National Aeronautics and Space Administration

NASA-CR-174892
19860012461

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

LIBRARY COPY
MAY 28 1986

Hampton, Virginia
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 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 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, Isoparametric, Finite Difference, Transfer Module

For sale by the National Technical Information Service, Springfield, Virginia 22161
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0   SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>2.0   INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>3.0   DESCRIPTION OF TASKS</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Task I - Current Status of Thermal Load Modeling</td>
<td>5</td>
</tr>
<tr>
<td>3.2 Task II - Assessment of Thermal Load Modeling</td>
<td>5</td>
</tr>
<tr>
<td>3.3 Task III - Development of Thermal Load Transfer Module</td>
<td>6</td>
</tr>
<tr>
<td>3.4 Task IV - Verification and Documentation of Automated Automated Thermal Transfer Load Module</td>
<td>6</td>
</tr>
<tr>
<td>4.0   TASK I</td>
<td>8</td>
</tr>
<tr>
<td>4.1 Thermal Analysis Programs</td>
<td>8</td>
</tr>
<tr>
<td>4.2 Structural Analysis Codes</td>
<td>9</td>
</tr>
<tr>
<td>5.0   TASK II</td>
<td>11</td>
</tr>
<tr>
<td>5.1 2D Titan Evaluation</td>
<td>11</td>
</tr>
<tr>
<td>5.2 Merlin Evaluation</td>
<td>12</td>
</tr>
<tr>
<td>5.3 Lomap Evaluation</td>
<td>12</td>
</tr>
<tr>
<td>5.4 Evaluation of Transferring Criteria</td>
<td>13</td>
</tr>
<tr>
<td>6.0   DEVELOPMENT AND DOCUMENTATION OF THE TRANSFER MODULE</td>
<td>17</td>
</tr>
<tr>
<td>6.1 General Flow of Code</td>
<td>17</td>
</tr>
<tr>
<td>6.2 Input and Output</td>
<td>19</td>
</tr>
<tr>
<td>6.3 Finite Difference Consideration</td>
<td>21</td>
</tr>
<tr>
<td>6.4 3D Search and Weighting Coefficient Routines</td>
<td>24</td>
</tr>
<tr>
<td>6.5 Exterior Point Surfacing Routine</td>
<td>31</td>
</tr>
<tr>
<td>7.0   VERIFICATION AND DOCUMENTATION OF THE TRANSFER MODULE</td>
<td>34</td>
</tr>
<tr>
<td>7.1 Verification of the Transfer Module</td>
<td>34</td>
</tr>
<tr>
<td>7.2 Documentation of the Transfer</td>
<td>39</td>
</tr>
<tr>
<td>8.0   CONCLUSIONS</td>
<td>51</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>53</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>66</td>
</tr>
<tr>
<td>APPENDIX C</td>
<td>75</td>
</tr>
<tr>
<td>APPENDIX D</td>
<td>180</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>203</td>
</tr>
</tbody>
</table>
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 element 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>
<thead>
<tr>
<th>Criteria</th>
<th>MERLIN</th>
<th>LOMAP</th>
<th>TITAN</th>
<th>TRANCITS Proposed Transfer Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent Meshes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2-D Capability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3-D Capability</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Thermal Mapping a Function of Heat Transfer Elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordinate Transformations</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>'Out-of-Box'</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Provision Isoparametric</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mapping Geometrical/Temporal</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windowing Capability For Time Interpolation</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
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
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.
INITIALIZE INTERNAL VARIABLES

READ/PROCESS USER CONTROL DATA

CREATE INTERNAL FILE STRUCTURE

READ/STORE HEAT TRANSFER INPUT DATA

READ STRESS MODEL INPUT DATA

PROCESS GEOMETRICAL DATA (ELEMENT TRANSFORMATION, ELIMINATION, ERROR CHECKING)

COMPUTE ADDITIONAL GEOMETRY FOR FINITE DIFFERENCE PROBLEMS

SEARCH FOR CONTAINING HEAT TRANSFER ELEMENT

COMPUTE EQUIVALENT SURFACE LOCATION FOR STRESS POINTS OUTSIDE HEAT TRANSFER MODEL

COMPUTE CORNER TEMPERATURES FOR FINITE DIFFERENCE ELEMENT

COMPUTE WEIGHTING COEFFICIENTS FOR ALL STRESS POINTS

COMPUTE TEMPERATURES FOR ALL STRESS POINTS FOR ALL DESIRED TIMES

FORMAT OUTPUT TEMPERATURES

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, geometrical 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 tempera-
tures 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 iso-
parametric shape functions. The equations that must be solved are of the
form:

\[ T = [1, x, y, z] [c]^{-1} \bar{T} \]

where

- \( T \) = temperature at vertex
- \( x \)
- \( y = \) coordinates of vertex
- \( z \)
- \( \bar{T} = \) 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

- \( X_{IJKL} \)
- \( Y_{IJKL} = \) coordinate of the tetrahedral
- \( Z_{IJKL} \)

Note that \([C]^{-1}\) is only a function of geometry and therefore must only be com-
puted 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 corre-
sponding tetrahedral.

Note that during this initial stage the actual temperatures of the ver-
tices are not directly computed, only the coefficients that relate the tempera-
tures 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
vertex. This is done by weighting all of the vertex 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:

\[
\begin{align*}
X_s &< X_{\text{min}} \\
X_s &> X_{\text{max}} \\
Y_s &< Y_{\text{min}} \\
Y_s &> Y_{\text{max}} \\
Z_s &< Z_{\text{min}} \\
Z_s &> Z_{\text{max}}
\end{align*}
\]

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:

\[
\begin{align*}
X &= \sum_{i=1}^{\eta} \alpha_i X_i \\
Y &= \sum_{i=1}^{\eta} \alpha_i Y_i \\
Z &= \sum_{i=1}^{\eta} \alpha_i Z_i \\
T &= \sum_{i=1}^{\eta} \alpha_i T_i
\end{align*}
\]

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, and \(\alpha_i\) 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 $X$, $Y$ to $Y$ and $Z$ to $Z$ yields a set of non-linear equations with $r$, $s$ and $t$ as unknowns. The local coordinates become functions of each other, global vertex coordinates and the coordinates of the stress point.

That is:

$$
\begin{align*}
    r_p &= f(r_p, s_p, t, X_i, Y_i, Z_i, X_p, Y_p, Z_p) \\
    s_o &= f(r_p, s, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p) \quad 6.4.3 \\
    t_p &= f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p)
\end{align*}
$$

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.

$$
\begin{align*}
    -1 > r_p > 1 \\
    -1 > s_p > 1 \\
    -1 > t_p > 1
\end{align*}
$$

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:

26
Figure 6-4. Isoparametric Mapping of Global Coordinates To Local Coordinates.
These initial values correspond to the correct values of \( r, s, t \) 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

\[ s = 1 \]

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.
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 a 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 7-7. 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 7-14. Combustor 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.
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.
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 1

#### Part 1 - Analytical Capabilities of the Program

<table>
<thead>
<tr>
<th></th>
<th>ANSY</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRAN</th>
<th>SINDA</th>
<th>CHTI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Goal of Program System</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Purpose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Educational</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2. Method of Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finite Elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finite Differences</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Integral Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perturbation Technique</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid Analytical - Numerical Technique (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3. Space Dimensionality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Three-Dimensional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-Dimensional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One-Dimensional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Extracted from "Survey of Computer Programs for Heat Analysis" by A.K. Noor.
<table>
<thead>
<tr>
<th>Solids of Revolution</th>
<th>ANSYS</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRAN</th>
<th>NX-TRAN</th>
<th>SINDA</th>
<th>THD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Scalar Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Point Contact Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

4. Range of Applications and Phenomena

<table>
<thead>
<tr>
<th>Linear Steady State</th>
<th>ANSYS</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRAN</th>
<th>NX-TRAN</th>
<th>SINDA</th>
<th>THD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonlinear Steady State</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Thermal Frequencies and Mode Shapes</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Linear Transient Response</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Nonlinear Transient Response</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

5. Formulation

a) Fundamental Unknowns

<table>
<thead>
<tr>
<th>Temperatures</th>
<th>ANSYS</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRAN</th>
<th>NX-TRAN</th>
<th>SINDA</th>
<th>THD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperatures and Flux</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Enthalpy</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

b) Elemental Matrices

<table>
<thead>
<tr>
<th>Conduction</th>
<th>ANSYS</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRAN</th>
<th>NX-TRAN</th>
<th>SINDA</th>
<th>THD</th>
</tr>
</thead>
<tbody>
<tr>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Capacitance</td>
<td>ANSYS</td>
<td>HEATING</td>
<td>NASC</td>
<td>NAS-TRAM</td>
<td>SINDA</td>
<td>THTD</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>a) Consistent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Lumped</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Free</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Forced</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interelement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convection and</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Supplied</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elements (see</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Material Properties and Material Models

<table>
<thead>
<tr>
<th>Type</th>
<th>ANSYS</th>
<th>HEATING</th>
<th>NASC</th>
<th>NAS-TRAM</th>
<th>SINDA</th>
<th>THTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isotropic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anisotropic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multilayered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dependent</td>
<td>Conductivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td>Specific Heat</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Density</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Absorptivity (Emissivity Factors)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Convection Coefficients</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perfect Conductors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Via Multipoint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thermal Properties</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latent Heat and Phase Change Effects</td>
<td>A/SYS</td>
<td>HEATING</td>
<td>MARC</td>
<td>NAS-TRANS</td>
<td>SANDA</td>
<td>THID</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Material Added or Removed During Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Supplied (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Initial Conditions

