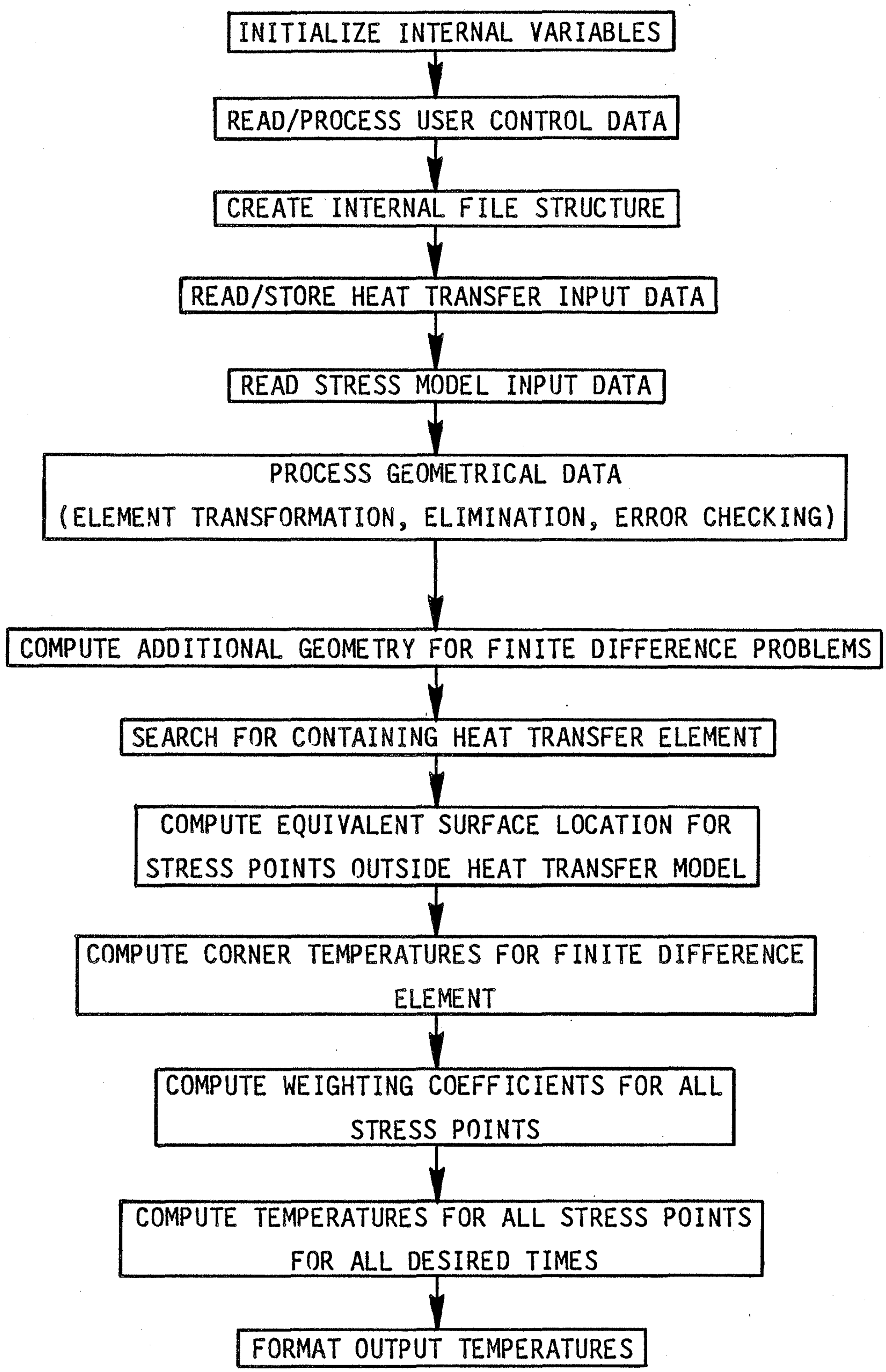


Figure 6-1. Overall Modular Flow of Code

These features not only make the code very portable but also allow the same basic structure to be used in the future to transfer other types of loading. Additional heat transfer elements can easily be added to this scheme.

6.2 INPUT AND OUTPUT

The input and output information can be grouped into three basic categories. They are geometrical, thermal and user control data. Figure 6-2 illustrates how these data interact with the transfer module.

The user control data are input to the transfer module with a namelist input file. This file contains all of the variables that can be input by the user to control the interfacing procedure and is described in detail in the TRANCITS Users manual (Reference 2). Options such as geometry and temporal windowing, coordinate transformations, time step tolerancing and information about the type of input and type of output desired are entered through this file and help provide for the user-friendliness of the code.

The other two categories of input are associated with the geometrical data for both the heat transfer and stress models and the temperature data that is the output from a thermal code. The transfer module has been structured to accept output of the SINDA finite difference code and the thermal output of the MARC finite element program. In order to satisfy these requirements and at the same time produce a module that can be used with a variety of heat transfer codes, a neutral heat transfer input form was developed. This input form will support both finite difference and finite element thermal results and is the primary input to TRANCITS. Compatibility with SINDA and MARC is accomplished by internally converting the output of these codes to this neutral form.

The heat transfer input file consists of six partitions of data. These data are described in detail in the users manual. It stores all the sizing, identification, qeometrical and thermal data associated with the heat transfer analysis. It is important to note that the geometrical data is an essential part of this input. The geometry (nodal coordinates and element connectivity) is the only link between the heat transfer and stress models. Without geometrical data an accurate interface cannot be accomplished. The thermal data in this file varies depending on the type of heat transfer analysis. Nodal temperatures are stored if the thermal code is a finite element type and elemental temperatures are stored if the thermal code is a finite difference formulation.

The automatic interface with SINDA and MARC makes use of output files created by these programs. The MARC program produces an output file known as the MARC cost file. This output contains all of the data required by the transfer module to perform the interface. The SINDA code also has the capability to produce an output file. However, since SINDA is a finite difference program, there is no geometry stored on the output file. Indeed, the SINDA code does not require geometry as input. Only volumes, areas and distances are needed to predict the thermal distributions. Therefore, the raw SINDA output

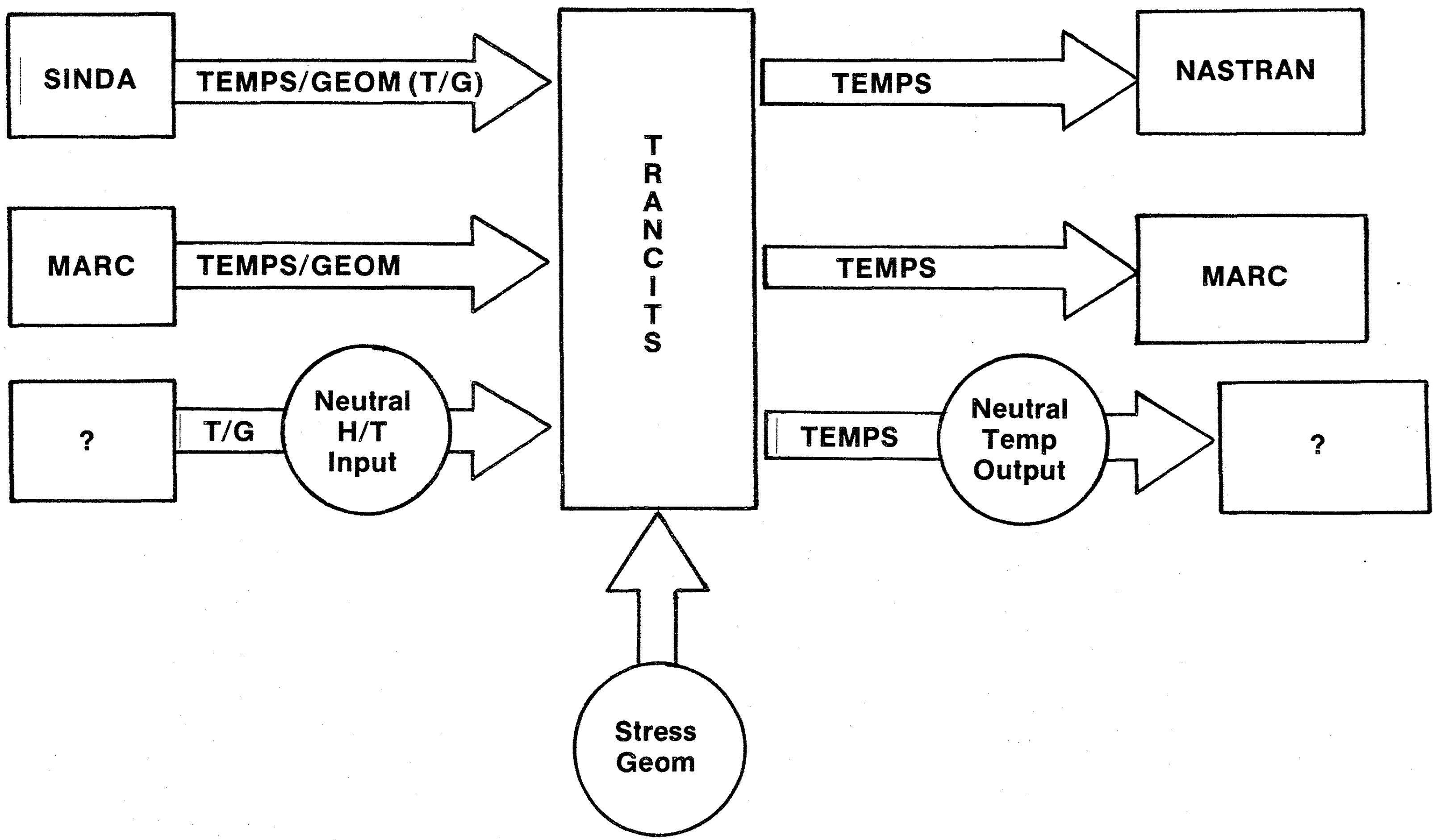


Figure 6-2. TRANCITS Schematic.

is not adequate for the transfer procedure. This file must be augmented by the SINDA geometry file. At first glance this seems to be a severe limitation of using TRANCITS in conjunction with any finite difference code. This is not the case because in practice large finite difference models are not created manually. Pre-processors automatically generate the volume and area data needed for the finite difference analysis. The input to these preprocessors is grid point coordinates and element connectivity. This input is exactly what is needed for the geometry portion of the TRANCITS input.

The input associated with the stress model is much simpler than that of the heat transfer model. To interface nodal temperature for the stress model all that is required is a file containing the name and rectangular cartesian coordinates of the stress points to be interfaced. This information is contained in the stress model coordinate file. If elemental centroid or elemental gauss point temperatures are requested, such as required by MARC, an additional file containing the stress element connectivity is also required.

The primary output of the transfer module is a neutral temperature output file. This file consists of the name of the stress node and its corresponding temperature for every transient time point requested. This file can easily be input into simple formatting routines and the stress point temperatures can be configured into any form required. In addition to this neutral output file, the program has the capability to automatically format the temperatures compatible with the NASTRAN or the MARC analysis codes.

6.3 FINITE DIFFERENCE CONSIDERATION

There are two primary differences between using a finite difference code versus a finite element code in conjunction with the transfer module. The first is the potential lack of geometry associated with the finite difference codes. This has been discussed in the previous section. The second is the elemental locations at which the thermal results are predicted. The basic output for a finite element heat transfer code is the temperature at the nodal points. This is ideally suited for the interfacing procedure, since these temperatures along with the assumptions associated with the thermal field inside the element can be used to accurately interpolate data within the element domain. On the other hand, finite difference codes produce temperatures at the element centroids and perhaps at the center of the element faces. Temperatures at these locations are much more difficult to use effectively and pose a problem. The approach taken in the transfer module is to convert these elemental temperatures into accurate temperatures at the elemental vertices. Once this has been accomplished, the interfacing technique proceeds in a manner similar to the one used for finite element heat transfer codes. This conversion of the elemental temperatures to nodal temperatures is performed in two steps. The first step is to map on an element by element basis the centroidal and face center temperatures to the element vertices. This mapping must be done in a fashion that is consistent with the assumptions governing the thermal distribution within the element. For a 3D linear finite difference volume, the assumption is made that the temperatures can vary linearly in all 3 directions between the face centers and the centroid of the element. Given the temperatures at the 6 face centers and the centroid, the temperature, at the eight vertices of the element can be computed making use of the 3D tetrahedral isoparametric shape functions. The equations that must be solved are of the form:

$$T = [1, x, y, z] \; [c]^{-1} \; \overline{T}$$

where

T = temperature at vertice

$$\left.\begin{matrix} x \\ y \\ z \end{matrix}\right\} = \text{coordinates of vertice}$$

$\overline{T}$ = known temperatures at the centroid and the three adjacent face centers.

where

$$[C]^{-1} = \begin{bmatrix} 1 & X_I & Y_I & Z_I \\ 1 & X_J & Y_J & Z_J \\ 1 & X_K & Y_K & Z_K \\ 1 & X_L & Y_L & Z_L \end{bmatrix}^{-1}$$

where

$$\left.\begin{matrix} X_{IJKL} \\ Y_{IJKL} \\ Z_{IJKL} \end{matrix}\right\} = \text{coordinate of the tetrahedral}$$

Note that $[C]^{-1}$ is only a function of geometry and therefore must only be computed once regardless of the number of transient time steps. This inverse is computed in closed form in the transfer module and is therefore efficient and accurate. Figure 6-3 illustrates the finite difference volume and the corresponding tetrahedral.

Note that during this initial stage the actual temperatures of the vertices are not directly computed, only the coefficients that relate the temperatures at the vertices to those at the known locations are computed.

The second step in this process is to compute a unique temperature at each vertice. This is done by weighting all of the vertice coefficients by a scale

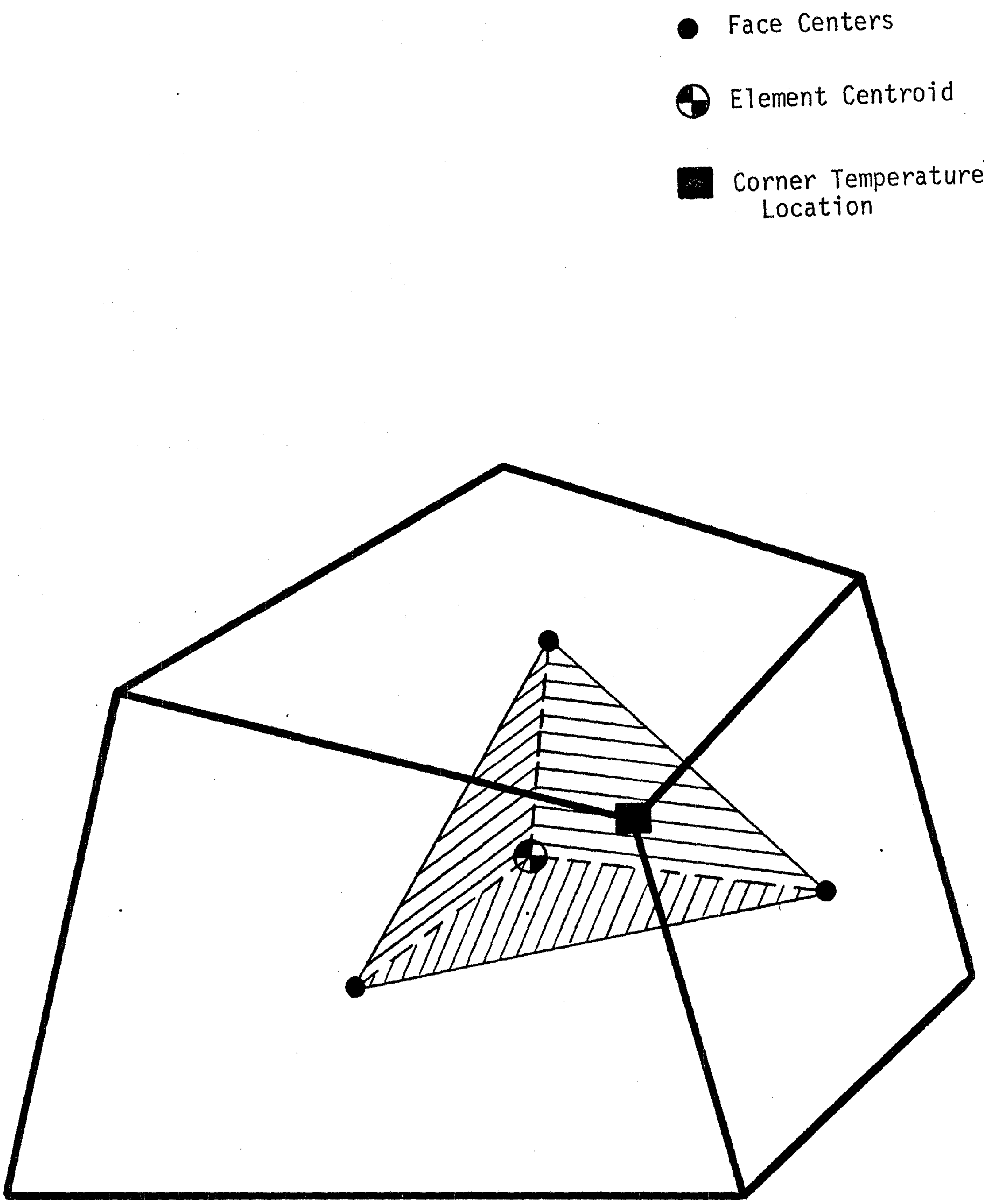


Figure 6-3. Finite Difference Volume and Corner Tetrahedral.

factor that is inversely proportional to the distance between the centroid of each element adjacent to the vertice and the vertice itself. This technique makes use of all of the surrounding temperature information to eventually produce a unique temperature at the vertice of each heat transfer element. Note that if the finite difference results are only at the centroid, the transfer module assumes the entire element was at that temperature. The weighting process however will generate different vertice temperatures since the temperature of the vertice depend on the geometry and the number of elements associated with that vertice.

6.4 3D SEARCH AND WEIGHTING COEFFICIENT ROUTINES

Two of the criteria that must be met by any transfer module are that the interfacing procedure be efficient and accurate. In order to satisfy the efficiency requirements, techniques were developed that rapidly determine which heat transfer element contains the stress points of interest. The approach taken was to invoke a "multi-step filtering" process. This process was developed and implemented such that most of the element candidates were eliminated using a very simple algorithm. The elements that passed through the initial "coarse" filter were then subjected to a more sophisticated test ("fine filter") that determined if the stress point was contained within the bounds of the element. This search technique, like the vertice coefficient routine is only a function of geometry and therefore must only be performed once regardless of the number of transient solutions interfaced. The simple coarse filter requires that the minimum and maximum values of the x, y, z coordinates be stored for each heat transfer element. This procedure is outlined below.

The stress point is outside of the heat transfer element if any of the following conditions are met:

$$X_s < X_{min}$$

$$X_s > X_{max}$$

$$Y_s < Y_{min}$$

$$Y_s > X_{max}$$

$$Z_s < Z_{min}$$

$$Z_s > Z_{max}$$

where X_s, Y_s, Z_s are the coordinates of the stress point. For most models this technique drastically reduces the number of heat transfer elements that could possibly contain the stress point. Note that this procedure is valid for any order 3D heat transfer element as long as the maximum and minimum values are based on all of the nodes associated with the element.

The technique employed as the "fine filter" actually performs two tasks. First it determines if the stress point lies outside of the element, in which case the next element is considered. Second, if it is determined that the stress point is contained within the element, it automatically returns coefficients that relate the known temperatures of the vertices of the element to the desired temperature at the stress point location. This "fine filter" or weighting coefficient routine is based on the inversion of the isoparametric shape functions. The choice of these interpolating functions was based on many factors. These functions are widely used for many finite element heat transfer element formulations. They can be formed to represent a variety of temperature fields and therefore can be used to describe the variation within a finite difference volume. They are mathematically easy to manipulate and understand. The inversion process involves solving for the so-called local isoparametric coordinates (r, s, t) based on the global coordinates of the vertices of the element and coordinates of the stress point. Regardless of the order of the element the following equations can be written:

$$X = \sum_{i=1}^{\eta} \alpha_i X_i$$

$$Y = \sum_{i=1}^{\eta} \alpha_i Y_i \qquad 6.4.2$$

$$Z = \sum_{i=1}^{\eta} \alpha_i Z_i$$

$$T = \sum_{i=1}^{\eta} \alpha_i T_i$$

where X, Y, Z represents the geometry

T represents the normal field

X_i Y_i Z_i are the coordinates of the vertices of the heat transfer element

N is the number of vertices

α_i are the isoparametric shape functions

The problem is to solve for local coordinates (r, s, t) that correspond to the global coordinates of the stress point (X_p, Y_p, Z_p). The relationships between the shape functions and the local coordinates are well known once the order of the element has been fixed. Substitution of these relations into the above summation along with the equating of X to X_p Y to Y_p and Z to Z_p yields a set of non-linear equations with r, s and t as unknowns. The local coordinates become functions of each other, global vertice coordinates and the coordinates of the stress point.

That is:

$$r_p = f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p)$$

$$s_o = f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p) \qquad 6.4.3$$

$$t_p = f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p)$$

These equations can now be solved numerically to yield values of r_p, s_p, t_p. Once the values of the local coordinates are known, a simple test determines if the stress point is contained in the heat transfer element. The stress point is not contained in the element if any of the following conditions are met.

$$-1 > r_p > 1$$

$$-1 > s_p > 1$$

$$-1 > t_p > 1$$

If the point is found to be contained with local coordinates can be substituted into the expressions for the shape functions and the thermal weighting coefficient can then be computed.

This is an ideal method for this interfacing procedure, it finds the correct heat transfer element and at the same time produces accurate weighting coefficients. Figure 6-4 illustrates this mapping technique.

For this study the 3D linear isoparametric formulation was chosen as the interpolating functions for the heat transfer elements. The set of non-linear equations described in Equation 6.4.3 were solved by using a fixed point iteration technique in conjunction with an "educated initial value" procedure. These initial values are computed as:

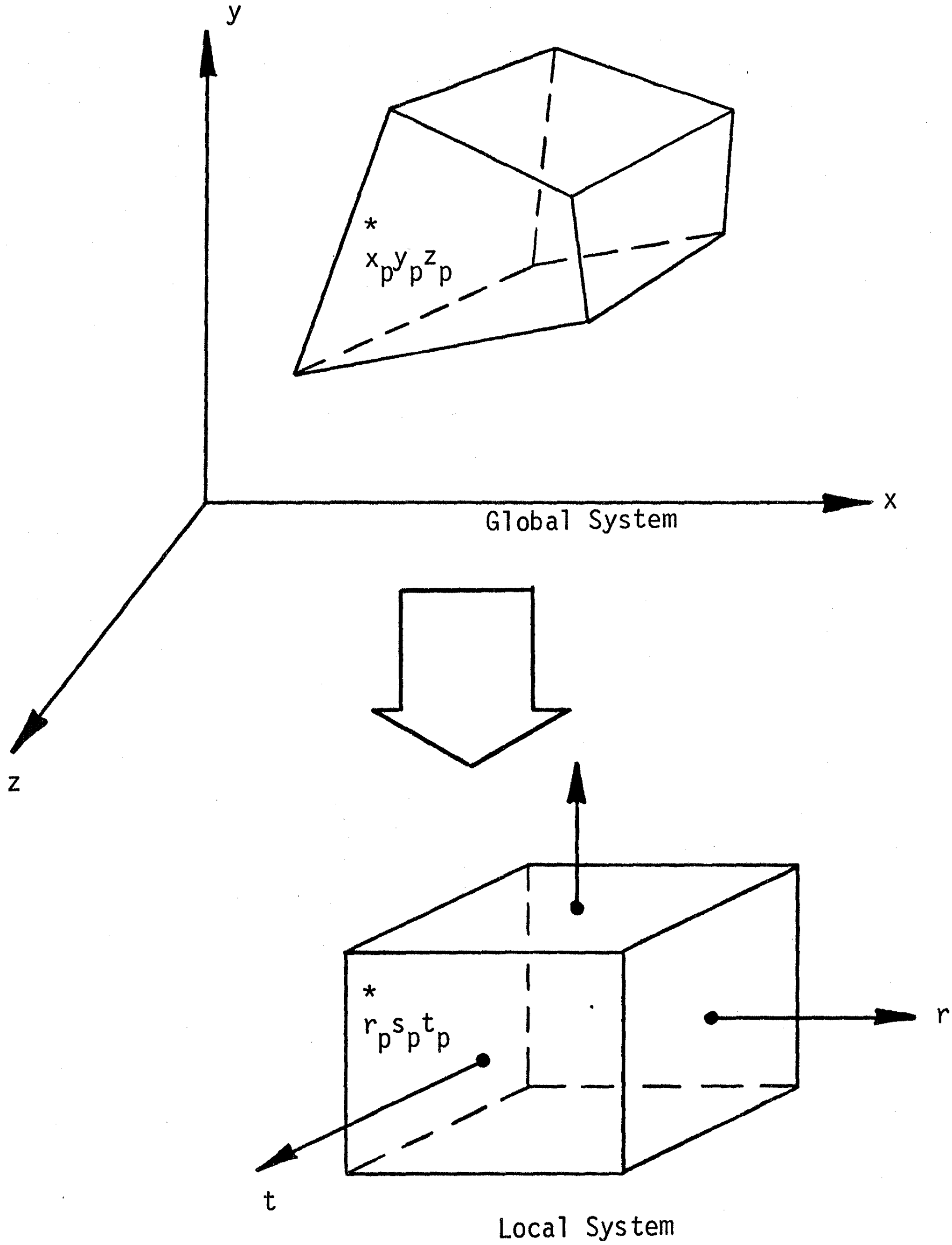


Figure 6-4. Isoparametric Mapping of Global Coordinates To Local Coordinates.

$$
\begin{aligned}
r_{pi} &= \frac{X_p - X_{min}}{X_{max} - X_{min}} \\
s_{pi} &= \frac{Y_p - Y_{min}}{Y_{max} - Y_{min}} \qquad 6.4.5 \\
t_{pi} &= \frac{Z_p - Z_{min}}{Z_{max} - Z_{min}}
\end{aligned}
$$

These initial values correspond to the correct values of r_p, s_p, t_p if the heat transfer element is a rectangular parallel piped. The initial guesses are used in conjunction with the righthand side of the non-linear equation to produce new values of the local coordinates. This iteration continues until convergence is achieved. The initial implementation of this technique showed it to be very efficient and fairly insensitive to element distortion. Several highly distorted 8-node 3D elements were tested. Figure 6-5 shows two of these elements. Convergence was achieved for all points attempted using these elements. Once this technique was installed into the transfer module and actual problems were interfaced, it was discovered that although the iterative procedure was not sensitive to the shape of the element, it was very sensitive to the orientation of the element with respect to the global axis system. It was found that for some configurations, the technique became unstable and either diverged or converged to an undesirable root of the equations.

Logic was added to the transfer module to avoid these stability problems. First, a coordinate transformation was applied to the heat transfer element to align the direction defined by I and J joints of the element with the global X axis. The iterative technique is then used to solve for the local coordinates of the stress point. If it fails to converge to the correct root, the program reorients the element such that a new vertex of the element is now the I joint. The alignment transformation is then performed and the iterative solution is repeated. If it still fails to converge a new vertex is chosen for the I joint and the process is again repeated. If convergence is not achieved after all the vertices have been used, the stress point is not contained in this heat transfer element.

A 400 element airfoil model, shown in Figure 6-6 was run using this version of the transfer module. The stress points used were the centroids of the elements and 8 additional points per element that were very near the corners of the element. There were a total of 3735 stress points. After the first transformation 3728 of the stress points converged. The other 7 stress points converged after the additional reorientation of the element was completed.

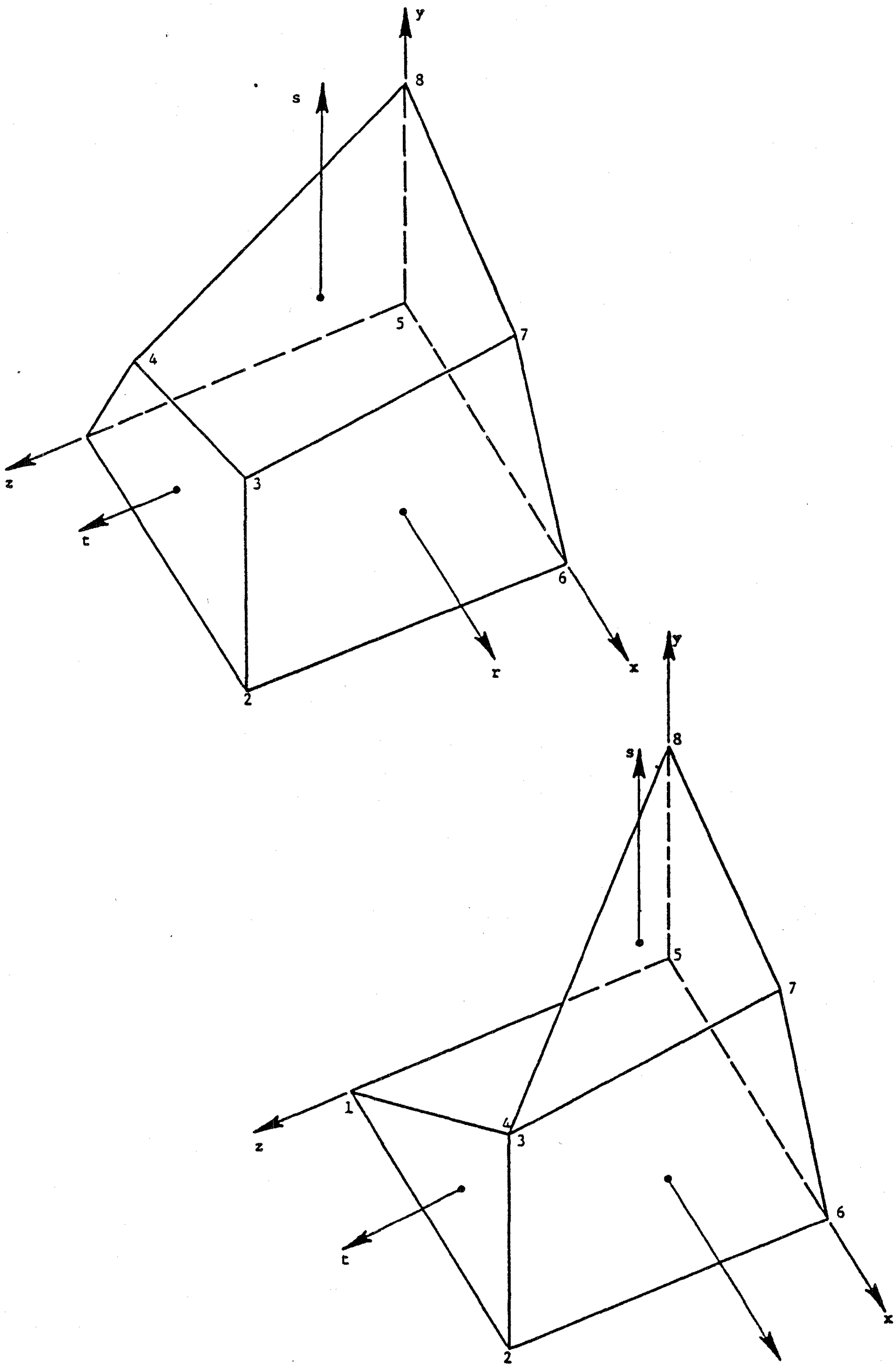


Figure 6-5. Highly Distorted Elements Tested.

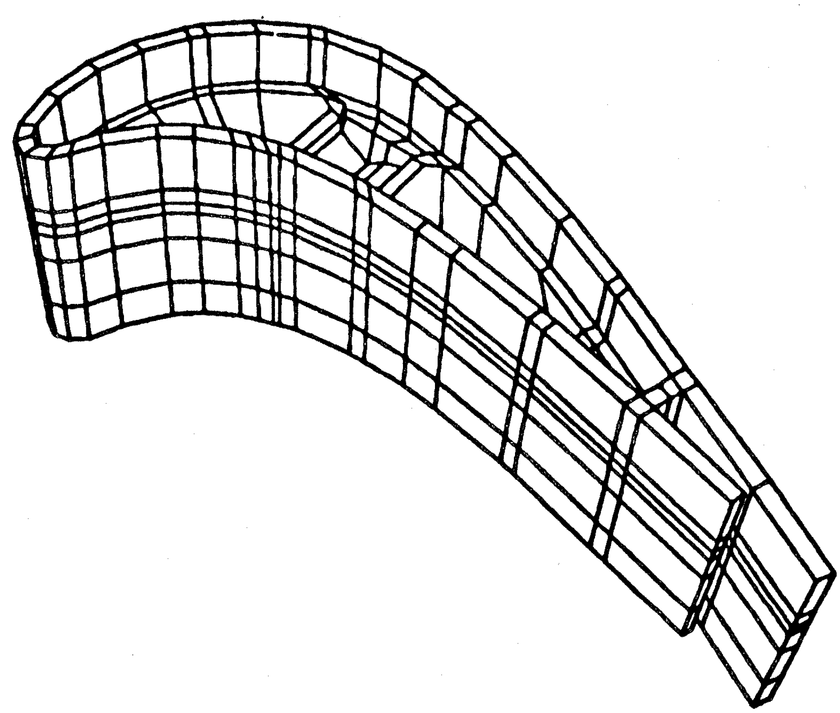


Figure 6-6. 400-3D Element Air Foil Model Used As Test Case.

These results show that the initial transformation is adequate for most of the stress points and that the additional computation required for the vertex reorientation is only necessary for a very small percentage of stress points.

As stated earlier the existing transfer module inverts the 3D linear isoparametric shape function. The technique used, however, can be applied to any order element with only small impact on the coding of the transfer module.

6.5 EXTERIOR POINT SURFACING ROUTINE

In order for the transfer module to be a useful tool, it must deal with stress points that lie slightly outside of the heat transfer module. This situation occurs for many reasons in practice, the most common being part tolerances used to create the heat transfer model and the stress model. It is not the intention of the transfer module to produce temperatures for stress Points that are in areas outside of the domain of the thermal analysis and therefore no extrapolation of temperatures are performed. On the other hand, if a stress point exists that physically represents the surface of the part and that point because of slight tolerance mismatches is not contained in the heat transfer element on the surface, the transfer module should produce a temperature that is representative of the surface temperatures. This has been accomplished in TRANCITS by implementing an exterior surfacing routine. This technique also makes use of the local coordinates (r, s, t). If the stress point lies outside the element, one of the local coordinates will be greater than one. For example,

$r = .6$

$s = -1.2$

$t = -.8$

This point is not contained in a heat transfer element since

$s \quad 1$

A point that lies on the surface near the stress point would have local coordinate values of

$r \ = .6$

$s \ = -1$

$t \ = .8$

The distance between (r, s, t) and (r , s , t) can easily be computed in physical space and stored. The heat transfer element with the smallest distance can then be used as the closest element and the temperature of the surface point can be produced. This allows the transfer module to predict an approximate temperature for the stress points that lie outside the heat transfer model. The program flags all stress points that exhibit this behavior and also prints out the distance between the actual coordinates of the stress point and the point on the surface for which the temperatures were computed. Users should pay close attention to these points and judge if the distance outside the model is acceptable for the analysis being performed. Figure 6-7 illustrates this surfacing technique along with the options of built in transformation and isoparametric mapping techniques.

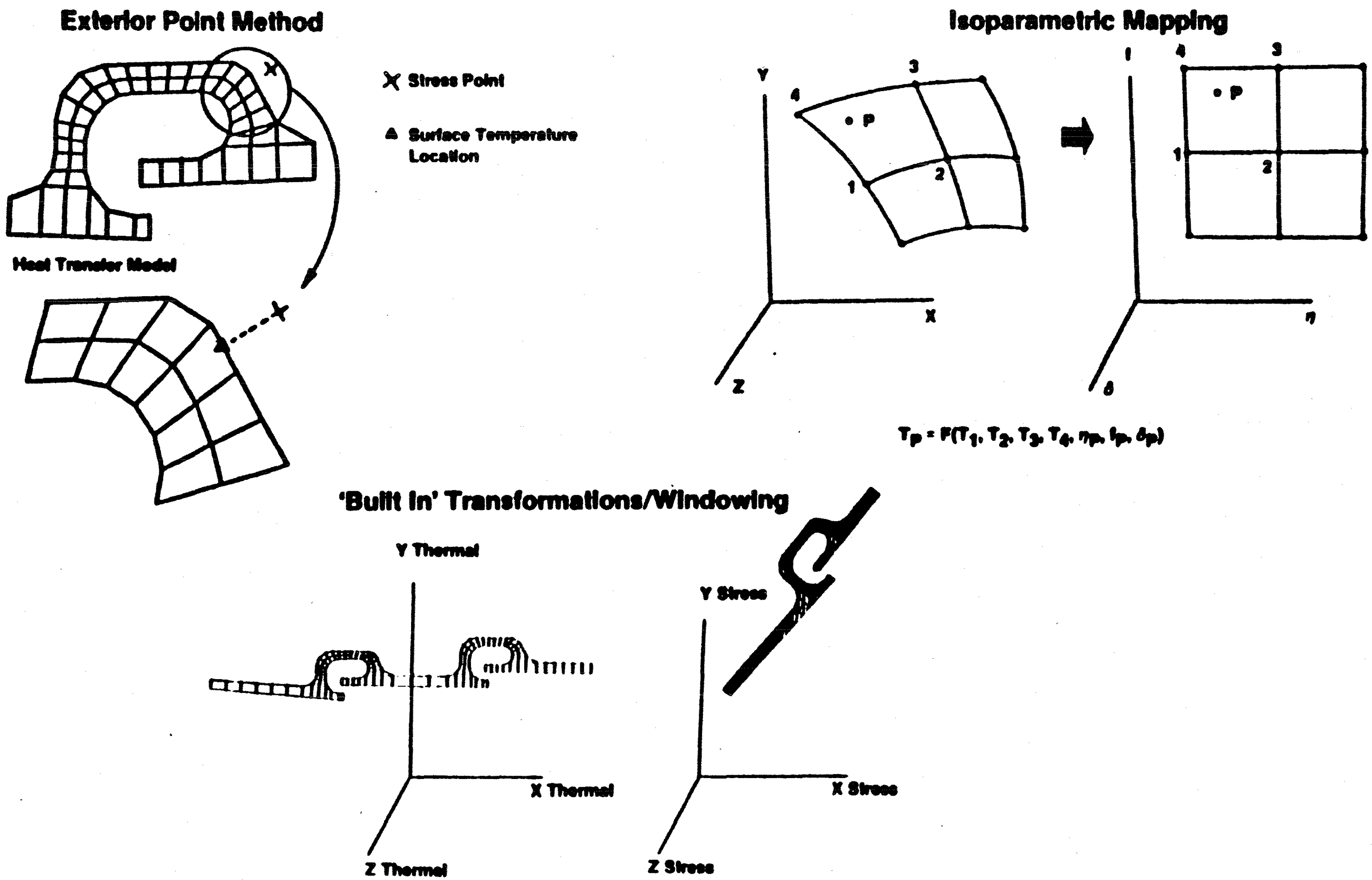


Figure 6-7. Enhanced Program Features.