<table>
<thead>
<tr>
<th>Homogeneous</th>
<th>A/SYS</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRANS</th>
<th>SANDA</th>
<th>THID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Varying Throughout the Region</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Enthalpy (for Phase Change)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User-Supplied</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Boundary Conditions and Thermal Loads

<table>
<thead>
<tr>
<th>Prescribed Temperatures</th>
<th>a) Steady State</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b) Time Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thermal Flux Input</td>
<td>a) Steady</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>b) Temperature Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>c) Time Varying</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convection from a Surface to its Surroundings</td>
<td>a) Steady State</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>b) Time Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convection from Surroundings to a Surface</td>
<td>a) Steady State</td>
<td>b) Time Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------------</td>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forced Convection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation from a Surface to its Surroundings</td>
<td>a) Steady State</td>
<td>b) Time Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation from Surroundings to a Surface</td>
<td>a) Steady State</td>
<td>b) Time Dependent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation Between Narrow Gaps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation Between Surfaces with</td>
<td>a) User-Supplied View Factors</td>
<td>b) Internally Calculated View Factors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prescribed Fluid Flow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Layer Convection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volumetric Heat Generation</td>
<td>a) On Element Level</td>
<td>b) On Node Level</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gap (Thermal Resistance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Conditions/Loads Added or Removed During Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Solution Techniques</td>
<td>ANSYS</td>
<td>HEATING</td>
<td>MARC</td>
<td>NAS-TRAN</td>
<td>STIDA</td>
<td>THID</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Linear Steady State</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Direct</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Iterative</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Others (see abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonlinear Steady State</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Incremental</td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Iterative</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Others (see abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transient</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Thermal Mode Superposition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Direct Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) Explicit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ii) Implicit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Specified Time Step</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Time Step Selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iii) Combined Explicit/Implicit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Finite Elements in the Time Domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d) Moving - Deforming Grids</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e) Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Other Capabilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thermal Stress Analysis Capability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Uncoupled</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Coupled</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>ANSYS</td>
<td>HEATING</td>
<td>MARC</td>
<td>NAS-TRAN</td>
<td>STIDA</td>
<td>THID</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Temperature Field Data Transmitted Directly from Heat Transfer Modules to Thermal Stress Modules</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enclosure Radiation with View Factor Calculation</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heat Input/Output at Constrained Boundaries</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyclic Symmetry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Substructuring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Repeated Use of Identical Substructures</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Mixing Linear and Nonlinear Substructures</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Mixing Substructures with Different Types of Nonlinearities</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Restart Capability</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
</tbody>
</table>

**II. Program Operational On**

<table>
<thead>
<tr>
<th>Vendor</th>
<th>ANSYS</th>
<th>HEATING</th>
<th>MARC</th>
<th>NAS-TRAN</th>
<th>STIDA</th>
<th>THID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDC</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>IBM</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>UNIVAC</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Honeywell</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Telefunken</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Amdahl</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Siemens</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>ICL</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>ANSYS</td>
<td>HEAT-</td>
<td>MARC</td>
<td>HAST-</td>
<td>SINDA</td>
<td>THED</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Minicomputers (see program abstracts)</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Supercomputers (see program abstracts)</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td><strong>12. Documentation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmer's Manual</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Theoretical Manual</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Data Preparation - Users' Manual</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Example Problem Manual</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Verification/Validation Manual</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Pre- and Post-Processors' Manual</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>
### Part II - User Interface and Modeling Capabilities

<table>
<thead>
<tr>
<th></th>
<th>ANSYS</th>
<th>HEATING</th>
<th>NARC</th>
<th>NAS-TRAN</th>
<th>STINDA</th>
<th>TRED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Input Form and Sequence</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Input Form</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed Format</td>
<td></td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free Form-List Directed Format</td>
<td></td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Oriented Language</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Input Sequence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Directed</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Directed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Supplied Subroutines (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2. Model Generation and Checking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Automatic or Semi-Automatic Generator for:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modal Point Coordinates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element Connectivities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints, Symmetry and Boundary Conditions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Substructure Connectivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repetition of Identical Segments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ANSYS</td>
<td>HEAT-INGS</td>
<td>MARC</td>
<td>NAS-TRAN</td>
<td>SINDA</td>
<td>T-HED</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------</td>
<td>-----------</td>
<td>------</td>
<td>----------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>b) Automatic or Semi-Automatic Generator For:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One-Dimensional Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Triangular Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Quadrilateral Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Body or Shell of Revolution Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Three-Dimensional Solid Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Two-Dimensional Shell Elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Curvilinear Finite Difference Grids</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td><strong>c) Data Checking Facilities</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line Printer</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Plotter</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Interactive Graphics</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td><strong>d) Plots and Graphics Display of Model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete Analysis Region</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Part of Analysis Region</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>&quot;Blow-Up&quot; Option</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Feature</td>
<td>ANSYS</td>
<td>HEAT-TEC</td>
<td>MARC</td>
<td>NAS-TRAN</td>
<td>STAN</td>
<td>EHL</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td>Hidden Lines or Surfaces</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orthographic Views</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perspective and Isometric Views</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Section View on Arbitrary Plane</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Other Facilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digitizer Input</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Renumbering of Nodes, Elements or Equations</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Lookup of Data</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Results Output Form</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Tabular Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed Set</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Defined Set and Sequences</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum and Minimum Quantities</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average and Maxima for Blocks of Nodes</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature or Flux Exceedances</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) File Output for User Post-Processing and Plotting</td>
<td>ANSYS</td>
<td>HEATING</td>
<td>MNSC</td>
<td>NAS-TRAN</td>
<td>STINDA</td>
<td>THED</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------</td>
<td>---------</td>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>c) Plots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Isotherm Plots (Contours) of Temperatures/Flux</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Surface Functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selective Output (e.g., by Elements or Regions)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Histories (e.g., Time History)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Others (see program abstracts)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

4. Interactive Input and Control

- Parameter Specification (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.
<table>
<thead>
<tr>
<th>TYPES OF ANALYSIS</th>
<th>ANALYTICAL CAPABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINEAR STATICS</td>
<td></td>
</tr>
<tr>
<td>MECHANICAL LOADS</td>
<td>✔</td>
</tr>
<tr>
<td>TEMPERATURE LOADS</td>
<td>✔</td>
</tr>
<tr>
<td>EULER BUCKLING</td>
<td></td>
</tr>
<tr>
<td>INERTIA RELIEF</td>
<td></td>
</tr>
<tr>
<td>DYNAMICS</td>
<td></td>
</tr>
<tr>
<td>MODE/FREQUENCY</td>
<td>✔</td>
</tr>
<tr>
<td>FREQUENCY RESPONSE</td>
<td></td>
</tr>
<tr>
<td>TRANSIENT RESPONSE</td>
<td>✔</td>
</tr>
<tr>
<td>SHOCK SPECTRA</td>
<td></td>
</tr>
<tr>
<td>RANDOM RESPONSE</td>
<td></td>
</tr>
<tr>
<td>NONLINEAR TRANSIENT</td>
<td></td>
</tr>
<tr>
<td>NONLINEAR STATICS</td>
<td></td>
</tr>
<tr>
<td>NONLINEAR BUCKLING</td>
<td></td>
</tr>
<tr>
<td>LARGE DISPLACEMENT</td>
<td></td>
</tr>
<tr>
<td>PLASTICITY</td>
<td></td>
</tr>
<tr>
<td>CREEP</td>
<td></td>
</tr>
<tr>
<td>VISCOELASTICITY</td>
<td></td>
</tr>
<tr>
<td>LARGE STRAINS</td>
<td></td>
</tr>
<tr>
<td>HEAT TRANSFER</td>
<td></td>
</tr>
<tr>
<td>STEADY STATE</td>
<td></td>
</tr>
<tr>
<td>TRANSIENT</td>
<td></td>
</tr>
<tr>
<td>SUBSTRUCTURES (SUPER-ELEMENTS)</td>
<td></td>
</tr>
<tr>
<td>STATIC</td>
<td></td>
</tr>
<tr>
<td>DYNAMIC</td>
<td></td>
</tr>
<tr>
<td>CYCLIC SYMMETRY</td>
<td></td>
</tr>
<tr>
<td>MISCELLANEOUS</td>
<td></td>
</tr>
<tr>
<td>FRACTURE MECHANICS</td>
<td></td>
</tr>
<tr>
<td>FLUIDS</td>
<td></td>
</tr>
<tr>
<td>ELECTRIC CIRCUITS</td>
<td></td>
</tr>
<tr>
<td>OPTIMIZATION</td>
<td></td>
</tr>
<tr>
<td>ACOUSTIC CAVITIES</td>
<td></td>
</tr>
<tr>
<td>FATIGUE DAMAGE</td>
<td></td>
</tr>
</tbody>
</table>
## STRUCTURAL ANALYSIS ELEMENT/MATRIX LIBRARY