7.0 VERIFICATION AND DOCUMENTATION OF THE TRANSFER MODULE

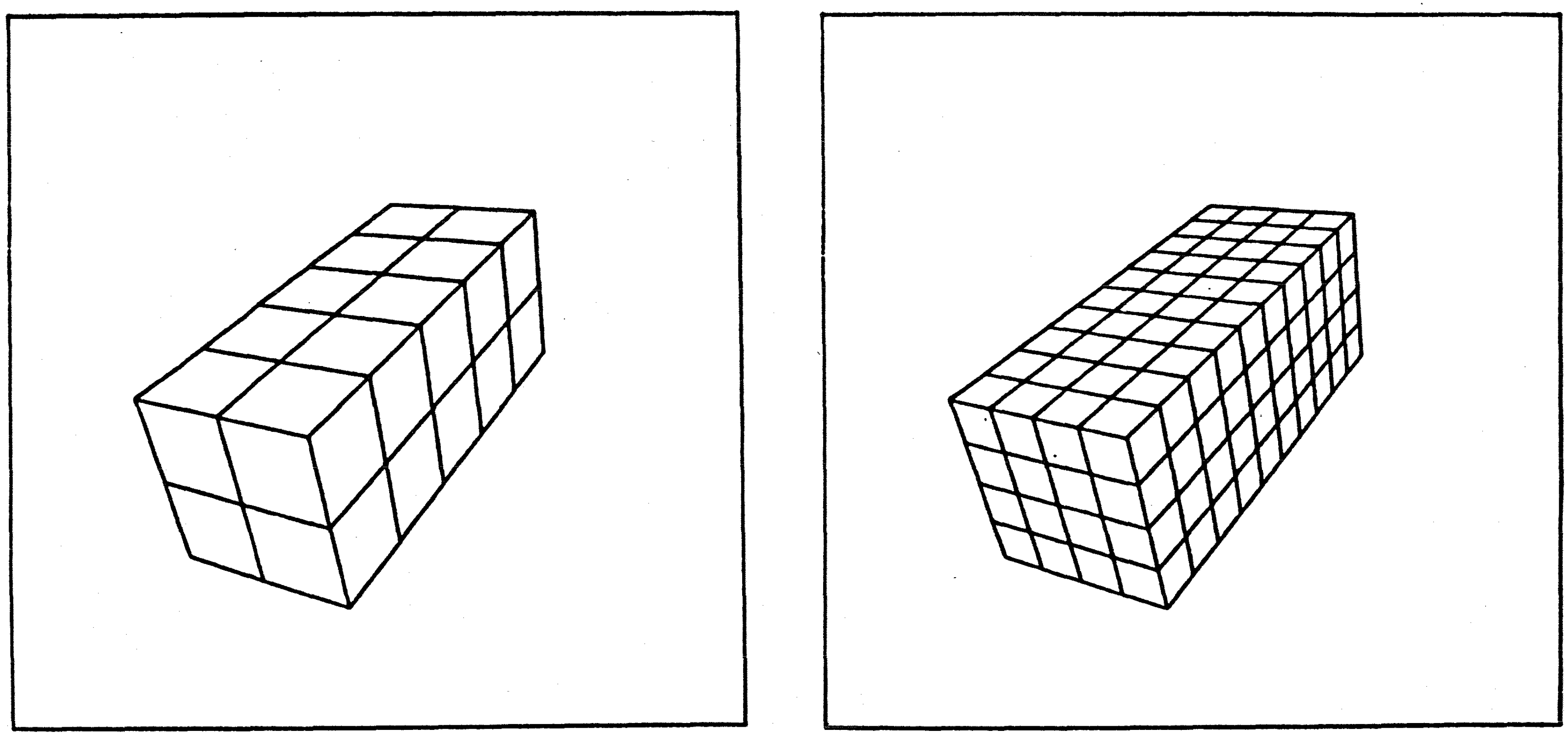
Two objectives were met by the verification and documentation phase of this contract. The verification subtask provided the necessary test cases to ensure that all of the options implemented were producing accurate and efficient results. Indeed many efficiency improvements were identified and implemented in this phase. The purpose of the documentation subtasks was to provide information required to use the transfer module as well as the documents needed to understand and possibly alter the coding for specific problems.

7.1 VERIFICATION OF THE TRANSFER MODULE

Verification of the transfer code was performed in parallel with the development effort. Each of the modules in the code were tested to ensure that the input and output of those modules were consistent with all file structures. All user options were executed and thermal results from both SINDA and MARC were directly input and processed by the transfer code. Nodal temperatures for NASTRAN were produced as well as elemental centroid temperatures for MARC.

Formal verification of the mapped thermal results were conducted using three different 3D models. The primary objective of these test cases was to ensure that the thermal data produced by the heat transfer codes was not degraded in the mapping to the stress analysis models. The verification procedure involved comparing the results of the heat transfer code to the interpolated temperatures produced by TRANCITS. Comparisons were made using two different methods. The first method plotted the heat transfer results and the TRANCITS results against the model geometry for various regions of the 3D models. These plots were used to verify that the thermal gradients predicted by the heat transfer analysis were indeed mapped into the stress models. The second comparison method utilized the PATRAN Reference 3 postprocessing program. Colored isotherms were created for the stress model using the interpolated temperatures and then for the heat transfer model using the original heat transfer results. These two sets of isotherms were then compared for a qualitative check on the thermal gradients.

The first test case was a simple model of a rectangular prism. The heat transfer mesh was composed of a 2x2x5 grid, the stress mesh was much finer and had a mesh density of 4x4x10. These models are shown in Figure 7-1. The boundary conditions applied to the heat transfer model give rise to a linear thermal gradient along the length of the prism. The transfer module was used to map these temperatures into the finer stress model and Figure 7-2 shows a plot of the heat transfer code temperatures and the TRANCITS temperatures versus the distance along the prism. The figure shows that for this simple case the interpolated values agree exactly with those predicted by the heat transfer code. Figure 7-3 and 7-4 show the isotherms for the heat transfer model and stress model respectively. Visual comparison of these isotherms again indicate the same thermal gradients in both models.



HEAT TRANSFER MODEL

STRESS MODEL

Figure 7-1. Prism Test Case.

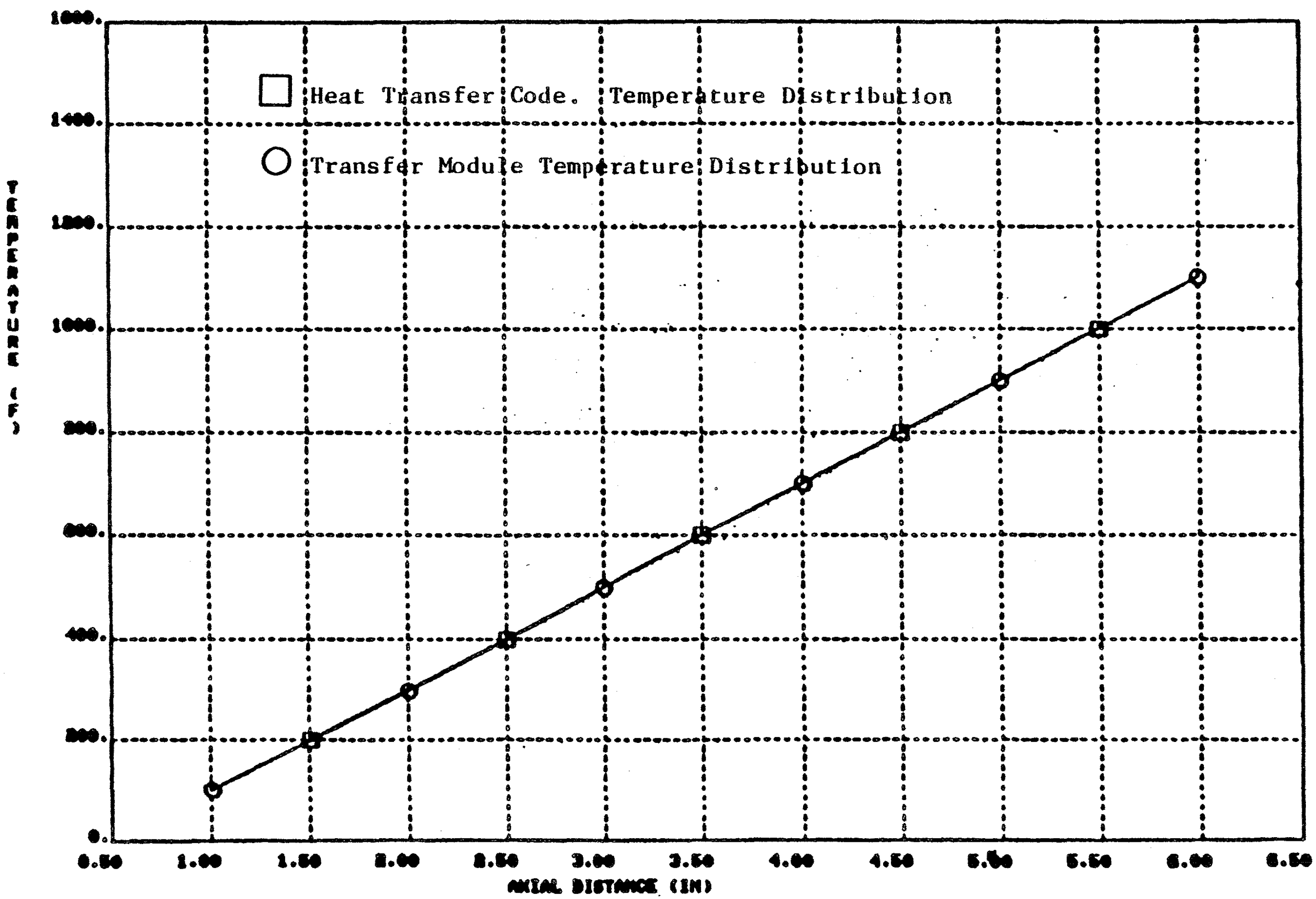


Figure 7-2. Temperature Versus Axial Distance For Beam Test Case.

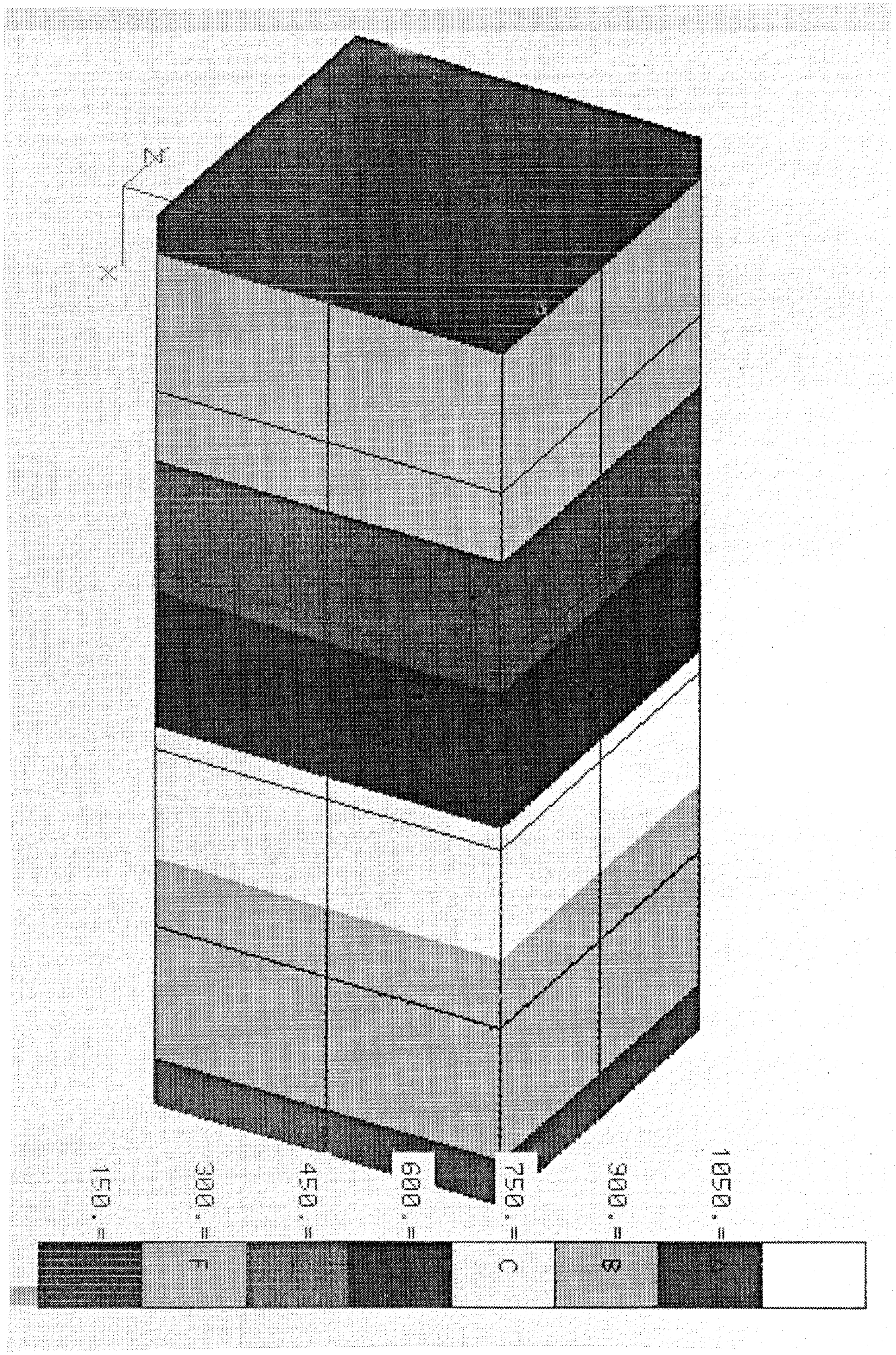


Figure 7-3. Heat Transfer Model Isotherm.

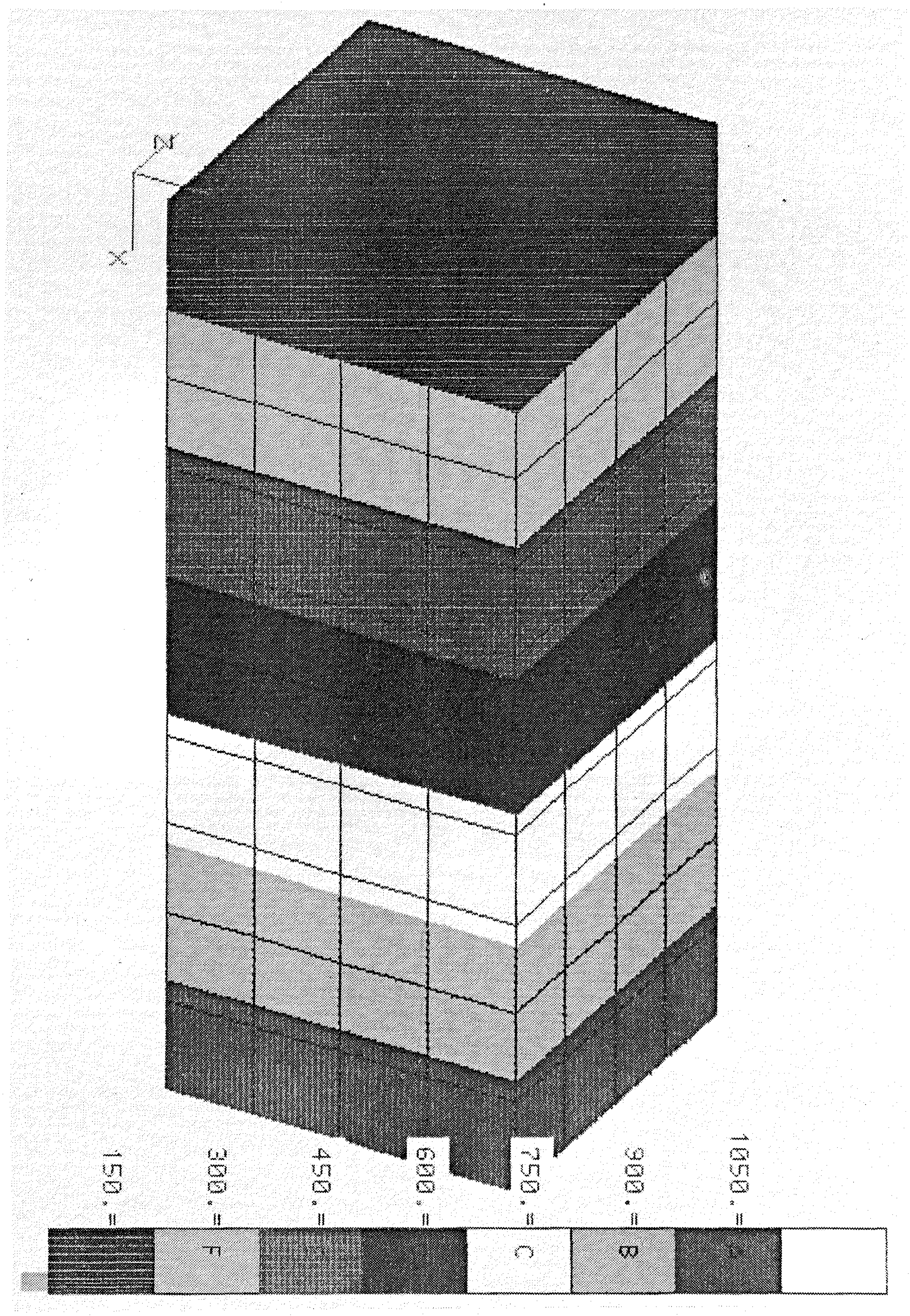


Figure 7-4. Stress Model Isotherm

The model used for the second test case is shown in Figure 7-5. It is finite difference model of the tip of a turbine blade composed of approximately 450 heat transfer elements. For this case the stress model mesh density was identical to the heat transfer model. The transfer module was used to map the heat transfer temperature at the centroid of the elements and the face centers of the element sides to the vertices of the element. The boundary condition applied to the heat transfer model represent realistic conditions for a turbine blade and give rise to gradients in all directions. Figures 7-6 thru 7-11 show plots of the original heat transfer temperatures and the transferred corner temperatures against a local distance parameter. These plots are for 3 spanwise sections through the airfoil (Tip, Middle, Root Sections) for both the concave and convex sides of the blade. The local distance parameter is measured from the leading edge on the concave side and the trailing edge on the convex side. The first two curves on each plot represent the temperatures for a layer of heat transfer elements just above and just below the corner node temperatures. As the plots show, the corner temperature distribution not only has the same trends as the original heat transfer temperature distribution but the value of the temperature at each corner node falls in between the temperature above and below the corner as computed by the heat transfer code.

These comparisons provide excellent verification of the accuracy of the transferring technique.

Models representing the final test case are shown in Figure 7-12. These models represent a 3D sector of a combustor liner. The mesh density for the heat transfer model is finer than the stress model in some locations and coarser in others. The boundary conditions used in the heat transfer model once again represent typical engine conditions. Figures 7-13 and 7-14 show isotherms for both heat transfer and stress grid respectively. These isotherms also verify that the gradients are essentially the same for both models.

7.2 DOCUMENTATION OF THE TRANSFER

The documentation of this code was divided into two segments. The first was the TRANCITS User's Manual. This manual describes in detail all of the input and output file contents for the options implemented in this version of the transfer module. It shows examples of these files and discusses how the code should be executed. It also has a complete description of all the user input variables and what defaults if any are used if these inputs are omitted. The manual specifies the current limitations of the code in regard to maximum size of problems that can be interfaced. The manual should provide all of the necessary information required to use the transfer modules.

The second involved the documentation of the coding of the transfer module. Listings and prologs of each subroutine used and a flow diagram of the entire code have been included in this report in Appendices C and D. These prologs are part of the source code and they give a brief description of the purpose of the subroutine. They also describe all of the calling arguments and whether these arguments are inputs or outputs to the routine. All files used in the routines are specified as well as any routines that are called

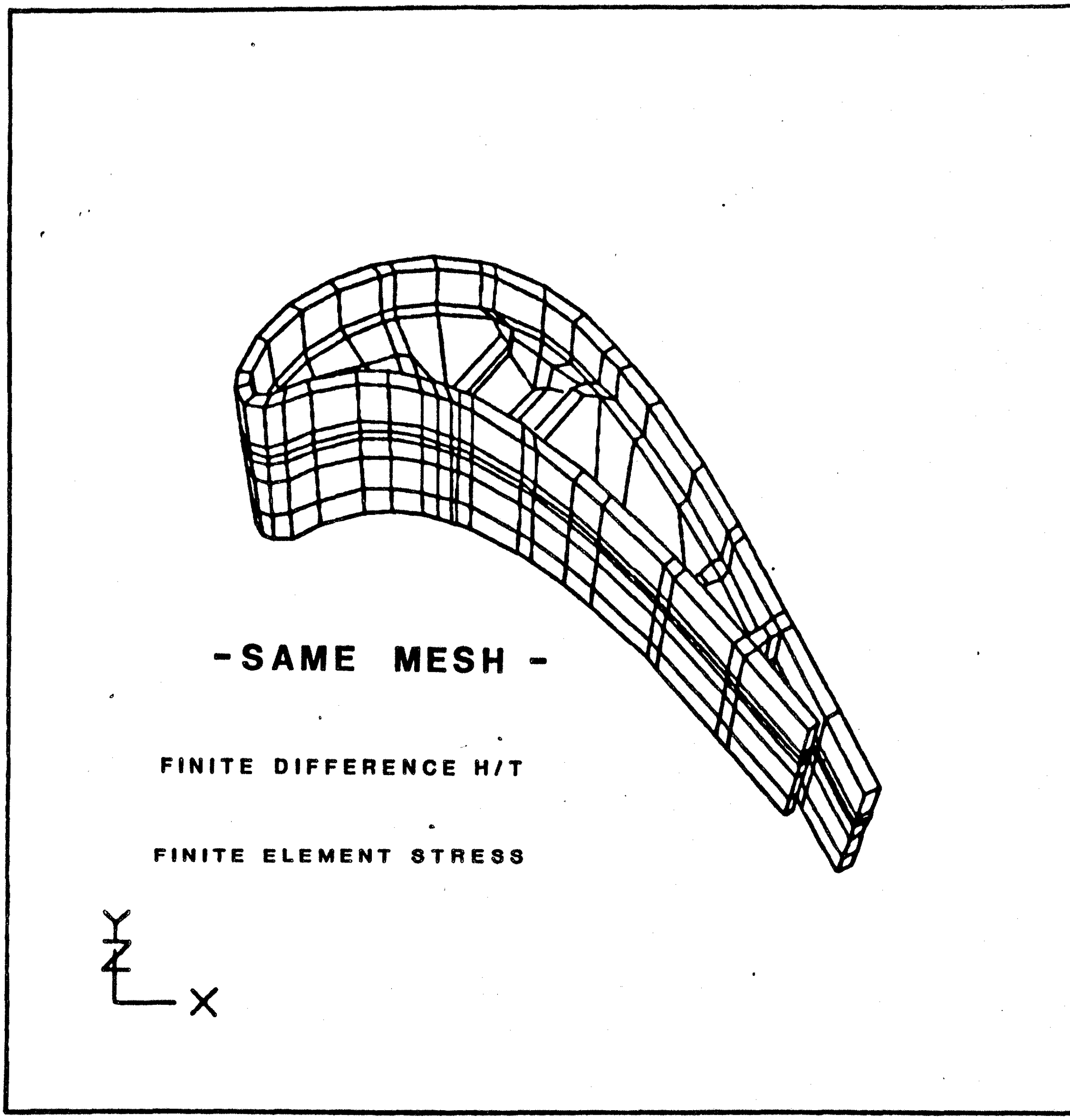


Figure 7-5. Airfoil Test Case Model.

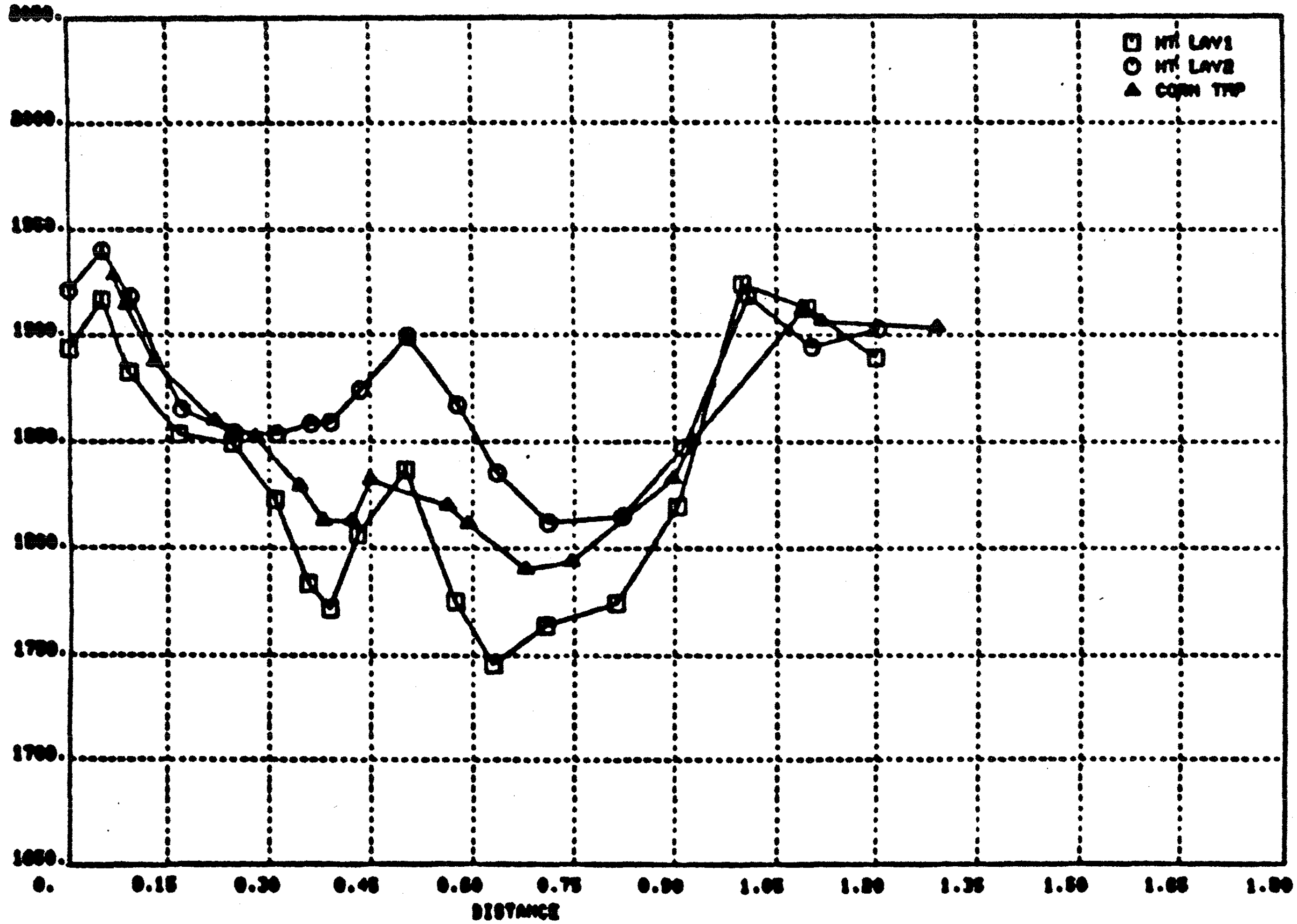


Figure 7-6. Temperature Verification Plots Concave Side Tip Section Large Model (Airfoil).

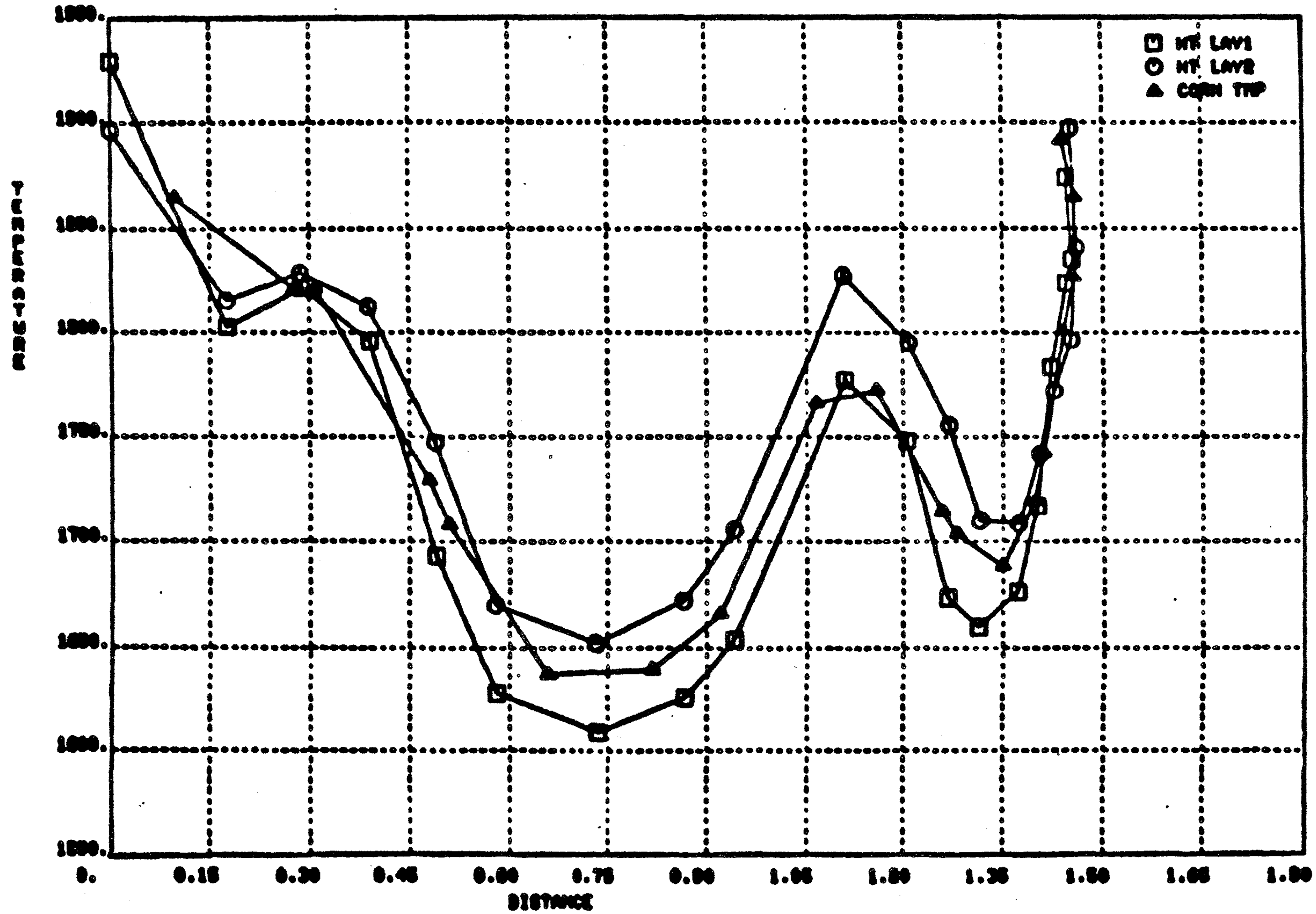


Figure 7-7. Temperature Verification Plots Convex Side Tip Section Large Model (Airfoil).

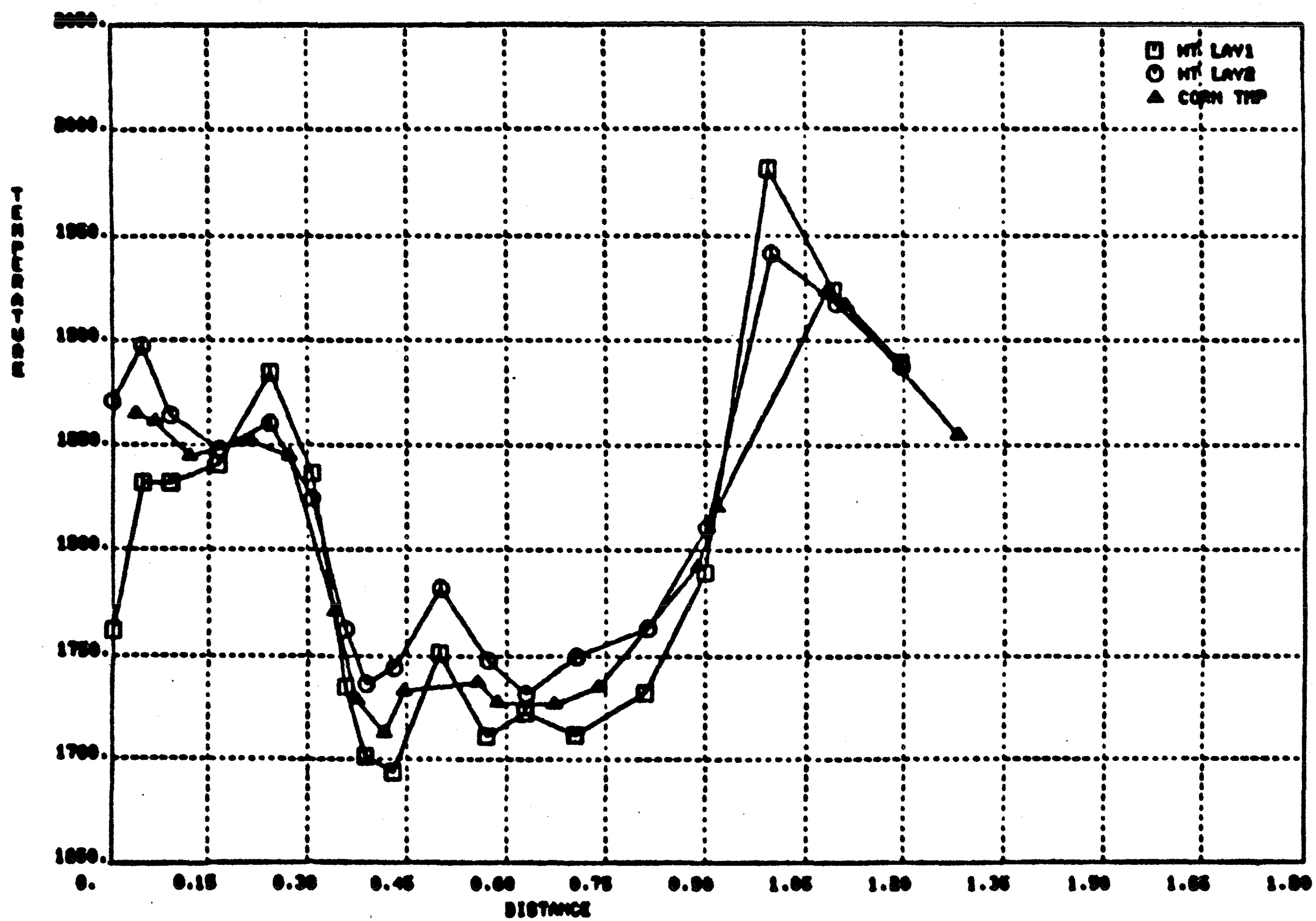


Figure 7-8. Temperature Verification Plots Concave Side Mid Section Large Model (Airfoil).

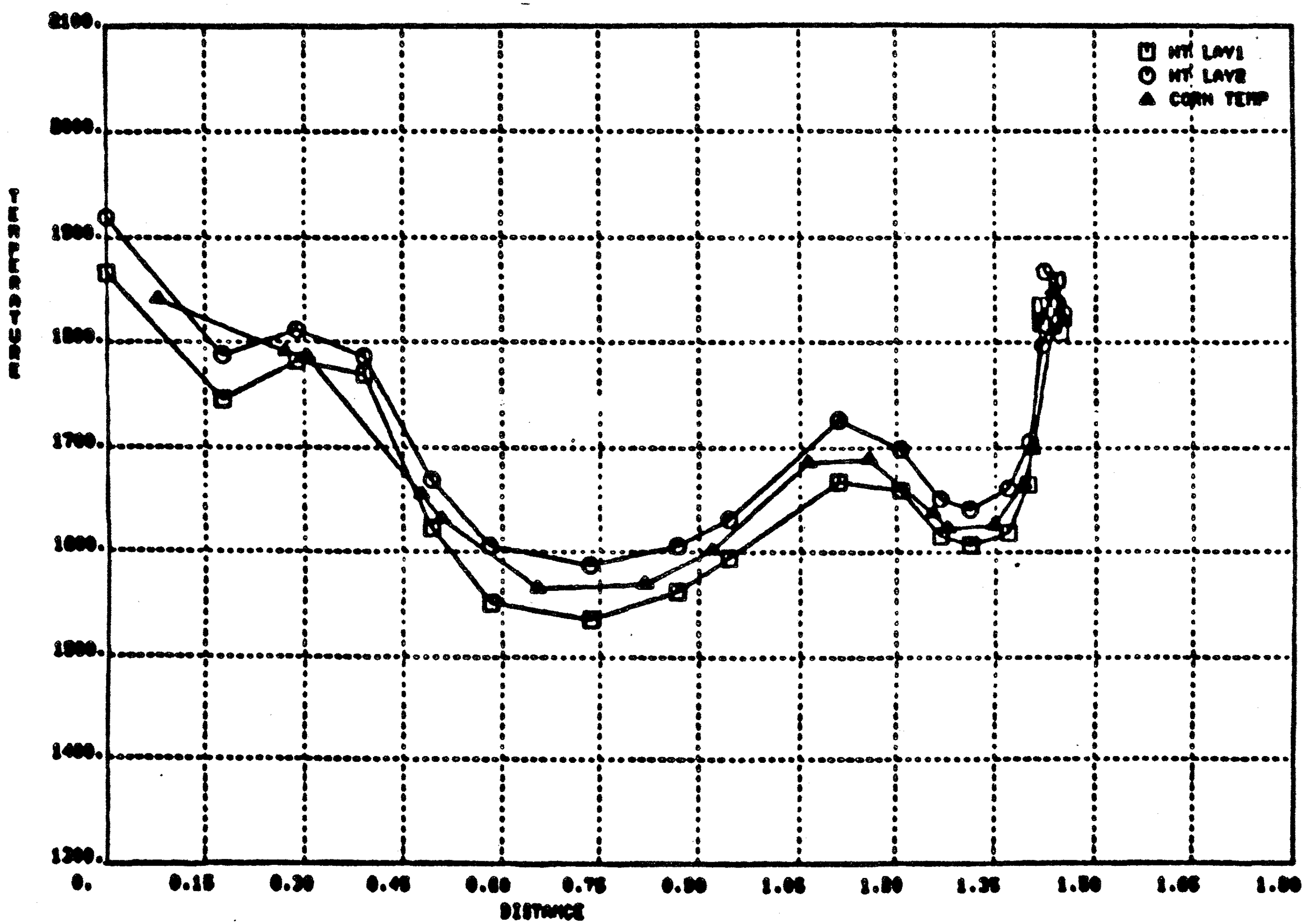


Figure 7-9. Temperature Verification Plots Convex Side Mid Section Large Model (Airfoil).