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>EASE2</th>
<th>STADYNE</th>
<th>NASTRAN</th>
<th>ANSYS</th>
<th>MARC</th>
<th>SUPER9</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROD</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>BEAM</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>TAPERED BEAM</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>OFFSET BEAM</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>PINNED END BEAM</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>CURVED BEAM</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>3 NODE TRIANGLE</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>6 NODE TRIANGLE</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>SHEAR PANEL</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>4 NODE QUAD</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>8 NODE QUAD</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>3 NODE TRIANGLE</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>6 NODE TRIANGLE</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>4 NODE QUAD</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>8 NODE QUAD</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
<tr>
<td>REDUCED THICK SHELL</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
<td>¥</td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>EASE2</th>
<th>STARDYNE</th>
<th>NASTRAN</th>
<th>ANSYS</th>
<th>MARC</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHELLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
</tr>
<tr>
<td>CURVED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIANGULAR RINGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>6 NODE</td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>QUAD RINGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 NODE</td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td>S, C</td>
<td></td>
</tr>
<tr>
<td>TETRAHEDRON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>WEDGES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 NODE</td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>15 NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>HEXAHEXEDRONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 NODE</td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td>S, C</td>
<td></td>
</tr>
<tr>
<td>PIPE ELEMENTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRAIGHT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELBOW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>EASE2</th>
<th>STARDYNE</th>
<th>NASTRAN</th>
<th>ANSYS</th>
<th>MARC</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GENERAL STIFFNESS ELEMENTS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRING</td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCALAR SPRING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 x 6 or 12 x 12 MATRIX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERAL MATRIX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MASSES</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NON-STRUCTURAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUMPED (DIAGONAL)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSISTENT</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DAMPING</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCALAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DASHPOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISCRETE VISCOUS ( [C] = [K] + \beta [M] )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRUCTURAL ( [1 + \beta i][K] )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODAL VISCOUS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERAL MATRIX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OTHER ELEMENTS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRICTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIGID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REBAR SOLID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELASTIC FOUNDATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRACK TIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAMINATED SHELL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOT ONLY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTES:**
1. See restraints
2. Generated from density
3. See constraints
## HEAT TRANSFER—CONDUCTING ELEMENTS

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>EASE2</th>
<th>STARDYNE</th>
<th>NASTRAN</th>
<th>ANSYS</th>
<th>MARC</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 NODE TRIANGLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P,C</td>
</tr>
<tr>
<td>4 NODE QUAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 NODE QUAD</td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td></td>
<td>S,C</td>
</tr>
<tr>
<td>PLANAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSVERSE CONDUCTING SHELL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXISYMMETRIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIANGULAR RING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P,C</td>
</tr>
<tr>
<td>4 NODE QUAD RING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 NODE QUAD RING</td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td></td>
<td>S,C</td>
</tr>
<tr>
<td>SOLID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TETRAHEDRON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEDGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>8 NODE BRICK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 NODE WEDGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td>20 NODE BRICK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S,C</td>
</tr>
<tr>
<td>GENERAL MATRIX INPUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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

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

**Notes:**

1. Performed by user subroutine
## Boundary Conditions

<table>
<thead>
<tr>
<th>Feature</th>
<th>EASE2</th>
<th>STARDYNE</th>
<th>NASTRAN</th>
<th>ANSYS</th>
<th>MARC</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>CONCENTRATED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DISTRIBUTED (BEAM)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pressure</td>
<td>PLATES/SHELLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AXISYMMETRIC ELEMENTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SOLIDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMPERATURE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACCELERATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ROTATIONAL VELOCITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>COMBINATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>AXI-SYMMETRIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AXISYMMETRIC SHELLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AXISYMMETRIC RINGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>TIME DEPENDENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FREQUENCY DEPENDENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PSD RANDOM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHOCK SPECTRUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>SINGLE POINT*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MULTI POINT*</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>SPECIFIED NONZERO DISPLACEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heat Transfer</td>
<td>HEAT SOURCE/SINK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONVENTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RADIATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPECIFIED TEMPERATURE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

*Single point constraint is enforced zero translation(s) and/or rotation(s) in coordinate(s) associated with a node point.

*Multi-point constraint is 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

<table>
<thead>
<tr>
<th>Feature</th>
<th>EASE2</th>
<th>STARDYNE</th>
<th>NASTRAN</th>
<th>ANSYS</th>
<th>MARC</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undeformed geometry</td>
<td>+</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node labels</td>
<td>+</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Element labels</td>
<td>+</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Property labels</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>2-D sections</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Boundary condition labels</td>
<td>+</td>
<td></td>
<td>•</td>
<td>+</td>
<td>•</td>
<td>+</td>
</tr>
<tr>
<td>Hidden lines removed</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deformed geometry</td>
<td>+</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Contours 2D structure</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Contours solid structure</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Time history</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Frequency response</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Power spectral density</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Arbitrary X VS. Y</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td><strong>Data generation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nodes</td>
<td>1</td>
<td>1.2</td>
<td>1.2.3</td>
<td>1.2</td>
<td>2.3</td>
<td>1.2</td>
</tr>
<tr>
<td>Elements</td>
<td>1</td>
<td>1</td>
<td>1.2.3</td>
<td>1.2</td>
<td>2.3</td>
<td>1.2</td>
</tr>
<tr>
<td>Restraints</td>
<td>1</td>
<td>1</td>
<td>1.2</td>
<td>1</td>
<td>2.3</td>
<td>1</td>
</tr>
<tr>
<td>Loads</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2.3</td>
</tr>
<tr>
<td><strong>Output sorting</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>By load cases</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>By element</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Max/Min summary</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Selected nodes and/or elements</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td><strong>Bandwidth minimization</strong></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>W</td>
<td>•,W</td>
<td>•,W</td>
</tr>
</tbody>
</table>

**Notes:**
1. Generates data in 1 "dimensions"    
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.
**DIRECTORY OF SUBROUTINES/FUNCTIONS - SORTED ALPHABETICALLY BY ROUTINE NAME**

<table>
<thead>
<tr>
<th>FILE NAME</th>
<th>SUBROUTINE/ FUNCTION</th>
<th>PAGE NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE/TRANCITS</td>
<td>AREAD</td>
<td>17</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CALCRD</td>
<td>12</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CONVEC</td>
<td>22</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CHKPT</td>
<td>92</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CHNDT</td>
<td>7</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CINV</td>
<td>37</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CKWND</td>
<td>53</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CORNCC</td>
<td>21</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CTRAN</td>
<td>9</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>CWCRNT</td>
<td>31</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>DATINT</td>
<td>75</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>EINCOR</td>
<td>49</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>ELTRAN</td>
<td>98</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>EMINMX</td>
<td>40</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>ERRRPT</td>
<td>84</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>ETCORE</td>
<td>64</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>FCRNTP</td>
<td>34</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>FILREW</td>
<td>48</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>FNDELM</td>
<td>50</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>FNDJAC</td>
<td>94</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>FTIME</td>
<td>41</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GENTMP</td>
<td>61</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GETCRD</td>
<td>23</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GETEWT</td>
<td>77</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GETMEP</td>
<td>26</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GETPD</td>
<td>13</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GETWNO</td>
<td>52</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GNSUR</td>
<td>46</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GPCOEF</td>
<td>101</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GPTEMP</td>
<td>102</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GTGAUS</td>
<td>19</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>GTNUMT</td>
<td>78</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>HTFIO</td>
<td>58</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>HTFPAS</td>
<td>67</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>HTICON</td>
<td>69</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>I3DSF</td>
<td>87</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>INITCM</td>
<td>99</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>JACBCK</td>
<td>96</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>KNTSPF</td>
<td>43</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>MARCRO</td>
<td>70</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>MARCTD</td>
<td>100</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>MAXTDF</td>
<td>35</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>MKRGF</td>
<td>54</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>MPERW</td>
<td>27</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>NTCDRE</td>
<td>65</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PARD</td>
<td>16</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PDDFSF</td>
<td>39</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PRCCGE</td>
<td>79</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PRCCSF</td>
<td>76</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PREPGM</td>
<td>8</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PRNTIM</td>
<td>39</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PROFAC</td>
<td>56</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>PUTWFI</td>
<td>24</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RDCNTL</td>
<td>59</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RDSPF</td>
<td>47</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RDTMP</td>
<td>30</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RDTIME</td>
<td>4</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>REAARY</td>
<td>89</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>REACRD</td>
<td>20</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>READR</td>
<td>71</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>REMELM</td>
<td>11</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RFIILCR</td>
<td>44</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RWCCF</td>
<td>25</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RWEMLMS</td>
<td>2</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RNWODS</td>
<td>1</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>RWSPCF</td>
<td>97</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SECTION</td>
<td>10</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SETFLP</td>
<td>29</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SFDIST</td>
<td>91</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SHAPEF</td>
<td>15</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SHPFV</td>
<td>95</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SINIT</td>
<td>73</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SKIP</td>
<td>57</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SKPREM</td>
<td>72</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SORT</td>
<td>68</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SORT2</td>
<td>90</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>SSURCH</td>
<td>42</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>STSTMP</td>
<td>62</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>TEMFOR</td>
<td>85</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>TFLSET</td>
<td>60</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>TIMCK</td>
<td>6</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>TPAST</td>
<td>86</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>UNPAKF</td>
<td>33</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>VOLM</td>
<td>14</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRCRNT</td>
<td>66</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRTMP</td>
<td>36</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRTMP</td>
<td>63</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRTDIR</td>
<td>81</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRTMP</td>
<td>83</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRTGEO</td>
<td>82</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>WRTNTF</td>
<td>80</td>
</tr>
<tr>
<td>GE/TRANCITS</td>
<td>ZERORF</td>
<td>45</td>
</tr>
</tbody>
</table>
SUBROUTINE RWNODS(IREC, NN, XYZ, IRW)

DATE/WRITTEN BY-- 8/04/82  RJ MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS I/O ROUTINE READS AND WRITES THE NODAL INFO
TO THE INTERNAL DATA BASE FILES
THIS FILE IS NOW IFL3
IN THE WRITE MODE IT AUTOMATICALLY ZEROS THIS FILE
FOR NODES NAMES THAT ARE NOT USED
THIS RANDOM FILE (IFL3) COULD EASILY BE REPLACED BY
AN ARRAY IF ENOUGH CORE WAS AVAILABLE

CALLING SEQUENCE-- CALL RWNODS(IREC, NN, XYZ, IRW)

CALLING ARGUMENTS--
IREC (I)  RECORD TO BE PROCESSED
NN (I/O)  NODE NUMBER(NAME)
XYZ (I/O) (3X1)  ARRAY OF COORDINATES
IRW (I)  READ/WRITE SWITCH
         IRW=0----  READ
         IRW=1----  WRITE

FILES USED--
IFL3 --- RANDOM FILE TO STORE COORDINATES

COMMONS USED--
AFIL ---  FILE CODE COMMON

FUNCTIONS/ROUTINES CALLED--
NONE

LIBRARIES ACCESSED--
NONE

LOCAL VARIABLES--
N/A

SPECIAL COMMENTS ABOUT THIS ROUTINE
NONE

SPECIAL REMARKS/INSTRUCTIONS--
HOST = H6000

END

-- GE/TRANCITS

---------- RWNODS  PAGE= 1
SUBROUTINE RWELMS(IREC,A,IRW,ICDE)

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

CPR

ROUTINE-- RWELMS

DATE/WRITTEN BY-- 8/04/82 RJ Maffeo

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE IS AN I/O ROUTINE

ITS PURPOSE IS TO READ/WRITE THE ELEMENT ATTRIBUTES TO THE

INTERNAL DATA BASE FILE

THIS DATA BASE FILE IS RANDOM FILE IFL4

THE ATTRIBUTES ARE

-- ELEMENT NAME

-- COORDINATES OF ALL 15 POINTS IN THE ELM

-- ELEMENT CONNECTIVITY OF THE 8 CORNERS

-- THE PACKED THRD FACE NUMBER

-- VOLUME OF ELEMENT

THE ROUTINE CAN PROCESS DIFFERENT AMOUNTS OF THESE

ATTRIBUTES BASED ON THE VALUE OF ICDE

THE RANDOM FILE IS NOT INITIALIZED PRESENTLY WITH THIS

ROUTINE

CALLING SEQUENCE-- CALL RWELMS(IREC,A,IRW,ICDE)

CALLING ARGUMENTS--

NAME ATTRIBUTES DEFINITION

IREC (I) RECORD NUMBER TO BE PROCESSED

A (I/O) (56X1) ARRAY OF ATTRIBUTES

IRW (I) READ/WRITE SWITCH

IRW=0---- READ

IRW=1---- WRITE

ICDE (I) SWITCH TO SET AMOUNT OF DATA TO PROCESS

ICDE=0 -- PROCESS ALL DATA

ICDE=1 -- PROCESS ELM NAME ONLY

ICDE=2 -- PROCESS ELM NAME AND CORNER COORDINATES

ICDE=4 -- PROCESS EVERYTHING EXCEPT VOLUME

ICDE=5 -- PROCESS ALL DATA

FILES USED--

IFL4 -- RANDOM FILE TO STORE ELM ATTRIBUTES

COMMONS USED--

AFIL -- COMMON TO DEFINE FILECODES

FUNCTIONS/ROUTINES CALLED--

NONE

LIBRARIES ACCESSED--

NONE

LOCAL VARIABLES--

N/A

SPECIAL COMMENTS ABOUT THIS ROUTINE

NONE
SPECIAL REMARKS/INSTRUCTIONS--
HOST = H6000
END
SUBROUTINE RDTIME

ROUTINE-- RDTIME

DATE/Written By-- 8/18/82  RJ MAFFEO
DATE/REvised By-- 9/13/83  SR MARTIN
REASON -- CONVERT FROM F4BT FILES TO THE NEW HEAT XFER OUTPUT FILE FORM

FUNCTION/PURPOSE--
THIS ROUTINE READS THE TIME VALUES FROM THE HEAT XFER OUTPUT FILE AND DETERMINES IF THEY ARE DESIRED FOR THIS RUN. IT AUTOMATICALLY ELIMINATES THE NON-STP TIME STEPS. IT ALSO CALLS THE REQUIRED ROUTINE TO PERFORM INITIAL CHECKS ON TIMES AND ELM COUNTS. THE HEAT TRANSFR OUTPUT FILE IS REWOUND BEFORE RETURNING FROM THIS ROUTINE.

CALLING SEQUENCE-- CALL RDTIME

CALLING ARGUMENTS--
NAME ATTRIBUTES DEFINITION
FILES USED--
INTITS -- HEAT XFER OUTPUT FILE (TITAN INPUT)
IERF -- ERROR REMARK FILE
INOT6 -- HARD COPY FILE

COMMONS USED--
AFIL -- FILE CODES
SIZE -- PROBLEM SIZE PARAMETERS
CNTLTMT -- TIME/TEMPERATURE CONTROL PARAMETERS

FUNCTIONS/Routines CALLED--
TIMCK
CHNODT

LIBRARIES ACCESSED--
NONE

LOCAL VARIABLES--
NSTPTM -- COUNT OF STP TIMES PROCESSED
NCNTT -- COUNT OF TOTAL TIMES PROCESSED
NTMDO -- USER SPECIFIED NUMBER OF STP TIMES TO DO
TIMT -- ARRAY OF TIMES PROCESSED. THIS ARRAY WILL BE USED AS A DIRECTORY TO POINT TO THE LOCATION ON THE HEAT XFER OUTPUT FILE FOR EACH TIME.

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS--
5570CPR
5580CPR
5590CPR
6640

HOST = H6000

***************************
SUBROUTINE TIMCK(TIME, ITIME)

ROUTINE -- TIMCK

DATE/WRITTEN BY -- 8/18/82  RJ MAFFEO

DATE/REVISED BY --

FUNCTION/PURPOSE --

THIS ROUTINE CHECKS TO SEE IF THE PRESENT TIME VALUE

IS ONE OF THE TIMES DESIRED

CALLING SEQUENCE -- CALL TIMCK(TIME, ITIME)

CALLING ARGUMENTS --

NAME  ATTRIBUTES  DEFINITION
TIME  (I)  PRESENT TIME VALUE OF TIME STEP
ITIME (I)  DESIRED SWITCH

ITIME=0  >>> NOT DESIRED
ITIME=1  >>> DESIRED

FILES USED -- NONE

COMMONS USED --

CNTLTM  --- TIME/TEMPERATURE CONTROL PARAMETERS

FUNCTIONS/ROUTINES CALLED -- NONE

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE CHNODT

FUNCTION/PURPOSE--

THIS ROUTINE MAKES TWO BASIC CHECKS ON THE INPUT

IT CHECKS TO SEE THAT THE NUMBER OF ELEMENTS IN EACH TIMESTEP IS GREATER THAN OR EQUAL TO THE NUMBER OF GEOMETRY ELEMENTS.

IT ALSO CHECKS TO INSURE THAT ALL ELMS ON THE NEUTRAL ELM FILE HAVE TEMPERATURES ON THE THT

CALLING SEQUENCE-- CALL CHNODT

CALLING ARGUMENTS--

FILES USED--

FILES USED--

COMMONS USED--

FUNCTIONS/Routines CALLED--

LIBRARIES ACCESSED--

LOCAL VARIABLES--

SPECIAL REMARKS/INSTRUCTIONS--

HOST • H6000

END
SUBROUTINE PREPGM

ROUTINE-- PREPGM

DATE/WRITTEN BY-- 8/18/82 RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE CALLS ALL ROUTINES NEEDED TO
TRANSFORM, SECTION (WINDOW) AND/OR
ELIMINATE UNWANTED ELMS FROM THE PROBLEM

CALLING SEQUENCE-- CALL PREPGM

CALLING ARGUMENTS-- NONE

NAME ATTRIBUTES DEFINITION
NONE NONE NONE

FILES USED-- NONE

COMMONS USED--

SIZE

ELMDAT

CNTLGM

LIBRARIES ACCESSED--

NONE

LOCAL VARIABLES--

JS LC -- WINDOW SWITCH

JS LC=1 >>> ELM IS IN WINDOW

JS LC=1 >>> ELM IS NOT IN WINDOW

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
SUBROUTINE CTRAN(XO,YO,ZO,C)

ROUTINE-- CTRAN

DATE/WRITTEN BY-- 8/18/82 RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE TRANSFORMS THE COORDINATES OF THE EIGHT CORNER POINTS OF EACH ELM.

TRANSFORMS ARE LIMITED TO TRANSLATIONS.

CALLING SEQUENCE-- CALL CTRAN(XO,YO,ZO,C)

CALLING ARGUMENTS--

NAME  ATTRIBUTES  DEFINITION
XO    (I)          X OFFSET
YO    (I)          Y OFFSET
ZO    (I)          Z OFFSET
C     (I/O) ARRAY  COORDINATE ARRAY

FILES USED-- NONE

COMMONS USED-- NONE

FUNCTIONS/ROUTINES CALLED-- NONE

LIBRARIES ACCESSED-- NONE

LOCAL VARIABLES--

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
SUBROUTINE SCTION(C,XMN,YMN,ZMN,XMX,YMX,ZMX,INOT)

ROUTINE-- SCTION

DATE/WRITTEN BY-- 8/18/82   RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE CHECKS TO SEE IF ANY OF THE CORNER
COORDINATES ARE OUTSIDE OF THE WINDOW SPECIFIED
BY THE MIN AND MAX VALUES OF X,Y,Z

NOTE IF ANY POINT OF AN ELM IS OUTSIDE THE WINDOW
THE ELM IS CONSIDERED OUT OF THE WINDOW.