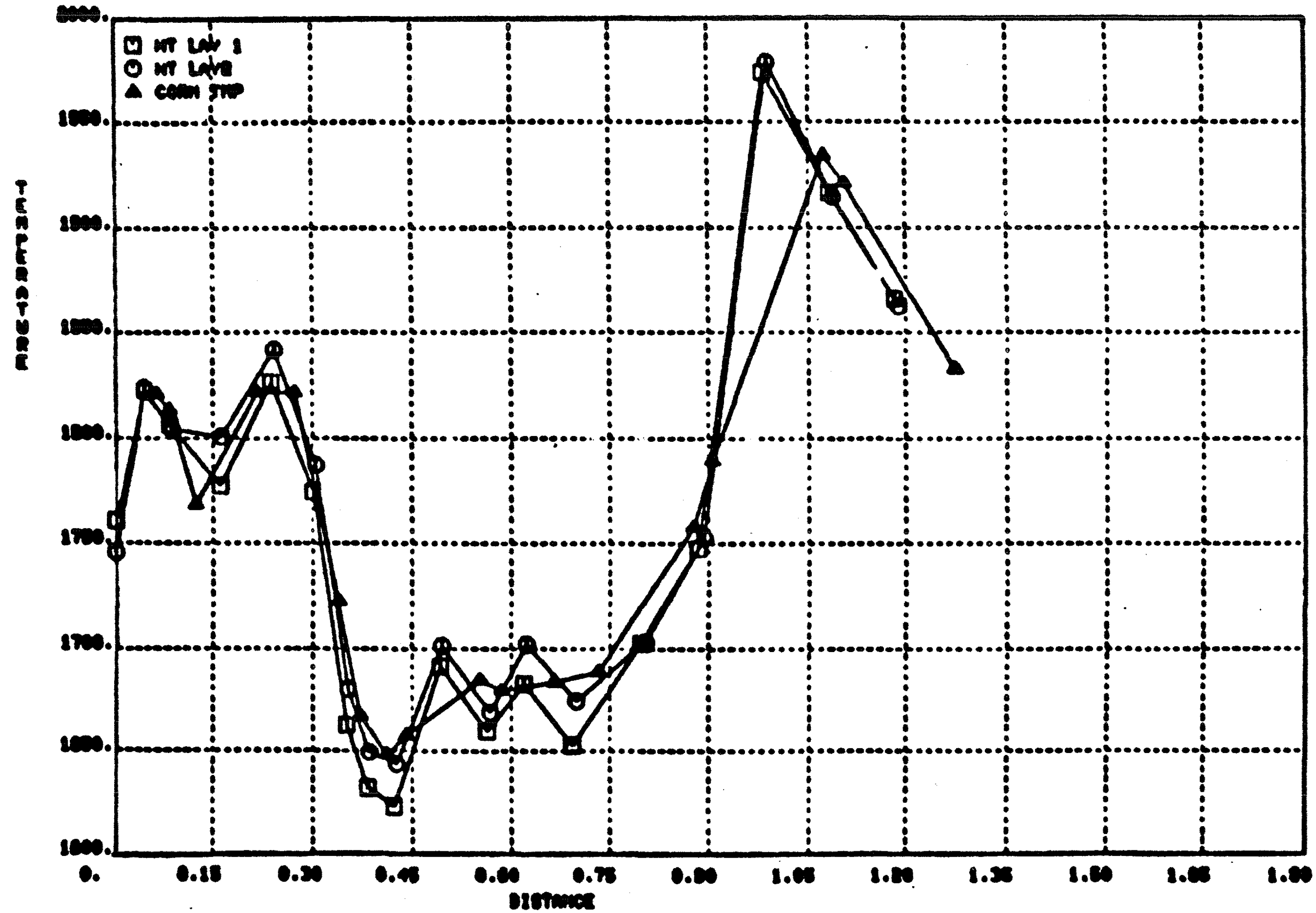


Figure 7-10. Temperature Verification Plots Concave Side Root Section Large Model (Airfoil).

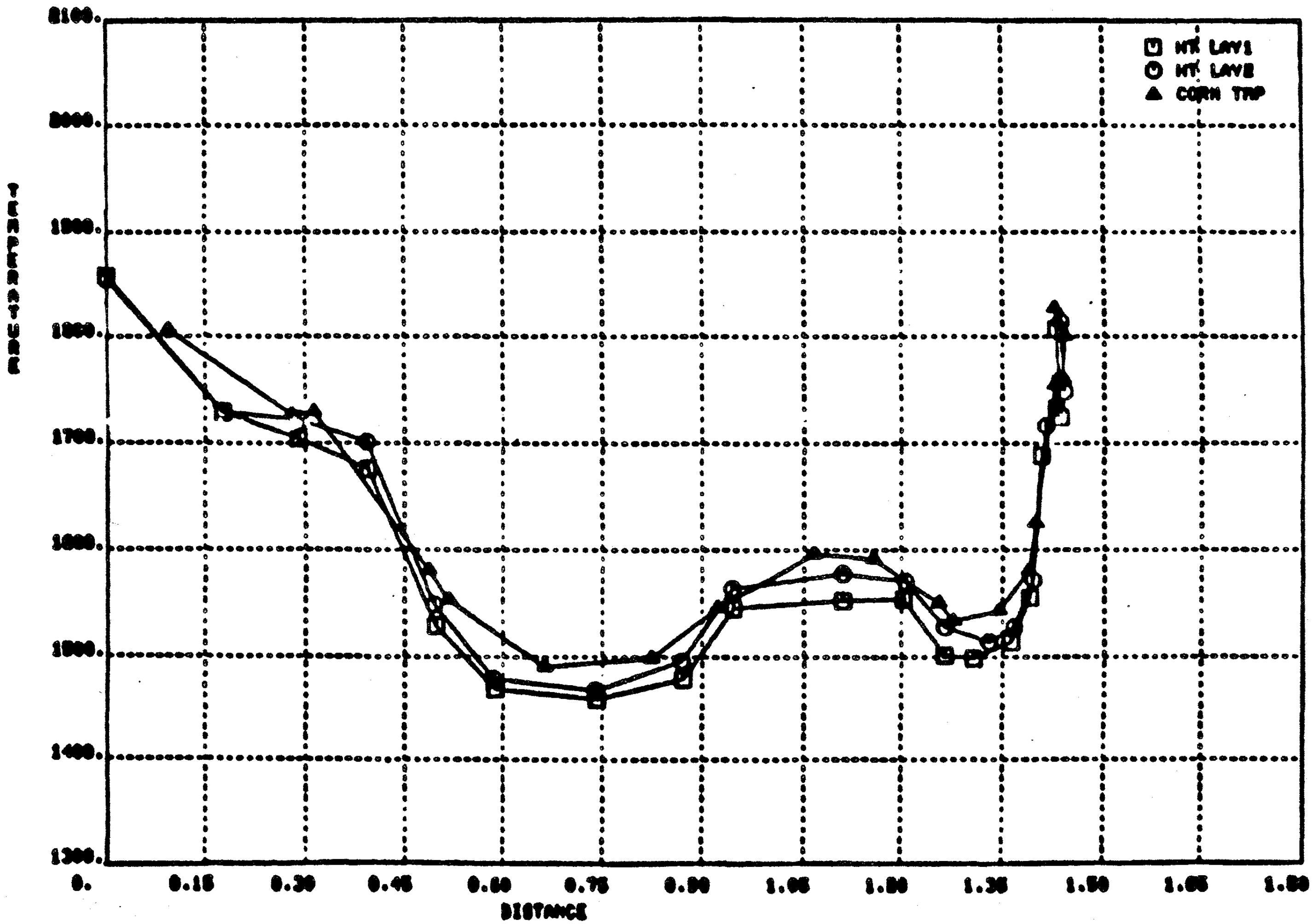
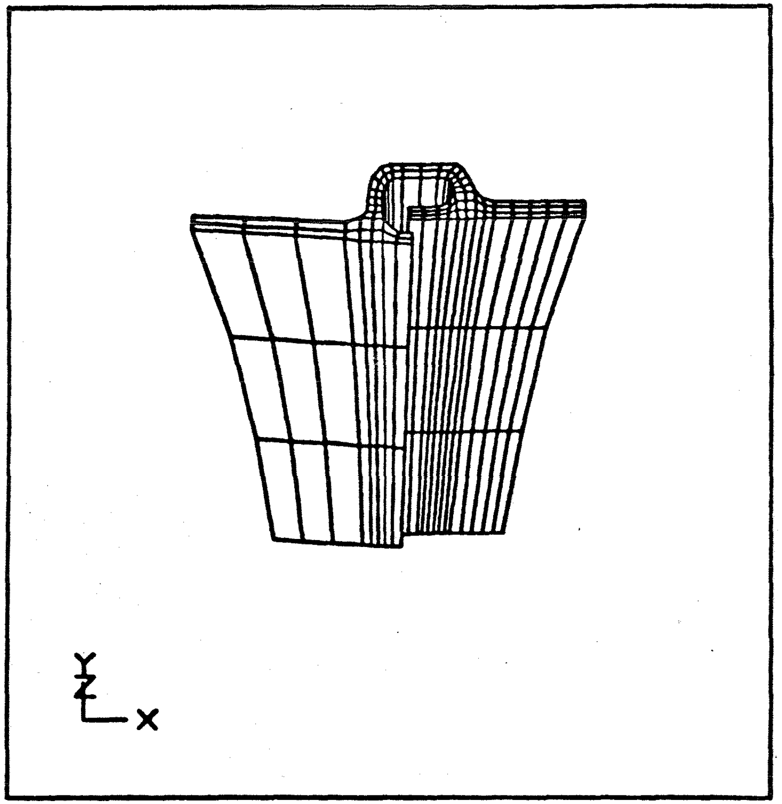
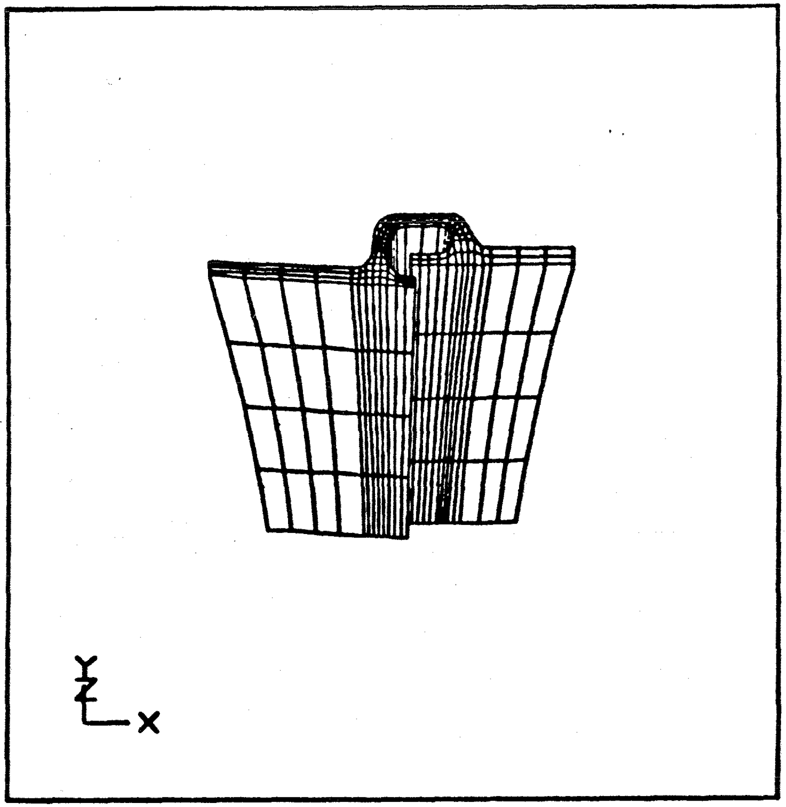


Figure 7-11. Temperature Verification Plots Convex Side Root Section Large Model (Airfoil).



HEAT TRANSFER MODEL



STRESS MODEL

Figure 7-12. Combustor Liner Test Case Models.

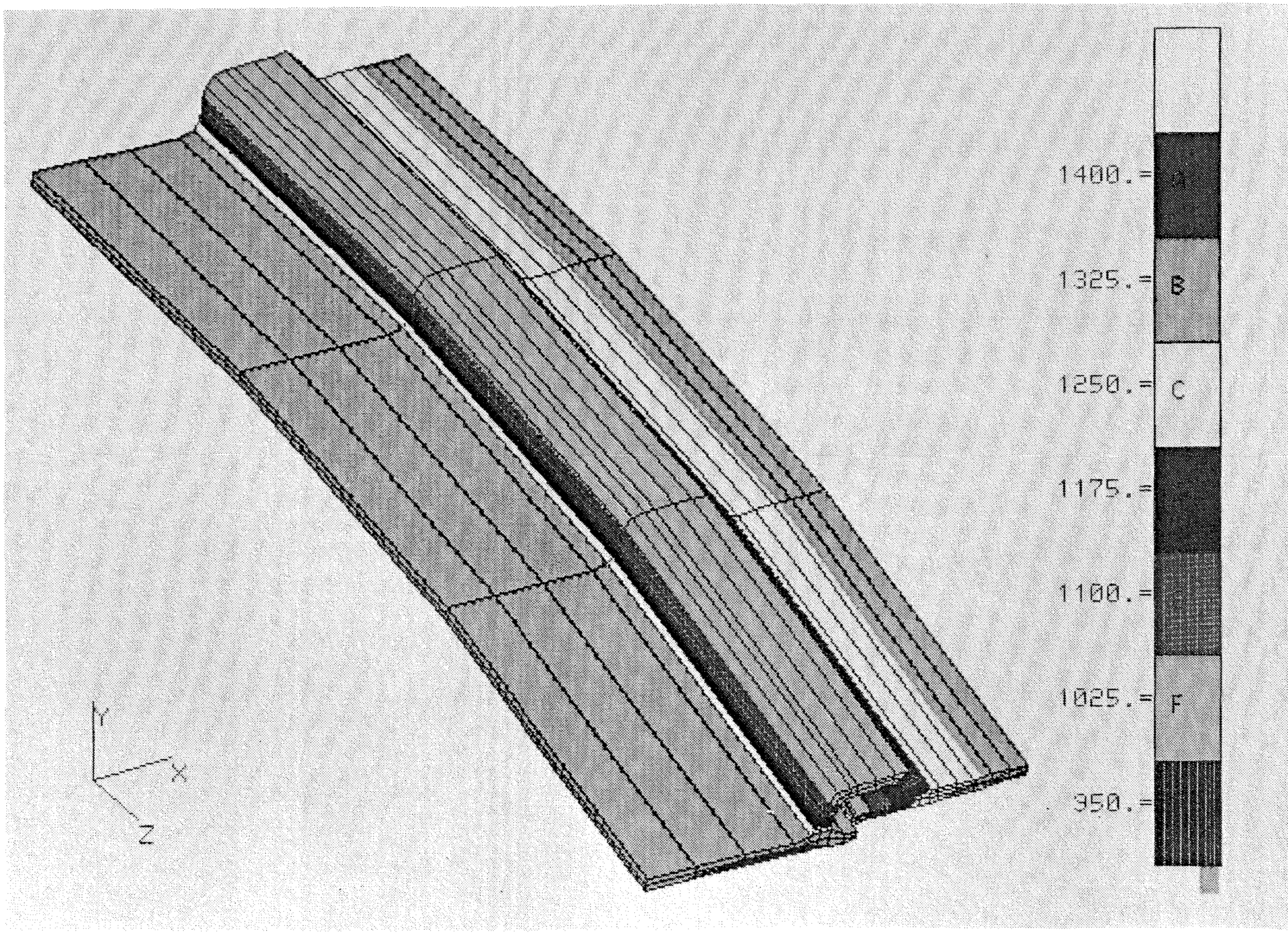


Figure 7-13. Combuster Liner Heat Transfer Model Isotherms.

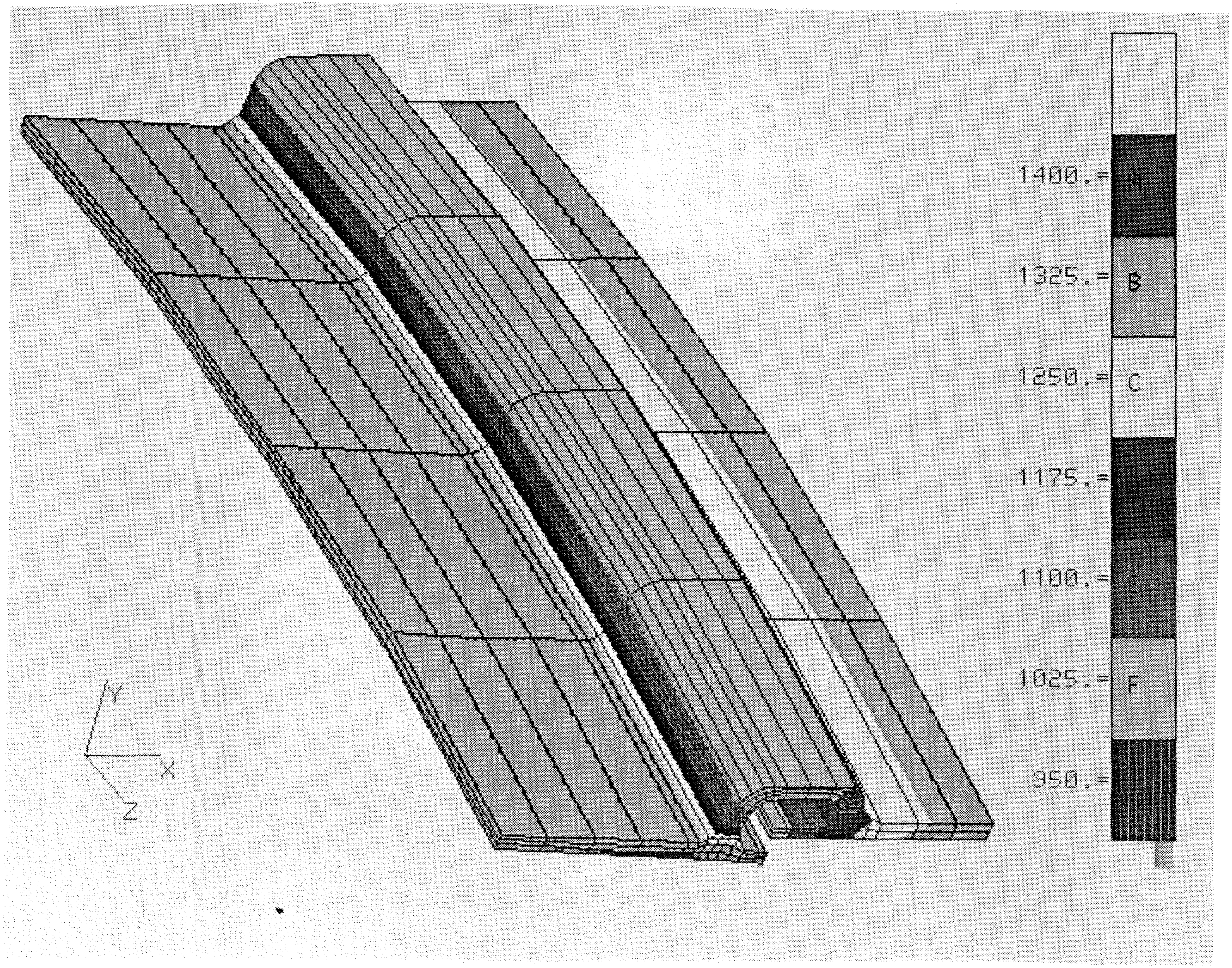


Figure 7-14. Combuster Liner Stress Model Isotherms.

from the subroutine. Commons that are used in the routine are also noted. If there are any special limitations or comments, these are described in the prolog.

The flow diagram gives a concise picture of the program flow and provides a convenient reference for which subroutines are a part of each module. This and the program documentation should enable any user to understand and perhaps adapt this transfer module to their individual needs.

8.0 CONCLUSIONS

The TRANCITS software package accurately and efficiently transfers thermal data from a 3D heat transfer study to a 3D stress analysis mesh of different elemental density. The heat transfer analysis can be performed using finite element or finite difference techniques. TRANCITS currently supports the output of the MARC and SINDA heat transfer codes directly and will format the thermal data compatible with the input requirements of NASTRAN and MARC. In addition, the neutral heat transfer input file and neutral temperature output file can be used to interface the results of any heat transfer code to any structural code. The transfer module has been written in a modular fashion and is easy to implement and modify. The technology developed to transfer thermal data forms an excellent foundation for the transfer of other engineering parameters such as pressures, loads and displacements.

This Page Intentionally Left Blank

APPENDIX A

Survey of Computer Programs for Heat Transfer
(A.K. Noor)

The following tables and discussions were extracted from A.K. Noor's paper "Survey of Computer Programs for Heat Transfer Analysis."

These tables illustrate the solution capabilities, the heat transfer elements supported, the pre and post-processing options, the type of boundary conditions supported and any interfacing options for six of the heat transfer codes considered.

Table I

Part I - Analytical Capabilities of the Program

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
1.	**Goal of Program System**							
	General Purpose	●		●	●	●	●	
	Commercial	●		●	●	●		
	Research		●	●	●	●		
	Educational	●		●	●	●		
	Others (see program abstracts)							
2.	**Method of Analysis**							
	Finite Elements	●		●	●			
	Finite Differences		●			●	●	
	Boundary Integral Method				●			
	Perturbation Technique							
	Hybrid Analytical - Numerical Technique (see program abstracts)							
	Others (see program abstracts)					●		
3.	**Space Dimensionality**							
	Three-Dimensional	●	●	●	●	●	●	
	Two-Dimensional	●	●	●	●	●	●	
	One-Dimensional	●	●	●	●	●	●	

*Extracted form "Survey of Computer Programs for Heat Analysis" By A.K. Noor.

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
	Solids of Revolution	●		●	●			
	Boundary Elements				●			
	Scalar Elements			●	●			
	Point Contact Elements	●		●	●			
	Others (see program abstracts)							
4.	Range of Applications and Phenomena							
	Linear Steady State	●	●	●	●	●		
	Nonlinear Steady State	●	●	●	●	●	●	
	Thermal Frequencies and Mode Shapes							
	Linear Transient Response	●	●	●	●	●		
	Nonlinear Transient Response	●	●	●	●	●	●	
	Others (see program abstracts)							
5.	Formulation							
	a) Fundamental Unknowns							
	Temperatures	●	●			●	●	
	Temperatures and Flux							
	Enthalpy							
	Others (see program abstracts)					●		
	b) Elemental Matrices							
	Conduction	●	●			●	●	

			ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
	Capacitance	a) Consistent	●		●	●		●	
		b) Lumped	●	●	●	●	●		
	Convection	a) Free	●		●	●	●	●	
		b) Forced	●		●	●	●	●	
	Radiation		●		●	●	●	●	
	Interelement Convection and Radiation		●			●	●	●	
	User Supplied Elements (see program abstracts)					●			
	Others (see program abstracts)		●						
6.	Material Properties and Material Models								
	Isotropic		●	●	●	●	●	●	
	Anisotropic		●	●	●	●	●		
	Multilayered			●					
	Temperature-Dependent Properties	Conductivity	●	●	●	●	●	●	
		Specific Heat	●	●	●		●	●	
		Density	●	●		●	●		
		Absorptivity (Emissivity Factors)	●	●	●	●	●	●	
		Convection Coefficients	●	●	●	●	●	●	
	Perfect Conductors (Via Multipoint Constraints)				●	●			
	Time Dependent Thermal Properties		●	●	●		●		

			ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
	Latent Heat and Phase Change Effects		●		●		●	●	
	Material Added or Removed During Analysis				●		●		
	User Supplied (see program abstracts)				●		●		
	Others (see program abstracts)				●				
7.	Initial Conditions								
	Homogeneous.		●	●	●	●	●		
	Varying Throughout the Region			●	●	●	●	●	
	Initial Enthalpy (for Phase Change)					●			
	User-Supplied			●	●	●	●		
	Others (see program abstracts)								
8.	Boundary Conditions and Thermal Loads								
	Prescribed Temperatures	a) Steady State	●	●	●	●	●		
		b) Time Dependent	●	●	●	●	●	●	
	Thermal Flux Input	a) Steady	●	●	●	●	●		
		b) Temperature Dependent		●	●	●	●		
		c) Time Varying	●	●	●	●	●	●	
	Convection from a Surface to Its Surroundings	a) Steady State	●	●	●	●	●		
		b) Time Dependent	●	●	●	●	●	●	

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
Convection from Surroundings to a Surface	a) Steady State	●	●	●	●	●		
	b) Time Dependent	●	●	●	●	●	●	
Forced Convection		●		●	●	●		
Radiation from a Surface to its Surroundings	a) Steady State	●	●	●	●	●		
	b) Time Dependent	●	●	●	●	●	●	
Radiation from Surroundings to a Surface	a) Steady State	●	●	●	●	●		
	b) Time Dependent	●	●	●	●	●	●	
Radiation Between Narrow Gaps		●	●	●	●	●		
Radiation Between n Surfaces with	a) User-Supplied View Factors	●	●		●	●	●	
	b) Internally Calculated View Factors				●			
Prescribed Fluid Flow		●			●	●		
Boundary Layer Convection				●	●			
Volumetric Heat Generation	a) On Element Level	●		●	●		●	
	b) On Node Level		●	●	●	●		
Gap (Thermal Resistance)		●		●	●	●	●	
Boundary Conditions/Loads Added or Removed During Analysis		●		●	●	●		
Others (see program abstracts)								

				ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
9.	Solution Techniques									
	Linear Steady State	a) Direct		●		●	●			
		b) Iterative			●			●		
		c) Others (see abstracts)					●			
	Nonlinear Steady State	a) Incremental				●				
		b) Iterative		●	●	●	●	●	●	
		c) Others (see abstracts)								
	Transient	a) Thermal Mode Superposition					●			
		b) Direct Integration	i) Explicit		●			●		
			ii) Implicit: User Specified Time Step	●	●	●	●	●	●	
			ii) Implicit: Automatic Time Step Selection				●	●		
			iii) Combined Explicit/Implicit							
		c) Finite Elements in the Time Domain								
		d) Moving - Deforming Grids								
		e) Others (see program abstracts)								
10.	Other Capabilities									
	Thermal Stress Analysis Capability	a) Uncoupled								
		b) Coupled								

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
	Temperature Field Data Transmitted Directly from Heat Transfer Modules to Thermal Stress Modules	●		●	●		●	
	Enclosure Radiation with View Factor Calculation				●			
	Heat Input/Output at Constrained Boundaries	●		●	●		●	
	Cyclic Symmetry				●			
	Substructuring: a) Repeated Use of Identical Substructures	●			●			
	Substructuring: b) Mixing Linear and Nonlinear Substructures				●			
	Substructuring: c) Mixing Substructures with Different Types of Nonlinearities	●	●	●	●	●	●	
	Restart Capability				●			
	Others (see program abstracts)							
II.	Program Operational On	●	●	●	●	●	●	
	CDC	●	●	●	●	●		
	IBM	●		●	●	●	●	
	UNIVAC	●						
	Honeywell							
	Telefunken	●		●	●			
	AMDAHL							
	SIEMENS							
	ICL							

		ANSYS	HEAT-INGS	MARC	NAST-RAN	SINDA	THTD	
	Minicomputers (see program abstracts)	●		●	●	●		
	Supercomputers (see program abstracts)	●		●	●	●		
	Others (see program abstracts)			●				
12.	Documentation							
	Programmer's Manual		●	●	●	●	●	
	Theoretical Manual		●	●	●		●	
	Data Preparation - Users' Manual	●	●	●	●	●	●	
	Example Problem Manual	●		●	●		●	
	Verification/Validation Manual	●		●	●			
	Pre- and Post-Processors' Manual	●		●	●	●	●	

Part II - User Interface and Modeling Capabilities

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
1.	Input Form and Sequence							
	a) Input Form							
	Fixed Format		●	●	●			
	Free Form-List Directed Format	●		●	●		●	
	Problem Oriented Language				●	●		
	Others (see program abstracts)							
	b) Input Sequence							
	User Directed	●		●			●	
	System Directed		●		●	●		
	User Supplied Subroutines (see program abstracts)			●	●	●		
2.	Model Generation and Checking							
	a) Automatic or Semi-Automatic Generator for:							
	Nodal Point Coordinates	●	●	●	●		●	
	Element Connectivities	●	●	●	●		●	
	Constraints, Symmetry and Boundary Conditions	●	●	●	●		●	
	Substructure Connectivity				●			
	Repetition of Identical Segments	●		●	●		●	
	Others (see program abstracts)							

	ANSYS	HEAT-INGS	MARC	NAS-TRAN	SINDA	THTD	
b) Automatic or Semi-Automatic Generator for:	●		●	●			
One-Dimensional Elements	●		●	●			
Triangular Elements	●		●	●			
Quadrilateral Elements	●		●				
Body or Shell of Revolution Elements	●		●	●		●	
Three-Dimensional Solid Elements	●			●			
Two-Dimensional Shell Elements							
Curvilinear Finite Difference Grids							
Others (see program abstracts)							
c) Data Checking Facilities	●	●	●	●	●	●	
Line Printer	●	●	●	●	●	●	
Plotter							
Interactive Graphics							
Others (see program abstracts)							
d) Plots and Graphics Display of Model		●					
Complete Analysis Region	●	●	●	●		●	
Part of Analysis Region	●	●	●	●		●	
"Blow-Up" Option	●	●	●	●		●	

		ANSYS	HEAT-INGS	MARC	NAS-TRAN	SINDA	THTD	
	Hidden Lines or Surfaces	●		●	●			
	Orthographic Views	●			●			
	Perspective and Isometric Views	●		●	●		●	
	Section View on Arbitrary Plane	●		●	●			
	Others (see program abstracts)							
	e) Other Facilities							
	Digitizer Input	●					●	
	Automatic Renumbering of Nodes, Elements or Equations	●		●	●		●	
	Table Lookup of Data	●					●	
	Others (see program abstracts)							
3.	Results Output Form							
	a) Tabular Output							
	Fixed Set	●		●		●		
	User Defined Set and Sequences	●	●	●	●	●	●	
	Maximum and Minimum Quantities	●	●		●	●		
	Average and Maxima for Blocks of Nodes							
	Temperature or Flux Exceedances	●			●			
	Others (see program abstracts)							

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
	b) File Output for User Post-Processing and Plotting	●	●	●	●	●		
	c) Plots							
	Isotherm Plots (Contours) of Temperatures/Flux	●	●	●	●		●	
	Surface Functions				●		●	
	Selective Output (e.g., by Elements or Regions)	●	●	●	●			
	Histories (e.g., Time History)	●	●	●	●	●	●	
	Others (see program abstracts)				●			
4.	Interactive Input and Control							
	Parameter Specicification (e.g., Flux or Time Steps)	●					●	
	Singularity Check			●				
	Error Correction/Recovery	●						
	User Control of Matrix Decomposition							
	Others (see program abstracts)							

APPENDIX B

Survey of Structural Analysis Codes

The following tables were extracted from a survey performed by R. Zirin of General Electric Gas Turbine Division.

These tables illustrate the solution capabilities, the element libraries, the types of loadings and the output features of the following structural codes: EASE2, STARDYNE, NASTRAN, ANSYS, MARC, SUPERB.

TYPES OF ANALYSIS

ANALYTICAL CAPABILITY		PROGRAM					
		EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
LINEAR STATICS	MECHANICAL LOADS	•	•	•	•	•	•
	TEMPERATURE LOADS	•	•	•	•	•	•
	EULER BUCKLING			•		•	
	INERTIA RELIEF			•			
DYNAMICS	MODE/FREQUENCY	•	•	•	•	•	✓
	FREQUENCY RESPONSE		•	•	•		
	TRANSIENT RESPONSE	•	•	•	•	•	
	SHOCK SPECTRA	•	•		•		✓
	RANDOM RESPONSE		•	•			
	NONLINEAR TRANSIENT			•	•	•	
NONLINEAR STATICS	NONLINEAR BUCKLING					•	
	LARGE DISPLACEMENT				•	•	
	PLASTICITY			•	•	•	
	CREEP				•	•	
	VISCOELASTICITY			•		•	
	LARGE STRAINS					•	
HEAT TRANSFER	STEADY STATE			•	•	•	•
	TRANSIENT			•	•	•	
SUBSTRUCTURES (SUPER-ELEMENTS)	STATIC		•	•	•		
	DYNAMIC		•	•	•		
	CYCLIC SYMMETRY			•			
MISCELLANEOUS	FRACTURE MECHANICS				•	•	
	FLUIDS			•	•	•	
	ELECTRIC CIRCUITS				•		
	OPTIMIZATION			•			
	ACOUSTIC CAVITIES			•			
	FATIGUE DAMAGE				•		

STRUCTURAL ANALYSIS ELEMENT/MATRIX LIBRARY

ELEMENT		EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
LINE ELEMENTS	ROD			•	•	•	
	BEAM	•	•	•	•	•	•
	TAPERED BEAM				•	•	
	OFFSET BEAM		•	•	•		•
	PINNED END BEAM	•	•	•			•
	CURVED BEAM					•	
FLAT MEMBRANES AND PLATES	3 NODE TRIANGLE	•	•	•	•	M	
	6 NODE TRIANGLE			M	•	M	C
	SHEAR PANEL			•			
	4 NODE QUAD	•	•	•	•	M	•
	8 NODE QUAD				S	M	S, C
CURVED SHELLS	3 NODE TRIANGLE					•	
	6 NODE TRIANGLE						C
	4 NODE QUAD					•	•
	8 NODE QUAD					•	S, C
	REDUCED THICK SHELL					•	

NOTES:

- M Membrane and/or plane strain only (no plate bending)
- S Includes sub-parametric forms with fewer nodes
- C Also includes cubic isoparametric element with two midside nodes

STRUCTURAL ANALYSIS
ELEMENT/MATRIX LIBRARY (continued)

ELEMENT			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
			PROGRAM					
AXI-SYMMETRIC ELEMENTS	SHELLS	CONICAL			•	•	D	
		CURVED			•		•	
	TRIANGULAR RINGS	3 NODE			•	•	•	D
		6 NODE			•	•	D	C
	QUAD RINGS	4 NODE			•	•	•	•
		8 NODE				S	•	S, C
SOLID ELEMENTS	TETRA-HEDRON	4 NODE		•	•	•	D	
	WEDGES	6 NODE	•	•	•	•	D	D
		15 NODE					•	C
	HEXA-HEDRONS	8 NODE	•	•	•	•	•	•
		20 NODE			S		•	S, C
PIPE ELEMENTS		STRAIGHT	•	•	•	•	•	
		ELBOW	•	•		•	•	
		TEE		•				

NOTES:

- **S** Includes subparametric forms with fewer nodes
- **C** Also includes cubic isoparametric element with two midside nodes
- **D** Degenerate case

STRUCTURAL ANALYSIS
ELEMENT/MATRIX LIBRARY (continued)

ELEMENT				EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
GENERAL STIFFNESS ELEMENTS		SPRING	K	1	•		•	•	•
		SCALAR SPRING				•			
		6 x 6 or 12 x 12 MATRIX			•	•	•		•
		GENERAL MATRIX				•			
MASSES	ELEMENT	LUMPED (DIAGONAL)		2	2	2	2		
		CONSISTENT				2	2	2	
	NON-STRUCTURAL	SCALAR (DOF)				•		•	
		NODAL		•	•	•	•		
		DISTRIBUTED				•			
		GUYAN REDUCTION			•	•	•		
		GENERAL MATRIX				•	•		
DAMPING		SCALAR				•			
		DASHPOT				•	•		
		DISCRETE VISCOUS $[C] = \alpha[K] + \beta[M]$		•		•	•	•	
		STRUCTURAL $(1 + ig)[K]$				•			
		MODAL VISCOUS		•	•	•		•	
		GENERAL MATRIX				•	•		
OTHER ELEMENTS		GAP					•	•	
		FRICTION					•	•	
		RIGID		3	•	•			
		REBAR SOLID						•	
		ELASTIC FOUNDATION					•	•	
		CRACK TIP					•	•	
		LAMINATED SHELL					•	•	
		PLOT ONLY				•			

The six right-hand columns are grouped under the heading PROGRAM.