CALLING ARGUMENTS--

NAME    ATTRIBUTES    DEFINITION

C        (I)          COORDINATE ARRAY
XMN >    (I)          MIN VALUE OF X,Y,Z FOR WINDOW
YMN >    (I)          MAX VALUE OF X,Y,Z FOR WINDOW
ZMN >    (I)          MIN VALUE OF X,Y,Z FOR WINDOW
ZMX >    (I)          MAX VALUE OF X,Y,Z FOR WINDOW

INOT    (O)          IN/OUT SWITCH

INOT=1 >> ELM IS IN THE WINDOW
INOT=-1 >> ELM IS OUT OF WINDOW

FILES USED-- NONE

COMMONS USED-- NONE

FUNCTIONS/ROUTINES CALLED-- NONE

LIBRARIES ACCESSED-- NONE

LOCAL VARIABLES--

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
SUBROUTINE REMELM

ROUTINE-- REMELM

DATE/WRITTEN BY-- 09/13/82  RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS SUB ELIMINATES UNDESIRED ELMS FROM THE RANDOM ELM GEOM FILE

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION

COMMONS USED --
ELMDAT
CNTLGM

FUNCTIONS/ROUTINES CALLED --
RWELMS

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE CALCRD

DATE/WRITTEN BY-- 8/08/82  RJ MAFFEO
DATE/REVISED BY--

THE PURPOSE OF THIS ROUTINE IS TO CALL ALL
REQUIRED ROUTINES TO COMPUTE THE COORDINATES OF THE
CENTROID OF THE ELMs AND THE COORDINATES OF THE
FACE CENTERS IT THEN CALLS THE ELM I/O ROUTINE TO STORE
THIS INFO INTO THE INTERNAL DATA BASE

CALLING SEQUENCE-- CALL CALCRD
CALLING ARGUMENTS-- NONE

FILES USED--
IERF -- ERROR REMARK FILE

COMMONS USED--
AFIL -- FILE CODE COMMON
ELMDAT -- ELM ATTRIBUTE COMMON
SIZE -- PROBLEM SIZE COMMON
WORK -- SCRATCH WORK SPACE COMMON
CRNDEF -- LOCAL COMMON TO COORD CALC ROUTINES

FUNCTIONS/ROUTINES CALLED--
GETPD
RWELMS
VOLM
AREAD
REACRD

LIBRARIES ACCESSED--

LOCAL VARIABLES--

SPECIAL COMMENTS ABOUT THIS ROUTINE

MANY OF THE ROUTINES CALLED BY CALCRD ARE ALSO USED
IN SIGMA THTD DECK GENERATOR THEY CAN COMPUTE MUCH MORE THAN
JUST COORDINATES(SUCH AS VOLUMES AREAS AND DELTA X).

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
SUBROUTINE GETPD(P3D,P2D)

DATE/Written BY-- 09/13/82  RJ MAFFEO

FUNCTION/PURPOSE--

THIS SUB COMPUTES AND STORES ALL OF THE PARTIAL DERIVATIVES OF THE SHAPE FUNCTIONS, BOTH 3D AND 2D

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
P3D (O) 3D PARTIAL DERIVATIVES
P2D (O) 2D PARTIAL DERIVATIVES

COMMONS USED --
CRNDEF --- SEE CALCRD

FUNCTIONS/ROUTINES CALLED --
PARD
GTGAUS

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE VOLM(X,Y,G.P,P3D,VOL,XC,YC,ZC,Q,IVER)

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
X (I) ARRAY OF X COORDS
Y (I) ' ' ' Y '
Z (I) ' ' ' Z '
GP (I) ARRAY OF GAUSS POINTS
P3D (I) 3D PARTIAL DERIVATIVES
VOL (O) VOLUME
XC (O) X COORD OF CENTROID
YC (O) Y '
ZC (O) Z '
Q (O) JACOBIAN
IVER (O) IVER=1 > ZERO VOLUME ELM

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --
SHAPEF
FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE SHAPEF(X,Y,Z,TETA,YNETA,ZETA,XP,YP,ZP)

ROUTINE--  SHAPEF

DATE/WRITTEN--  09/14/82  RJ MAFFEO

DATE/REVISED--

FUNCTION/PURPOSE--

THIS SUB COMPUTES THE VALUES OF X,Y,Z AT THE GAUSS POINTS
BASED ON LINEAR 3D SHAPE FUNCTIONS

CALLING ARGUMENTS --

NAME   ATTRIBUTES   DEFINITION
X (I) >   ARRAYS OF COORDINATES
Y (I) >>
Z (I) >
TETA (I) >
YNETA (I) >> LOCAL COORDINATES
ZETA (I) >
XP (0) >
YP (0) >> DESIRED GLOBAL COORDINATES
ZP (0) >

COMMONS USED --

CRNDEF

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE PARD(TETA,YNETA,ZETA,IGP,IND,PD)

ROUTINE-- PARD

DATE/Written by-- 09/14/82  RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS SUB COMPUTES THE DERIVATIVES OF THE
SHAPE FUNCTIONS WRT THE LOCAL SYSTEM
BOTH FOR VOLUMES(3D) AND AREAS(2D)

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

TETA (I) > LOCAL COORDINATES
YNETA (I) >>
ZETA (I) >
IGP (I) GAUSS POINT COUNTER
IND (I) 2D/3D INDICATOR
IND=O >> 3D
PD (O) IND=1 THRU 6 >> 2D IND=FACE NUMBER

COMMONS USED --

CRNDEF

LOCAL COORDINATES GAUSS 20/30
IND=FACE NUMBER

FILES USED --

FUNCTIONS/Routines CALLED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END

-- GE/TRANCITS

-- PARD PAGE= 16

93
SUBROUTINE AREAD(X,Y,Z,P2D,Q,ARA,DX2.

FUNCTION/PURPOSE--

THIS SUB COMPUTES THE FACE AREAS AND DELTA X FOR 3D THT ELMS

NOTE IT COMPUTES DELTA X TWO WAYS

IDX=1 DX=DIST FROM ELM CENTROID TO FACE CENTROID

IDX=2 , DX=PERPENDICULAR DIST FROM ELM CENT TO FACE

IDX=3 SWITCH TO JUST CALC CENTROID OF FACE 5

IT IS BEING USED HERE TO COMPUTE THE COORDINATES

OF THE FACE CENTERS

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

X (I) ARRAY OF CORNER COORDS

Y (I) ARRAY OF CORNER COORDS

Z (I) ARRAY OF CORNER COORDS

P2D (I) DERIVATIVES OF SHAPE FUNCTIONS

Q (0) JACOBIAN

ARA (0) ARRAY OF FACE AREAS

DX2 (0) ARRAY OF FACE DELTA XS

IDX (I) DELTA X COMPUTATION INDICATOR

XC (I) COORDS OF VOLUME CENTROID

YC (I) COORDS OF VOLUME CENTROID

ZC (I) COORDS OF VOLUME CENTROID

ZA (0) ARRAYS OF COORDS FOR FACE CENTERS

COMMONS USED --

CRNDEF

FUNCTIONS/ROUTINES CALLED --

GTGAUS SHAPEF

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
SUBROUTINE GTGAUS(IND, I1, I2, I3, IP)

FUNCTION/PURPOSE--

THIS SUB GETS APPROPRIATE GAUSS POINTS FOR EACH FACE OF THE 3D BRICK

Calling Arguments --

NAME ATTRIBUTES DEFINITION

IND (I) FACE INDICATOR
I1 (O) >
I2 (O) >
I3 (O) >> GAUSS POINT COUNTERS
IP (O) >

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

END
SUBROUTINE REACRD(CRD,XC,YC,ZC,XA,YA,ZA)

ROUTINE -- REACRD
DATE/WRITTEN BY -- 09/15/82 RJ MAFFEO
DATE/REVISED BY --
FUNCTION/PURPOSE --
THIS SUB REARRANGES THE FACE CENTER COORDINATE ARRAYS AND THE VOLUME CENTROIDS INTO THE CRD ARRAY

CALLING ARGUMENTS --
NAME | ATTRIBUTES | DEFINITION
-- | -- | --
CRD | (0) | BRICK COORD ARRAY
XC | (I) > | VOLUME CENTROIDS
YC | (I) >> | VOLUME CENTROIDS
ZC | (I) > | VOLUME CENTROIDS
XA | (I) > | FACE CENTER ARRAYS
YA | (I) >> | FACE CENTER ARRAYS
ZA | (I) > | FACE CENTER ARRAYS

COMMONS USED --
FUNCTIONS/ROUTINES CALLED --
FILES USED --
LIBRARIES ACCESSED --
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END

-- GE/TRANCITS
-- REACRD
PAGE = 20
SUBROUTINE CORNCC

FUNCTION/PURPOSE--
THIS SUB CALLS ALL ROUTINES NEEDED TO COMPUTE AND STORE THE INFORMATION ASSOCIATED WITH THE CORNER NODE WEIGHTING COEFFICIENTS

CALLING ARGUMENTS --
NAME     ATTRIBUTES     DEFINITION
NONE     

COMMONS USED --
AFIL     
ELMDAT   
SIZE     
WORK     
CNTLGM   

FUNCTIONS/ROUTINES CALLED --
RWELMS   
SORT     
CCNVEC   
GETCRD   
CINV     
PUTWFI   
RWCCF    

FILES USED --
IERF S   

LIBRARIES ACCESSED --
NONE     

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE CCNVEC(NOD,IRW,NLOC,NUMA,MCON,MXNCON,NSIZW)

FUNCTION/PURPOSE--

THIS SUB WRITES TO (CREATES) AND READS FROM THE 
NODE -CONNECTION VECTOR (MCON)

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
NOD (I) NODE NUMBER
IRW (I) READ/WRITE SWITCH(0-READ,1-WRITE)
NLOC (I/O) CURRENT CONNECTION TO NODE
NUMA (O) MAX # CONNECTIONS TO NODE
MCON (I/O) PACKED CONNECTION ARRAY
MXNCON (O) MAX # OF CONNECTIONS TO ANY NODE
NSIZW (I) SIZE OF WORK ARRAY (MCON HERE)