NOTES:
1 See restraints
2 Generated from density
3 See constraints

HEAT TRANSFER—CONDUCTING ELEMENTS

ELEMENT			*PROGRAM					
			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
LINEAR					•	•	•	•
PLANAR	3 NODE TRIANGLE				•	•	•	P,C
	4 NODE QUAD				•	•	•	•
	8 NODE QUAD					S	•	S,C
	TRANSVERSE CONDUCTING SHELL					•		
AXISYMMETRIC	TRIANGULAR RING				•	•	•	P,C
	4 NODE QUAD RING				•	•	•	•
	8 NODE QUAD RING					S	•	S,C
SOLID	TETRAHEDRON				•	•		
	WEDGE				•	•	D	
	8 NODE BRICK				•	•	•	•
	15 NODE WEDGE						D	C
	20 NODE BRICK						•	S,C
GENERAL MATRIX INPUT					•			

NOTES:
- S Contains subparametric forms with fewer number of nodes
- P Also contains parabolic isoparametric element with one midside node
- C Also contains cubic isoparametric element with two midside nodes
- D Degenerate case

COORDINATE SYSTEMS AND MATERIAL PROPERTIES

FEATURE			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
COORDINATE SYSTEMS	BASIC (GLOBAL)	CARTESIAN	●	●	●	●	●	●
		CYLINDRICAL	●	●	●	●		●
		SPHERICAL			●	●		●
		GENERAL					1	
	SKEWED (LOCAL)	CARTESIAN	●	●	●	●	●	●
		CYLINDRICAL	●	●	●	●		●
		SPHERICAL			●	●		●
		GENERAL					1	
		MIXED	●	●	●	●	●	●
MATERIAL PROPERTIES		ISOTROPIC	●	●	●	●	●	●
		2-D ORTHOTROPIC		●	●	●	1	●
		3-D ORTHOTROPIC				●	1	●
		TEMPERATURE DEPENDENT	●		●	●	●	●
		STRESS DEPENDENT			●	●	●	
		TIME DEPENDENT				●	●	
		NONLINEAR ELASTIC					●	
	WORK HARDENING	ISOTROPIC				●	●	
		KINEMATIC				●	●	
		COMBINED				●	●	
		ORNL 10 CYCLE				●	●	
		GENERAL					1	

NOTES:

1 Performed by user subroutine

BOUNDARY CONDITIONS

FEATURE			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
LOADING	STATIC	CONCENTRATED	•	•	•	•	•	•
		DISTRIBUTED (BEAM)	•	•	•	•	•	•
		PRESSURE — PLATES/SHELLS	•	•	•	•	•	•
		PRESSURE — AXISYMMETRIC ELEMENTS			•	•	•	•
		PRESSURE — SOLIDS	•	•	•	•	•	•
		TEMPERATURE	•	•	•	•	•	•
		ACCELERATION	•	•	•	•	•	•
		ROTATIONAL VELOCITY	•	•	•	•	1	•
		COMBINATION	•	•	•	•		+
		AXI-SYMMETRIC — AXISYMMETRIC SHELLS			•	•		
		AXI-SYMMETRIC — AXISYMMETRIC RINGS				•		
	DYNAMIC	TIME DEPENDENT	•	•	•	•	•	
		FREQUENCY DEPENDENT		•	•	•		
		PSD RANDOM		•	•			
		SHOCK SPECTRUM	•	•		•		
DISPLACEMENT CONSTRAINTS		SINGLE POINT*	•	•	•	•	•	•
		MULTI POINT*	2		•	•	•	3
		SPECIFIED NONZERO DISPLACEMENT	•	•	•	•	•	•
HEAT TRANSFER		HEAT SOURCE/SINK			•	•	•	•
		CONVENTION			•	•	•	•
		RADIATION			•	•	•	
		SPECIFIED TEMPERATURE			•	•	•	•

NOTES: *Single point constraint ≡ enforced zero translation(s) and/or rotation(s) in coordinate(s) associated with a node point
Multi-point constraint ≡ enforced linear constraint relationships between translation(s) and/or rotation(s) which may be associated with different node points

1 Applies to some elements
2 Specialized forms of rigid and interface coupling
3 Displacement components set equal on different nodes
+ Stand alone program

PRE- AND POST-PROCESSING

		FEATURE	EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
PLOTTING	INPUT	UNDEFORMED GEOMETRY	♦	●	●	●	●	●
		NODE LABELS	♦	♦	●	●	●	●
		ELEMENT LABELS	♦		●	●	●	●
		PROPERTY LABELS			●	●		●
		2-D SECTIONS				●	●	
		BOUNDARY CONDITION LABELS	♦		●			
		HIDDEN LINES REMOVED				✓	♦	
	OUTPUT	DEFORMED GEOMETRY	♦	●	●	●	●	●
		CONTOURS 2D STRUCTURE		♦	●	●	●	●
		CONTOURS SOLID STRUCTURE				●	●	●
		TIME HISTORY	4	●	●	●	♦	
		FREQUENCY RESPONSE		●	●,4	●		
		POWER SPECTRAL DENSITY		●	●,4			
		ARBITRARY X VS. Y				●	♦	
DATA GENERATION		NODES	1	1,2	1,2,3	1,2	2,3	1,2
		ELEMENTS	1	1	1,2,3	1,2	2,3	1,2
		RESTRAINTS	1	1	1,2	1	2,3	1
		LOADS	1	1	2	1	2,3	1
OUTPUT SORTING		BY LOAD CASES		●	●	●	●	●
		BY ELEMENT	●		●			
		MAX/MIN SUMMARY	●	●	●			●
		SELECTED NODES AND/OR ELEMENTS		●	●	●	●	
BANDWIDTH MINIMIZATION			●	●	●	W	●,W	♦,W

NOTES:

1 Generates data in 1 "dimension"
2 Generates data in 2 "dimensions"
3 Generates data in 3 "dimensions"
4 Printer plots
♦ Stand alone program
W Wavefront solution

APPENDIX C

Subroutine Prolog

This appendix lists the prologs for all of the subroutines used in the transfer module. The prologs represent a standard description of the purpose of the routine, the calling arguments and their attributes, the routines that are called, any commons that are used and special comments about the subroutine.

PROLOGS

DIRECTORY OF SUBROUTINES/FUNCTIONS - SORTED ALPHABETICALLY BY ROUTINE NAME

FILE NAME	SUBROUTINE/ FUNCTION	PAGE NO
GE/TRANCITS	AREAD	17
GE/TRANCITS	CALCRD	12
GE/TRANCITS	CCNVEC	22
GE/TRANCITS	CHKPT	92
GE/TRANCITS	CHNODT	7
GE/TRANCITS	CINV	37
GE/TRANCITS	CKWIND	53
GE/TRANCITS	CORNCC	21
GE/TRANCITS	CTRAN	9
GE/TRANCITS	CWCRNT	31
GE/TRANCITS	DATINT	75
GE/TRANCITS	EINCOR	49
GE/TRANCITS	ELTRAN	98
GE/TRANCITS	EMINMX	40
GE/TRANCITS	ERRPRT	84
GE/TRANCITS	ETCORE	64
GE/TRANCITS	FCRNTP	34
GE/TRANCITS	FILREW	48
GE/TRANCITS	FNDELM	50
GE/TRANCITS	FNDJAC	94
GE/TRANCITS	FTIME	41
GE/TRANCITS	GENTMP	61
GE/TRANCITS	GETCRD	23
GE/TRANCITS	GETEWT	77
GE/TRANCITS	GETMPE	26
GE/TRANCITS	GETPD	13
GE/TRANCITS	GETWND	52
GE/TRANCITS	GNLSUR	46
GE/TRANCITS	GPCOEF	101
GE/TRANCITS	GPTEMP	102
GE/TRANCITS	GTGAUS	19
GE/TRANCITS	GTNUMT	78
GE/TRANCITS	HTFIO	58
GE/TRANCITS	HTFPAS	67
GE/TRANCITS	HTICON	69
GE/TRANCITS	I3DSF	87
GE/TRANCITS	INITCM	99
GE/TRANCITS	JACBCK	96
GE/TRANCITS	KNTSPF	43
GE/TRANCITS	MARCRD	70
GE/TRANCITS	MARCTO	100
GE/TRANCITS	MAXTDF	35
GE/TRANCITS	MKRGF	54
GE/TRANCITS	MPERW	27
GE/TRANCITS	NTCORE	65
GE/TRANCITS	PARD	16
GE/TRANCITS	PDOFSF	93
GE/TRANCITS	PRCGEO	79
GE/TRANCITS	PRCSOF	76
GE/TRANCITS	PREPGM	8
GE/TRANCITS	PRNTIM	39