COMMONS USED --
AFIL

FUNCTIONS/ROUTINES CALLED --

FILES USED --
IERF S

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

THE NODE-CONNECTION VECTOR (MCON) IS KEYED
TO THE NODE NAME AND IS COMPOSED OF THE MAX
CONNECTION TO THAT NODE AND THE CURRENT CONNECTION TO
THAT NODE IN THE FORM --
MCON(NOD)*100*MAX CONNECTION + CURRENT CONNECTION

THE INITIAL CURRENT COUNT IS EQUAL TO THE MAX CONNECTION.
AS THE CONNECTIONS ARE PROCESSED THE CURRENT COUNT IS
REDUCED BY ONE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END

-- GE/TRANCITS

-- CCNVEC
SUBROUTINE GETCRD(NE, CRD, BRD, XCN, YCN, ZCN, JREC)

FUNCTION/PURPOSE--

THIS SUB EXTRACTS THE APPROPRIATE SUBSET OF
COORDINATES FOR EACH CORNER NODE FROM THE
FULL SET OF COORDS OF EACH ELM

COMMONS USED --
AFIL (I)

FUNCTIONS/ROUTINES CALLED --

FILES USED --
IERF S

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

THE ARRAY IDX IS USED TO DEFINE THE RELATIONSHIP
BETWEEN THE CORNER POINT AND THE APPROPRIATE 3 FACES
FOR EXAMPLE CORNER POINT 1 IS CONNECTED TO FACES
1, 4, 5
THEREFORE IDX(1,1)=1, IDX(1,2)=4, IDX(1,3)=5

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE PUTWFI(NOD, NE, NUMA, JREC, XSF, NPOS, ACWF)

FUNCTION/PURPOSE

THIS SUB POSITIONS THE WEIGHTING COEFF INFO INTO CORRECT LOCATIONS OF THE ACWF ARRAY

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
NOD (I) CORNER NODE NUMBER
NE (I) ELM NUMBER
NUMA (I) MAX NUMBER OF CONNECTIONS
JREC (I) PACKED CORNER - FACE #
XSF (I) WEIGHTING COEFFS
NPOS (I) STARTING POSITION FOR INFO
ACWF (O) ARRAY CONTAINING INFO

COMMONS USED --

AFIL

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE RWCCF(NOD, ACWF, NPOS, IRW)

ROUTINE-- RWCCF

DATE/Written by-- 11/11/82  RJ MAFFEO
DATE/Revised by--

FUNCTION/PURPOSE--

THIS ROUTINE READS AND WRITES THE CORNER NODE CODF INFO TO AND FROM THE DATA BASE

Calling Arguments --

NAME ATTRIBUTES DEFINITION

NOD  (I/O) NODE NAME (RECORD #)
ACWF (I/O) ARRAY OF INFO
NPOS (I) # OF TERMS TO BE READ/WRITTEN
IRW (I) READ/WRITE SWITCH(0-READ, 1-WRITE)

COMMONS Used--

AFIL

FILES USED--

IRCCF R

LIBRARIES Accessed--

NONE

LOCAL VARIABLES--

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END

-- GE/TRANCITS -- RWCCF PAGE= 25
SUBROUTINE GETMPE(MP,NWDMPE)

DATE/WRITTEN BY-- 09/19/83  R.J. MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE READS THE FIRST TIME STEP TEMPERATURE SOLUTION AND EXTRACTS THE ELM NAME AND RELATIVE POSITION OF THIS ELM IN THE OUTPUT LIST. IT THEN CALLS MPERW TO STORE THIS INFO INTO THE MP ARRAY.

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
MP           (I/O) ELM MATRIX POSITION ARRAY
NWDMPE      (I) # OF WORDS IN MP ARRAY

COMMONS USED --
AFIL SIZE CNTLFL

FUNCTIONS/ROUTINES CALLED --
TFLSET HTFIO MPERW

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE MPERW(MP,NE,NPOS,IRW)

DATE/Written BY-- 10/08/83  RJ MAFFEO
DATE/REVISED BY--
FUNCTION/PURPOSE--
THIS ROUTINE PACKS AND UNPACKS THE INFO NEEDED TO RELATE
THE ELEMENT NAME TO THE RELATIVE POSITION OF THE ELEMENT
IN THE OUTPUT LIST

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
MP (I/O) MATRIX POSITION ARRAY
NE (I) ELM NAME
NPOS (I/O) RELATIVE POSITION IN OUTPUT
IRW (I) SWITCH(0--READ,1--WRITE)

COMMONS USED -- NONE

FUNCTIONS/ROUTINES CALLED --
MOD (FORTRAN SUPPLIED)

FILES USED --
O6

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

FOR EXAMPLE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H8000

NOTE THAT MP(1) CONTAINS THE POSITIONS OF BOTH
ELMS 1 AND 2

THE ADVANTAGE OF THIS SCHEME IS THAT THE DIMENSION OF
MP NEED ONLY BE 4000 TO STORE ELMS UP TO 8000

MP(1) IS 09870079
THE VALUE STORED IN MP(1) IS 09870079
MP(1) CONTAINS THE POSITIONS OF BOTH

THE MAX POSITION THAT CAN BE PACKED IS NPOS=9999
THE POSITIONS ARE PACKED AS FOLLOWS --
POSITION 1 >> LEFT HAND SIDE OF WORD
POSITION 2 >> RIGHT HAND SIDE OF WORD
PER WORD OF THE MP ARRAY IT MAKES USE OF THE FORTRAN
SUPPLIED 'MOD' FUNCTION

THIS ROUTINE USES DECIMAL PACKING TO PACK 2 POSITIONS
PER WORD OF THE MP ARRAY

This routine uses decimal packing to pack 2 positions per word of the MP array. It makes use of the FORTRAN supplied 'MOD' function. The positions are packed as follows:

Position 1 >> Left hand side of word
Position 2 >> Right hand side of word

The max position that can be packed is NPOS=9999. The advantage of this scheme is that the dimension of MP need only be 4000 to store elms up to 8000.

For example, if the position of ELM 1 is 987 and the position of ELM 2 is 79, then the value stored in MP(1) is 09870079. Note that MP(1) contains the positions of both elms 1 and 2.

Special remarks/instructions --
Host = H8000
30960CPR
30970CPR
31620 END
SUBROUTINE SETFLP(NNEW, NOLD)

ROUTINE -- SETFLP

DATE/Written by -- 09/19/83  R. J. Maffeo
DATE/REVISED BY --

FUNCTION/PURPOSE--

THIS ROUTINE SETS THE H T INPUT FILE TO THE CORRECT
POSITION TO READ THE SOLUTION DESIRED

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

NNEW (I) FILE DESIRED
NOLD (I) CURRENT FILE COUNT

AFIL OF THT OUTPUT FILE

COMMONS USED --

AFIL

FUNCTIONS/ROUTINES CALLED --

TFLSET
HTFPAS

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

HOST = H6000

SPECIAL REMARKS/INSTRUCTIONS --

END

END
SUBROUTINE RDTEMP(TEMP,NFLVAL)

ROUTINE-- RDTEMP
DATE/WRITTEN BY-- 09/19/83  RJ MAFFEO
DATE/REVISED BY--
FUNCTION/PURPOSE--

THIS ROUTINE READS THE TEMPERATURE INFO FOR THE DESIRED TIME STEP FROM THE THT OUTPUT FILE AND STORES IT IN THE TEMP ARRAY

CALLING ARGUMENTS --
NAME  ATTRIBUTES  DEFINITION
TEMP    (D)    TEMPER STORAGE ARRAY
NFLVAL   (I)    CURRENT FILE COUNT

COMMONS USED --
AFIL
SIZE
CNTLFL

FUNCTIONS/ROUTINES CALLED --
HTFIO

FILES USED --
IERF

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
3369OC SUBROUTINE CWCRNT(MP,TEMP,TIMVAL,NNODMX,NNODMN,TCRIT,MXNCON)
3370C====================================================================
3371CPR
3372CPR====================================================================
3373CPR
3374CPR====================================================================
3375CPR
3376CPR--------- ROUTINE-- CWCRNT
3377CPR
3378CPR--------- DATE/WRITTEN BY-- 09/19/83 RJ MAFFEO
3379CPR
3380CPR--------- DATE/REVISED BY--
3381CPR
3382CPR--------- FUNCTION/PURPOSE--
3383CPR
3384CPR--------- THIS ROUTINE COMPUTES A UNIQUE TEMPERATURE FOR EACH
3385CPR
3386CPR--------- CORNER OF THE HEAT TRANSFER ELM BASED ON THE INDIVIDUAL
3387CPR
3388CPR--------- TEMPERATURES FOR EACH ELM CONNECTION
3389CPR
3390CPR
3391CPR
3392CPR
3393CPR
3394CPR--------- CALLING ARGUMENTS --
3395CPR
3396CPR--------- NAME         ATTRIBUTES     DEFINITION
3397CPR
3398CPR--------- MP           (I)            ELM MATRIX POSITION ARRAY
3399CPR--------- TEMP         (I)            ELM TEMPERATURE ARRAY
3400CPR--------- TIMVAL       (I)            CURRENT VALUE OF TIME
3401CPR--------- NNODMX       (I)            MAX NODE NAME
3402CPR--------- NNODMN       (I)            MIN NODE NAME
3403CPR--------- TCRIT        (I)            TEMPERATURE USED IN MAXTDF TO
3404CPR--------- MXNCON       (I)            MAX NUMBER CONNECTIONS TO
3405CPR
3406CPR--------- COMMONS USED--
3407CPR
3408CPR--------- AFIL
3409CPR
3410CPR--------- FUNCTIONS/ROUTINES CALLED--
3411CPR
3412CPR--------- RWCCF
3413CPR--------- UNPAKF
3414CPR--------- MPERW
3415CPR--------- FCRNTP
3416CPR--------- MAXTDF
3417CPR--------- WRITMP
3418CPR--------- FILREW
3419CPR
3420CPR--------- FILES USED--
3421CPR
3422CPR
3423CPR
3424CPR
3425CPR--------- LIBRARIES ACCESSED--
3426CPR--------- NONE
3427CPR
3428CPR--------- LOCAL VARIABLES--
3429CPR
3430CPR--------- SPECIAL COMMENTS ABOUT THIS ROUTINE
3431CPR
3432CPR--------- SPECIAL REMARKS/INSTRUCTIONS--
3433CPR--------- HOST = H6000
3434CPR
3435CPR
3436CPR
3437CPR
3438CPR---------GE/TRANSCITS -- CWCRNT PAGE= 31
SUBROUTINE UNPAKF(NPACK,NAME,IFACE)

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
NPACK (I) PACKED CORNER-FACE #
NAME (D) CORNER NAME
IFACE (D) FACE INFO ARRAY

COMMONS USED --
AFIL

FUNCTIONS/ROUTINES CALLED --
IERF S

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE
NOTE PACK CORNER-FACE # = CORNER NAME*1000000+IFACPC WHERE IFACPC IS A SINGLE INTEGER THAT CONTAINS ALL THE FACE NUMBERS

FOR EXAMPLE
IF THE FACE NUMBERS WERE 6,5,4,3,2,1 THEN IFACPC=654321

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE FCRNTP(TEMP,MPOS,IFACE,NCORN,CTP,COF)

THIS ROUTINE EXTRACTS THE 7 TEMPERATURES ASSOCIATED WITH A CONNECTING ELM. IT THEN FINDS THE 4 TEMPERATURES ASSOCIATED WITH THE CORNER OF INTEREST AND DOES THE MULTIPLICATION TO COMPUTE THE TEMPERATURE FOR THIS CORNER BASED ON THIS ELM CONNECTION.

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

TEMP (I) ELM TEMPERATURE ARRAY
MPOS (I) MATRIX POSITION OF THE ELM
IFACE (I) ELM FACE ARRAY
NCORN (I) CORNER NUMBER
CTP (O) CORNER TEMPERATURE
COF (I) CORNER COEFFICIENT ARRAY

COMMONS USED --

AFIL

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE MAXTDF(NEL, CRNT, NCON, NDONE, TIMVAL, TCRIT, IREC)

ROUTINE -- MAXTDF

DATE/WRITTEN BY -- 11/30/82  RJ MAFFEO
DATE/REVISED BY --
FUNCTION/PURPOSE --

THIS ROUTINE COMPUTES THE MAX DIFFERENCE BETWEEN
ALL THE POSSIBLE CORNER TEMPERATURES AT A CORNER
AND FLAGS IT IF IT IS GREATER THAN TCRIT

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
NEL (I) ARRAY OF ELMS CONNECTED TO THIS NODE
CRNT (I) ARRAY OF TEMPS FOR THIS NODE
NCON (I) NUMBER OF ELM CONNECTED TO THIS NODE
NDONE (I) PAGE HEADER COUNTER
TIMVAL (I) (=0 WRITE HEADER, NOT = 0 DON'T WRITE)
TCRIT (I) CRITICAL TEMP FOR PRINTOUT
IREC (I) RECORD NUMBER (NODE NUMBER) OF CORNER

COMMONS USED --

AFIL

FUNCTIONS/ROUTINES CALLED --

SORT

FILES USED --

INOT6 $S

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
THIS ROUTINE WRITES THE VALUES OF THE TEMPERATURES AT THE STRESS POINTS TO HARD COPY AND TO THE OUTPUT FILE.

CALLING ARGUMENTS --

NAME    ATTRIBUTES    DEFINITION

TBAR    (I)            TEMPERATURE AT THE STRESS POINT
TIMVAL  (I)            CURRENT VALUE OF TIME
NOD     (I)            STRESS POINT NAME(NODE #)
NDONE   (I)            SWITCH

=0      -- WRITER PAGE HEADER
=1      -- DO NOT WRITE HEADER
= -1    -- FLUSH BUFFERS

COMMONS USED --

AFIL

FUNCTIONS/ROUTINES CALLED --

FILES USED --

INOT6  S
IOUTF  S

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END

-- WRITMP  PAGE= 36
SUBROUTINE CINV(CD,XSP,YSP,ZSP,XSF,NE,JCN)

ROUTINE--
CINV

DATE/Written by-- 9/10/82 RJ MAFFEO

DATE/REVISED BY-- 01/12/83 RJ MAFFEO

FUNCTION/PURPOSE--

THIS SUB FIND THE VALUES OF THE SHAPE FUNCTIONS FOR ANY POINT NEAR A TETRAHEDRAL ELEMENT. THE SHAPE FUNCTIONS ARE BASED ON THE SIMPLEX TETRAHEDRAL ELEMENT. IT FIRST FINDS THE INVERSE OF THE COORDINATE MATRIX (-C-) AND THEN MULTIPLIES THIS INVERSE INTO THE COORD VECTOR (1 X Y Z) TO FIND THE SHAPE FUNCTIONS. THIS ROUTINE CAN BE USED TO COMPUTE THE WEIGHTING COEFF FOR THE CORNERS OF A TETRAHEDRAL ELEMENT BASED ON THE TEMPERATURES AT THE CENTER OF THE FACES AND THE CENTROID.

CALLING ARGUMENTS--

NAME ATTRIBUTES DEFINITION
CD (I) 12X1 ARRAY COORDINATE ARRAY
XSP (I) X, Y, Z COORDS OF CORNERS OF TETRA
YSF (I) COORDS OF THE POINT FOR WHICH THE SHAPE FUNCTIONS ARE DESIRED
ZSF (I) VECTOR OF SHAPE FUNCTIONS
NE (I) ELEMENT NAME
JCN (I) CORNER NUMBER

FILES USED--
AFIL

COMMONS USED--

Routines Called --

Libraries Accessed--

Local Variables--

INX 18X1 THIS VECTOR IS USED TO SELECT THE CORRECT INDICES OF THE C MATRIX FOR EACH ELEMENT OF THE C INVERSE MATRIX.
CI 4X4 INVERSE OF THE COORD MATRIX (-C-)
C 5X4 20 MATRIX CONTAINING CORRDS OF TETRA

Special Comments About This Routine--

NOTE THAT THE COORD MATRIX (-C-) IS AUGMENTED WITH A ROW OF ONES. THIS ROW OF ONES IS USED IN THE COMPUTATION OF THE INVERSE.
NOTE XSF(5)=1/DIST;
DIST=DISTANCE FROM CENTROID TO CORNER.
NOTE A LOCAL COORD SYSTEM IS ESTABLISHED FOR EACH ELM

WHERE Q(I) IS THE COORDS OF THE VERTICES

AND QC IS THE COORDS OF THE CENTROID OF THE ELM

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
42740  SUBROUTINE PRNTIM(ISTOP,I1,TB,TE)
42930  END
SUBROUTINE EMINMX(XC, YC, ZC, CADD, XYZV, NEMNMX, NE, NNPE)

FUNCTION/PURPOSE--

THIS ROUTINE FINDS THE MIN AND MAX XYZ COORDS
OF THE HEAT TRANSFER ELEMENT AND STORES THIS INFO
INTO THE XYZV ARRAY
IT ALSO KEEPS TRACK OF THE ELMS WITH THE SMALLEST
AND LARGEST XYZ COORDS AND STORES THEM IN THE
NEMNMX ARRAY

CALLING ARGUMENTS --
NAME      DEFINITION
XC        (I) CONSTANT COORD ADDER
YC        (I) CURRENT ELM NAME
ZC        (I) # OF NODES PER ELM
CADD      (I)
NE        (I) CURRENT ELM NAME
NNPE      (I) # OF NODES PER ELM

FUNCTIONS/ROUTINES CALLED --
SORT

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

END
SUBROUTINE FTIME(A1)

ROUTINE-- FTIME

DATE/Written By-- 04/05/83 RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE CALLS THE APPROPRIATE SYSTEM ROUTINE

TO MEASURE THE PROCESSOR TIME

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

A1 (0) PROCESSOR TIME

COMMONS USED -- NONE

FUNCTIONS/ROUTINES CALLED --

PTIME -- HONEY WELL TIMER

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END

--- GE/TRANCITS --- FTIME PAGE= 41
SUBROUTINE SSURCH

ROUTINE -- SSURCH

DATE/Written by -- 04/05/83  RJ MAFFEO

DATE/REVISED BY --

FUNCTION/PURPOSE --

THIS ROUTINE IS THE MINI-EXEC FOR THE STRESS POINT
TO HEAT TRANSFER ELM SEARCH ROUTINES IT CALLS ALL
REQUIRED ROUTINES TO FIND WHICH HT ELM CONTAINS
EACH STRESS POINT AND THEN FINDS THE WEIGHTING COEFFS
FOR THAT STRESS POINT

CALLING ARGUMENTS --

NAME  ATTRIBUTES

COMMONS USED --

AFIL
SIZE
SPDAT

FUNCTIONS/ROUTINES CALLED --

KNTSPF
RFILCR
ZERORF
GNLSUR

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END

-- GE/TRANCITS

-- SSURCH

PAGE= 42
SUBROUTINE KNTSPF(NUMSP,MXNSP)

This routine counts the number of stress points to be interfaced from the stress node file and also finds the largest stress node name.