GE/TRANCITS	PROFAC	56
GE/TRANCITS	PUTWFI	24
GE/TRANCITS	RDCNTL	59
GE/TRANCITS	RDSPF	47
GE/TRANCITS	RDTEMP	30
GE/TRANCITS	RDTIME	4
GE/TRANCITS	REAARY	89
GE/TRANCITS	REACRD	20
GE/TRANCITS	READR	71
GE/TRANCITS	REMELM	11
GE/TRANCITS	RFILCR	44
GE/TRANCITS	RWCCF	25
GE/TRANCITS	RWELMS	2
GE/TRANCITS	RWNODS	1
GE/TRANCITS	RWSPCF	97
GE/TRANCITS	SCTION	10
GE/TRANCITS	SETFLP	29
GE/TRANCITS	SFDIST	91
GE/TRANCITS	SHAPEF	15
GE/TRANCITS	SHAPFV	95
GE/TRANCITS	SINTIT	73
GE/TRANCITS	SKIP	57
GE/TRANCITS	SKPREC	72
GE/TRANCITS	SORT	68
GE/TRANCITS	SORT2	90
GE/TRANCITS	SSURCH	42
GE/TRANCITS	STSTMP	62
GE/TRANCITS	TEMFOR	85
GE/TRANCITS	TFLSET	60
GE/TRANCITS	TIMCK	6
GE/TRANCITS	TPNAST	86
GE/TRANCITS	UNPAKF	33
GE/TRANCITS	VOLM	14
GE/TRANCITS	WRCRNT	66
GE/TRANCITS	WRITMP	36
GE/TRANCITS	WRSTMP	63
GE/TRANCITS	WRTDIR	81
GE/TRANCITS	WRTEMP	83
GE/TRANCITS	WRTGEO	82
GE/TRANCITS	WRTNTF	80
GE/TRANCITS	ZERORF	45

```
C==================================================
      SUBROUTINE RWNODS(IREC,NN,XYZ,IRW)
C==================================================
CPR
CPR
CPR   ROUTINE--   RWNODS
CPR
CPR   DATE/WRITTEN BY--  8/04/82   RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS I/O ROUTINE READS AND WRITES THE NODAL INFO
CPR   TO THE INTERNAL DATA BASE FILES
CPR   THIS FILE IS NOW IFL3
CPR   IN THE WRITE MODE IT AUTOMATICALLY ZEROS THIS FILE
CPR   FOR NODES NAMES THAT ARE NOT USED
CPR   THIS RANDOM FILE(IFL3) COULD EASILY BE REPLACED BY
CPR   AN ARRAY IF ENOUGH CORE WAS AVAIALABEL
CPR
CPR   CALLING SEQUENCE-- CALL RWNODS(IREC,NN,XYZ,IRW)
CPR
CPR   CALLING ARGUMENTS--
CPR        NAME      ATTRIBUTES      DEFINITION
CPR       IREC          (I)          RECORD TO BE PROCESSED
CPR       NN            (I/O)        NODE NUMBER(NAME)
CPR       XYZ           (I/O) (3X1)  ARRAY OF COORDINATES
CPR       IRW           (I)          READ/WRITE SWITCH
CPR                                    IRW=0---- READ
CPR                                    IRW=1---- WRITE
CPR
CPR
CPR
CPR   FILES USED--
CPR            IFL3 --- RANDOM FILE TO STORE COORDINATES
CPR
CPR
CPR   COMMONS USED--
CPR           AFIL ---  FILE CODE COMMON
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED--
CPR        NONE
CPR
CPR   LIBRARIES ACCESSED--
CPR        NONE
CPR
CPR   LOCAL VARIABLES--
CPR        N/A
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR        NONE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS--
CPR        HOST = H6000
CPR
CPR       **********************************
      END
```

```
C==============================================================
      SUBROUTINE RWELMS(IREC,A,IRW,ICODE)
C==============================================================
CPR
CPR
CPR   ROUTINE--   RWELMS
CPR
CPR   DATE/WRITTEN BY--  8/04/82   RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR    THIS ROUTINE IS AN I/O ROUTINE
CPR    ITS PURPOSE IS TO READ/WRITE THE ELEMENT ATTRIBUTES TO THE
CPR    INTERNAL DATA BASE FILE
CPR    THIS DATA BASE FILE IS RANDOM FILE IFL4
CPR    THE ATTRIBUTES ARE
CPR                      -- ELEMENT NAME
CPR                      -- COORDINATES OF ALL 15 POINTS IN THE ELM
CPR                      -- ELEMENT CONNECTIVITY OF THE 8 CORNERS
CPR                      -- THE PACKED THTD FACE NUMBER
CPR                      -- VOLUME OF ELEMENT
CPR
CPR    THE ROUTINE CAN PROCESS DIFFERENT AMOUNTS OF THESE
CPR    ATTRIBUTES BASED ON THE VALUE OF ICODE
CPR    THE RANDOM FILE IS NOT INITIALIZED PRESENTLY WITH THIS
CPR    ROUTINE
CPR
CPR
CPR   CALLING SEQUENCE--  CALL RWELMS(IREC,A,IRW,ICODE)
CPR
CPR   CALLING ARGUMENTS--
CPR        NAME      ATTRIBUTES      DEFINITION
CPR       IREC      (I)             RECORD NUMBER TO BE PROCESSED
CPR       A         (I/O) (56X1)    ARRAY OF ATTRIBUTES
CPR       IRW       (I)             READ/WRITE SWITCH
CPR                                  IRW=0---- READ
CPR                                  IRW=1---- WRITE
CPR       ICODE     (I)     SWITCH TO SET AMOUNT OF DATA TO PROCESS
CPR            ICODE=0 -- PROCESS ALL DATA
CPR            ICODE=1 -- PROCESS ELM NAME ONLY
CPR            ICODE=2 -- PROCESS ELM NAME AND CORNER COORDINATES
CPR            ICODE=4 -- PROCESS EVERYTHING EXCEPT VOLUME
CPR            ICODE=5 -- PROCESS ALL DATA
CPR
CPR
CPR
CPR   FILES USED--
CPR    IFL4 -- RANDOM FILE TO STORE ELM ATTRIBUTES
CPR
CPR
CPR   COMMONS USED--
CPR    AFIL -- COMMON TO DEFINE FILECODES
CPR
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED--
CPR    NONE
CPR
CPR
CPR   LIBRARIES ACCESSED--
CPR    NONE
CPR
CPR   LOCAL VARIABLES--
CPR    N/A
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR    NONE
```

```
4350CPR
4360CPR
4370CPR   SPECIAL REMARKS/INSTRUCTIONS--
4380CPR        HOST = H6000
4390CPR
4400CPR      ************************************
4850      END
```

```
C==================================================
      SUBROUTINE RDTIME
C==================================================
CPR
CPR
CPR   ROUTINE--   RDTIME
CPR
CPR   DATE/WRITTEN BY--  8/18/82   RJ MAFFEO
CPR   DATE/REVISED BY--  9/13/83   SR MARTIN
CPR        REASON    --  CONVERT FROM F4BT FILES TO THE NEW
CPR                      HEAT XFER OUTPUT FILE FORM
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE READS THE TIME VALUES FROM THE HEAT XFER OUTPUT FILE
CPR   AND DETERMINES IF THEY ARE DESIRED FOR THIS RUN  IT
CPR   AUTOMATICALLY ELIMINATES THE NON-STP TIME STEPS  IT ALSO
CPR   CALLS THE REQUIRED ROUTINE TO PERFORM INITIAL CHECKS ON
CPR   TIMES AND ELM COUNTS
CPR   THE HEAT TRANSFR OUTPUT FILE IS REWOUND BEFORE RETURNING
CPR   FROM THIS ROUTINE
CPR
CPR
CPR   CALLING SEQUENCE-- CALL RDTIME
CPR
CPR   CALLING ARGUMENTS--
CPR        NAME     ATTRIBUTES      DEFINITION
CPR
CPR   FILES USED--
CPR
CPR    INTITS -- HEAT XFER OUTPUT FILE (TITAN INPUT)
CPR    IERF   -- ERROR REMARK FILE
CPR    INOT6  -- HARD COPY FILE
CPR
CPR
CPR   COMMONS USED--
CPR
CPR   AFIL     -- FILE CODES
CPR   SIZE     -- PROBLEM SIZE PARAMETERS
CPR   CNTLTM  -- TIME/TEMPERATURE CONTROL PARAMETERS
CPR
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED--
CPR
CPR   TIMCK
CPR   CHNODT
CPR
CPR
CPR   LIBRARIES ACCESSED--
CPR     NONE
CPR
CPR
CPR   LOCAL VARIABLES--
CPR
CPR   NSTPTM    -- COUNT OF STP TIMES PROCESSED
CPR   NCNTT     -- COUNT OF TOTAL TIMES PROCESSED
CPR   NTMDO     -- USER SPECIFIED NUMBER OF STP TIMES TO DO
CPR   TIMT      -- ARRAY OF TIMES PROCESSED  THIS ARRAY WILL
CPR             -- BE USED AS A DIRECTORY TO POINT TO THE
CPR             -- LOCATION ON THE HEAT XFER OUTPUT FILE FOR EACH TIME
CPR
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS--
```

```
5570CPR        HOST = H6000
5580CPR
5590CPR       ***************************
6640     END
```

```
C===========================================================
      SUBROUTINE TIMCK(TIME,ITIME)
C===========================================================
CPR
CPR
CPR     ROUTINE--   TIMCK
CPR
CPR     DATE/WRITTEN BY--  8/18/82   RJ MAFFEO
CPR     DATE/REVISED BY--
CPR
CPR     FUNCTION/PURPOSE--
CPR
CPR     THIS ROUTINE CHECKS TO SEE IF THE PRESENT TIME VALUE
CPR     IS ONE OF THE TIMES DESIRED
CPR
CPR
CPR     CALLING SEQUENCE--   CALL TIMCK(TIME,ITIME)
CPR
CPR     CALLING ARGUMENTS--
CPR          NAME     ATTRIBUTES     DEFINITION
CPR
CPR          TIME        (I)       PRESENT TIME VALUE OF TIME STEP
CPR          ITIME       (O)       DESIRED SWITCH
CPR                                  ITIME=0  >>> NOT DESIRED
CPR                                  ITIME=1  >>> DESIRED
CPR
CPR
CPR
CPR     FILES USED--   NONE
CPR
CPR     COMMONS USED--
CPR
CPR      CNTLTM   --- TIME/TEMPERATURE CONTROL PARAMATERS
CPR
CPR
CPR
CPR     FUNCTIONS/ROUTINES CALLED--  NONE
CPR
CPR     LIBRARIES ACCESSED--
CPR       NONE
CPR
CPR
CPR     LOCAL VARIABLES--
CPR
CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR     SPECIAL REMARKS/INSTRUCTIONS--
CPR          HOST = H6000
CPR
CPR        ********************************
      END
```

```
7860C==================================================
7880      SUBROUTINE CHNODT
7910C==================================================
7920CPR
7930CPR
7940CPR     ROUTINE--   CHNODT
7950CPR
7960CPR     DATE/WRITTEN BY--  8/18/82   RJ MAFFEO
7970CPR     DATE/REVISED BY--
7980CPR
7990CPR     FUNCTION/PURPOSE--
8000CPR
8010CPR     THIS ROUTINE MAKES TWO BASIC CHECKS ON THE INPUT
8020CPR     IT CHECKS TO SEE THAT THE NUMBER OF ELEMENTS IN EACH TIMESTEP
8030CPR     IS GREATER THAN OR EQUAL TO THE NUMBER OF GEOMETRY ELEMENTS
8040CPR     IT ALSO CHECKS TO INSURE THAT ALL ELMS ON THE
8050CPR     NEUTRAL ELM FILE HAVE TEMPERATURES ON THE THT
8060CPR     OUTPUT FILE
8070CPR
8080CPR
8090CPR     CALLING SEQUENCE--   CALL CHNODT
8100CPR
8110CPR     CALLING ARGUMENTS--
8120CPR          NAME     ATTRIBUTES      DEFINITION
8130CPR
8140CPR
8150CPR
8160CPR     FILES USED--
8170CPR
8180CPR     IFLO   -- THT OUTPUT FILE
8190CPR     IERF   -- ERROR REMARK FILE
8200CPR
8210CPR     COMMONS USED--
8220CPR
8230CPR         AFIL
8240CPR         WORK
8250CPR         SIZE
8260CPR         ELMDAT
8270CPR         CNTLFL
8280CPR
8290CPR
8300CPR
8310CPR     FUNCTIONS/ROUTINES CALLED--
8320CPR
8330CPR       RDMANY
8340CPR       RWELMS
8350CPR
8360CPR     LIBRARIES ACCESSED--
8370CPR       NONE
8380CPR
8390CPR
8400CPR     LOCAL VARIABLES--
8410CPR
8420CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE
8430CPR
8440CPR
8450CPR     SPECIAL REMARKS/INSTRUCTIONS--
8460CPR          HOST = H6000
8470CPR
8480CPR      ******************************
9510      END
```

```
C================================================
      SUBROUTINE PREPGM
C=========================================================
CPR
CPR
CPR     ROUTINE--   PREPGM
CPR
CPR     DATE/WRITTEN BY--  8/18/82   RJ MAFFEO
CPR     DATE/REVISED BY--
CPR
CPR     FUNCTION/PURPOSE--
CPR
CPR     THIS ROUTINE CALLS ALL ROUTINES NEEDED TO
CPR     TRANSFORM,SECTION (WINDOW) AND/OR
CPR     ELIMINATE UNWANTED ELMS FROM THE PROBLEM
CPR
CPR
CPR     CALLING SEQUENCE--   CALL PREPGM
CPR
CPR     CALLING ARGUMENTS--   NONE
CPR          NAME      ATTRIBUTES       DEFINITION
CPR          NONE         NONE              NONE
CPR
CPR     FILES USED--    NONE
CPR
CPR     COMMONS USED--
CPR
CPR       SIZE
CPR       ELMDAT
CPR       CNTLGM
CPR
CPR
CPR
CPR     FUNCTIONS/ROUTINES CALLED--
CPR
CPR       RWELMS
CPR       CTRAN
CPR       SECTION
CPR       REMELM
CPR
CPR
CPR     LIBRARIES ACCESSED--
CPR       NONE
CPR
CPR
CPR     LOCAL VARIABLES--
CPR
CPR       JSLC  -- WINDOW SWITCH
CPR                 JSLC=1  >>> ELM IS IN WINDOW
CPR                 JSLC=-1 >>> ELM IS NOT IN WINDOW
CPR
CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR     SPECIAL REMARKS/INSTRUCTIONS--
CPR          HOST = H6000
CPR
CPR        ********************************
      END
```

```
C=============================================================
      SUBROUTINE CTRAN(XO,YO,ZO,C)
C=============================================================
CPR
CPR
CPR     ROUTINE--   CTRAN
CPR
CPR     DATE/WRITTEN BY--  8/18/82   RJ MAFFEO
CPR     DATE/REVISED BY--
CPR
CPR     FUNCTION/PURPOSE--
CPR
CPR     THIS ROUTINE TRANSFORMS THE COORDINATES OF THE EIGHT
CPR     CORNER POINTS OF EACH ELM
CPR     TRANSFORMS ARE LIMITED TO TRANSLATIONS
CPR
CPR
CPR     CALLING SEQUENCE--   CALL CTRAN(XO,YO,ZO,C)
CPR
CPR     CALLING ARGUMENTS--
CPR          NAME      ATTRIBUTES      DEFINITION
CPR         XO        (I)             X OFFSET
CPR         YO        (I)             Y OFFSET
CPR         ZO        (I)             Z OFFSET
CPR         C         (I/O) ARRAY     COORDINATE ARRAY
CPR
CPR
CPR
CPR     FILES USED--   NONE
CPR
CPR     COMMONS USED--   NONE
CPR
CPR
CPR     FUNCTIONS/ROUTINES CALLED--   NONE
CPR
CPR     LIBRARIES ACCESSED--
CPR       NONE
CPR
CPR
CPR     LOCAL VARIABLES--
CPR
CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR     SPECIAL REMARKS/INSTRUCTIONS--
CPR          HOST = H6000
CPR
CPR       *********************************
      END
```

```
C=====================================================
      SUBROUTINE SCTION(C,XMN,YMN,ZMN,XMX,YMX,ZMX,INOT)
C======================================================
CPR
CPR
CPR    ROUTINE--   SCTION
CPR
CPR    DATE/WRITTEN BY--  8/18/82   RJ MAFFEO
CPR    DATE/REVISED BY--
CPR
CPR    FUNCTION/PURPOSE--
CPR
CPR    THIS ROUTINE CHECKS TO SEE IF ANY OF THE CORNER
CPR    COORDINATES ARE OUTSIDE OF THE WINDOW SPECIFIED
CPR    BY THE MIN AND MAX VALUES OF X,Y,Z
CPR    NOTE IF ANY POINT OF AN ELM IS OUTSIDE THE WINDOW
CPR    THE ELM IS CONSIDERED OUT OF THE WINDOW.
CPR
CPR
CPR
CPR    CALLING ARGUMENTS--
CPR         NAME     ATTRIBUTES      DEFINITION
CPR
CPR         C        (I)            COORDINATE ARRAY
CPR         XMN >
CPR         YMN >>   (I)            MIN VALUE OF X,Y,Z FOR WINDOW
CPR         ZMN >
CPR         XMX >
CPR         YMX >>   (I)            MAX VALUE OF X,Y,Z FOR WINDOW
CPR         ZMX >
CPR
CPR         INOT     (O)            IN/OUT SWITCH
CPR                            INOT=1  >> ELM IS IN THE WINDOW
CPR                            INOT=-1 >> ELM IS OUT OF WINDOW
CPR
CPR
CPR
CPR    FILES USED--       NONE
CPR
CPR    COMMONS USED--    NONE
CPR
CPR
CPR    FUNCTIONS/ROUTINES CALLED--  NONE
CPR
CPR    LIBRARIES ACCESSED--
CPR      NONE
CPR
CPR
CPR    LOCAL VARIABLES--
CPR
CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR    SPECIAL REMARKS/INSTRUCTIONS--
CPR         HOST = H6000
CPR
CPR    **********************************
      END
```

```
C=============================================
      SUBROUTINE REMELM
C=====================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          REMELM
CPR
CPR   DATE/WRITTEN BY-- 09/13/82      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB ELIMINATES UNDESIRED ELMS FROM THE RANDOM
CPR   ELM GEOM FILE
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       ELMDAT
CPR       CNTLGM
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       RWELMS
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=================================================
      END
```

```
C=============================================================
      SUBROUTINE CALCRD
C=============================================================
CPR
CPR
CPR    ROUTINE--   CALCRD
CPR
CPR    DATE/WRITTEN BY--  8/08/82   RJ MAFFEO
CPR    DATE/REVISED BY--
CPR
CPR    FUNCTION/PURPOSE--
CPR
CPR     THE PURPOSE OF THIS ROUTINE IS TO CALL ALL
CPR     REQUIRED ROUTINES TO COMPUTE THE COORDINATES OF THE
CPR     CENTROID OF THE ELMS AND THE COORDINATES OF THE
CPR     FACE CENTERS IT THEN CALLS THE ELM I/O ROUTINE TO STORE
CPR     THIS INFO INTO THE INTERNAL DATA BASE
CPR
CPR
CPR    CALLING SEQUENCE--   CALL CALCRD
CPR
CPR    CALLING ARGUMENTS--     NONE
CPR
CPR         NAME     ATTRIBUTES      DEFINITION
CPR         NONE       NONE             NONE
CPR
CPR    FILES USED--
CPR
CPR      IERF     -- ERROR REMARK FILE
CPR
CPR    COMMONS USED--
CPR
CPR     AFIL     -- FILE CODE COMMON
CPR     ELMDAT   -- ELM ATTRIBUTE COMMON
CPR     SIZE     -- PROBLEM SIZE COMMON
CPR     WORK     -- SCRATCH WORK SPACE COMMON
CPR     CRNDEF  -- LOCAL COMMON TO COORD CALC  ROUTINES
CPR
CPR    FUNCTIONS/ROUTINES CALLED--
CPR
CPR     GETPD
CPR     RWELMS
CPR     VOLM
CPR     AREAD
CPR     REACRD
CPR
CPR    LIBRARIES ACCESSED--
CPR      NONE
CPR
CPR    LOCAL VARIABLES--
CPR
CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR    MANY OF THE ROUTINES CALLED BY CALCRD ARE ALSO USED
CPR    IN SIGMA THTD DECK GENERATOR THEY CAN COMPUTE MUCH MORE THAN
CPR    JUST COORDINATES(SUCH AS VOLUMES AREAS AND DELTA X).
CPR
CPR
CPR
CPR    SPECIAL REMARKS/INSTRUCTIONS--
CPR         HOST = H6000
CPR
CPR      ********************************
      END
```

```
C==================================================
      SUBROUTINE GETPD(P3D,P2D)
C=====================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          GETPD
CPR
CPR   DATE/WRITTEN BY-- 09/13/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB COMPUTES AND STORES ALL OF THE PARTIAL DERIVATIVES
CPR   OF THE SHAPE FUNCTIONS.BOTH 3D AND 2D
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       P3D       (O)            3D PARTIAL DERIVATIVES
CPR       P2D       (O)            2D PARTIAL DERIVATIVES
CPR
CPR   COMMONS USED --
CPR
CPR       CRNDEF  --- SEE CALCRD
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       PARD
CPR       GTGAUS
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
      END
```

```
C======================================================
      SUBROUTINE VOLM(X,Y,Z,GP,P3D,VOL,XC,YC,ZC,Q,IVER)
C============================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--           VOLM
CPR
CPR   DATE/WRITTEN BY-- 09/13/82      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB DOES NUMERICAL INTEGRATION FOR VOL,XC,YC,ZC
CPR   IT COMPUTES DETJ AND THE JACOBIAN
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       X          (I)            ARRAY OF X COORDS
CPR       Y          (I)              '    '  Y   '
CPR       Z          (I)              '    '  Z   '
CPR       GP         (I)            ARRAY OF GAUSS POINTS
CPR       P3D        (I)            3D PARTIAL DERIVATIVES
CPR       VOL        (O)            VOLUME
CPR       XC         (O)            X COORD OF CENTROID
CPR       YC         (O)            Y        '
CPR       ZC         (O)            Z        '
CPR       Q          (O)            JACOBIAN
CPR       IVER       (O)            IVER=1 > ZERO VOLUME ELM
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       SHAPEF
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C=====================================================
      SUBROUTINE SHAPEF(X,Y,Z,TETA,YNETA,ZETA,XP,YP,ZP)
C=====================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          SHAPEF
CPR
CPR   DATE/WRITTEN BY-- 09/14/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR  THIS SUB COMPUTES THE VALUES OF X,Y,Z AT THE GUASS POINTS
CPR  BASED ON LINEAR 3D SHAPE FUNCTIONS
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       X         (I) >
CPR       Y         (I) >>        ARRAYS OF COORDINATES
CPR       Z         (I) >
CPR       TETA      (I) >
CPR       YNETA     (I) >>        LOCAL COORDINATES
CPR       ZETA      (I) >
CPR       XP        (O) >
CPR       YP        (O) >>        DESIRED GLOBAL COORDINATES
CPR       ZP        (O) >
CPR
CPR   COMMONS USED --
CPR
CPR       CRNDEF
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
17970C=====================================================
17990      SUBROUTINE PARD(TETA,YNETA,ZETA,IGP,IND,PD)
18010C=====================================================
18030CPR
18040CPR===================================================
18050CPR
18060CPR
18070CPR   ROUTINE--         PARD
18080CPR
18090CPR   DATE/WRITTEN BY-- 09/14/82     RJ MAFFEO
18100CPR   DATE/REVISED BY--
18110CPR
18120CPR   FUNCTION/PURPOSE--
18130CPR
18140CPR   THIS SUB COMPUTES THE DERIVATIVES OF THE
18150CPR   SHAPE FUNCTIONS WRT THE LOCAL SYSTEM
18160CPR   BOTH FOR VOLUMES(3D) AND AREAS(2D)
18170CPR
18180CPR
18190CPR
18200CPR   CALLING ARGUMENTS --
18210CPR       NAME      ATTRIBUTES     DEFINITION
18220CPR
18230CPR       TETA      (I) >
18240CPR       YNETA     (I) >>         LOCAL COORDINATES
18250CPR       ZETA      (I) >
18260CPR       IGP       (I)            GAUSS POINT COUNTER
18270CPR       IND       (I)            2D/3D INDICATOR
18280CPR                                IND=0 >> 3D
18290CPR                                IND=1 THRU 6 >> 2D IND=FACE NUMBER
18300CPR       PD        (O)            ARRAY OF DERIVATIVES
18310CPR
18320CPR   COMMONS USED --
18330CPR
18340CPR       CRNDEF
18350CPR
18360CPR   FUNCTIONS/ROUTINES CALLED --
18370CPR
18380CPR
18390CPR   FILES USED --
18400CPR
18410CPR
18420CPR   LIBRARIES ACCESSED --
18430CPR   NONE
18440CPR
18450CPR
18460CPR   LOCAL VARIABLES --
18470CPR
18480CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
18490CPR
18500CPR
18510CPR   SPECIAL REMARKS/INSTRUCTIONS --
18520CPR      HOST = H6000
18530CPR
18540CPR
18550CPR===================================================
18950      END
```

```
C================================================
      SUBROUTINE AREAD(X,Y,Z,P2D,Q,ARA,DX2,
C================================================
CPR
CPR===================================================
CPR
CPR
CPR   ROUTINE--          AREAD
CPR
CPR   DATE/WRITTEN BY-- 09/14/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB COMPUTES THE FACE AREAS AND DELTA X
CPR   FOR 3D THT ELMS
CPR   NOTE IT COMPUTES DELTA X TWO WAYS
CPR   IDX=1      DX=DIST FROM ELM CENTROID TO FACE CENTROID
CPR   IDX=2    . DX=PERPENDICULAR DIST FROM ELM CENT TO FACE
CPR   IDX=3      SWITCH TO JUST CALC CENTROID OF FACE 5
CPR
CPR   IT IS BEING USED HERE TO COMPUTE THE COORDINATES
CPR   OF THE FACE CENTERS
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       X         (I) >
CPR       Y         (I) >>         ARRAY OF CORNER COORDS
CPR       Z         (I) >
CPR       P2D       (I)            DERIVATIVES OF SHAPE FUNCTIONS
CPR       Q         (O)            JACOBIAN
CPR       ARA       (O)            ARRAY OF FACE AREAS
CPR       DX2       (O)            ARRAY OF FACE DELTA XS
CPR       IDX       (I)            DELTA X COMPUTATION INDICATOR
CPR       XC        (I) >
CPR       YC        (I) >>         COORDS  OF VOLUME CENTROID
CPR       ZC        (I) >
CPR       XA        (O) >
CPR       YA        (O) >>         ARRAYS OF COORDS FOR FACE CENTERS
CPR       ZA        (O) >
CPR
CPR   COMMONS USED --
CPR
CPR       CRNDEF
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       GTGAUS
CPR       SHAPEF
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
```

```
19680CPR      HOST = H6000
19690CPR
19700CPR
19710CPR==================================================
20780       END
```

```
C============================================
      SUBROUTINE GTGAUS(IND,I1,I2,I3,IP)
C============================================
CPR
CPR================================================
CPR
CPR
CPR   ROUTINE--          GTGAUS
CPR
CPR   DATE/WRITTEN BY-- 09/14/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB GETS APPROPRIATE GAUSS POINTS
CPR   FOR EACH FACE OF THE 3D BRICK
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       IND        (I)            FACE INDICATOR
CPR       I1         (O) >
CPR       I2         (O) >
CPR       I3         (O) >>       GAUSS POINT COUNTERS
CPR       IP         (O) >
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
      END
```

```
C==================================================
      SUBROUTINE REACRD(CRD,XC,YC,ZC,XA,YA,ZA)
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          REACRD
CPR
CPR   DATE/WRITTEN BY-- 09/15/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB REARRANGES THE FACE CENTER COORDINATE ARRAYS
CPR   AND THE VOLUME CENTROIDS INTO THE CRD ARRAY
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       CRD        (O)            BRICK COORD ARRAY
CPR       XC         (I)  >
CPR       YC         (I)  >>        VOLUME CENTROIDS
CPR       ZC         (I)  >
CPR       XA         (I)  >
CPR       YA         (I)  >>        FACE CENTER ARRAYS
CPR       ZA         (I)  >
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C=================================================
      SUBROUTINE CORNCC
C=================================================
CPR
CPR================================================
CPR
CPR
CPR   ROUTINE--          CORNCC
CPR
CPR   DATE/WRITTEN BY-- 11/10/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR     THIS SUB CALLS ALL ROUTINES NEEDED TO COMPUTE
CPR     AND STORE THE INFORMATION ASSOCIATED WITH THE
CPR     CORNER NODE WEIGHTING COEFFICIENTS
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR          NONE
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       ELMDAT
CPR       SIZE
CPR       WORK
CPR       CNTLGM
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       RWELMS
CPR       SORT
CPR       CCNVEC
CPR       GETCRD
CPR       CINV
CPR       PUTWFI
CPR       RWCCF
CPR
CPR   FILES USED --
CPR
CPR       IERF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
      END
```

```
C=====================================================
      SUBROUTINE CCNVEC(NOD,IRW,NLOC,NUMA,MCON,MXNCON,NSIZW)
C=====================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          CCNVEC
CPR
CPR   DATE/WRITTEN BY-- 11/10/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR     THIS SUB WRITES TO (CREATES) AND READS FROM THE
CPR     NODE -CONNECTION VECTOR (MCON)
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       NOD       (I)            NODE NUMBER
CPR       IRW       (I)            READ/WRITE SWITCH(O-READ,1-WRITE)
CPR       NLOC      (I/O)          CURRENT CONNECTION TO NODE
CPR       NUMA      (O)            MAX # CONNECTIONS TO NODE
CPR       MCON      (I/O)          PACKED CONNECTION ARRAY
CPR       MXNCON    (O)            MAX # OF CONNECTIONS TO ANY NODE
CPR       NSIZW     (I)            SIXE OF WORK ARRAY (MCON HERE)
CPR
CPR   COMMONS USED --
CPR                   AFIL
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IERF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR    THE NODE-CONNECTION VECTOR (MCON) IS KEYED
CPR    TO THE NODE NAME AND IS COMPOSED OF THE MAX
CPR    CONNECTION TO THAT NODE AND THE CURRENT CONNECTION TO
CPR    THAT NODE IN THE FORM --
CPR
CPR       MCON(NOD)=1000*MAX CONNECTION + CURRENT CONNECTION
CPR
CPR    THE INITIAL CURRENT COUNT IS EQUAL TO THE MAX CONNECTION.
CPR    AS THE CONNECTIONS ARE PROCESSED THE CURRENT COUNT IS
CPR    REDUCED BY ONE
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR     HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
C==============================================
      SUBROUTINE GETCRD(NE,CRD,BRD,XCN,YCN,ZCN,JREC)
C==============================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--         GETCRD
CPR
CPR   DATE/WRITTEN BY-- 11/11/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB EXTRACTS THE APPROPRIATE SUBSET OF
CPR   COORDINATES FOR EACH CORNER NODE FORM THE
CPR   FULL SET OF COORDS OF EACH ELM
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       NE          (I)           ELMEMENT NAME
CPR       CRD         (I)           FULL SET OF COORDS
CPR       BRD         (O)           SUBSET OF COORDS
CPR       XCN         (O)  >>>
CPR       YCN         (O)  >>>>>>>>> COORDS OF CORNER
CPR       ZCN         (O)  >>>
CPR       JREC        (I)           CORNER LABEL
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IERF      S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR   THE ARRARY IDX IS USED TO DEFINE THE RELATIONSHIP
CPR   BETWEEN THE CORNER POINT AND THE APPROPRIATE 3 FACES
CPR   FOR EXAMPLE  CORNER POINT 1 IS CONNECTED TO FACES
CPR        1,4,5
CPR   THEREFORE IDX(1,1)=1,IDX(1,2)=4,IDX(1,3)=5
CPR
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
27360C==============================================
27380      SUBROUTINE PUTWFI(NOD,NE,NUMA,JREC,XSF,NPOS,ACWF)
27400C=======================================================
27420CPR
27430CPR===================================================
27440CPR
27450CPR
27460CPR   ROUTINE--          PUTWFI
27470CPR
27480CPR   DATE/WRITTEN BY-- 11/11/82     RJ MAFFEO
27490CPR   DATE/REVISED BY--
27500CPR
27510CPR   FUNCTION/PURPOSE--
27520CPR
27530CPR   THIS SUB POSITIONS THE WEIGHTING COEFF
27540CPR   INFO INTO CORRECT LOCATIONS OF THE ACWF ARRAY
27550CPR
27560CPR
27570CPR
27580CPR
27590CPR
27600CPR   CALLING ARGUMENTS --
27610CPR       NAME      ATTRIBUTES     DEFINITION
27620CPR
27630CPR       NOD       (I)            CORNER NODE NUMBER
27640CPR       NE        (I)            ELM NUMBER
27650CPR       NUMA      (I)            MAX NUMBER OF CONNECTIONS
27660CPR       JREC      (I)            PACKED CORNER - FACE #
27670CPR       XSF       (I)            WEIGHTING COEFFS
27680CPR       NPOS      (I)            STARTING POSITION FOR INFO
27690CPR       ACWF      (O)            ARRAY CONTAINING INFO
27700CPR
27710CPR   COMMONS USED --
27720CPR
27730CPR       AFIL
27740CPR
27750CPR   FUNCTIONS/ROUTINES CALLED --
27760CPR
27770CPR
27780CPR   FILES USED --
27790CPR
27800CPR
27810CPR   LIBRARIES ACCESSED --
27820CPR   NONE
27830CPR
27840CPR
27850CPR   LOCAL VARIABLES --
27860CPR
27870CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
27880CPR
27890CPR
27900CPR   SPECIAL REMARKS/INSTRUCTIONS --
27910CPR      HOST = H6000
27920CPR
27930CPR
27940CPR===================================================
28260      END
```

```
C=============================================
      SUBROUTINE RWCCF(NOD,ACWF,NPOS,IRW)
C=============================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          RWCCF
CPR
CPR   DATE/WRITTEN BY-- 11/11/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE READS AND WRITES THE CORNER NODE
CPR   COEFF INFO TO AND FROM THE DATA BASE
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       NOD        (I/O)          NODE NAME (RECORD #)
CPR       ACWF       (I/O)          ARRAY OF INFO
CPR       NPOS       (I)            # OF TERMS TO BE READ/WRITTEN
CPR       IRW        (I)            READ/WRITE SWITCH(O-READ,1-WRITE)
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IRCCF    R
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C=============================================
      SUBROUTINE GETMPE(MP,NWDMPE)
C=============================================
CPR
CPR================================================
CPR
CPR
CPR   ROUTINE--          GETMPE
CPR
CPR   DATE/WRITTEN BY-- 09/19/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE READS THE FIRST TIME STEP TEMPERATURE
CPR   SOLUTION AND EXTRACTS THE ELM NAME AND RELATIVE POSITION
CPR   OF THIS ELM IN THE OUTPUT LIST.IT THEN CALLS MPERW TO
CPR   STORE THIS INFO INTO THE MP ARRAY
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       MP        (I/O)          ELM MATRIX POSITION ARRAY
CPR       NWDMPE    (I)            # OF WORDS IN MP ARRAY
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       SIZE
CPR       CNTLFL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       TFLSET
CPR       HTFIO
CPR       MPERW
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
      END
```

```
C=====================================================
      SUBROUTINE MPERW(MP,NE,NPOS,IRW)
C=====================================================
CPR
CPR=================================================
CPR
CPR
CPR   ROUTINE--          MPERW
CPR
CPR   DATE/WRITTEN BY-- 10/08/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE PACKS AND UNPACKS THE INFO NEEDED TO RELATE
CPR   THE ELEMENT NAME TO THE RELATIVE POSITION OF THE ELEMENT
CPR   IN THE OUTPUT LIST
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       MP         (I/O)            MATRIX POSITION ARRAY
CPR       NE         (I)              ELM NAME
CPR       NPOS       (I/O)            RELATIVE POSITION IN OUTPUT
CPR       IRW        (I)              SWITCH(0--READ,1--WRITE)
CPR
CPR   COMMONS USED --  NONE
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       MOD  (FORTRAN SUPPLIED)
CPR
CPR   FILES USED --
CPR
CPR       06        S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR   THIS ROUTINE USES DECIMAL PACKING TO PACK 2 POSITIONS
CPR   PER WORD OF THE MP ARRAY  IT MAKES USE OF THE FORTRAN
CPR   SUPPLIED 'MOD' FUNCTION
CPR   THE POSITIONS ARE PACKED AS FOLLOWS --
CPR      POSITION 1  >>>   LEFT HAND SIDE OF WORD
CPR      POSITION 2  >>>   RIGHT HAND SIDE OF WORD
CPR   THE MAX POSITION THAT CAN BE PACKED IS NPOS=9999
CPR   THE ADVANTAGE OF THIS SCHEME IS THAT THE DIMENSION OF
CPR   MP NEED ONLY BE 4000 TO STORE ELMS UP TO 8000
CPR
CPR   FOR EXAMPLE
CPR    IF THE POSITION OF ELM 1 IS 987
CPR    AND THE POSITION OF ELM 2 IS 79
CPR    THEN THE VALUE STORED IN MP(1) IS 09870079
CPR    NOTE THAT MP(1) CONTAINS THE POSITIONS OF BOTH
CPR    ELMS 1 AND 2
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
```

```
30960CPR
30970CPR=====================================================
31620       END
```

```
C=================================================
      SUBROUTINE SETFLP(NNEW,NOLD)
C=================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--           SETFLP
CPR
CPR   DATE/WRITTEN BY-- 09/19/83      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE SETS THE H T  INPUT FILE TO THE CORRECT
CPR   POSITION TO READ THE SOLUTION DESIRED
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       NNEW      (I)            FILE DESIRED
CPR       NOLD      (I)            CURRENT FILE COUNT
CPR                                OF THT OUTPUT FILE
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       TFLSET
CPR       HTFPAS
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR====================================================
      END
```

```
C================================================
      SUBROUTINE RDTEMP(TEMP,NFLVAL)
C================================================
CPR
CPR===================================================
CPR
CPR
CPR   ROUTINE--          RDTEMP
CPR
CPR   DATE/WRITTEN BY-- 09/19/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE READS THE TEMPERATURE INFO FOR THE DESIRED
CPR   TIME STEP FROM THE THT OUTPUT FILE AND STORES IT IN
CPR   THE TEMP ARRAY
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       TEMP       (O)            TEMPER  STORAGE ARRAY
CPR       NFLVAL     (I)            CURRENT FILE COUNT
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       SIZE
CPR       CNTLFL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       HTFIO
CPR
CPR   FILES USED --
CPR
CPR       IERF      S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR===================================================
      END
```

```
C==================================================
      SUBROUTINE CWCRNT(MP,TEMP,TIMVAL,NNODMX,NNODMN,TCRIT,MXNCON)
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          CWCRNT
CPR
CPR   DATE/WRITTEN BY-- 09/19/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE COMPUTES A UNIQUE TEMPERATURE FOR EACH
CPR   CORNER OF THE HEAT TRANSFER ELM BASED ON THE INDIVIDUAL
CPR   TEMPERATURES FOR EACH ELM CONNECTION
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES      DEFINITION
CPR
CPR       MP         (I)             ELM MATRIX POSITION ARRAY
CPR       TEMP       (I)             ELM TEMPERATURE ARRAY
CPR       TIMVAL     (I)             CURRENT VALUE OF TIME
CPR       NNODMX     (I)             MAX NODE NAME
CPR       NNODMN     (I)             MIN NODE NAME
CPR       TCRIT      (I)             TEMPERATURE USED IN MAXTDF TO
CPR                                  DETERMINE IF CORNER TEMPERATURE
CPR                                  DIFFERENCE SHOULD BE OUTPUT
CPR       MXNCON     (I)             MAX NUMBER CONNECTIONS TO
CPR                                  ANY NODE
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       RWCCF
CPR       UNPAKF
CPR       MPERW
CPR       FCRNTP
CPR       MAXTDF
CPP       WRITMP
CPR       FILREW
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
```

```
35450       END
```

```
C=====================================================
      SUBROUTINE UNPAKF(NPACK,NAME,IFACE)
C=====================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          UNPAKF
CPR
CPR   DATE/WRITTEN BY-- 11/29/82      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE UNPACKS THE CORNER-FACE NUMBER INTO AN CORNER
CPR   NUMBER(NAME) AND THE FACE ARRAY(IFACE)
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES      DEFINITION
CPR
CPR       NPACK      (I)             PACKED CORNER-FACE #
CPR       NAME       (O)             CORNER NAME
CPR       IFACE      (O)             FACE INFO ARRAY
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IERF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR   NOTE  PACK CORNER-FACE # = CORNER NAME*1000000+IFACPC
CPR         WHERE IFACPC IS A SINGLE INTEGER THAT CONTAINS ALL
CPR         ALL THE FACE NUMBERS
CPR   FOR EXAMPLE
CPR      IF THE FACE NUMBERS WERE 6,5,4,3,2,1
CPR      THEN IFACPC=654321
CPR
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
C==================================================
      SUBROUTINE FCRNTP(TEMP,MPOS,IFACE,NCORN,CTP,COF)
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          FCRNTP
CPR
CPR   DATE/WRITTEN BY-- 11/29/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE EXTRACTS THE 7 TEMPERATURES ASSC  WITH A
CPR   CONNECTING ELM IT THEN FINDS THE 4 TEMPS ASSC  WITH
CPR   THE CORNER OF INTEREST AND DOES THE MULTIPLICATION
CPR   TO COMPUTE THE TEMPERATURE FOR THIS CORNER BASED ON
CPR   THIS ELM CONNECTION
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       TEMP      (I)            ELM TEMPER  ARRAY
CPR       MPOS      (I)            MATRIX POS  OF THE ELM
CPR       IFACE     (I)            ELM FACE ARRAY
CPR       NCORN     (I)            CORNER NUMBER
CPR       CTP       (O)            CORNER TEMPERATURE
CPR       COF       (I)            CORNER COEFF  ARRAY
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C==================================================
      SUBROUTINE MAXTDF(NEL,CRNT,NCON,NDONE,TIMVAL,TCRIT,IREC)
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--           MAXTDF
CPR
CPR   DATE/WRITTEN BY-- 11/30/82      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE COMPUTES THE MAX DIFFERENCE BETWEEN
CPR   ALL THE POSSIBLE CORNER TEMPERATURES AT A CORNER
CPR   AND FLAGS IT IF IT IS GREATER THAN TCRIT
CPR
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       NEL        (I)         ARRAY OF ELMS CONNECTED TO THIS NODE
CPR       CRNT       (I)         ARRAY OF TEMPS FOR THIS NODE
CPR       NCON       (I)         NUMBER OF ELM CONNECTED TO THIS NODE
CPR       NDONE      (I)         PAGE HEADER COUNTER
CPR                              (=O WRITE HEADER, NOT = O DONT WRITE)
CPR       TIMVAL     (I)         CURRENT VALUE OF TIME
CPR       TCRIT      (I)         CRITICAL TEMP FOR PRINTOUT
CPR                              IF MAX DIFF < TCRIT -- DONT WRITE
CPR                              IF MAX DIFF > TCRIT -- WRITE
CPR       IREC       (I)         RECORD NUMBER(NODE NUMBER)OF CORNER
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       SORT
CPR
CPR   FILES USED --
CPR
CPR       INOT6    S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR     HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
39180C================================================
39200       SUBROUTINE WRITMP(TBAR,TIMVAL,NOD,NDONE)
39220C================================================
39240CPR
39250CPR==================================================
39260CPR
39270CPR
39280CPR    ROUTINE--          WRITMP
39290CPR
39300CPR    DATE/WRITTEN BY-- 11/30/82     RJ MAFFEO
39310CPR    DATE/REVISED BY--
39320CPR
39330CPR    FUNCTION/PURPOSE--
39340CPR
39350CPR    THIS ROUTINE WRITES THE VALUES OF THE TEMPERATURES AT THE
39360CPR    STRESS POINTS TO HARD COPY AND TO THE OUTPUT FILE
39370CPR
39380CPR
39390CPR
39400CPR
39410CPR
39420CPR    CALLING ARGUMENTS --
39430CPR        NAME       ATTRIBUTES     DEFINITION
39440CPR
39450CPR        TBAR       (I)            TEMPERATURE AT THE STRESS POINT
39460CPR        TIMVAL     (I)            CURRENT VALUE OF TIME
39470CPR        NOD        (I)            STRESS POINT NAME(NODE #)
39480CPR        NDONE      (I)            SWITCH
39490CPR                                  =0      -- WRITER PAGE HEADER
39500CPR                                  NOT 0   -- DO NOT WRITE HEADER
39510CPR                                  =-1     -- FLUSH BUFFERS
39520CPR
39530CPR    COMMONS USED --
39540CPR
39550CPR        AFIL
39560CPR
39570CPR    FUNCTIONS/ROUTINES CALLED --
39580CPR
39590CPR
39600CPR    FILES USED --
39610CPR
39620CPR        INOT6    S
39630CPR        IOUTF    S
39640CPR
39650CPR    LIBRARIES ACCESSED --
39660CPR    NONE
39670CPR
39680CPR
39690CPR    LOCAL VARIABLES --
39700CPR
39710CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE
39720CPR
39730CPR
39740CPR    SPECIAL REMARKS/INSTRUCTIONS --
39750CPR       HOST = H6000
39760CPR
39770CPR
39780CPR==================================================
40580       END
```

```
C=====================================================
      SUBROUTINE CINV(CD,XSP,YSP,ZSP,XSF,NE,JCN)
C=====================================================
CPR
CPR=====================================================
CPR
CPR
CPR ROUTINE--   CINV
CPR
CPR DATE/WRITTEN BY--  9/10/82   RJ MAFFEO
CPR DATE/REVISED BY--  01/12/83  RJ MAFFEO
CPR
CPR FUNCTION/PURPOSE--
CPR   THIS SUB FIND THE VALUES OF THE SHAPE FUNCTIONS FOR
CPR   ANY POINT NEAR A TETRAHEDRAL ELEMENT THE SHAPE FUNCTIONS
CPR   ARE BASED ON THE SIMPLEX TETRAHEDRAL ELEMENT
CPR   IT FIRST FINDS THE INVERSE OF THE COORDINATE MATRIX (-C-)
CPR   IT THEN THENS MULTIPLIES THIS INVERSE INTO THE COORD
CPR   VECTOR (1 X Y Z) TO FIND THE SHAPE FUNCTIONS
CPR   THIS ROUTINE CAN BE USED TO COMPUTE THE WEIGHTING
CPR   COEFF  FOR THE CORNERS OF A THTD ELM BASED ON THE TEMPERATURES
CPR   AT THE CENTER OF THE FACES AND THE CENTROID
CPR
CPR CALLING ARGUMENTS--
CPR        NAME     ATTRIBUTES       DEFINITION
CPR        CD       (I) 12X1 ARRAY    COORDINATE ARRAY
CPR                                   X,Y,Z COORDS OF CORNERS OF TETRA
CPR        XSP      (I)
CPR        YSP      (I)
CPR        ZSP      (I)               COORDS OF THE POINT FOR WHICH
CPR                                   THE SHAPE FUNCTIONS ARE DESIRED
CPR        XSF      (O) 5X1 ARRAY     VECTOR OF SHAPE FUNCTIONS
CPR         NE      (I)               ELMEMENT NAME
CPR        JCN      (I)               CORNER NUMBER
CPR
CPR
CPR
CPR FILES USED--
CPR     AFIL
CPR
CPR COMMONS USED--
CPR     NONE
CPR
CPR
CPR ROUTINES CALLED --
CPR     NONE
CPR
CPR LIBRARIES ACCESSED--
CPR     NONE
CPR
CPR LOCAL VARIABLES--
CPR     INX      18X1     THIS VECTOR IS USED TO SELECT THE
CPR                       CORRECT INDICES OF THE C MATRIX FOR
CPR                       EACH ELEMENT OF THE C INVERSE MATRIX
CPR     CI       4X4      INVERSE OF THE COORD MATRIX(-C-)
CPR     C        5X4      2D MATRIX CONTAINING CORRDS OF TETRA
CPR
CPR
CPR SPECIAL COMMENTS ABOUT THIS ROUTINE --
CPR
CPR     NOTE THAT THE COORD MATRIX(-C-) IS AUGMENTED WITH
CPR     A ROW OF ONES THIS ROW OF ONES IS USED IN THE COMPUTATION
CPR     OF THE INVERSE
CPR
CPR     NOTE XSF(5)=1/DIST
CPR          DIST=DISTANCE FROM CENTROID TO CORNER
CPR
```

```
41300CPR     NOTE A LOCAL COORD SYSTEM IS ESTABLISHED FOR EACH ELM
41310CPR          THIS SYSTEM IS FORMED AS Q(I)-QC
41320CPR          WHERE Q(I) IS THE COORDS OF THE VERTICES
41330CPR          AND QC IS THE COORDS OF THE CENTROID OF THE ELM
41340CPR
41350CPR
41360CPR
41370CPR SPECIAL REMARKS/INSTRUCTIONS--
41380CPR      HOST = H6000
41390CPR
41400CPR*****************************************************************************
42720        END
```

```
42740CC=================================================
42760      SUBROUTINE PRNTIM(ISTOP,I1,TB,TE)
42930      END
```

```
C===================================================
      SUBROUTINE EMINMX(XC,YC,ZC,CADD,XYZV,NEMNMX,NE,NNPE)
C===================================================
CPR
CPR=====================================================
CPR
CPR
CPR  ROUTINE--          EMINMX
CPR
CPR  DATE/WRITTEN BY-- 04/05/83     RJ MAFFEO
CPR  DATE/REVISED BY--
CPR
CPR  FUNCTION/PURPOSE--
CPR
CPR  THIS ROUTINE FINDS THE MIN AND MAX XYZ COORDS
CPR  OF THE HEAT TRANSFER ELEMENT AND STORES THIS INFO
CPR  INTO THE XYZV ARRAY
CPR  IT ALSO KEEPS TRACK OF THE ELMS WITH THE SMALLEST
CPR  AND LARGEST XYZ COORDS AND STORES THEM IN THE
CPR  NEMNMX ARRAY
CPR
CPR
CPR  CALLING ARGUMENTS --
CPR      NAME       ATTRIBUTES     DEFINITION
CPR
CPR      XC        (I) >>
CPR      YC        (I) >>>>  COORD ARRAYS FOR HT ELM VERTICES
CPR      ZC        (I) >>
CPR      CADD      (I)       CONSTANT COORD ADDER
CPR      NE        (I)       CURRENT ELM NAME
CPR      NNPE      (I)       # OF NODES PER ELM
CPR
CPR  COMMONS USED --
CPR
CPR
CPR  FUNCTIONS/ROUTINES CALLED --
CPR
CPR      SORT
CPR
CPR  FILES USED --
CPR
CPR
CPR  LIBRARIES ACCESSED --
CPR  NONE
CPR
CPR
CPR  LOCAL VARIABLES --
CPR
CPR  SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR  SPECIAL REMARKS/INSTRUCTIONS --
CPR     HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
C====================================================
      SUBROUTINE FTIME(A1)
C====================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          FTIME
CPR
CPR   DATE/WRITTEN BY-- 04/05/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE CALLS THE APPROPRIATE SYSTEM ROUTINE
CPR   TO MEASURE THE PROCESSOR TIME
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       A1          (O)          PROCESSOR TIME
CPR
CPR   COMMONS USED --      NONE
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       PTIME  -- HONEY WELL TIMER
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C==================================================
      SUBROUTINE SSURCH
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          SSURCH
CPR
CPR   DATE/WRITTEN BY-- 04/05/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE IS THE MINI-EXEC FOR THE STRESS POINT
CPR   TO HEAT TRANSFER ELM SEARCH ROUTINES IT CALLS ALL
CPR   REQUIRED ROUTINES TO FIND WHICH HT ELM CONTAINS
CPR   EACH STRESS POINT AND THEN FINDS THE WEIGHTING COEFFS
CPR   FOR THAT STRESS POINT
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       SIZE
CPR       SPDAT
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       KNTSPF
CPR       RFILCR
CPR       ZERORF
CPR       GNLSUR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C=================================================
      SUBROUTINE KNTSPF(NUMSP,MXNSP)
C=================================================
CPR
CPR=================================================
CPR
CPR
CPR   ROUTINE--          KNTSPF
CPR
CPR   DATE/WRITTEN BY-- 04/05/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE COUNTS THE NUMBER OF STRESS POINTS
CPR   TO BE INTERFACED FROM THE STRESS NODE FILE AND
CPR   ALSO FINDS THE LARGEST STRESS NODE NAME
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       NUMSP     (O)            # OF STRESS NODES
CPR       MXNSP     (O)            LARGEST STRESS NODE NAME
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       FILREW
CPR       RDSPF
CPR
CPR   FILES USED --
CPR
CPR       IERF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=================================================
      END
```

```
C==================================================
      SUBROUTINE RFILCR
C==================================================
CPR
CPR===================================================
CPR
CPR
CPR   ROUTINE--          RFILCR
CPR
CPR   DATE/WRITTEN BY-- 04/06/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE SIZES ALL THE RANDOM FILES
CPR   IT CALLS THE APPROPRIATE SYSTEM ROUTINE TO DEFINE
CPR   THE WIDTH OF THE FILE
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       RANSIZ  -- HONEYWELL ROUTINE TO SET WIDTH OF FILE
CPR       DEFINE FILE -- IBM ROUTINE TO SET WIDTH
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C=================================================
      SUBROUTINE ZERORF(IFC,MXN,NWD,A)
C=================================================
CPR
CPR=================================================
CPR
CPR
CPR   ROUTINE--          ZERORF
CPR
CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE INITIALIZES RANDOM FILES
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       IFC       (I)        FILECODE
CPR       MXN       (I)        MAX RECORD NUMBER
CPR       NWD       (I)        # OF WORDS IN -A- ARRAY
CPR       A         (I)        INITIALIZEING ARRAY
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IFC       R
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=================================================
      END
```

```
C============================================
      SUBROUTINE GNLSUR
C============================================
CPR
CPR===================================================
CPR
CPR
CPR   ROUTINE--          GNLSUR
CPR
CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE IS CALL BY THE SEARCH MINI-EXEC (SSURCH)
CPR   IT INITIATES THE SEARCH THRU THE HEAT TRANSFER ELMS
CPR   TO FIND WHICH ELM CONTAINS EACH STRESS POINT
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       SIZE
CPR       WORK
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       EINCOR
CPR       RDSPF
CPR       FNDELM
CPR
CPR   FILES USED --
CPR
CPR       INOT6    S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR===================================================
      END
```

```
C================================================
      SUBROUTINE RDSPF(NN,X,Y,Z,IERC)
C================================================
CPR
CPR================================================
CPR
CPR
CPR   ROUTINE--         RDSPF
CPR
CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE READS THE STRESS POINT COORD INFO
CPR   FROM THE STRESS POINT NODE FILE
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       NN         (O)            NODE NAME
CPR       X          (O)            X COORD
CPR       Y          (O)            Y COORD
CPR       Z          (O)            Z COORD
CPR       IERC       (O)            ERROR CODES
CPR                               0 -- NO ERRORS
CPR                               1 -- END OF FILE
CPR                               2 -- READ ERROR
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       ISPNF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
      END
```

```
C==================================================
      SUBROUTINE FILREW(IFC)
C==================================================
CPR
CPR=================================================
CPR
CPR
CPR   ROUTINE--          FILREW
CPR
CPR   DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE REWINDS FILES
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       IFC        (I)            FILE CODE
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=================================================
      END
```

```
C==================================================
        SUBROUTINE EINCOR(ENX,NELMN,NELMX)
C==================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          EINCOR
CPR
CPR   DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   COORDS INTO CORE FROM THE RANDOM ELM FILE
CPR   THE DATA STORED IN CORE IS --
CPR       ELM NUMBER
CPR       XMIN
CPR       YMIN
CPR       ZMIN
CPR       XMAX
CPR       YMAX
CPR       ZMAX
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES      DEFINITION
CPR
CPR       ENX        (O)           ARRAY CONTAINING WINDOW INFO
CPR       NELMN      (I)           MIN ELM NAME
CPR       NELMX      (I)           MAX ELM NAME
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       ELMDAT
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       RWELMS
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
        END
```

```
C==============================================
      SUBROUTINE FNDELM(NSP,XP,YP,ZP,IFOND)
C==============================================
CPR
CPR=================================================
CPR
CPR
CPR   ROUTINE--          FNDELM
CPR
CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE DOES THE ACTUAL SEARCH THRU THE HEAT
CPR   TRANSFER ELMS TO FIND WHICH ELM CONTAINS THE STRESS
CPR   POINT
CPR   IT CALLS THE REQUIRED ROUTINES TO GET THE ELM
CPR   COORDINATES,CHECKS TO SEE IF THE STRESS POINT
CPR   IS WITHIN THE ELM AND IF IT IS COMPUTES
CPR   AND WRITES THE WEIGHTING COEFFS FOR THIS STRESS
CPR   POINT WITH RESPECT TO THE VERTICES OF THE
CPR   HEAT TRANSFER ELEMENT
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       NSP       (I)         STRESS NODE NAME
CPR       XP        (I) >>
CPR       YP        (I) >>>>    STRESS POINT COORDS
CPR       ZP        (I) >>
CPR       IFOND     (O)         INDICATOR
CPR                              0 -- DID NOT FIND CONTAINING ELM
CPR                              1 -- DID     FIND CONTAINING ELM
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       ELMDAT
CPR       SIZE
CPR       SPDAT
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       CKWIND
CPR       GETWND
CPR       I3DSF
CPR       RWSPCF
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
```

```
52720CPR=====================================================
54420       END
```

```
C=====================================================
      SUBROUTINE GETWND(IDONE,IWHR)
C=====================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--         GETWND
CPR
CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   OF THE HEAT TRANSFER ELMS AND THE COORDS OF THE
CPR   CORNERS OF THE ELM
CPR   IT CAN GET THIS INFO FROM FILE OR FROM CORE
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       IDONE      (I)       SEQUENTIAL POINTER TO HT ELM DATA
CPR       IWHR       (I)       INDICATOR
CPR                             1 -- GET ONLY WINDOW COORDS
CPR                             2 -- GET CORNER COORDS
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       ELMDAT
CPR       SIZE
CPR       WORK
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       RWELMS
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR   THE COORD DATA IS PASSED BACK TO THE ROUTINE -FNDELM-
CPR   VIA THE COMMON CALLED ELMDAT IN THE ARRAY AED
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
55750C================================================
55770       SUBROUTINE CKWIND(XP,YP,ZP,V,INOT)
55790C================================================
55810CPR
55820CPR===================================================
55830CPR
55840CPR
55850CPR   ROUTINE--          CKWIND
55860CPR
55870CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
55880CPR   DATE/REVISED BY--
55890CPR
55900CPR   FUNCTION/PURPOSE--
55910CPR
55920CPR   THIS ROUTINE CHECKS TO SEE IF THE COORDS OF THE
55930CPR   STRESS POINT ARE WITHIN THE WINDOW OF THE HEAT TRANSFER
55940CPR   ELEMENT
55950CPR
55960CPR
55970CPR
55980CPR   CALLING ARGUMENTS --
55990CPR       NAME      ATTRIBUTES     DEFINITION
56000CPR
56010CPR       XP >>
56020CPR       YP >>>>    (I)      COORDS OF STRESS POINT
56030CPR       ZP >>
56050CPR                           COORDS OF HEAT TRAN ELM
56060CPR       INOT       (O)      IN - OUT INDICATOR
56070CPR                             O -- OUT OF WINDOW
56080CPR                             1 -- IN  WINDOW
56090CPR
56100CPR   COMMONS USED --
56110CPR
56120CPR
56130CPR   FUNCTIONS/ROUTINES CALLED --
56140CPR
56150CPR
56160CPR   FILES USED --
56170CPR
56180CPR
56190CPR   LIBRARIES ACCESSED --
56200CPR   NONE
56210CPR
56220CPR
56230CPR   LOCAL VARIABLES --
56240CPR
56250CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
56260CPR
56270CPR
56280CPR   SPECIAL REMARKS/INSTRUCTIONS --
56290CPR      HOST = H6000
56300CPR
56310CPR
56320CPR=================================================
56610       END
```

```
      SUBROUTINE MKRGF
C=======================================================
CPR
CPR
CPR   ROUTINE--   MKRGF
CPR
CPR   DATE/WRITTEN BY--  8/04/82    RJ MAFFEO
CPR   DATE/REVISED BY--  4/04/83    RJ MAFFEO
CPR        REASON --     ADDED MIN MAX WINDOW LOGIC
CPR   DATE/REVISED BY--  8/30/83    SR MARTIN
CPR        REASON -- CHANGED I/O TO READ NEW INPUT FILE FORM
CPR
CPR   FUNCTION/PURPOSE--
CPR    THIS ROUTINE DOES THE PROCESSING OF THE NEUTRAL
CPR    NODE AND ELEMENT FILES INTO THE INTERNAL DATA BASE FILES
CPR    FOR THIS VERSION OF 3D TITS.
CPR    IT CALLS RWNODS TO PROCESS THE NODAL INFO.
CPR    IT CALLS RWELMS TO PROCESS THE ELM INFO
CPR    IT CALLS PROFAC TO PROCESS THE FACE INFORMATION
CPR
CPR
CPR   CALLING SEQUENCE--  CALL MKRGF
CPR
CPR   CALLING ARGUMENTS--  NONE
CPR        NAME      ATTRIBUTES      DEFINITION
CPR
CPR
CPR   FILES USED--
CPR    IFL3  -- RANDOM FILE TO STORE THE NODAL COORDINATES
CPR    IFL4  -- RANDOM FILE TO STORE THE ELM ATTRIBUTES
CPR    INTITS -- HEAT TRANSFER OUTPUT FILE
CPR    INOT6 -- OUTPUT FILE FOR MESSAGES
CPR    IERF  -- ERROR REMARK FILE
CPR
CPR
CPR   COMMONS USED--
CPR    AFIL     -- FILECODE COMMON
CPR    ELMDAT  -- ELM ATTRIBUTE COMMON
CPR    SIZE    -- SIZE VARIABLES COMMON
CPR    CNTLFL  -- INFO WHICH DESCRIBES ALL ELEMENTS
CPR
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED--
CPR    RWNODS
CPR    HEADEL
CPR    RWELMS
CPR    READPS
CPR    PROFAC
CPR    EMINMX
CPR
CPR
CPR   LIBRARIES ACCESSED--
CPR     NONE
CPR
CPR
CPR   LOCAL VARIABLES--
CPR    NNODMX     --  MAXIMUM NODE NAME
CPR    NNODMN     --  MINIMUM NODE NAME
CPR    NUMNOD     --  NUMBER OF NODES
CPR    NELMX      --  MAXIMUM ELM NAME
CPR    NELMN      --  MINIMUM ELM NAME
CPR    NUME       --  NUMBER OF ELEMENTS
CPR    CRD(45)    --  ARRARY OF COORDINATES FOR THE 15 POINTS IN A ELM
CPR    IFACPC    --  PACKED THTD FACE NUMBERS
CPR    IE(8)      --  ARRAY OF CONNECTIVITY FOR ELM CORENER POINTS
CPR    AED(56)    --  ELM ATTRIBUTE ARRAY
```

```
57300CPR                THIS ARRAY CONTAINS
57310CPR                NE,CRD,IE,IFACPC,VOL
57320CPR   NE       --  ELM NAME
57330CPR   VOL      --  VOLUME OF THE ELM
57340CPR   XYZ(3)   --  COORDINATE ARRAY FOR EACH NODE
57350CPR
57360CPR
57370CPR  SPECIAL COMMENTS ABOUT THIS ROUTINE
57380CPR   NOTE THE 15 POINTS FOR EACH ELM ARE
57390CPR   8 CORNERS , 6 MIDSIDES AND 1 CENTROID
57400CPR
57410CPR
57420CPR  SPECIAL REMARKS/INSTRUCTIONS--
57430CPR       HOST = H6000
57440CPR
57450CPR        ***********************************
58690     END
```

```
      SUBROUTINE PROFAC
C=========================================================
CPR
CPR
CPR     ROUTINE--   PROFAC
CPR
CPR     DATE/WRITTEN BY--  8/04/82   RJ MAFFEO
CPR     DATE/REVISED BY--
CPR
CPR     FUNCTION/PURPOSE--
CPR      THIS ROUTINE READS THE PACKED THTD FACE NUMBER FROM THE
CPR      END OF THE NEUTRAL ELEMENT FILE AND STORES THEM ON
CPR      THE INTERNAL DATA BASE FILE
CPR      THIS DATA BASE FILE IS RANDOM FILE IFL4,
CPR      THE ELEMENTAL ATTRIBUTE FILE
CPR      NOTE THE PACKED FACE NUMBERS EXIST ON THE HEAT
CPR      TRANSFER OUTPUT FILE
CPR      IN RECORDS OF THE FORM- <ELMT #> <FACE INFO>
CPR
CPR
CPR     CALLING SEQUENCE--   CALL PROFAC
CPR
CPR     CALLING ARGUMENTS--  NONE
CPR          NAME      ATTRIBUTES       DEFINITION
CPR
CPR
CPR     FILES USED--
CPR      IFL4 -- RANDOM ELEMENT ATTRIBUTE FILE
CPR      INTITS -- HEAT TRANSFER OUTPUT FILE (INPUT TO TITAN)
CPR      IERF -- ERROR REMARK FILE
CPR
CPR
CPR     COMMONS USED--
CPR      AFIL   -- FILECODE COMMON
CPR      ELMDAT -- ELEMENT ATTRIBUTE COMMON
CPR      SIZE   -- SIZE OF PROBLEM VARIABLES IN THIS COMMON
CPR
CPR
CPR
CPR     FUNCTIONS/ROUTINES CALLED--
CPR      RWELMS
CPR
CPR
CPR     LIBRARIES ACCESSED--
CPR      NONE
CPR
CPR     LOCAL VARIABLES--
CPR     NRED      --- NUMMBER OF NUMBERS READ FROM NEF FOR ONE SET OF
CPR                   ELEMENT FACE DEFINITIONS (LIMIT IS 200)
CPR     IFACPC  --- PACKED FACE NUMBER
CPR     NCNT      --- TOTAL NUMBER OF ELMS THAT HAVE FACES DEFINED
CPR
CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR     SPECIAL REMARKS/INSTRUCTIONS--
CPR          HOST = H6000
CPR
CPR               ********************************
      END
```

```
C==================================================
      SUBROUTINE SKIP (IFC,ISETS)
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          SKIP
CPR
CPR   DATE/WRITTEN BY-- 08/31/83          SR MARTIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR      THIS ROUTINE ALLOWS THE CALLING ROUTINE TO SKIP PAST
CPR      UNNEEDED TIMESTEP AND TEMPERATURE DATA
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR      IFC        INPUT,INTEGER   FILE CODE
CPR      ISETS      INPUT,INTEGER   NUMBER OF SETS TO SKIP OVER
CPR
CPR   COMMONS USED --
CPR       CNTLFL
CPR       SIZE
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       FC       S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR       DUMMY (CHARACTER) USED TO DUMMY READ LINES IN THE FILE
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C==================================================
      SUBROUTINE HTFIO (IFC,DATA,ICNT,ISTAT)
C==================================================
CPR==================================================
CPR
CPR
CPR   ROUTINE--       HTFIO
CPR
CPR   DATE/WRITTEN BY-- 09/13/83     SR MARTIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE READS INFO FROM THE HEAT TRANSFER INPUT FILE
CPR
CPR   CALLING ARGUMENTS --
CPR    IFC (INPUT/INTEGER) FILE CODE OF FILE TO READ FROM
CPR    DATA (OUTPUT/REAL) ARRAY OF DATA PASSED BACK
CPR    ICNT (INPUT/INTEGER) AMOUNT OF DATA IN 'DATA' ARRAY
CPR    ISTAT (OUTPUT/INTEGER) RETURN STATUS, VALUES ARE--
CPR                     0-OK, 1-END OF FILE, 2-READ ERROR
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IFC      S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
61280      SUBROUTINE RDCNTL
61290CPR
61300CPR=====================================================
61310CPR
61320CPR
61330CPR   ROUTINE--          RDCNTL
61340CPR
61350CPR   DATE/WRITTEN BY-- 09/13/83     SR MARTIN
61360CPR   DATE/REVISED BY--
61370CPR
61380CPR   FUNCTION/PURPOSE--
61390CPR      READ USER HEADER INFORMATION AND FILE COUNTERS
61400CPR
61410CPR
61420CPR
61430CPR
61440CPR   CALLING ARGUMENTS --
61450CPR       NAME       ATTRIBUTES     DEFINITION
61460CPR
61470CPR
61480CPR   COMMONS USED --
61490CPR
61500CPR       AFIL
61510CPR       CNTLFL
61520CPR       CNTLTM
61530CPR       SIZE
61540CPR
61550CPR   FUNCTIONS/ROUTINES CALLED --
61560CPR
61570CPR       HTFIO
61580CPR
61590CPR   FILES USED --
61600CPR
61610CPR       INTITS   S
61620CPR       INOT6    S
61630CPR
61640CPR   LIBRARIES ACCESSED --
61650CPR   NONE
61660CPR
61670CPR
61680CPR   LOCAL VARIABLES --
61690CPR
61700CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
61710CPR
61720CPR
61730CPR   SPECIAL REMARKS/INSTRUCTIONS --
61740CPR      HOST = H6000
61750CPR
61760CPR
61770CPR=====================================================
62640      END
```

```
      SUBROUTINE TFLSET
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--         TFLSET
CPR
CPR   DATE/WRITTEN BY-- 09/14/83     SR MARTIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR       THIS ROUTINE REWINDS THE HEAT TRANSFER FILE AND REPOSITIONS
CPR       THE FILE POINTER TO THE FIRST RECORD OF THE TIME/TEMP
CPR       INFORMATION
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       CNTLTM
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       INTITS   S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR     HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
C==================================================
      SUBROUTINE GENTMP
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          GENTMP
CPR
CPR   DATE/WRITTEN BY-- 09/26/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE IS THE MINI-EXEC USED TO CALL ALL
CPR   ROUTINES ASSOCIATED WITH COMPUTING THE FINAL VALUES
CPR   OF TEMPERATURE FOR EACH STRESS NODE
CPR   IT CAN DEAL WITH BOTH FINITE DIFFERENCE AND
CPR   FINITE ELEMNT HEAT TRANSFER CODES
CPR   IF THE HEAT TRANSFER CODE IS FINITE DIFFERENCE
CPR   THE TEMPERATURES OF THE CORNERS OF THE HEAT TRANSFER
CPR   ELEMENTS ARE FOUND VIA CALLS FROM THIS RUOTINE
CPR
CPR   CALLING ARGUMENTS --   NONE
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       MAFCOM
CPR       AFIL
CPR       MAFCOM
CPR       SIZE
CPR       MAFCOM
CPR       WORK
CPR       MAFCOM
CPR       CNTLTM
CPR       CNTLFL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       GETMPE
CPR       SETFLP
CPR       RDTEMP
CPR       CWCRNT
CPR       ETCORE
CPR       NTCORE
CPR       STSTMP
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C==================================================
      SUBROUTINE STSTMP(TEMP,TIMVAL)
C==================================================
CPR
CPR===================================================
CPR
CPR
CPR    ROUTINE--          STSTMP
CPR
CPR    DATE/WRITTEN BY-- 09/28/83     RJ MAFFEO
CPR    DATE/REVISED BY--
CPR
CPR    FUNCTION/PURPOSE--
CPR
CPR    THIS ROUTINE READS THE COEFF INFO FOR EACH STRESS POINT
CPR    (VIA CALL RWSPCF) GETS THE CORNER TEMPS FROM THE TEMP ARRAY
CPR    AND COMPUTES THE TEMPERATURE OF THE STRESS POINT BY
CPR    SUMMING THE COEFFS TIMES THE CORNER TEMPS
CPR
CPR    CALLING ARGUMENTS --
CPR        NAME      ATTRIBUTES     DEFINITION
CPR
CPR        TEMP       (I)      ARRAY CONTAINING TEMPS OF HEAT
CPR                            TRANSFER CORNERS
CPR        TIMVAL     (I)      TIME STEP VALUE
CPR
CPR    COMMONS USED --
CPR
CPR        MAFCOM
CPR        AFIL
CPR        MAFCOM
CPR        SIZE
CPR        SPDAT
CPR
CPR    FUNCTIONS/ROUTINES CALLED --
CPR
CPR        RWSPCF
CPR        WRSTMP
CPR
CPR    FILES USED --
CPR
CPR
CPR    LIBRARIES ACCESSED --
CPR    NONE
CPR
CPR
CPR    LOCAL VARIABLES --
CPR
CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR    SPECIAL REMARKS/INSTRUCTIONS --
CPR       HOST = H6000
CPR
CPR
CPR===================================================
      END
```

```
C================================================
      SUBROUTINE WRSTMP(NAME,TEMP,TIMVAL,NDONE)
C================================================
CPR
CPR================================================
CPR
CPR
CPR   ROUTINE--         WRSTMP
CPR
CPR   DATE/WRITTEN BY-- 09/28/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE WRITES THE STRESS POINT TEMPERATURES
CPR   TO THE OUTPUT FILE (IN NEUTRAL TEMPERATURE FILE FORM)
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR       ----      ----------     ----------
CPR       NAME      (I)      STRESS POINT LABEL
CPR       TEMP      (I)      TEMPERATURE OF STRESS POINT
CPR       TIMVAL    (I)      TIME VALUE OF STEP
CPR       NDONE     (I)      # OF NODES WRITTEN
CPR
CPR   COMMONS USED --
CPR
CPR       MAFCOM
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED -- NONE
CPR
CPR
CPR   FILES USED --
CPR
CPR       IOUTF    S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
      END
```

```
C=================================================
      SUBROUTINE ETCORE(ETMP)
C=================================================
CPR
CPR===================================================
CPR
CPR
CPR   ROUTINE--         ETCORE
CPR
CPR   DATE/WRITTEN BY-- 09/29/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE TRANSFERS THE FINITE DIFFERENCE ELEMENTAL
CPR   CORNER TEMPERATURES FROM SCRATCH FILE TO CORE
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       ETMP        (O)         ARRAY CONTAINING ELM CORNER TEMPS
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       SIZE
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       FILREW
CPR
CPR   FILES USED --
CPR
CPR       ICRNTF   S
CPR       IERF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR     HOST = H6000
CPR
CPR
CPR===================================================
      END
```

```
C==================================================
      SUBROUTINE NTCORE(TEMP)
C==================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--         MAIN
CPR
CPR   DATE/WRITTEN BY-- 09/19/83     SR MARTIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR       THIS ROUTINE ASSUMES THE FILE POINTER IS POINTING AT THE
CPR       FIRST NODE NUMBER, TEMPERATURE PAIR OF A
CPR       TIMESTEP  IT READS IN EACH PAIR AND STORES THE
CPR       TEMPERATURE (FROM DATA(2)) INTO THE 'TEMP' ARRAY
CPR       IN POSITION <NODE NUMBER> (IE--DATA(1))
CPR       THE RESULT BEING THAT, FOR SAY NODE 13, THE TEMPERATURE
CPR       WOULD BE IN RECORD 13 OF THE 'TEMP' ARRAY
CPR       UPON RETURN, THE FILE POINTER WILL POINT
CPR       TO THE NEXT TIMESTEP VALUE
CPR
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR       TEMP      REAL,OUTPUT     ARRAY WHICH WILL CONTAIN
CPR                                 TEMPERATURE VALUES  NOTE--
CPR                                  TEMP(I) WILL CONTAIN THE
CPR                                 TEMPERATURE VALUE FOR NODE 'I'
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       SIZE
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       HTFIO
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR       DATA (REAL,ARRAY) USED TO RETRIEVE THE NODE NUMBER,
CPR           TEMPERATURE PAIR FORM THE INPUT FILE
CPR       NODNUM (INTEGER) USED TO STORE THE NODE NUMBER
CPR           AS AS INTEGER AFTER RETRIEVING IT VIA THE
CPR         'DATA' ARRAY.
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
69220C==================================================
69240      SUBROUTINE WRCRNT(TBAR,NODE)
69260C==================================================
69280CPR
69290CPR==================================================
69300CPR
69310CPR
69320CPR   ROUTINE--          WRCRNT
69330CPR
69340CPR   DATE/WRITTEN BY-- 09/29/83     RJ MAFFEO
69350CPR   DATE/REVISED BY--
69360CPR
69370CPR   FUNCTION/PURPOSE--
69380CPR
69390CPR   THIS ROUTINE WRITES THE FINITE DIFFERENCE CORNER
69400CPR   TEMPS TO THE SCRATCH FILE.
69410CPR
69420CPR   CALLING ARGUMENTS --
69430CPR       NAME       ATTRIBUTES     DEFINITION
69440CPR
69450CPR       TBAR        (I)        VALUE OF TEMPERATURE
69460CPR       NODE        (I)        H T  ELM CORNER NAME
69470CPR
69480CPR   COMMONS USED --
69490CPR
69500CPR       AFIL
69510CPR
69520CPR   FUNCTIONS/ROUTINES CALLED --
69530CPR
69540CPR
69550CPR   FILES USED --
69560CPR
69570CPR       ICRNTF   S
69580CPR
69590CPR   LIBRARIES ACCESSED --
69600CPR   NONE
69610CPR
69620CPR
69630CPR   LOCAL VARIABLES --
69640CPR
69650CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
69660CPR
69670CPR
69680CPR   SPECIAL REMARKS/INSTRUCTIONS --
69690CPR      HOST = H6000
69700CPR
69710CPR
69720CPR==================================================
69930      END
```

```
C==================================================
      SUBROUTINE HTFPAS(IFC,NFP,IFILP)
C==================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--         HTFPAS
CPR
CPR   DATE/WRITTEN BY-- 09/29/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE SKIPS PAST PORTIONS OF THE
CPR   HEAT TRANSFER INPUT FILE
CPR   IT CURRENTLY ONLY CAN SKIP PAST THE TIME-TEMPERATURE
CPR   PORTION(IFILP=6)
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       IFC       (I)      FILE CODE OF FILE TO BE SKIPED
CPR       NFP       (I)      # OF PARTITIONS TO SKIP
CPR       IFILP     (I)      PORTION OF FILE(DATA TYPE)TO BE SKIPPED
CPR                          IFILP=1 THRU 5  NOT CURRENTLY USED
CPR                          IFILP=6 SKIP TIME-TEMPERATURE RECORDS
CPR
CPR   COMMONS USED --
CPR
CPR       SIZE
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IFC      S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
70860C==================================================
70880      SUBROUTINE SORT(SEEDS,LAX,NDUM,IKEY)
70900C==================================================
70910CPR
70920CPR====================================================
70930CPR
70940CPR
70950CPR   ROUTINE--          SORT
70960CPR
70970CPR   DATE/WRITTEN BY-- ????????     GE SORT ROUTINE
70980CPR   DATE/REVISED BY--
70990CPR
71000CPR   FUNCTION/PURPOSE--
71010CPR
71020CPR  THIS ROUTINE SORTS 1 DIMENSIONAL ARRAYS               .
71030CPR
71040CPR   CALLING ARGUMENTS --
71050CPR       NAME      ATTRIBUTES     DEFINITION
71060CPR
71070CPR       SEEDS    (I/O)       ARRAY TO BE SORTED
71080CPR       LAX      (I)         NUMBER OF ITEMS IN ARRAY
71090CPR       NDUM     DUMMY       DUMMY
71100CPR       IKEY     DUMMY       DUMMY
71110CPR
71120CPR   COMMONS USED --    NONE
71130CPR
71140CPR   FUNCTIONS/ROUTINES CALLED -- NONE
71150CPR
71160CPR   FILES USED --    NONE
71170CPR
71180CPR   LIBRARIES ACCESSED --
71190CPR   NONE
71200CPR
71210CPR   LOCAL VARIABLES --
71220CPR
71230CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
71240CPR
71250CPR
71260CPR   SPECIAL REMARKS/INSTRUCTIONS --
71270CPR      HOST = H6000
71280CPR
71290CPR
71300CPR===================================================
71550      END
```

```
C=================================================================
      SUBROUTINE HTICON(IHTIC)
C=================================================================
CPR
CPR
CPR
CPR   ROUTINE--          HTICON
CPR
CPR   DATE/WRITTEN BY-- 01/14/84      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE CONVERTS FROM THE HEAT TRANSFER OUTPUT FORM
CPR   TO THE INTERNAL 3D TITAN HEAT TRANSFER INPUT FORM
CPR   VALID HEAT TRANSFER INPUT CODES ARE
CPR
CPR    0   --- INTERNAL 3D TITAN HEAT TRANSFER INPUT FORM
CPR    1   --- SINDA HEAT TRANSFER INPUT FORM
CPR    2   --- MARC HEAT TRANSFER INPUT FORM  (FORMATTED)
CPR    3   --- MARC HEAT TRANSFER INPUT FORM  (SEQUENTIAL BINARY)
CPR    4   --- THTD HEAT TRANSFER INPUT FORM
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       IHTIC       (I)        HEAT TRANSFER INPUT CODE
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       SINTIT
CPR       MARCRD
CPR
CPR   FILES USED --
CPR
CPR       IERF      S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
      SUBROUTINE MARCRD(IFTYP,NSTEPS)
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--         MARCRD
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RD MCCLAIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR       READ MARC POST TAPE AND WRITE TRANSFER MODULE
CPR       INPUT FILE
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       IFTYP        INPUT          FILE TYPE OF MARC POST TAPE
CPR                                   O=FORMATTED, 1=SEQ  BINARY
CPR       NSTEPS       INPUT          NUMBER OF STEPS TO READ FROM
CPR                                   MARC POST TAPE
CPR
CPR   COMMONS USED --
CPR         AFIL -- FILE CODES
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       READR     READS DATA FROM POST TAPE
CPR       SKPREC    SKIPS RECORDS ON POST TAPE
CPR
CPR   FILES USED --
CPR
CPR       INTITS     S   TRANSFER MODULE INPUT FILE
CPR       IHTIN      S   MARC INPUT FILE (SEQ BIN OR FORMATTED)
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
      SUBROUTINE READR(LFN,DATA,NDATA,CHAR,NWRDS,ITYPE,IFTYP,IEND)
CPR
CPR====================================================================
CPR
CPR
CPR   ROUTINE--           READR
CPR
CPR   DATE/WRITTEN BY-- 01/11/84      RD MCCLAIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR      READ FLOATING POINT,INTEGER, OR CHARACTER DATA
CPR      FROM MARC POST TAPE
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       LFN          INPUT          FILE CODE TO READ FROM
CPR       DATA         OUTPUT         FLOATING POINT ARRAY
CPR       NDATA        OUTPUT         INTEGER ARRAY
CPR       CHAR         OUTPUT         CHARACTER ARRAY
CPR       NWRDS        INPUT          NUMBER OF ITEMS TO READ
CPR       ITYPE        INPUT          TYPE OF DATA TO READ
CPR                                   1=LOAD FP DATA INTO ARRAY DATA
CPR                                   2=LOAD INTEGER DATA INTO ARRAY NDATA
CPR                                   3=LOAD CHAR DATA INTO ARRAY CHAR
CPR       IFTYP        INPUT          FILE TYPE
CPR                                   0=FORMATTED
CPR                                   1=SEQUENTIAL BINARY
CPR
CPR   COMMONS USED --
CPR         AFIL -- FILE CODES
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       LFN      S     FILE CODE TO BE READ
CPR       INOT6    S     STANDARD OUTPUT FILE
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR====================================================================
      END
```

```
      SUBROUTINE SKPREC(IFILE,NUMREC,IFTYP)
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          SKPREC
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RD MCCLAIN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR       SKIP RECORDS ON A SEQUENTIAL FILE
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       IFILE        INPUT        FILE CODE TO SKIP RECORDS ON
CPR       NUMREC       INPUT        NUMBER OF RECORDS TO SKIP
CPR       IFTYP        INPUT        FILE TYPE OF MARC POST TAPE
CPR                                 O=FORMATTED, 1=BINARY
CPR
CPR   COMMONS USED --  NONE
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
      END
```

```
C================================================
      SUBROUTINE SINTIT
C================================================
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          SINTIT
CPR
CPR   DATE/WRITTEN BY-- 01/09/84     RN PITTMAN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR      MAIN EXECUTIVE ROUTINE FOR CREATION OF SINDA-TO-3DTITAN
CPR      INTERFACE FILE
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME     ATTRIBUTES     DEFINITION
CPR               -----NONE-----
CPR
CPR   COMMONS USED --
CPR       WORK   - STORES INTERNAL ARRAY DATA
CPR                -------- ----- ----
CPR                   TEMP       ARRAY OF ELEMENTAL TEMPERATURES
CPR                              CORRESPONDING TO 'ITELM'
CPR                  IPKFAC      ARRAY OF PACKED FACE NUMBERS
CPR                              CORRESPONDING TO 'IGELM'
CPR                  IGELM       ARRAY OF ELEMENTS WITH GEOMETRY
CPR                  ITELM       ARRAY OF ELEMENTS WITH TEMPERATURES
CPR
CPR       AFIL - FILE CODES
CPR                ---- -----
CPR                 ELEMENTS       DEFINITION
CPR                 --------       ----------
CPR                   IHTIN       SINDA TEMPERATURE OUTPUT FILE (INPUT)
CPR                   ISINGF       SINDA GEOMETRY FILE (INPUT)
CPR                   INTITS       INTERFACE FILE (OUTPUT)
CPR                   IERF       HARDCOPY FILE (OUTPUT)
CPR
CPR       KNTDAT - COUNTER DATA
CPR                ------- ----
CPR                 ELEMENTS       DEFINITION
CPR                 --------       ----------
CPR                  NUMNOD      NUMBER OF NODES (CORNER POINTS)
CPR                              IN MODEL
CPR                  NUMEWG      NUMBER OF ELEMENTS WITH GEOMETRY
CPR                  NUMTIM      NUMBER OF SOLUTION TIMES
CPR                   NNPE       NUMBER OF NODES PER ELEMENT
CPR                   ECODE      ELEMENT CODE
CPR                              =1  - 2D MODEL (NNPE=4)
CPR                              =11 - 3D MODEL (NNPE=8)
CPR                   ACODE      ANALYSIS CODE
CPR                              =1  - FINITE DIFFERENCE ANALYSIS
CPR                  NUMEWT      NUMBER OF ELEMENTS WITH TEMPERATURES
CPR                  NUMSKP      NUMBER OF RECORDS TO SKIP ON 'IHTIN'
CPR                              PRIOR TO BEGINNING OF TEMPERATURE DATA
CPR                  NUMFUL      NUMBER OF FULL RECORDS OF DATA
CPR                  NUMREM      NUMBER OF WORDS ON FINAL RECORD
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       DATINT
CPR       PRCSOF
CPR       PRCGEO
CPR       WRTNTF
CPR
CPR   FILES USED --
CPR       NONE
```

```
77430CPR
77440CPR   LIBRARIES ACCESSED --
77450CPR       NONE
77460CPR
77470CPR   LOCAL VARIABLES --
77480CPR       NONE
77490CPR
77500CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
77510CPR       PROGRAM IS CURRENTLY DIMENSIONED TO PERMIT A MAXIMUM OF
77520CPR       5000 ELEMENTS WITH GEOMETRY AND 5000 ELEMENTS WITH
77530CPR       TEMPERATURES
77540CPR
77550CPR   SPECIAL REMARKS/INSTRUCTIONS --
77560CPR      HOST = HIS
77570CPR
77580CPR
77590CPR=================================================
78060      END
```

```
      SUBROUTINE DATINT
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          DATINT
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR      THIS ROUTINE INITIALIZES ALL OF THE GLOBAL COMMON VARIABLES
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR               -----NONE-----
CPR
CPR   COMMONS USED --
CPR       WORK   -- STORES INTERNAL ARRAY DATA
CPR       AFIL -- FILE CODES (MAIN)
CPR       KNTDAT -- COUNTER DATA (MAIN)
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       NONE
CPR
CPR   FILES USED --
CPR       NONE
CPR
CPR   LIBRARIES ACCESSED --
CPR       NONE
CPR
CPR   LOCAL VARIABLES --
CPR       NONE
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
CPR       NONE
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR       HOST = H6000
CPR
CPR
CPR=====================================================
C  .  END OF SUBROUTINE 'DATINT'
      END
```

```
79090        SUBROUTINE PRCSOF
79130CPR
79140CPR=================================================
79150CPR
79160CPR
79170CPR     ROUTINE--          PRCSOF
79180CPR
79190CPR     DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
79200CPR     DATE/REVISED BY--
79210CPR
79220CPR     FUNCTION/PURPOSE--
79230CPR        THIS ROUTINE CONTROLS THE INITIAL PROCESSING OF THE
79240CPR        'SINDA OUTPUT FILE' (THE FILE WHICH CONTAINS THE
79250CPR        RESULTS OF THE SINDA ANALYSIS)
79260CPR
79270CPR     CALLING ARGUMENTS --
79280CPR         NAME       ATTRIBUTES     DEFINITION
79290CPR                 -----NONE-----
79300CPR
79310CPR     COMMONS USED --
79320CPR         AFIL -- FILE CODES (MAIN)
79330CPR
79340CPR     FUNCTIONS/ROUTINES CALLED --
79350CPR         GETEWT
79360CPR         GTNUMT
79370CPR
79380CPR     FILES USED --
79390CPR         NONE
79400CPR
79410CPR     LIBRARIES ACCESSED --
79420CPR         NONE
79430CPR
79440CPR     LOCAL VARIABLES --
79450CPR         NONE
79460CPR
79470CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE --
79480CPR         NONE
79490CPR
79500CPR     SPECIAL REMARKS/INSTRUCTIONS --
79510CPR         HOST = H6000
79520CPR
79530CPR
79540CPR=================================================
79770C     . END OF SUBROUTINE 'PRCSOF'
79800        END
```

```
79840        SUBROUTINE GETEWT
79880CPR
79890CPR=================================================
79900CPR
79910CPR
79920CPR     ROUTINE--         GETEWT
79930CPR
79940CPR     DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
79950CPR     DATE/REVISED BY--
79960CPR
79970CPR     FUNCTION/PURPOSE--
79980CPR        1   READS THE SINDA ELEMENT NUMBERS FROM 'IHTIN' INTO 'ITELM'
79990CPR        2   READS FROM 'IHTIN' THE TOTAL NUMBER OF SINDA ELEMENTS
80000CPR            WITH TEMPERATURES INTO 'NUMEWT'
80010CPR        3   COUNTS THE NUMBER OF RECORDS PRECEEDING THE FIRST TEMP-
80020CPR            ERATURE SOLUTION ON 'IHTIN' AND STORES THIS VALUE IN
80030CPR            'NUMSKP'.
80040CPR
80050CPR     CALLING ARGUMENTS --
80060CPR         NAME       ATTRIBUTES     DEFINITION
80070CPR                 -----NONE-----
80080CPR
80090CPR     COMMONS USED --
80100CPR         WORK    --STORES INTERNAL ARRAY DATA
80110CPR         AFIL -- FILE CODES (MAIN)
80120CPR         KNTDAT -- COUNTER DATA (MAIN)
80130CPR
80140CPR     FUNCTIONS/ROUTINES CALLED --
80150CPR         ERRPRT
80160CPR
80170CPR     FILES USED --
80180CPR         IHTIN    (S)   SINDA OUTPUT FILE (INPUT TO THIS ROUTINE)
80190CPR
80200CPR     LIBRARIES ACCESSED --
80210CPR         NONE
80220CPR
80230CPR     LOCAL VARIABLES --
80240CPR         IPOS   -  POSITION OF THE BEGINNING OF EACH GROUP OF 10
80250CPR                   ELEMENT NUMBERS IN THE 'ITELM' ARRAY
80260CPR         ***NOTE***
80270CPR         IN THIS ROUTINE, THE GLOBAL VARIABLES 'NUMFUL' AND NUMREM'
80280CPR         ARE REDEFINED AS FOLLOWS-
80290CPR           NUMFUL - NUMBER OF 'FULL' 10-WORD RECORDS OF SINDA
80300CPR                    ELEMENT NUMBERS CONTAINED ON 'IHTIN'
80310CPR           NUMREM - NUMBER OF ELEMENT NUMBERS REMAINING ON FINAL
80320CPR                    RECORD OF ELEMENT NUMBERS
80330CPR
80340CPR     SPECIAL COMMENTS ABOUT THIS ROUTINE --
80350CPR          NONE
80360CPR
80370CPR     SPECIAL REMARKS/INSTRUCTIONS --
80380CPR         HOST = H6000
80390CPR
80400CPR
80410CPR=================================================
81210C .     END OF SUBROUTINE 'GETEWT'
81240        END
```

```
      SUBROUTINE GTNUMT
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--         GTNUMT
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR      THIS ROUTINE COUNTS THE TOTAL NUMBER OF SOLUTION BLOCKS
CPR      CONTAINED IN THE 'SINDA OUTPUT FILE' AND STORES THIS VALUE
CPR      IN THE GLOBAL VARIABLE 'NUMTIM'
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR               -----NONE-----
CPR
CPR   COMMONS USED --
CPR       WORK   --STORES  INTERNAL ARRAY DATA
CPR       AFIL -- FILE CODES (MAIN)
CPR       KNTDAT -- COUNTER DATA (MAIN)
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       ERRPRT
CPR
CPR   FILES USED --
CPR       IHTIN    (S)    SINDA OUTPUT FILE (INPUT TO THIS PROGRAM)
CPR
CPR   LIBRARIES ACCESSED --
CPR       NONE
CPR
CPR   LOCAL VARIABLES --
CPR       DUMMY -  VARIABLE USED IN DUMMY READS TO PASS THROUGH EACH
CPR                SOLUTION BLOCK TO THE NEXT
CPR       KNT   -  CURRENT COUNT OF THE NUMBER OF SOLUTION BLOCKS
CPR                ENCOUNTERED ON 'IHTIN'
CPR       ***NOTE***
CPR       IN THIS ROUTINE, THE GLOBAL VARIABLES 'NUMFUL' AND 'NUMREM'
CPR       ARE REDEFINED AS FOLLOWS-
CPR         NUMFUL - NUMBER OF 'FULL' 6-WORD RECORDS OF TEMPERATURE
CPR                  DATA IN EACH SOLUTION BLOCK ON 'IHTIN'
CPR         NUMREM - NUMBER OF TEMPERATURES ON FINAL RECORD OF EACH
CPR                  SOLUTION BLOCK
CPR       THESE DEFINITIONS ARE NOT ALTERED FURTHER IN THIS PROGRAM
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
CPR       EACH TEMPERATURE SOLUTION BLOCK ON 'IHTIN' IS PASSED THROUGH
CPR       WITH DUMMY READS UNTIL THE END OF FILE IS ENCOUNTERED
CPR       THE SOLUTION COUNTER, 'KNT', IS INCREMENTED BY ONE AT THE
CPR       BEGINNING OF EACH BLOCK
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
C . . END OF SUBROUTINE 'GTNUMT'
      END
```

```
82620      SUBROUTINE PRCGEO
82660CPR
82670CPR====================================================
82680CPR
82690CPR
82700CPR   ROUTINE--         PRCGEO
82710CPR
82720CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
82730CPR   DATE/REVISED BY--
82740CPR
82750CPR   FUNCTION/PURPOSE--
82760CPR      1  READS THE VALUES CONTAINED IN THE 'DIRECTORY' RECORD OF
82770CPR         THE 'SINDA GEOMETRY FILE' INTO THE GLOBAL COMMON VALUES
82780CPR         'NUMNOD', 'NUMEWG', AND 'NNPE'
82790CPR      2  SETS THE VALUE FOR 'ECODE' BASED UPON THE VALUE OF 'NNPE'
82800CPR
82810CPR   CALLING ARGUMENTS --
82820CPR       NAME      ATTRIBUTES     DEFINITION
82830CPR               -----NONE-----
82840CPR
82850CPR   COMMONS USED --
82860CPR       AFIL -- FILE CODES (MAIN)
82870CPR       KNTDAT -- CCOUNTER DATA (MAIN)
82880CPR
82890CPR   FUNCTIONS/ROUTINES CALLED --
82900CPR       ERRPRT
82910CPR
82920CPR   FILES USED --
82930CPR       ISINGF    (S)   SINDA GEOMETRY FILE (INPUT)
82940CPR
82950CPR   LIBRARIES ACCESSED --
82960CPR       NONE
82970CPR
82980CPR   LOCAL VARIABLES --
82990CPR       TITLE - CHARACTER ARRAY INTO WHICH THE 'ISINGF' TITLE RECORD
83000CPR               IS READ
83010CPR
83020CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
83030CPR       NONE
83040CPR
83050CPR   SPECIAL REMARKS/INSTRUCTIONS --
83060CPR       HOST = H6000
83070CPR
83080CPR
83090CPR====================================================
83560C ... END OF SUBROUTINE 'PRCGEO'
83590      END
```

```
      SUBROUTINE WRTNTF
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          WRTNTF
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR      THIS ROUTINE CONTROLS THE WRITING OF THE 'INTERFACE FILE'.
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR               -----NONE-----
CPR
CPR   COMMONS USED --
CPR       WORKS  --STORES  INTERNAL ARRAY DATA
CPR       AFIL -- FILE CODES (MAIN)
CPR       KNTDAT -- COUNTER DATA (MAIN)
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       WRTDIR
CPR       WRTGEO
CPR       WRTEMP
CPR       ERRPRT
CPR
CPR   FILES USED --
CPR       INTITS    (S)   INTERFACE FILE (OUTPUT)
CPR       ISINGF    (S)   SINDA GEOMETRY FILE (INPUT)
CPR       IHTIN    (S)   SINDA OUTPUT FILE (INPUT)
CPR
CPR   LIBRARIES ACCESSED --
CPR       NONE
CPR
CPR   LOCAL VARIABLES --
CPR       DUMMY  -  VARIABLE USED IN 'DUMMY' READS TO SKIP OVER ALL
CPR                 DATA ON 'IHTIN' PRECEEDING THE TEMPERATURE
CPR                 SOLUTION BLOCKS
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
CPR       NONE
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR       HOST = H6000
CPR
CPR
CPR==================================================
C   .  END OF SUBROUTINE 'WRTNTF'
       END
```

```
84920      SUBROUTINE WRTDIR
84960CPR
84970CPR====================================================
84980CPR
84990CPR
85000CPR   ROUTINE--          WRTDIR
85010CPR
85020CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
85030CPR   DATE/REVISED BY--
85040CPR
85050CPR   FUNCTION/PURPOSE--
85060CPR      THIS ROUTINE WRITES THE TITLE (RECORDS 1-4) AND DIRECTORY
85070CPR      (RECORD 5) RECORDS OF THE 'INTERFACE FILE'
85080CPR
85090CPR   CALLING ARGUMENTS --
85100CPR       NAME      ATTRIBUTES     DEFINITION
85110CPR               -----NONE-----
85120CPR
85130CPR   COMMONS USED --
85140CPR       WORK   --STORES INTERNAL ARRAY DATA
85150CPR       AFIL -- FILE CODES (MAIN)
85160CPR       KNTDAT -- COUNTER DATA (MAIN)
85170CPR
85180CPR   FUNCTIONS/ROUTINES CALLED --
85190CPR       ERRPRT
85200CPR
85210CPR   FILES USED --
85220CPR       INTITS    (S)   INTERFACE FILE (OUTPUT)
85230CPR       ISINGF    (S)   SINDA GEOMETRY FILE (INPUT)
85240CPR
85250CPR   LIBRARIES ACCESSED --
85260CPR       NONE
85270CPR
85280CPR   LOCAL VARIABLES --
85290CPR       TITLE  -  80-CHARACTER TITLE CONTAINED ON RECORD 1 OF 'ISINGF'
85300CPR                 AND WRITTEN ON RECORD 4 OF 'INTITS'
85310CPR       USRTTL -  80-CHARACTER ARRAY USED TO 'DUMMY-OUT' RECORDS
85320CPR                 1-3 OF 'INTITS'
85330CPR
85340CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
85350CPR       NONE
85360CPR
85370CPR   SPECIAL REMARKS/INSTRUCTIONS --
85380CPR       HOST = H6000
85390CPR
85400CPR
85410CPR====================================================
86090C    . END OF SUBROUTINE 'WRTDIR'
86120      END
```

```
      SUBROUTINE WRTGEO
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--         WRTGEO
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR      THIS ROUTINE WRITES THE GEOMETRICAL DATA (COORDINATE,
CPR      CONNECTIVITY, AND FACES) ON THE INTERFACE FILE
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES    DEFINITION
CPR               -----NONE-----
CPR
CPR   COMMONS USED --
CPR       WORK   --STORES INTERNAL ARRAY DATA
CPR       AFIL -- FILE CODES (MAIN)
CPR       KNTDAT -- COUNTER DATA (MAIN)
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       ERRPRT
CPR
CPR   FILES USED --
CPR       ISINGF    (S)   SINDA GEOMETRY FILE (INPUT)
CPR       INTITS    (S)   INTERFACE FILE (OUTPUT)
CPR
CPR   LIBRARIES ACCESSED --
CPR       NONE
CPR
CPR   LOCAL VARIABLES --
CPR       DUMMY   -  USED IN 'DUMMY READ' TO POSITION 'ISINGF' AT
CPR                  BEGINNING OF COORDINATE DATA
CPR       N       -  ARRAY OF NODE NUMBERS DEFINING CONNECTIVITY
CPR                  OF 'IGELM(I)'
CPR       NODE    -  CURRENT NODE (CORNER POINT) NUMBER
CPR       X       -  X-COORDINATE OF 'NODE'
CPR       Y       -  Y-COORDINATE OF 'NODE'
CPR       Z       -  Z-COORDINATE OF 'NODE'
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
CPR       NONE
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR       HOST = H6000
CPR
CPR
CPR==================================================
C ..  END OF SUBROUTINE 'WRTGEO'
      END
```

```
      SUBROUTINE WRTEMP
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          WRTEMP
CPR
CPR   DATE/WRITTEN BY-- 01/11/84     RN PITTMAN
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR       THIS ROUTINE WRITES THE TIME AND TEMPERATURE DATA ON THE
CPR       'INTERFACE FILE'
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR               -----NONE-----
CPR
CPR   COMMONS USED --
CPR       WORK   --STORES INTERNAL ARRAY DATA
CPR       AFIL -- FILE CODES (MAIN)
CPR       KNTDAT -- COUNTER DATA (MAIN)
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR       ERRPRT
CPR
CPR   FILES USED --
CPR       IHTIN    (S)   SINDA OUTPUT FILE (INPUT)
CPR       INTITS   (S)   INTERFACE FILE (OUTPUT)
CPR
CPR   LIBRARIES ACCESSED --
CPR       NONE
CPR
CPR   LOCAL VARIABLES --
CPR       KNT  - POSITION IN THE 'TEMP' ARRAY OF THE NEXT GROUP OF
CPR              SIX TEMPERATURES TO BE WRITTEN TO 'INTITS'
CPR       TIME - CURRENT VALUE OF TIME READ FROM 'IHTIN' FOR THIS
CPR              TEMPERATURE BLOCK
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE --
CPR       NONE
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR       HOST = H6000
CPR
CPR
CPR==================================================
C     END OF SUBROUTINE 'WRTEMP'
      END
```

```
88910       SUBROUTINE ERRPRT (IROUT)
88950CPR
88960CPR=====================================================
88970CPR
88980CPR
88990CPR    ROUTINE--          ERRPRT
89000CPR
89010CPR    DATE/WRITTEN BY-- 01/12/84     RN PITTMAN
89020CPR    DATE/REVISED BY--
89030CPR
89040CPR    FUNCTION/PURPOSE--
89050CPR        THIS ROUTINE PROCESSES ALL FATAL ERROR CONDITIONS BY
89060CPR        PRINTING AN EXPLANATION OF THE ERROR ON THE HARDCOPY
89070CPR        OUTPUT FILE AND TERMINATING PROGRAM EXECUTION
89080CPR
89090CPR    CALLING ARGUMENTS --
89100CPR        NAME      ATTRIBUTES     DEFINITION
89110CPR        ----      ----------     ----------
89120CPR        IROUT    INTEGER/INPUT  ROUTING VARIABLE FOR APPROPRIATE
89130CPR                                 ERROR MESSAGE
89140CPR
89150CPR    COMMONS USED --
89160CPR        AFIL -- FILE CODES (MAIN)
89170CPR
89180CPR    FUNCTIONS/ROUTINES CALLED --
89190CPR        NONE
89200CPR
89210CPR    FILES USED --
89220CPR        IERF     (S)   HARDCOPY FILE (OUTPUT)
89230CPR
89240CPR    LIBRARIES ACCESSED --
89250CPR        NONE
89260CPR
89270CPR    LOCAL VARIABLES --
89280CPR        NONE
89290CPR
89300CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE --
89310CPR        NONE
89320CPR
89330CPR    SPECIAL REMARKS/INSTRUCTIONS --
89340CPR        HOST = H6000
89350CPR
89360CPR
89370CPR=====================================================
91950       END
```

```
91970C================================================
91990      SUBROUTINE TEMFOR(ISPOC)
92010C================================================
92030CPR
92040CPR==================================================
92050CPR
92060CPR
92070CPR   ROUTINE--          TEMFOR
92080CPR
92090CPR   DATE/WRITTEN BY-- 01/14/84     RJ MAFFEO
92100CPR   DATE/REVISED BY--
92110CPR
92120CPR   FUNCTION/PURPOSE--
92130CPR   THIS ROUTINE CALLS ALL THE REQUIRED ROUTINES TO FORMAT THE
92140CPR   OUTPUT TEMPERATURES INTO FORMED NEEDED FOR THE STRESS ANALYSIS
92150CPR   PROGRAM
92160CPR         VALID CODES ARE
92170CPR           0  --- NEUTRAL TEMPERATURE OUTPUT FILE ONLY
92180CPR           1  --- NASTRAN FORMATTED TEMPERATURES
92190CPR         2,3  --- MARC    FORMATTED TEMPERATURES
92200CPR
92210CPR   CALLING ARGUMENTS --
92220CPR       NAME      ATTRIBUTES     DEFINITION
92230CPR
92240CPR       ISPOC         (I)          STRESS PROGRAM OUTPUT CODE
92250CPR
92260CPR   COMMONS USED --
92270CPR
92280CPR       AFIL
92290CPR       SIZE
92300CPR
92310CPR   FUNCTIONS/ROUTINES CALLED --
92320CPR
92330CPR       TPNAST
92340CPR       MARCTO
92350CPR
92360CPR   FILES USED --
92370CPR
92380CPR       IERF     S
92390CPR
92400CPR   LIBRARIES ACCESSED --
92410CPR   NONE
92420CPR
92430CPR
92440CPR   LOCAL VARIABLES --
92450CPR
92460CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
92470CPR
92480CPR
92490CPR   SPECIAL REMARKS/INSTRUCTIONS --
92500CPR      HOST = H6000
92510CPR
92520CPR
92530CPR==================================================
93110      END
```

```
93140       SUBROUTINE TPNAST(NUMNOD,NUMTMP)
93170CPR
93180CPR=====================================================
93190CPR
93200CPR
93210CPR    ROUTINE--          TPNAST
93220CPR
93230CPR    DATE/WRITTEN BY-- 01/14/84     RD MCCLAIN
93240CPR    DATE/REVISED BY--
93250CPR
93260CPR    FUNCTION/PURPOSE--
93270CPR
93280CPR        WRITE NASTRAN BULK DATA TEMPERATURE CARDS
93290CPR
93300CPR
93310CPR    CALLING ARGUMENTS --
93320CPR        NAME       ATTRIBUTES     DEFINITION
93330CPR
93340CPR        NUMNOD       INPUT         NUMBER OF NODES ON NEUT TEMP FILE
93350CPR        NUMTMP       INPUT         NUMBER OF TEMPERATURE SETS
93360CPR
93370CPR    COMMONS USED --
93380CPR
93390CPR        AFIL   -- FILE CODES
93400CPR
93410CPR    FUNCTIONS/ROUTINES CALLED --
93420CPR
93430CPR
93440CPR    FILES USED --
93450CPR
93460CPR        IOUTF     S    NEUTRAL OUTPUT FILE
93470CPR        ITEMP     S    NASTRAN BULK DATA FILE
93480CPR
93490CPR    LIBRARIES ACCESSED --
93500CPR    NONE
93510CPR
93520CPR
93530CPR    LOCAL VARIABLES --
93540CPR
93550CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE
93560CPR
93570CPR
93580CPR    SPECIAL REMARKS/INSTRUCTIONS --
93590CPR       HOST = H6000
93600CPR
93610CPR
93620CPR=====================================================
93910       END
```

```
C======================================================
      SUBROUTINE I3DSF(CMNMX,CRD,XP,YP,ZP,NSP,
CPR
CPR====================================================
CPR
CPR
CPR   ROUTINE--          I3DSF
CPR
CPR   DATE/WRITTEN BY-- 11/28/83     R J  MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE COMPUTES THE VALUES OF THE LOCAL COORDINATES
CPR   OF A STRESS POINT WRT THE VERTICES OF THE 8-NODED HEAT
CPR   TRANSFER ELEMENT.IF THE STRESS POINT LIES IN THE ELM
CPR   IT COMPUTES THE 8 WEIGHTING COEFFICIENTS FOR THIS POINT
CPR   IF THE POINT LIES OUTSIDE THE ELM IT COMPUTES THE SURFACE
CPR   COORDINATES OF THE POINT PROJECTED IN THE LOCAL COORD SYSTEM
CPR   AND COMPUTES THE WEIGHTING COEFFS BASED ON THESE SURFACE
CPR   COORDINATES IT ALSO COMPUTES THE DISTANCE THE POINT IS AWAY
CPR   FROM THE SURFACE
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME  ATTRIBUTES     DEFINITION
CPR
CPR       CMNMX    (I)      ARRAY WITH MIN AND MAX WINDOW COORDS
CPR       CRD      (I)      ARRAY WITH ELM VERTICE COORDS
CPR       XP       (I) >>
CPR       YP       (I) >>>> STRESS POINT COORDS
CPR       ZP       (I) >>
CPR       NSP      (I)      STRESS POINT NAME
CPR       XLOC     (O) >>
CPR       YLOC     (O) >>>> LOCAL COORDS OF STRESS POINT
CPR       ZLOC     (O) >>
CPR       SF       (O)      WEIGHTING COEFFICIENTS
CPR       ICONV    (O)      CONVERGENCE PARAMETER
CPR                         O --- DID NOT CONVERGE
CPR                         1 --- CONVERGED TO POINT IN THE ELM
CPR                         2 --- CONVERGED TO POINT OUT OF ELM
CPR       IPAS     (I)      SEARCH TYPE PARAMETER
CPR                         -1 --- NO REARRANGE,NO ROTATION
CPR                                NO SURFACING
CPR                          O --- NO REARRANGE,ROTATE
CPR                                NO SURFACING
CPR                          1 --- REAARANGE,ROTATE
CPR                                PERFORM SURFACING
CPR       DIST     (O)       DISTANCE FORM OUTSIDE POINT TO SURFACE
CPR       XSUR     (O) >>
CPR       YSUR     (O) >>>>  SURFACE COORDINATES
CPR       ZSUR     (O) >>
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       REAARY
CPR       SHAPFV
CPR       CHKPT
CPR       JACBCK
CPR       SFDIST
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
```

```
94620CPR   NONE
94630CPR
94640CPR
94650CPR   LOCAL VARIABLES --
94660CPR        IPOINT --- ACCURACY INDICATOR
94670CPR                  0 -- STRESS POINT NOT RECALCULATED ACCURATELY
94680CPR                  1 -- STRESS POINT     RECALCULATED ACCURATELY
94690CPR        IJACB  --- JACOBIAN CHECK PARAMETER
94700CPR                  0 -- JACOBIAN NOT POSITIVE
94710CPR                  1 -- JACOBIAN IS  POSITIVE
94720CPR
94730CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
94740CPR
94750CPR
94760CPR   SPECIAL REMARKS/INSTRUCTIONS --
94770CPR      HOST = H6000
94780CPR
94790CPR
94800CPR=====================================================
97960      END
```

```
C=================================================
      SUBROUTINE REAARY(X,Y,Z,XP,YP,ZP,SF,ICLOS,ITRYC,DTOCN,XL,YL,ZL
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          REAARY
CPR
CPR   DATE/WRITTEN BY-- 11/28/83     R J  MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE REARRANGES THE CONNECTIVITY COORDS AND
CPR   THE SHAPE FUNCTION ARRAYS IN AN ATTEMPT TO GET AN
CPR   ORDER THAT WILL ENHANCE CONVERGENCE OF THE INVERSE
CPR   SHAPE FUNCTION ROUTINE(I3DSF) IT ALSO COMPUTES THE
CPR   DISTANCE FROM THE STRESS POINT TO THE NEAREST ELM
CPR   CORNER OPTIONALY IT WILL "UN-REARRANGE" THE
CPR   SHAPE FUNCTION ARRAY AND THE LOCAL COORDS
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       X        (I/O) >>
CPR       Y        (I/O) >>>>   CONNECTIVITY COORD ARRAYS
CPR       Z        (I/O) >>
CPR       XP       (I)  >>
CPR       YP       (I)  >>>>    COORDS OF STRESS POINTS
CPR       ZP       (I)  >>
CPR       SF       (I/O)        SHAPE FUNCTION ARRAY
CPR       ICLOS    (I/O)        VALUE OF NEW I JOINT
CPR       ITRYC    (I)          CONTROL ARGUMENT
CPR                              -1 -- REARRANGE SHAPE FUNCTION ARRAY
CPR                               0 -- PICK I JOINT AS JOINT NEAREST TO
CPR                                     STRESS POINT
CPR                             N>0 -- USE VALUE OF N AS I JOINT
CPR       DTOCN    (O)          DISTANCE TO NEAREST CORNER
CPR       XL       (I/O) >>>
CPR       YL       (I/O) >>>>>>  VALUES OF LOCAL COORDINATES
CPR       ZL       (I/O) >>>
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       SORT2
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR   NOTE THE IJ ARRAY IS USED TO RELATE THE REARRANGED
CPR   CONNECTIVITY ARRAY TO THE JOINT PICKED AS THE I JOINT
CPR
      END
```

```
C==================================================
      SUBROUTINE SORT2(SEEDS,FOLLO,LAX)
C==================================================
CPR   ROUTINE--         SORT2
CPR
CPR   DATE/WRITTEN BY-- ?????????    GE DOUBLE SORT
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE SORTS TWO 1 DIMENSIONAL ARRAYS
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       SEEDS    (I/O)      PRIMARY ARRAY TO BE SORTED
CPR       FOLLO    (I/O)      ARRAY TO BE SORTED LIKE THE PRIMARY
CPR       LAX      (I)        NUMBER OF ITEMS IN ARRAYS
CPR
CPR   COMMONS USED -- NONE
CPR
CPR   FUNCTIONS/ROUTINES CALLED -- NONE
CPR
CPR   FILES USED -- NONE
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR================================================
       END
```

```
C==================================================
      SUBROUTINE SFDIST(SF,DIST,XNC,YNC,ZNC,XSUR,YSUR,ZSUR,XP,YP,ZP,
C==================================================
CPR
CPR====================================================
CPR
CPR
CPR   ROUTINE--         SFDIST
CPR
CPR   DATE/WRITTEN BY-- 11/15/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE COMPUTES THE DISTANCE FROM STRESS
CPR   POINTS THAT ARE OUTSIDE A HEAT TRANSFER ELM TO THE
CPR   SURFACE OF THE HEAT TRAN ELM IT ALSO COMPUTES THE
CPR   SURFACE COORDIANTES THIS SURFACE POINT IS FOUND BY
CPR   'TRUNCATING' THE LOCAL COORDINATES TO 1 OR -1
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       SF       (I)            SHAPE FUNCTION ARRAY
CPR       DIST     (O)            COMPUTED DISTANCE
CPR       XNC      (I) >>>
CPR       YNC      (I) >>>>>>>>> LOCAL VALUES OF STRESS POINT
CPR       ZNC      (I) >>>
CPR       XSUR     (O) >>>
CPR       YSUR     (O) >>>>>>>>> COORDS OF SURFACE POINT
CPR       ZSUR     (O) >>>
CPR       XP       (I) >>>
CPR       YP       (I) >>>>>>>>> COORDS OF STRESS POINT
CPR       ZP       (I) >>>
CPR       NNPE     (I)           NUMBER OF NODES PER ELM
CPR       CRD      (I)           COORDS OF VERTICES OF ELM
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR   SHAPFV
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR===================================================
      END
```

```
C==================================================
      SUBROUTINE CHKPT(SF,X,Y,Z,XS,YS,ZS,IPOINT)
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          CHKPT
CPR
CPR   DATE/WRITTEN BY-- 11/28/83     R J. MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE RECOMPUTES THE STRESS POINT COORDS BASED
CPR   ON THE VALUES OF THE LOCAL COORDINATES TO CHECK FOR
CPR   ACCURACY
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       SF        (I)      ARRAY OF SHAPE FUNCTIONS
CPR       X         (I) >>
CPR       Y         (I) >>>> ARRAY OF ELM CORNER COORDS
CPR       Z         (I) >>
CPR       XS        (I) >>
CPR       YS        (I) >>>> ORIGINAL STRESS POINT COORDS
CPR       ZS        (I) >>
CPR       IPOINT    (O)      ACCURACY INDCATOR
CPR                          O -- DID NOT REPRODUCE COORD
CPR                          1 -- DID     REPRODUCE COORD
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR==================================================
C==================================================
      END
```

```
C=====================================================
      SUBROUTINE PDOFSF(TETA,YNETA,ZETA,IGP,IND,PD)
C=====================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          PDOFSF
CPR
CPR   DATE/WRITTEN BY-- 09/14/82     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS SUB COMPUTES THE DERIVATIVES OF THE
CPR   SHAPE FUNCTIONS WRT THE LOCAL SYSTEM
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES      DEFINITION
CPR
CPR       TETA       (I) >
CPR       YNETA      (I) >>          LOCAL COORDINATES
CPR       ZETA       (I) >
CPR       IGP        (I)             GAUSS POINT COUNTER
CPR       IND        (I)             2D/3D INDICATOR
CPR                                  IND=0 >> 3D
CPR                                  IND=1 THRU 6 >> 2D IND=FACE NUMBER
CPR       PD         (O)             ARRAY OF DERIVATIVES
CPR
CPR   COMMONS USED --
CPR
CPR       CORNDEF
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
      END
```

```
C===========================================================
      SUBROUTINE FNDJAC(X,Y,Z,P3D,XJAC)
CPR
CPR==================================================
CPR
CPR
CPR   ROUTINE--          FNDJAC
CPR
CPR   DATE/WRITTEN BY-- 11/28/83     R.J  MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE COMPUTES THE VALUE OF THE DETERMINANT OF
CPR   THE JACOBIAN EVALUATED AT THE STRESS POINT
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       X       (I) >>
CPR       Y       (I) >>>> ARRAY OF ELM CORNER COORDS
CPR       Z       (I) >>
CPR       P3D     (I)      ARRAY WITH PARTIAL DERIVATIVES OF
CPR                        LOCAL COORD WRT THE GLOBAL COORD
CPR       XJAC    (O)      VALUE OF DETERMINANT OF JACOBIAN
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR**************************************************
C***********************************************************
      END
```

```
C=============================================================
      SUBROUTINE SHAPFV(SF,XNC,YNC,ZNC)
CPR
CPR=======================================================
CPR
CPR
CPR   ROUTINE--          SHAPFV
CPR
CPR   DATE/WRITTEN BY-- 11/28/83     R J  MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE FINDS THE VALUES OF THE SHAPE FUNCTIONS
CPR   FOR THE 8-NODED ISOPARAMETRIC ELM BASED ON THE VALUES
CPR   OF THE LOCAL COORDINATES
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       SF      (O)          ARRAY WITH SHAPE FUNCTIONS
CPR       XNC     (I) >>
CPR       YNC     (I) >>>>     LOCAL COORDS
CPR       ZNC     (I) >>
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
C=============================================================
      END
```

```
C===========================================================
      SUBROUTINE JACBCK(XL,YL,ZL,X,Y,Z,XJAC,IJACB)
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          JACBCK
CPR
CPR   DATE/WRITTEN BY-- 11/28/83     R.J  MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR   THIS ROUTINE CALLS THE REQUIRED ROUTINES TO CHECK THE
CPR   VALUE OF THE DETERMINANT OF THE JACOBIAN AT THE STRESS
CPR   POINT
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR       XL        (I) >>
CPR       YL        (I) >>>>    LOCAL COORDS
CPR       ZL        (I) >>
CPR       X         (I) >>
CPR       Y         (I) >>>>   ARRAY OF ELM CORNER COORDS
CPR       Z         (I) >>
CPR       XJAC      (O)        VALUE OF DETERMINANT OF JACOBIAN
CPR       IJACB     (O)        INTERGER VARIABLE
CPR                            O -- JACOBIAN NOT POSITIVE
CPR                            1 -- JACOBIAN  IS POSITIVE
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       PDOFSF
CPR       FNDJAC
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=====================================================
C===========================================================
      END
```

```
C====================================================
      SUBROUTINE RWSPCF(IREC,B,IRW)
C====================================================
CPR
CPR====================================================
CPR
CPR
CPR   ROUTINE--          RWSPCF
CPR
CPR   DATE/WRITTEN BY-- 04/07/83     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE READS AND WRITES DATA TO THE
CPR   THE RANDOM STRESS POINT COEFF FILE
CPR   THIS FILE CONTAINS THE FOLLOWING INFO PER RECORD
CPR     -  STRESS NODE NAME   (1)
CPR     -  HEAT TRAN ELM CONTAINING THIS STRESS POINT  (1)
CPR     -  CONNECTIVITY OF THE HEAT TRAN ELM  (8)
CPR     -  WEIGHTING COEFF FOR THIS STRESS NODE (8)
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES      DEFINITION
CPR
CPR       IREC       (I)             FILE RECORD NUMBER
CPR       B          (I/O)           DATA ARRAY
CPR       IRW        (I)             READ/WRITE OPTION
CPR                                 (O-READ  1-WRITE)
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR       IRSPCF   R
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR====================================================
      END
```

```
106190C==================================================
106210      SUBROUTINE ELTRAN
106230C==================================================
106250CPR
106260CPR==================================================
106270CPR
106280CPR
106290CPR   ROUTINE--          ELTRAN
106300CPR
106310CPR   DATE/WRITTEN BY-- 01/27/84     RJ MAFFEO
106320CPR   DATE/REVISED BY--
106330CPR
106340CPR   FUNCTION/PURPOSE--
106350CPR
106360CPR Validate the values input for IECODE  and IACODE then assign
106370CPR the values to NNPE and NTPE depending on the values of
106380CPR IECODE and IACODE
106390CPR
106400CPR
106410CPR
106420CPR
106430CPR   CALLING ARGUMENTS --
106440CPR       NAME       ATTRIBUTES     DEFINITION
106450CPR
106460CPR
106470CPR   COMMONS USED --
106480CPR
106490CPR       AFIL
106500CPR       CNTLFL
106510CPR
106520CPR   FUNCTIONS/ROUTINES CALLED --
106530CPR
106540CPR
106550CPR   FILES USED --
106560CPR
106570CPR       IERF      S
106580CPR
106590CPR   LIBRARIES ACCESSED --
106600CPR   NONE
106610CPR
106620CPR
106630CPR   LOCAL VARIABLES --
106640CPR
106650CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
106660CPR
106670CPR
106680CPR   SPECIAL REMARKS/INSTRUCTIONS --
106690CPR      HOST = H6000
106700CPR
106710CPR
106720CPR==================================================
107330      END
```

```
C==================================================
      SUBROUTINE INITCM
C==================================================
CPR
CPR=================================================
CPR
CPR
CPR   ROUTINE--           INITCM
CPR
CPR   DATE/WRITTEN BY-- 01/31/84     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE INITIALIZES THE CRNDEF COMMON
CPR   THIS COMMON IS USED IN CALCRD AND I3DSF ROUTINES
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME      ATTRIBUTES     DEFINITION
CPR
CPR
CPR   COMMONS USED --
CPR
CPR       CRNDEF
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=================================================
      END
```

```
C============================================================
      SUBROUTINE MARCTO(ISPOC,NUMSP,NUMSOL)
C============================================================
CPR
CPR====================================================
CPR
CPR
CPR   ROUTINE--          MARCTO
CPR
CPR   DATE/WRITTEN BY-- 03/08/84     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE CALLS ALL REQUIRED ROUTINES TO COMPUTE
CPR   THE MARC 8-NODED ELM GAUSS POINT TEMPERATURES(OR THE
CPR   ELM CENTROIDAL TEMPERATURE)
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR       ISPOC     (I)   DESIRED ELM LOCATION
CPR                       ISPOC=2 --- CENTROIDAL TEMPERATURE
CPR                       ISPOC=3 --- 2X2X2 GAUSS POINT TEMPERATURES
CPR       NUMSP     (I)   NUMBER OF STRESS MODEL NODAL POINTS
CPR       NUMSOL    (I)   NUMBER OF TRANSIENT SOLUTIONS
CPR
CPR   COMMONS USED --
CPR
CPR       AFIL
CPR       WORK
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR       GPCOEF
CPR       GPTEMP
CPR
CPR   FILES USED --
CPR
CPR       ISNEF    S
CPR       IOUTF    S
CPR       ITEMP    S
CPR       IERF     S
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR====================================================
      END
```

```
C=============================================================
      SUBROUTINE GPCOEF(GPF)
C=============================================================
CPR
CPR=====================================================
CPR
CPR
CPR   ROUTINE--          GPCOEF
CPR
CPR   DATE/WRITTEN BY-- 03/07/84     RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE COMPUTES AND STORES THE WEIGHTING
CPR   COEFFS ASSOCIATED WITH THE 2X2X2 GAUSS POINTS
CPR   USED FOR THE MARC SOLID ELM(ELM TYPE 43)
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES     DEFINITION
CPR
CPR      GPF        (O)     ARRAY WITH GAUSS PT COEFF
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR====================================================
      END
```

```
C====================================================================
      SUBROUTINE GPTEMP(TA,IE,GPF,ISPOC,TV)
C====================================================================
CPR
CPR=================================================================
CPR
CPR
CPR   ROUTINE--          GPTEMP
CPR
CPR   DATE/WRITTEN BY-- 03/17/84      RJ MAFFEO
CPR   DATE/REVISED BY--
CPR
CPR   FUNCTION/PURPOSE--
CPR
CPR   THIS ROUTINE COMPUTES THE VALUE OF THE ELM TEMPERATURE
CPR   FOR THE 8-NODED 3D MARC STRESS ELM  IT WILL COMPUTE
CPR   TEMPS AT THE 8 2X2X2 GAUSS POINTS OR AT THE ELM
CPR   CENTROID
CPR
CPR
CPR
CPR   CALLING ARGUMENTS --
CPR       NAME       ATTRIBUTES      DEFINITION
CPR
CPR       TA        (I)          NODAL TEMPERATURE ARRAY
CPR       IE        (I)          ELM CONNECTIVITY ARRAY
CPR       GPF       (I)          GAUSS PT COEFF ARRAY
CPR       ISPOC     (I)          TEMPERATURE LOCATION PARAMETER
CPR       TV        (O)          ELEMENTAL TEMPERATURE ARRAY
CPR                            NOTE IF ISPOC=2
CPR                                 TV(1)=CENTROIDAL TEMPERATURE
CPR
CPR   COMMONS USED --
CPR
CPR
CPR   FUNCTIONS/ROUTINES CALLED --
CPR
CPR
CPR   FILES USED --
CPR
CPR
CPR   LIBRARIES ACCESSED --
CPR   NONE
CPR
CPR
CPR   LOCAL VARIABLES --
CPR
CPR   SPECIAL COMMENTS ABOUT THIS ROUTINE
CPR
CPR
CPR   SPECIAL REMARKS/INSTRUCTIONS --
CPR      HOST = H6000
CPR
CPR
CPR=================================================================
      END
```

APPENDIX D

Transfer Module Flow Diagram

This appendix contains a program flow diagram of the transfer module. This diagram shows the general flow of the code and which subroutines are subordinate to each other.

```
                                                  **********************
                                                  *                    *
                                                  *  THE SAVAM STACKER *
                                                  *                    *
                                                  **********************

                                   THIS IS CALLING STACK STARTING FROM  $$$ MAI

==========
MAIN
==========
 \
 \
 \   **********
 \\\*INITCM   *
 \   **********
 \          001
 \
 \
 \   **********
 \\\*RFILCR   *
 \   **********
 \         \ 002
 \         \
 \         \
 \         \         **********
 \         \\\\\\\\*RANSIZ    *
 \         \         **********
 \         \                L003
 \         \
 \         \
 \         \         **********
 \         \\\\\\\\*ZERORF    *
 \                   **********
 \                          004
 \
 \
 \   **********
 \\\*FTIME    *
 \   **********
 \          005
 \
 \
 \   **********
 \\\*HTICON   *
 \   **********
 \         \ 006
 \         \


 \         \
 \         \         **********
 \         \\\\\\\\*SINTIT    *
 \         \         **********
```

```
  \         \               \ 007
  \         \               \
  \         \               \
  \         \               \         **********
  \         \               \\\\\\\\*DATINT   *
  \         \               \         **********
  \         \               \                008
  \         \               \
  \         \               \
  \         \               \         **********
  \         \               \\\\\\\\*PRCSOF   *
  \         \               \         **********
  \         \               \               \ 009
  \         \               \               \
  \         \               \               \
  \         \               \               \         **********
  \         \               \               \\\\\\\\*GETEWT   *
  \         \               \               \         **********
  \         \               \               \               \ 010
  \         \               \               \               \
  \         \               \               \               \
  \         \               \               \               \         **********
  \         \               \               \               \\\\\\\\*ERRPRT   *
  \         \               \               \                         **********
  \         \               \               \                                011
  \         \               \               \
  \         \               \               \
  \         \               \               \         **********
  \         \               \               \\\\\\\\*GTNUMT   *
  \         \               \                         **********
  \         \               \                               \ 012
  \         \               \                               \
  \         \               \                               \
  \         \               \                               \         ==========
  \         \               \                               \\\\\\\\=ERRPRT   =
  \         \               \                                         ==========
  \         \               \                                               RO11
  \         \               \
  \         \               \
  \         \               \         **********
  \         \               \\\\\\\\*PRCGEO   *
  \         \               \         **********
  \         \               \               \ 013
  \         \               \               \
  \         \               \               \
  \         \               \               \         ==========
  \         \               \               \\\\\\\\=ERRPRT   =
  \         \               \                         ==========
  \         \               \                               RO11
  \         \               \




  \         \               \
  \         \               \         **********
  \         \               \\\\\\\\*WRTNTF   *
  \         \                         **********
  \         \                               \ 014
  \         \                               \
  \         \                               \
  \         \                               \         **********
  \         \                               \\\\\\\\*WRTDIR   *
  \         \                               \         **********
```

```
 \      \                       \           \ 015
 \      \                       \           \
 \      \                       \           \
 \      \                       \           \      ==========
 \      \                       \           \\\\\\\=ERRPRT  =
 \      \                       \                  ==========
 \      \                       \                        R011
 \      \                       \
 \      \                       \
 \      \                       \      **********
 \      \                       \\\\\\\*WRTGEO  *
 \      \                       \      **********
 \      \                       \           \ 016
 \      \                       \           \
 \      \                       \           \
 \      \                       \           \      ==========
 \      \                       \           \\\\\\\=ERRPRT  =
 \      \                       \                  ==========
 \      \                       \                        R011
 \      \                       \
 \      \                       \
 \      \                       \      **********
 \      \                       \\\\\\\*WRTEMP  *
 \      \                       \      **********
 \      \                       \           \ 017
 \      \                       \           \
 \      \                       \           \
 \      \                       \           \      ==========
 \      \                       \           \\\\\\\=ERRPRT  =
 \      \                       \                  ==========
 \      \                       \                        R011
 \      \                       \
 \      \                       \
 \      \                       \      ==========
 \      \                       \\\\\\\=ERRPRT  =
 \      \                              ==========
 \      \                                    R011
 \      \
 \      \
 \      \      **********
 \      \\\\\\\*MARCRD  *
 \             **********
 \                  \ 018
 \                  \


 \                  \
 \                  \      **********
 \                  \\\\\\\*READR   *
 \                  \      **********
 \                  \             019
 \                  \
 \                  \
 \                  \      **********
 \                  \\\\\\\*SKPREC  *
 \                         **********
 \                                020
 \
 \
 \  **********
 \\\*RDCNTL  *
 \  **********
```

```
 \       \ 021
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*HTFIO    *
 \       \       **********
 \       \              022
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*ELTRAN   *
 \       \       **********
 \                      023
 \
 \
 \  **********
 \\\*MKRGF    *
 \  **********
 \       \ 024
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*RWNODS   *
 \       \       **********
 \       \              025
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*EMINMX   *
 \       \       **********
 \       \              \ 026
 \       \              \
 \       \              \
 \       \              \       **********
 \       \              \\\\\\\\*SORT     *
 \       \                      **********
 \       \                             027
 \       \



 \       \
 \       \       **********
 \       \\\\\\\\*RWELMS   *
 \       \       **********
 \       \              028
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*PROFAC   *
 \               **********
 \                      \ 029
 \                      \
 \                      \
 \                      \       ==========
 \                      \\\\\\\\=RWELMS   =
 \                              ==========
 \                                    R028
 \
 \
 \  **********
 \\\*PRNTIM   *
 \  **********
```

```
 \         030
 \
 \
 \  **********
 \\\*RDTIME  *
 \  **********
 \       \ 031
 \       \
 \       \
 \       \       ==========
 \       \\\\\\\\=HTFIO    =
 \       \       ==========
 \       \             R022
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*SKIP     *
 \       \       **********
 \       \              032
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*TIMCK    *
 \       \       **********
 \       \              033
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*CHNODT   *
 \               **********
 \                     \ 034
 \                     \


 \                     \
 \                     \       ==========
 \                     \\\\\\\\=HTFIO    =
 \                     \       ==========
 \                     \             R022
 \                     \
 \                     \
 \                     \       ==========
 \                     \\\\\\\\=RWELMS   =
 \                             ==========
 \                                   R028
 \
 \
 \  **********
 \\\*PREPGM  *
 \  **********
 \       \ 035
 \       \
 \       \
 \       \       ==========
 \       \\\\\\\\=RWELMS   =
 \       \       ==========
 \       \             R028
 \       \
 \       \
 \       \       **********
 \       \\\\\\\\*CTRAN    *
 \       \       **********
```

```
\       \               036
\       \
\       \
\       \       **********
\       \\\\\\\\*SCTION  *
\       \       **********
\       \               037
\       \
\       \
\       \       **********
\       \\\\\\\\*REMELM  *
\               **********
\                     \ 038
\                     \
\                     \
\                     \       ==========
\                     \\\\\\\\=RWELMS  =
\                             ==========
\                                   R028
\
\
\  **********
\\\*CALCRD  *
\  **********
\       \ 039
\       \
```

```
\       \
\       \       **********
\       \\\\\\\\*GETPD   *
\       \       **********
\       \             \ 040
\       \             \
\       \             \
\       \             \       **********
\       \             \\\\\\\\*PARD    *
\       \             \       **********
\       \             \              041
\       \             \
\       \             \
\       \             \       **********
\       \             \\\\\\\\*GTGAUS  *
\       \                     **********
\       \                            042
\       \
\       \
\       \       ==========
\       \\\\\\\\=RWELMS  =
\       \       ==========
\       \             R028
\       \
\       \
\       \       **********
\       \\\\\\\\*VOLM    *
\       \       **********
\       \             \ 043
\       \             \
\       \             \
\       \             \       **********
\       \             \\\\\\\\*SHAPEF  *
\       \                     **********
```

```
\       \                          044
\       \
\       \
\       \      **********
\       \\\\\\\*AREAD    *
\       \      **********
\       \           \ 045
\       \           \
\       \           \
\       \           \      **********
\       \           \\\\\\\*GTGAUS   *
\       \           \      **********
\       \           \            R042
\       \           \
\       \           \
\       \           \      **********
\       \           \\\\\\\*SHAPEF   *
\       \                  **********
\       \                        R044
\       \



\       \
\       \      **********
\       \\\\\\\*REACRD   *
\              **********
\                     046
\
\
\  **********
\\*SSURCH   *
\  **********
\       \ 047
\       \
\       \
\       \      **********
\       \\\\\\\*KNTSPF   *
\       \      **********
\       \           \ 048
\       \           \
\       \           \
\       \           \      **********
\       \           \\\\\\\*FILREW   *
\       \           \      **********
\       \           \             049
\       \           \
\       \           \
\       \           \      **********
\       \           \\\\\\\*RDSPF    *
\       \                  **********
\       \                         050
\       \
\       \
\       \      **********
\       \\\\\\\*GNLSUR   *
\              **********
\                   \ 051
\                   \
\                   \
\                   \      **********
\                   \\\\\\\*EINCOR   *
\                   \      **********
```

```
\               \         \ 052
\               \         \
\               \         \
\               \         \      ==========
\               \         \\\\\\\=RWELMS  =
\               \                ==========
\               \                      R028
\               \
\               \
\               \      ==========
\               \\\\\\\=RDSPF   =
\               \      ==========
\               \            R050
\               \



\               \
\               \      **********
\               \\\\\\\*FNDELM  *
\                      **********
\                         \ 053
\                         \
\                         \
\                         \      **********
\                         \\\\\\\*GETWND  *
\                         \      **********
\                         \         \ 054
\                         \         \
\                         \         \
\                         \         \      ==========
\                         \         \\\\\\\=RWELMS  =
\                         \                ==========
\                         \                      R028
\                         \
\                         \
\                         \      **********
\                         \\\\\\\*CKWIND  *
\                         \      **********
\                         \             055
\                         \
\                         \
\                         \      **********
\                         \\\\\\\*I3DSF   *
\                         \      **********
\                         \         \ 056
\                         \         \
\                         \         \
\                         \         \      **********
\                         \         \\\\\\\*REAARY  *
\                         \         \      **********
\                         \         \         \ 057
\                         \         \         \
\                         \         \         \
\                         \         \         \      **********
\                         \         \         \\\\\\\*SORT2   *
\                         \         \                **********
\                         \         \                       058
\                         \         \
\                         \         \
\                         \         \      **********
\                         \         \\\\\\\*SHAPFV  *
\                         \         \      **********
```

```
  \                                 \             \             059
  \                                 \             \
  \                                 \             \
  \                                 \             \       **********
  \                                 \             \\\\\\\\*CHKPT    *
  \                                 \             \       **********
  \                                 \             \             060
  \                                 \             \



  \                                 \             \
  \                                 \             \       **********
  \                                 \             \\\\\\\\*JACBCK   *
  \                                 \             \       **********
  \                                 \             \             \ 061
  \                                 \             \             \
  \                                 \             \             \
  \                                 \             \             \       **********
  \                                 \             \             \\\\\\\\*PDOFSF   *
  \                                 \             \             \       **********
  \                                 \             \             \             062
  \                                 \             \             \
  \                                 \             \             \
  \                                 \             \             \       **********
  \                                 \             \             \\\\\\\\*FNDJAC   *
  \                                 \             \                     **********
  \                                 \             \                           063
  \                                 \             \
  \                                 \             \
  \                                 \             \       **********
  \                                 \             \\\\\\\\*SFDIST   *
  \                                 \                     **********
  \                                 \                           \ 064
  \                                 \                           \
  \                                 \                           \
  \                                 \                           \       ==========
  \                                 \                           \\\\\\\\=SHAPFV   =
  \                                 \                                   ==========
  \                                 \                                        R059
  \                                 \
  \                                 \
  \                                 \       ==========
  \                                 \\\\\\\\=RWELMS   =
  \                                 \       ==========
  \                                 \            R028
  \                                 \
  \                                 \
  \                                 \       **********
  \                                 \\\\\\\\*RWSPCF   *
  \                                         **********
  \                                              065
  \
  \
  \  **********
  \\\*CORNCC   *
  \  **********
  \       \ 066
  \       \
  \       \
  \       \       ==========
  \       \\\\\\\\=RWELMS   =
  \       \       ==========
```

```
  \        \                 R028
  \        \




  \        \
  \        \         ==========
  \        \\\\\\\\=SORT     =
  \        \         ==========
  \        \                R027
  \        \
  \        \
  \        \         **********
  \        \\\\\\\\*CCNVEC   *
  \        \         **********
  \        \                067
  \        \
  \        \
  \        \         **********
  \        \\\\\\\\*GETCRD   *
  \        \         **********
  \        \                068
  \        \
  \        \
  \        \         **********
  \        \\\\\\\\*CINV     *
  \        \         **********
  \        \                069
  \        \
  \        \
  \        \         **********
  \        \\\\\\\\*RWCCF    *
  \        \         **********
  \        \                070
  \        \
  \        \
  \        \         **********
  \        \\\\\\\\*PUTWFI   *
  \                  **********
  \                         071
  \
  \   **********
  \\\*GENTMP   *
  \   **********
  \        \ 072
  \        \
  \        \
  \        \         **********
  \        \\\\\\\\*TFLSET   *
  \        \         **********
  \        \               \ 073
  \        \               \
  \        \               \
  \        \               \         ==========
  \        \               \\\\\\\\=SKIP     =
  \        \                         ==========
  \        \                                R032
  \        \
```

```
 \         \
 \         \          **********
 \         \\\\\\\\*GETMPE   *
 \         \          **********
 \         \               \ 074
 \         \               \
 \         \               \
 \         \               \          ==========
 \         \               \\\\\\\\=HTFIO    =
 \         \               \          ==========
 \         \               \                R022
 \         \               \
 \         \               \
 \         \               \          **********
 \         \               \\\\\\\\*MPERW    *
 \         \                          **********
 \         \                                 075
 \         \
 \         \
 \         \          **********
 \         \\\\\\\\*SETFLP   *
 \         \          **********
 \         \               \ 076
 \         \               \
 \         \               \
 \         \               \          ==========
 \         \               \\\\\\\\=TFLSET   =
 \         \               \          ==========
 \         \               \                R073
 \         \               \
 \         \               \
 \         \               \          **********
 \         \               \\\\\\\\*HTFPAS   *
 \         \                          **********
 \         \                                 077
 \         \
 \         \
 \         \          **********
 \         \\\\\\\\*RDTEMP   *
 \         \          **********
 \         \               \ 078
 \         \               \
 \         \               \
 \         \               \          ==========
 \         \               \\\\\\\\=HTFIO    =
 \         \                          ==========
 \         \                                R022
 \         \
 \         \
 \         \          **********
 \         \\\\\\\\*CWCRNT   *
 \         \          **********
 \         \               \ 079
 \         \               \


 \         \               \
 \         \               \          ==========
 \         \               \\\\\\\\=FILREW   =
```

```
\       \             \      ==========
\       \             \            R049
\       \             \
\       \             \
\       \             \      ==========
\       \             \\\\\\\=RWCCF    =
\       \             \      ==========
\       \             \            R070
\       \             \
\       \             \
\       \             \      **********
\       \             \\\\\\\*UNPAKF   *
\       \             \      **********
\       \             \             080
\       \             \
\       \             \
\       \             \      ==========
\       \             \\\\\\\=MPERW    =
\       \             \      ==========
\       \             \            R075
\       \             \
\       \             \
\       \             \      **********
\       \             \\\\\\\*FCRNTP   *
\       \             \      **********
\       \             \             081
\       \             \
\       \             \
\       \             \      **********
\       \             \\\\\\\*MAXTDF   *
\       \             \      **********
\       \             \            \ 082
\       \             \            \
\       \             \            \
\       \             \            \      ==========
\       \             \            \\\\\\\=SORT     =
\       \             \                   ==========
\       \             \                         R027
\       \             \
\       \             \
\       \             \      **********
\       \             \\\\\\\*WRCRNT   *
\       \             \      **********
\       \             \             083
\       \
\       \
\       \      **********
\       \\\\\\\*ETCORE   *
\       \      **********
\       \             \ 084
\       \             \
```

```
\       \             \
\       \             \      ==========
\       \             \\\\\\\=FILREW   =
\       \                    ==========
\       \                          R049
\       \
\       \      **********
\       \\\\\\\*NTCORE   *
```

```
  \        \        **********
  \        \             \ 085
  \        \             \
  \        \             \
  \        \             \        ==========
  \        \             \\\\\\\\=HTFIO    =
  \        \                      ==========
  \        \                            R022
  \        \
  \        \
  \        \        **********
  \        \\\\\\\\*STSTMP   *
  \                 **********
  \                      \ 086
  \                      \
  \                      \
  \                      \        ==========
  \                      \\\\\\\\=RWSPCF   =
  \                      \        ==========
  \                      \              R065
  \                      \
  \                      \
  \                      \        **********
  \                      \\\\\\\\*WRSTMP   *
  \                               **********
  \                                     087
  \
  \
  \   **********
   \\*TEMFOR   *
      **********
           \ 088
           \
           \
           \        **********
           \\\\\\\\*TPNAST   *
           \        **********
           \               089
           \
           \
           \        **********
           \\\\\\\\*MARCTO   *
                    **********
                         \ 090
                         \




                         \
                         \        **********
                         \\\\\\\\*GPCOEF   *
                         \        **********
                         \               091
                         \
                         \
                         \        **********
                         \\\\\\\\*GPTEMP   *
                                  **********
                                         092
```

MAIN	SU	1		MAIN		NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		AFIL	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		ELMDAT	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		SIZE	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		WORK	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		CNTLGM	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		CNTLTM	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		SPDAT	C	NEWSRC	100001
MAIN	CO	3		CNTLFL	C	NEWSRC	100001
MAIN	CO	3		MAFCOM	M	NEWSRC	100001
MAIN	CO	3		CRNDEF	C	NEWSRC	100001
MAIN	SU	5	101	INITCM		NEWSRC	100001
MAIN	SU	5	102	RFILCR		NEWSRC	100001
MAIN	SU	5	103	FTIME		NEWSRC	100001
MAIN	SU	5	104	HTICON		NEWSRC	100001
MAIN	SU	5	105	RDCNTL		NEWSRC	100001
MAIN	SU	5	106	MKRGF		NEWSRC	100001
MAIN	SU	5	108	PRNTIM		NEWSRC	100001
MAIN	SU	5	109	RDTIME		NEWSRC	100001
MAIN	SU	5	112	PREPGM		NEWSRC	100001
MAIN	SU	5	115	CALCRD		NEWSRC	100001
MAIN	SU	5	118	SSURCH		NEWSRC	100001
MAIN	SU	5	121	CORNCC		NEWSRC	100001
MAIN	SU	5	124	GENTMP		NEWSRC	100001
MAIN	SU	5	127	TEMFOR		NEWSRC	100001
MAIN	RD	7		YES		NEWSRC	100001
MAIN	WR	8		YES		NEWSRC	100001
RWNODS	SU	1		RWNODS		NEWSRC	100002
RWNODS	CO	3		MAFCOM	M	NEWSRC	100002
RWNODS	CO	3		AFIL	C	NEWSRC	100002
RWNODS	RD	7		YES		NEWSRC	100002
RWNODS	WR	8		YES		NEWSRC	100002
RWELMS	SU	1		RWELMS		NEWSRC	100003
RWELMS	CO	3		MAFCOM	M	NEWSRC	100003
RWELMS	CO	3		AFIL	C	NEWSRC	100003
RWELMS	RD	7		YES		NEWSRC	100003
RWELMS	WR	8		YES		NEWSRC	100003
RDTIME	SU	1		RDTIME		NEWSRC	100004
RDTIME	CO	3		MAFCOM	M	NEWSRC	100004
RDTIME	CO	3		AFIL	C	NEWSRC	100004
RDTIME	CO	3		MAFCOM	M	NEWSRC	100004
RDTIME	CO	3		SIZE	C	NEWSRC	100004
RDTIME	CO	3		MAFCOM	M	NEWSRC	100004
RDTIME	CO	3		CNTLTM	C	NEWSRC	100004
RDTIME	CO	3		CNTLFL	C	NEWSRC	100004
RDTIME	SU	5	101	HTFIO		NEWSRC	100004
RDTIME	SU	5	102	SKIP		NEWSRC	100004
RDTIME	SU	5	104	TIMCK		NEWSRC	100004
RDTIME	SU	5	105	CHNODT		NEWSRC	100004
RDTIME	WR	8		YES		NEWSRC	100004
TIMCK	SU	1		TIMCK		NEWSRC	100005
TIMCK	CO	3		MAFCOM	M	NEWSRC	100005
TIMCK	CO	3		CNTLTM	C	NEWSRC	100005
CHNODT	SU	1		CHNODT		NEWSRC	100006
CHNODT	CO	3		MAFCOM	M	NEWSRC	100006
CHNODT	CO	3		AFIL	C	NEWSRC	100006
CHNODT	CO	3		CNTLFL	C	NEWSRC	100006
CHNODT	CO	3		MAFCOM	M	NEWSRC	100006
CHNODT	CO	3		ELMDAT	C	NEWSRC	100006
CHNODT	CO	3		MAFCOM	M	NEWSRC	100006

CHNODT	CO	3		SIZE	C	NEWSRC	100006
CHNODT	CO	3		MAFCOM	M	NEWSRC	100006
CHNODT	CO	3		WORK	C	NEWSRC	100006
CHNODT	SU	5	101	HTFIO		NEWSRC	100006
CHNODT	SU	5	102	RWELMS		NEWSRC	100006
CHNODT	WR	8		YES		NEWSRC	100006
PREPGM	SU	1		PREPGM		NEWSRC	100007
PREPGM	CO	3		MAFCOM	M	NEWSRC	100007
PREPGM	CO	3		ELMDAT	C	NEWSRC	100007
PREPGM	CO	3		MAFCOM	M	NEWSRC	100007
PREPGM	CO	3		SIZE	C	NEWSRC	100007
PREPGM	CO	3		MAFCOM	M	NEWSRC	100007
PREPGM	CO	3		CNTLGM	C	NEWSRC	100007
PREPGM	SU	5	101	RWELMS		NEWSRC	100007
PREPGM	SU	5	102	CTRAN		NEWSRC	100007
PREPGM	SU	5	103	SCTION		NEWSRC	100007
PREPGM	SU	5	105	REMELM		NEWSRC	100007
CTRAN	SU	1		CTRAN		NEWSRC	100008
SCTION	SU	1		SCTION		NEWSRC	100009
REMELM	SU	1		REMELM		NEWSRC	100010
REMELM	CO	3		MAFCOM	M	NEWSRC	100010
REMELM	CO	3		ELMDAT	C	NEWSRC	100010
REMELM	CO	3		MAFCOM	M	NEWSRC	100010
REMELM	CO	3		CNTLGM	C	NEWSRC	100010
REMELM	SU	5	101	RWELMS		NEWSRC	100010
CALCRD	SU	1		CALCRD		NEWSRC	100011
CALCRD	CO	3		MAFCOM	M	NEWSRC	100011
CALCRD	CO	3		AFIL	C	NEWSRC	100011
CALCRD	CO	3		MAFCOM	M	NEWSRC	100011
CALCRD	CO	3		ELMDAT	C	NEWSRC	100011
CALCRD	CO	3		MAFCOM	M	NEWSRC	100011
CALCRD	CO	3		SIZE	C	NEWSRC	100011
CALCRD	CO	3		MAFCOM	M	NEWSRC	100011
CALCRD	CO	3		WORK	C	NEWSRC	100011
CALCRD	CO	3		CRNDEF	C	NEWSRC	100011
CALCRD	SU	5	101	GETPD		NEWSRC	100011
CALCRD	SU	5	102	RWELMS		NEWSRC	100011
CALCRD	SU	5	103	VOLM		NEWSRC	100011
CALCRD	SU	5	104	AREAD		NEWSRC	100011
CALCRD	SU	5	105	REACRD		NEWSRC	100011
CALCRD	WR	8		YES		NEWSRC	100011
GETPD	SU	1		GETPD		NEWSRC	100012
GETPD	CO	3		CRNDEF	C	NEWSRC	100012
GETPD	SU	5	101	PARD		NEWSRC	100012
GETPD	SU	5	102	GTGAUS		NEWSRC	100012
VOLM	SU	1		VOLM		NEWSRC	100013
VOLM	SU	5	101	SHAPEF		NEWSRC	100013
SHAPEF	SU	1		SHAPEF		NEWSRC	100014
SHAPEF	CO	3		CRNDEF	C	NEWSRC	100014
PARD	SU	1		PARD		NEWSRC	100015
PARD	CO	3		CRNDEF	C	NEWSRC	100015
AREAD	SU	1		AREAD		NEWSRC	100016
AREAD	CO	3		CRNDEF	C	NEWSRC	100016
AREAD	SU	5	101	GTGAUS		NEWSRC	100016
AREAD	SU	5	102	SHAPEF		NEWSRC	100016
GTGAUS	SU	1		GTGAUS		NEWSRC	100017
REACRD	SU	1		REACRD		NEWSRC	100018
CORNCC	SU	1		CORNCC		NEWSRC	100019
CORNCC	CO	3		MAFCOM	M	NEWSRC	100019
CORNCC	CO	3		AFIL	C	NEWSRC	100019
CORNCC	CO	3		MAFCOM	M	NEWSRC	100019
CORNCC	CO	3		ELMDAT	C	NEWSRC	100019
CORNCC	CO	3		MAFCOM	M	NEWSRC	100019
CORNCC	CO	3		SIZE	C	NEWSRC	100019
CORNCC	CO	3		MAFCOM	M	NEWSRC	100019
CORNCC	CO	3		WORK	C	NEWSRC	100019
CORNCC	CO	3		MAFCOM	M	NEWSRC	100019

CORNCC	CO	3		CNTLGM	C	NEWSRC	100019
CORNCC	SU	5	101	RWELMS		NEWSRC	100019
CORNCC	SU	5	102	SORT		NEWSRC	100019
CORNCC	SU	5	103	CCNVEC		NEWSRC	100019
CORNCC	SU	5	105	GETCRD		NEWSRC	100019
CORNCC	SU	5	106	CINV		NEWSRC	100019
CORNCC	SU	5	108	RWCCF		NEWSRC	100019
CORNCC	SU	5	109	PUTWFI		NEWSRC	100019
CORNCC	WR	8		YES		NEWSRC	100019
CCNVEC	SU	1		CCNVEC		NEWSRC	100020
CCNVEC	CO	3		MAFCOM	M	NEWSRC	100020
CCNVEC	CO	3		AFIL	C	NEWSRC	100020
CCNVEC	WR	8		YES		NEWSRC	100020
GETCRD	SU	1		GETCRD		NEWSRC	100021
GETCRD	CO	3		MAFCOM	M	NEWSRC	100021
GETCRD	CO	3		AFIL	C	NEWSRC	100021
GETCRD	WR	8		YES		NEWSRC	100021
PUTWFI	SU	1		PUTWFI		NEWSRC	100022
PUTWFI	CO	3		MAFCOM	M	NEWSRC	100022
PUTWFI	CO	3		AFIL	C	NEWSRC	100022
RWCCF	SU	1		RWCCF		NEWSRC	100023
RWCCF	CO	3		MAFCOM	M	NEWSRC	100023
RWCCF	CO	3		AFIL	C	NEWSRC	100023
RWCCF	RD	7		YES		NEWSRC	100023
RWCCF	WR	8		YES		NEWSRC	100023
GETMPE	SU	1		GETMPE		NEWSRC	100024
GETMPE	CO	3		MAFCOM	M	NEWSRC	100024
GETMPE	CO	3		AFIL	C	NEWSRC	100024
GETMPE	CO	3		CNTLFL	C	NEWSRC	100024
GETMPE	CO	3		MAFCOM	M	NEWSRC	100024
GETMPE	CO	3		SIZE	C	NEWSRC	100024
GETMPE	SU	5	101	HTFIO		NEWSRC	100024
GETMPE	SU	5	103	MPERW		NEWSRC	100024
MPERW	SU	1		MPERW		NEWSRC	100025
MPERW	WR	8		YES		NEWSRC	100025
SETFLP	SU	1		SETFLP		NEWSRC	100026
SETFLP	CO	3		MAFCOM	M	NEWSRC	100026
SETFLP	CO	3		AFIL	C	NEWSRC	100026
SETFLP	SU	5	101	TFLSET		NEWSRC	100026
SETFLP	SU	5	102	HTFPAS		NEWSRC	100026
RDTEMP	SU	1		RDTEMP		NEWSRC	100027
RDTEMP	CO	3		MAFCOM	M	NEWSRC	100027
RDTEMP	CO	3		AFIL	C	NEWSRC	100027
RDTEMP	CO	3		MAFCOM	M	NEWSRC	100027
RDTEMP	CO	3		SIZE	C	NEWSRC	100027
RDTEMP	CO	3		MAFCOM	M	NEWSRC	100027
RDTEMP	CO	3		CNTLFL	C	NEWSRC	100027
RDTEMP	SU	5	101	HTFIO		NEWSRC	100027
RDTEMP	WR	8		YES		NEWSRC	100027
CWCRNT	SU	1		CWCRNT		NEWSRC	100028
CWCRNT	CO	3		MAFCOM	M	NEWSRC	100028
CWCRNT	CO	3		AFIL	C	NEWSRC	100028
CWCRNT	SU	5	101	FILREW		NEWSRC	100028
CWCRNT	SU	5	102	RWCCF		NEWSRC	100028
CWCRNT	SU	5	103	UNPAKF		NEWSRC	100028
CWCRNT	SU	5	104	MPERW		NEWSRC	100028
CWCRNT	SU	5	105	FCRNTP		NEWSRC	100028
CWCRNT	SU	5	106	MAXTDF		NEWSRC	100028
CWCRNT	SU	5	107	WRCRNT		NEWSRC	100028
UNPAKF	SU	1		UNPAKF		NEWSRC	100029
UNPAKF	CO	3		MAFCOM	M	NEWSRC	100029
UNPAKF	CO	3		AFIL	C	NEWSRC	100029
UNPAKF	WR	8		YES		NEWSRC	100029
FCRNTP	SU	1		FCRNTP		NEWSRC	100030
FCRNTP	CO	3		MAFCOM	M	NEWSRC	100030
FCRNTP	CO	3		AFIL	C	NEWSRC	100030
MAXTDF	SU	1		MAXTDF		NEWSRC	100031

MAXTDF	CO	3		MAFCOM	M	NEWSRC	100031
MAXTDF	CO	3		AFIL	C	NEWSRC	100031
MAXTDF	SU	5	101	SORT		NEWSRC	100031
MAXTDF	WR	8		YES		NEWSRC	100031
WRITMP	SU	1		WRITMP		NEWSRC	100032
WRITMP	CO	3		MAFCOM	M	NEWSRC	100032
WRITMP	CO	3		AFIL	C	NEWSRC	100032
WRITMP	WR	8		YES		NEWSRC	100032
CINV	SU	1		CINV		NEWSRC	100033
CINV	CO	3		MAFCOM	M	NEWSRC	100033
CINV	CO	3		AFIL	C	NEWSRC	100033
CINV	WR	8		YES		NEWSRC	100033
PRNTIM	SU	1		PRNTIM		NEWSRC	100034
PRNTIM	CO	3		MAFCOM	M	NEWSRC	100034
PRNTIM	CO	3		AFIL	C	NEWSRC	100034
PRNTIM	WR	8		YES		NEWSRC	100034
EMINMX	SU	1		EMINMX		NEWSRC	100035
EMINMX	SU	5	101	SORT		NEWSRC	100035
FTIME	SU	1		FTIME		NEWSRC	100036
SSURCH	SU	1		SSURCH		NEWSRC	100037
SSURCH	CO	3		MAFCOM	M	NEWSRC	100037
SSURCH	CO	3		AFIL	C	NEWSRC	100037
SSURCH	CO	3		MAFCOM	M	NEWSRC	100037
SSURCH	CO	3		SIZE	C	NEWSRC	100037
SSURCH	CO	3		MAFCOM	M	NEWSRC	100037
SSURCH	CO	3		SPDAT	C	NEWSRC	100037
SSURCH	SU	5	101	KNTSPF		NEWSRC	100037
SSURCH	SU	5	102	GNLSUR		NEWSRC	100037
KNTSPF	SU	1		KNTSPF		NEWSRC	100038
KNTSPF	CO	3		MAFCOM	M	NEWSRC	100038
KNTSPF	CO	3		AFIL	C	NEWSRC	100038
KNTSPF	SU	5	101	FILREW		NEWSRC	100038
KNTSPF	SU	5	102	RDSPF		NEWSRC	100038
KNTSPF	WR	8		YES		NEWSRC	100038
RFILCR	SU	1		RFILCR		NEWSRC	100039
RFILCR	CO	3		MAFCOM	M	NEWSRC	100039
RFILCR	CO	3		ELMDAT	C	NEWSRC	100039
RFILCR	SU	5	101	RANSIZ		NEWSRC	100039
RFILCR	SU	5	102	ZERORF		NEWSRC	100039
ZERORF	SU	1		ZERORF		NEWSRC	100040
ZERORF	WR	8		YES		NEWSRC	100040
GNLSUR	SU	1		GNLSUR		NEWSRC	100041
GNLSUR	CO	3		MAFCOM	M	NEWSRC	100041
GNLSUR	CO	3		AFIL	C	NEWSRC	100041
GNLSUR	CO	3		MAFCOM	M	NEWSRC	100041
GNLSUR	CO	3		SIZE	C	NEWSRC	100041
GNLSUR	CO	3		MAFCOM	M	NEWSRC	100041
GNLSUR	CO	3		WORK	C	NEWSRC	100041
GNLSUR	SU	5	101	EINCOR		NEWSRC	100041
GNLSUR	SU	5	102	RDSPF		NEWSRC	100041
GNLSUR	SU	5	103	FNDELM		NEWSRC	100041
GNLSUR	WR	8		YES		NEWSRC	100041
RDSPF	SU	1		RDSPF		NEWSRC	100042
RDSPF	CO	3		MAFCOM	M	NEWSRC	100042
RDSPF	CO	3		AFIL	C	NEWSRC	100042
RDSPF	RD	7		YES		NEWSRC	100042
FILREW	SU	1		FILREW		NEWSRC	100043
EINCOR	SU	1		EINCOR		NEWSRC	100044
EINCOR	CO	3		MAFCOM	M	NEWSRC	100044
EINCOR	CO	3		AFIL	C	NEWSRC	100044
EINCOR	CO	3		MAFCOM	M	NEWSRC	100044
EINCOR	CO	3		ELMDAT	C	NEWSRC	100044
EINCOR	SU	5	101	RWELMS		NEWSRC	100044
FNDELM	SU	1		FNDELM		NEWSRC	100045
FNDELM	CO	3		MAFCOM	M	NEWSRC	100045
FNDELM	CO	3		AFIL	C	NEWSRC	100045
FNDELM	CO	3		MAFCOM	M	NEWSRC	100045

FNDELM	CO	3		ELMDAT	C	NEWSRC	100045
FNDELM	CO	3		MAFCOM	M	NEWSRC	100045
FNDELM	CO	3		SIZE	C	NEWSRC	100045
FNDELM	CO	3		MAFCOM	M	NEWSRC	100045
FNDELM	CO	3		SPDAT	C	NEWSRC	100045
FNDELM	SU	5	101	GETWND		NEWSRC	100045
FNDELM	SU	5	102	CKWIND		NEWSRC	100045
FNDELM	SU	5	104	I3DSF		NEWSRC	100045
FNDELM	SU	5	105	RWELMS		NEWSRC	100045
FNDELM	SU	5	109	RWSPCF		NEWSRC	100045
FNDELM	WR	8		YES		NEWSRC	100045
GETWND	SU	1		GETWND		NEWSRC	100046
GETWND	CO	3		MAFCOM	M	NEWSRC	100046
GETWND	CO	3		AFIL	C	NEWSRC	100046
GETWND	CO	3		MAFCOM	M	NEWSRC	100046
GETWND	CO	3		ELMDAT	C	NEWSRC	100046
GETWND	CO	3		MAFCOM	M	NEWSRC	100046
GETWND	CO	3		SIZE	C	NEWSRC	100046
GETWND	CO	3		MAFCOM	M	NEWSRC	100046
GETWND	CO	3		WORK	C	NEWSRC	100046
GETWND	SU	5	101	RWELMS		NEWSRC	100046
CKWIND	SU	1		CKWIND		NEWSRC	100047
MKRGF	SU	1		MKRGF		NEWSRC	100048
MKRGF	CO	3		MAFCOM	M	NEWSRC	100048
MKRGF	CO	3		AFIL	C	NEWSRC	100048
MKRGF	CO	3		MAFCOM	M	NEWSRC	10Q048
MKRGF	CO	3		ELMDAT	C	NEWSRC	100048
MKRGF	CO	3		MAFCOM	M	NEWSRC	100048
MKRGF	CO	3		SIZE	C	NEWSRC	100048
MKRGF	CO	3		MAFCOM	M	NEWSRC	100048
MKRGF	CO	3		CNTLFL	C	NEWSRC	100048
MKRGF	SU	5	101	RWNODS		NEWSRC	100048
MKRGF	SU	5	103	EMINMX		NEWSRC	100048
MKRGF	SU	5	104	RWELMS		NEWSRC	100048
MKRGF	SU	5	105	PROFAC		NEWSRC	100048
MKRGF	RD	7		YES		NEWSRC	100048
MKRGF	WR	8		YES		NEWSRC	100048
PROFAC	SU	1		PROFAC		NEWSRC	100049
PROFAC	CO	3		MAFCOM	M	NEWSRC	100049
PROFAC	CO	3		AFIL	C	NEWSRC	100049
PROFAC	CO	3		MAFCOM	M	NEWSRC	100049
PROFAC	CO	3		ELMDAT	C	NEWSRC	100049
PROFAC	CO	3		MAFCOM	M	NEWSRC	100049
PROFAC	CO	3		SIZE	C	NEWSRC	100049
PROFAC	SU	5	101	RWELMS		NEWSRC	100049
PROFAC	RD	7		YES		NEWSRC	100049
PROFAC	WR	8		YES		NEWSRC	100049
SKIP	SU	1		SKIP		NEWSRC	100050
SKIP	CO	3		MAFCOM	M	NEWSRC	100050
SKIP	CO	3		SIZE	C	NEWSRC	100050
SKIP	RD	7		YES		NEWSRC	100050
HTFIO	SU	1		HTFIO		NEWSRC	100051
HTFIO	RD	7		YES		NEWSRC	100051
RDCNTL	SU	1		RDCNTL		NEWSRC	100052
RDCNTL	CO	3		MAFCOM	M	NEWSRC	100052
RDCNTL	CO	3		AFIL	C	NEWSRC	100052
RDCNTL	CO	3		MAFCOM	M	NEWSRC	100052
RDCNTL	CO	3		CNTLFL	C	NEWSRC	100052
RDCNTL	CO	3		MAFCOM	M	NEWSRC	100052
RDCNTL	CO	3		CNTLTM	C	NEWSRC	100052
RDCNTL	CO	3		MAFCOM	M	NEWSRC	100052
RDCNTL	CO	3		SIZE	C	NEWSRC	100052
RDCNTL	SU	5	101	HTFIO		NEWSRC	100052
RDCNTL	SU	5	102	ELTRAN		NEWSRC	100052
RDCNTL	RD	7		YES		NEWSRC	100052
RDCNTL	WR	8		YES		NEWSRC	100052
TFLSET	SU	1		TFLSET		NEWSRC	100053

TFLSET	CO	3		MAFCOM	M	NEWSRC	100053
TFLSET	CO	3		AFIL	C	NEWSRC	100053
TFLSET	CO	3		MAFCOM	M	NEWSRC	100053
TFLSET	CO	3		CNTLTM	C	NEWSRC	100053
TFLSET	SU	5	101	SKIP		NEWSRC	100053
TFLSET	RD	7		YES		NEWSRC	100053
GENTMP	SU	1		GENTMP		NEWSRC	100054
GENTMP	CO	3		MAFCOM	M	NEWSRC	100054
GENTMP	CO	3		MAFCOM	M	NEWSRC	100054
GENTMP	CO	3		AFIL	C	NEWSRC	100054
GENTMP	CO	3		MAFCOM	M	NEWSRC	100054
GENTMP	CO	3		SIZE	C	NEWSRC	100054
GENTMP	CO	3		MAFCOM	M	NEWSRC	100054
GENTMP	CO	3		WORK	C	NEWSRC	100054
GENTMP	CO	3		MAFCOM	M	NEWSRC	100054
GENTMP	CO	3		CNTLTM	C	NEWSRC	100054
GENTMP	CO	3		CNTLFL	C	NEWSRC	100054
GENTMP	SU	5	101	TFLSET		NEWSRC	100054
GENTMP	SU	5	102	GETMPE		NEWSRC	100054
GENTMP	SU	5	103	SETFLP		NEWSRC	100054
GENTMP	SU	5	104	RDTEMP		NEWSRC	100054
GENTMP	SU	5	105	CWCRNT		NEWSRC	100054
GENTMP	SU	5	106	ETCORE		NEWSRC	100054
GENTMP	SU	5	107	NTCORE		NEWSRC	100054
GENTMP	SU	5	108	STSTMP		NEWSRC	100054
STSTMP	SU	1		STSTMP		NEWSRC	100055
STSTMP	CO	3		MAFCOM	M	NEWSRC	100055
STSTMP	CO	3		MAFCOM	M	NEWSRC	100055
STSTMP	CO	3		AFIL	C	NEWSRC	100055
STSTMP	CO	3		MAFCOM	M	NEWSRC	100055
STSTMP	CO	3		SIZE	C	NEWSRC	100055
STSTMP	CO	3		SPDAT	C	NEWSRC	100055
STSTMP	SU	5	101	RWSPCF		NEWSRC	100055
STSTMP	SU	5	102	WRSTMP		NEWSRC	100055
WRSTMP	SU	1		WRSTMP		NEWSRC	100056
WRSTMP	CO	3		MAFCOM	M	NEWSRC	100056
WRSTMP	CO	3		MAFCOM	M	NEWSRC	100056
WRSTMP	CO	3		AFIL	C	NEWSRC	100056
WRSTMP	WR	8		YES		NEWSRC	100056
ETCORE	SU	1		ETCORE		NEWSRC	100057
ETCORE	CO	3		MAFCOM	M	NEWSRC	100057
ETCORE	CO	3		MAFCOM	M	NEWSRC	100057
ETCORE	CO	3		AFIL	C	NEWSRC	100057
ETCORE	CO	3		MAFCOM	M	NEWSRC	100057
ETCORE	CO	3		SIZE	C	NEWSRC	100057
ETCORE	SU	5	101	FILREW		NEWSRC	100057
ETCORE	RD	7		YES		NEWSRC	100057
NTCORE	SU	1		NTCORE		NEWSRC	100058
NTCORE	CO	3		MAFCOM	M	NEWSRC	100058
NTCORE	CO	3		MAFCOM	M	NEWSRC	100058
NTCORE	CO	3		AFIL	C	NEWSRC	100058
NTCORE	CO	3		MAFCOM	M	NEWSRC	100058
NTCORE	CO	3		SIZE	C	NEWSRC	100058
NTCORE	SU	5	101	HTFIO		NEWSRC	100058
WRCRNT	SU	1		WRCRNT		NEWSRC	100059
WRCRNT	CO	3		MAFCOM	M	NEWSRC	100059
WRCRNT	CO	3		MAFCOM	M	NEWSRC	100059
WRCRNT	CO	3		AFIL	C	NEWSRC	100059
WRCRNT	WR	8		YES		NEWSRC	100059
HTFPAS	SU	1		HTFPAS		NEWSRC	100060
HTFPAS	CO	3		MAFCOM	M	NEWSRC	100060
HTFPAS	CO	3		SIZE	C	NEWSRC	100060
HTFPAS	RD	7		YES		NEWSRC	100060
SORT	SU	1		SORT		NEWSRC	100061
HTICON	SU	1		HTICON		NEWSRC	100062
HTICON	CO	3		AFIL	C	NEWSRC	100062
HTICON	SU	5	101	SINTIT		NEWSRC	100062

HTICON	SU	5	102	MARCRD		NEWSRC	100062
HTICON	WR	8		YES		NEWSRC	100062
MARCRD	SU	1		MARCRD		NEWSRC	100063
MARCRD	CO	3		AFIL	C	NEWSRC	100063
MARCRD	SU	5	101	READR		NEWSRC	100063
MARCRD	SU	5	103	SKPREC		NEWSRC	100063
MARCRD	WR	8		YES		NEWSRC	100063
READR	SU	1		READR		NEWSRC	100064
READR	CO	3		AFIL	C	NEWSRC	100064
READR	RD	7		YES		NEWSRC	100064
READR	WR	8		YES		NEWSRC	100064
SKPREC	SU	1		SKPREC		NEWSRC	100065
SKPREC	CO	3		AFIL	C	NEWSRC	100065
SKPREC	RD	7		YES		NEWSRC	100065
SKPREC	WR	8		YES		NEWSRC	100065
SINTIT	SU	1		SINTIT		NEWSRC	100066
SINTIT	CO	3		MAFCOM	M	NEWSRC	100066
SINTIT	CO	3		WORK	C	NEWSRC	100066
SINTIT	CO	3		MAFCOM	M	NEWSRC	100066
SINTIT	CO	3		AFIL	C	NEWSRC	100066
SINTIT	CO	3		KNTDAT	C	NEWSRC	100066
SINTIT	SU	5	101	DATINT		NEWSRC	100066
SINTIT	SU	5	102	PRCSOF		NEWSRC	100066
SINTIT	SU	5	103	PRCGEO		NEWSRC	100066
SINTIT	SU	5	104	WRTNTF		NEWSRC	100066
DATINT	SU	1		DATINT		NEWSRC	100067
DATINT	CO	3		MAFCOM	M	NEWSRC	100067
DATINT	CO	3		WORK	C	NEWSRC	100067
DATINT	CO	3		MAFCOM	M	NEWSRC	100067
DATINT	CO	3		AFIL	C	NEWSRC	100067
DATINT	CO	3		KNTDAT	C	NEWSRC	100067
PRCSOF	SU	1		PRCSOF		NEWSRC	100068
PRCSOF	CO	3		MAFCOM	M	NEWSRC	100068
PRCSOF	CO	3		AFIL	C	NEWSRC	100068
PRCSOF	SU	5	101	GETEWT		NEWSRC	100068
PRCSOF	SU	5	102	GTNUMT		NEWSRC	100068
GETEWT	SU	1		GETEWT		NEWSRC	100069
GETEWT	CO	3		MAFCOM	M	NEWSRC	100069
GETEWT	CO	3		WORK	C	NEWSRC	100069
GETEWT	CO	3		MAFCOM	M	NEWSRC	100069
GETEWT	CO	3		AFIL	C	NEWSRC	100069
GETEWT	CO	3		KNTDAT	C	NEWSRC	100069
GETEWT	SU	5	101	ERRPRT		NEWSRC	100069
GETEWT	RD	7		YES		NEWSRC	100069
GTNUMT	SU	1		GTNUMT		NEWSRC	100070
GTNUMT	CO	3		MAFCOM	M	NEWSRC	100070
GTNUMT	CO	3		WORK	C	NEWSRC	100070
GTNUMT	CO	3		MAFCOM	M	NEWSRC	100070
GTNUMT	CO	3		AFIL	C	NEWSRC	100070
GTNUMT	CO	3		KNTDAT	C	NEWSRC	100070
GTNUMT	SU	5	101	ERRPRT		NEWSRC	100070
GTNUMT	RD	7		YES		NEWSRC	100070
PRCGEO	SU	1		PRCGEO		NEWSRC	100071
PRCGEO	CO	3		MAFCOM	M	NEWSRC	100071
PRCGEO	CO	3		AFIL	C	NEWSRC	100071
PRCGEO	CO	3		KNTDAT	C	NEWSRC	100071
PRCGEO	SU	5	101	ERRPRT		NEWSRC	100071
PRCGEO	RD	7		YES		NEWSRC	100071
WRTNTF	SU	1		WRTNTF		NEWSRC	100072
WRTNTF	CO	3		MAFCOM	M	NEWSRC	100072
WRTNTF	CO	3		WORK	C	NEWSRC	100072
WRTNTF	CO	3		MAFCOM	M	NEWSRC	100072
WRTNTF	CO	3		AFIL	C	NEWSRC	100072
WRTNTF	CO	3		KNTDAT	C	NEWSRC	100072
WRTNTF	SU	5	101	WRTDIR		NEWSRC	100072
WRTNTF	SU	5	102	WRTGEO		NEWSRC	100072
WRTNTF	SU	5	103	WRTEMP		NEWSRC	100072

WRTNTF	SU	5	104	ERRPRT		NEWSRC	100072
WRTNTF	RD	7		YES		NEWSRC	100072
WRTDIR	SU	1		WRTDIR		NEWSRC	100073
WRTDIR	CO	3		MAFCOM	M	NEWSRC	100073
WRTDIR	CO	3		WORK	C	NEWSRC	100073
WRTDIR	CO	3		MAFCOM	M	NEWSRC	100073
WRTDIR	CO	3		AFIL	C	NEWSRC	100073
WRTDIR	CO	3		KNTDAT	C	NEWSRC	100073
WRTDIR	SU	5	101	ERRPRT		NEWSRC	100073
WRTDIR	RD	7		YES		NEWSRC	100073
WRTDIR	WR	8		YES		NEWSRC	100073
WRTGEO	SU	1		WRTGEO		NEWSRC	100074
WRTGEO	CO	3		MAFCOM	M	NEWSRC	100074
WRTGEO	CO	3		WORK	C	NEWSRC	100074
WRTGEO	CO	3		MAFCOM	M	NEWSRC	100074
WRTGEO	CO	3		AFIL	C	NEWSRC	100074
WRTGEO	CO	3		KNTDAT	C	NEWSRC	100074
WRTGEO	SU	5	101	ERRPRT		NEWSRC	100074
WRTGEO	RD	7		YES		NEWSRC	100074
WRTGEO	WR	8		YES		NEWSRC	100074
WRTEMP	SU	1		WRTEMP		NEWSRC	100075
WRTEMP	CO	3		MAFCOM	M	NEWSRC	100075
WRTEMP	CO	3		WORK	C	NEWSRC	100075
WRTEMP	CO	3		MAFCOM	M	NEWSRC	100075
WRTEMP	CO	3		AFIL	C	NEWSRC	100075
WRTEMP	CO	3		KNTDAT	C	NEWSRC	100075
WRTEMP	SU	5	101	ERRPRT		NEWSRC	100075
WRTEMP	RD	7		YES		NEWSRC	100075
WRTEMP	WR	8		YES		NEWSRC	100075
ERRPRT	SU	1		ERRPRT		NEWSRC	100076
ERRPRT	CO	3		MAFCOM	M	NEWSRC	100076
ERRPRT	CO	3		AFIL	C	NEWSRC	100076
ERRPRT	WR	8		YES		NEWSRC	100076
TEMFOR	SU	1		TEMFOR		NEWSRC	100077
TEMFOR	CO	3		MAFCOM	M	NEWSRC	100077
TEMFOR	CO	3		AFIL	C	NEWSRC	100077
TEMFOR	CO	3		MAFCOM	M	NEWSRC	100077
TEMFOR	CO	3		SIZE	C	NEWSRC	100077
TEMFOR	SU	5	101	TPNAST		NEWSRC	100077
TEMFOR	SU	5	102	MARCTO		NEWSRC	100077
TEMFOR	WR	8		YES		NEWSRC	100077
TPNAST	SU	1		TPNAST		NEWSRC	100078
TPNAST	CO	3		AFIL	C	NEWSRC	100078
TPNAST	RD	7		YES		NEWSRC	100078
TPNAST	WR	8		YES		NEWSRC	100078
I3DSF	SU	1		I3DSF		NEWSRC	100079
I3DSF	CO	3		MAFCOM	M	NEWSRC	100079
I3DSF	CO	3		AFIL	C	NEWSRC	100079
I3DSF	SU	5	101	REAARY		NEWSRC	100079
I3DSF	SU	5	102	SHAPFV		NEWSRC	100079
I3DSF	SU	5	103	CHKPT		NEWSRC	100079
I3DSF	SU	5	104	JACBCK		NEWSRC	100079
I3DSF	SU	5	106	SFDIST		NEWSRC	100079
REAARY	SU	1		REAARY		NEWSRC	100080
REAARY	SU	5	101	SORT2		NEWSRC	100080
SORT2	SU	1		SORT2		NEWSRC	100081
SFDIST	SU	1		SFDIST		NEWSRC	100082
SFDIST	SU	5	101	SHAPFV		NEWSRC	100082
CHKPT	SU	1		CHKPT		NEWSRC	100083
PDOFSF	SU	1		PDOFSF		NEWSRC	100084
PDOFSF	CO	3		CRNDEF	C	NEWSRC	100084
FNDJAC	SU	1		FNDJAC		NEWSRC	100085
SHAPFV	SU	1		SHAPFV		NEWSRC	100086
JACBCK	SU	1		JACBCK		NEWSRC	100087
JACBCK	SU	5	101	PDOFSF		NEWSRC	100087
JACBCK	SU	5	102	FNDJAC		NEWSRC	100087
RWSPCF	SU	1		RWSPCF		NEWSRC	100088

RWSPCF	CO	3		MAFCOM	M	NEWSRC	100088
RWSPCF	CO	3		AFIL	C	NEWSRC	100088
RWSPCF	RD	7		YES		NEWSRC	100088
RWSPCF	WR	8		YES		NEWSRC	100088
ELTRAN	SU	1		ELTRAN		NEWSRC	100089
ELTRAN	CO	3		MAFCOM	M	NEWSRC	100089
ELTRAN	CO	3		AFIL	C	NEWSRC	100089
ELTRAN	CO	3		MAFCOM	M	NEWSRC	100089
ELTRAN	CO	3		CNTLFL	C	NEWSRC	100089
ELTRAN	WR	8		YES		NEWSRC	100089
INITCM	SU	1		INITCM		NEWSRC	100090
INITCM	CO	3		CRNDEF	C	NEWSRC	100090
MARCTO	SU	1		MARCTO		NEWSRC	100091
MARCTO	CO	3		MAFCOM	M	NEWSRC	100091
MARCTO	CO	3		AFIL	C	NEWSRC	100091
MARCTO	CO	3		MAFCOM	M	NEWSRC	100091
MARCTO	CO	3		WORK	C	NEWSRC	100091
MARCTO	SU	5	101	GPCOEF		NEWSRC	100091
MARCTO	SU	5	102	GPTEMP		NEWSRC	100091
MARCTO	RD	7		YES		NEWSRC	100091
MARCTO	WR	8		YES		NEWSRC	100091
GPCOEF	SU	1		GPCOEF		NEWSRC	100092
GPTEMP	SU	1		GPTEMP		NEWSRC	100093

REFERENCES

1. "Survey of Computer Programs For Heat Transfer Analysis"

    A.R. Noor

2. 3D TITAN User's Manual

    R.J. Maffeo

3. PATRAN

    PDA Engineering

    Software Products Division

    1560 Brookhollow Drive

    Santa Ana, California 92705-5475

End of Document