Calling arguments:
- NAME
- ATTRIBUTES
- NUMSP
- MXNSP

Commons used:
- AFIL

Functions/routines called:
- FILREW
- RDSPF

Files used:
- IERF

Libraries accessed:
- NONE

Local variables:
- LOCAL VARIABLES

Special comments about this routine:
- SPECIAL REMARKS/INSTRUCTIONS
  - HOST = H6000

End
SUBROUTINE RFILCR

ROUTINE-- RFILCR

DATE/Written by-- 04/06/83  RJ Maffeo

FUNCTION/PURPOSE--

THIS ROUTINE SIZES ALL THE RANDOM FILES
IT CALLS THE APPROPRIATE SYSTEM ROUTINE TO DEFINE
THE WIDTH OF THE FILE

CALLING ARGUMENTS --

NAME  ATTRIBUTES  DEFINITION

COMMONS USED --

FUNCTIONS/RUTINES CALLED --

RANSIZ -- Honeywell routine to set width of file
DEFINE FILE -- IBM routine to set width

FILES USED --

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

END
**SUBROUTINE ZERORF(IFC, MXN, NWD, A)**

**ROUTINE--** ZERORF

**DATE/Written by--** 04/07/83  R.J. Maffeo

**FUNCTION/Purpose--**

THIS ROUTINE INITIALIZES RANDOM FILES

**Calling Arguments --**

<table>
<thead>
<tr>
<th>NAME</th>
<th>Attributes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFC</td>
<td>(I)</td>
<td>FILECODE</td>
</tr>
<tr>
<td>MXN</td>
<td>(I)</td>
<td>MAX RECORD NUMBER</td>
</tr>
<tr>
<td>NWD</td>
<td>(I)</td>
<td># OF WORDS IN -A- ARRAY</td>
</tr>
<tr>
<td>A</td>
<td>(I)</td>
<td>INITIALIZING ARRAY</td>
</tr>
</tbody>
</table>

**COMMONS Used --**

**FILES Used --**

**LIBRARIES Accessed --**

**LOCAL VARIABLES --**

**SPECIAL COMMENTS ABOUT THIS ROUTINE**

**Special Remarks/Instructions --**

**END**
SUBROUTINE GNLSUR

DATE/Written By-- 04/07/83 RJ MAFFEO
DATE/Revised By--

FUNCTION/PURPOSE--

THIS ROUTINE IS CALL BY THE SEARCH MINI-EXEC (SSURCH)

IT INITIATES THE SEARCH THRU THE HEAT TRANSFER ELMS

TO FIND WHICH ELM CONTAINS EACH STRESS POINT

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

COMMONS Used --

AFIL SIZE

WORK

FUNCTIONS/Routines CALLED --

EINCOR

RDSPF

FNDELM

FILES USED --

INOT6 S

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE RDSPF(NN,X,Y,Z,IERC)

ROUTINE-- RDSPF

DATE/Written By-- 04/07/83 RJ Maffeo
DATE/REVISED By--

FUNCTION/PURPOSE--
THIS ROUTINE READS THE STRESS POINT COORD INFO
FROM THE STRESS POINT NODE FILE

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

NN (O) NODE NAME
X (O) X COORD
Y (O) Y COORD
Z (O) Z COORD
IERC (O) ERROR CODES

0 -- NO ERRORS
1 -- END OF FILE
2 -- READ ERROR

COMMONS USED --
AFIL

FUNCTIONS/ROUTINES CALLED --

FILES USED --
ISPNF S
LIBRARIES ACCESSED --
NONE
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE FILREW(IFC)

COMMONS USED --
FUNCTIONS/RUTINES CALLED --
FILES USED --
LIBRARIES ACCESSED --
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE EINCOR(ENX, NELMN, NELMX)

FUNCTION/PURPOSE--
COORDS INTO CORE FROM THE RANDOM ELM FILE
THE DATA STORED IN CORE IS--

CALLING ARGUMENTS--
NAME ATTRIBUTES DEFINITION
ENX (O) ARRAY CONTAINING WINDOW INFO
NELMN (I) MIN ELM NAME
NELMX (I) MAX ELM NAME

COMMONS USED--
AFIL
ELMDAT

FUNCTIONS/ROUTINES CALLED--
RWELMS

FILES USED--

LIBRARIES ACCESSED--
NONE

LOCAL VARIABLES--

SPECIAL REMARKS/INSTRUCTIONS--
HOST = H6000

END
52020C-------------------------------------------------------
52040    SUBROUTINE FNDELM(NSP,XP,YP,ZP,IFOND)
52060C-------------------------------------------------------
52080CPR 52090CPR
52110CPR
52120CPR ROUTINE--       FNDELM
52130CPR
52140CPR DATE/WRITTEN BY-- 04/07/83       RU MAFFEO
52150CPR DATE/REVISED BY--
52160CPR
52170CPR FUNCTION/PURPOSE--
52180CPR
52190CPR THIS ROUTINE DOES THE ACTUAL SEARCH THRU THE HEAT
52200CPR TRANSFER ELMS TO FIND WHICH ELM CONTAINS THE STRESS
52210CPR POINT
52220CPR IT CALLS THE REQUIRED ROUTINES TO GET THE ELM
52230CPR COORDINATES, CHECKS TO SEE IF THE STRESS POINT
52240CPR IS WITHIN THE ELM AND IF IT IS COMPUTES
52250CPR AND WRITES THE WEIGHTING COEFFS FOR THIS STRESS
52260CPR POINT WITH RESPECT TO THE VERTICES OF THE
52270CPR HEAT TRANSFER ELEMENT
52280CPR
52290CPR
52300CPR
52310CPR CALLING ARGUMENTS --
52320CPR NAME       ATTRIBUTES      DEFINITION
52330CPR
52340CPR NSP       (I)     STRESS NODE NAME
52350CPR XP        (I) >    > STRESS POINT COORDS
52360CPR YP        (I) >>   >
52370CPR ZP        (I) >>   >
52380CPR IFOND     (O)     INDICATOR
52390CPR 0 -- DID NOT FIND CONTAINING ELM
52400CPR 1 -- DID FIND CONTAINING ELM
52410CPR
52420CPR COMMONS USED --
52430CPR
52440CPR AFIL
52450CPR ELMDAT
52460CPR SIZE
52470CPR SPDAT
52480CPR
52490CPR FUNCTIONS/ROUTINES CALLED --
52500CPR
52510CPR CKWIND
52520CPR GETWND
52530CPR IGOSF
52540CPR RWSPCF
52550CPR
52560CPR FILES USED --
52570CPR
52580CPR
52590CPR LIBRARIES ACCESSED --
52600CPR NONE
52610CPR
52620CPR
52630CPR LOCAL VARIABLES --
52640CPR
52650CPR SPECIAL COMMENTS ABOUT THIS ROUTINE
52660CPR
52670CPR
52680CPR SPECIAL REMARKS/INSTRUCTIONS --
52690CPR HOST = H6000
52700CPR
52710CPR

-- GE/TRANCITS -- FNDELM PAGE= 50 127
END
SUBROUTINE GETWND(IDONE, IWHR)

COMMONS USED --
AFIL
ELMDAT
SIZE
WORK

FUNCTIONS/ROUTINES CALLED --
RWELMS
 FILES USED --

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE
THE COORD DATA IS PASSED BACK TO THE ROUTINE -FNDLEM- VIA THE COMMON CALLED ELMDAT IN THE ARRAY AED

END
SUBROUTINE CKWIND(XP,YP,ZP,V,INOT)

THIS ROUTINE CHECKS TO SEE IF THE COORDS OF THE STRESS POINT ARE WITHIN THE WINDOW OF THE HEAT TRANSFER ELEMENT

CALLING ARGUMENTS --

NAME       ATTRIBUTES   DEFINITION
XP          »           COORDS OF STRESS POINT
YP          »»          COORDS OF HEAT TRANS ELM
ZP          »           IN - OUT INDICATOR
INOT        (0)         0 -- OUT OF WINDOW
               1 -- IN WINDOW

COMMONS USED --
FUNCTIONS/ROUTINES CALLED --
FILES USED --
LIBRARIES ACCESSED --
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE MKRGF

DATE/WRITTEN BY-- 8/04/82 RJ MAFFEO
DATE/REVISED BY-- 4/04/83 RJ MAFFEO
REASON -- ADDED MIN MAX WINDOW LOGIC

DATE/REVISED BY-- 8/30/83 SR MARTIN
REASON -- CHANGED I/O TO READ NEW INPUT FILE FORM

FUNCTION/PURPOSE--
THIS ROUTINE DOES THE PROCESSING OF THE NEUTRAL
NODE AND ELEMENT FILES INTO THE INTERNAL DATA BASE FILES
FOR THIS VERSION OF 3D TITS.
IT CALLS RWNODS TO PROCESS THE NODAL INFO.
IT CALLS RWELMS TO PROCESS THE ELM INFO
IT CALLS PROFAC TO PROCESS THE FACE INFORMATION

CALLING SEQUENCE-- CALL MKRGF
CALLING ARGUMENTS-- NONE
NAME ATTRIBUTES DEFINITION
FILES USED--
IFL3 -- RANDOM FILE TO STORE THE NODAL COORDINATES
IFL4 -- RANDOM FILE TO STORE THE ELM ATTRIBUTES
INTITS -- HEAT TRANSFER OUTPUT FILE
INOT6 -- OUTPUT FILE FOR MESSAGES
IERF -- ERROR REMARK FILE
COMMONS USED--
AFIL -- FILECODE COMMON
ELMDAT -- ELM ATTRIBUTE COMMON
SIZE -- SIZE VARIABLES COMMON
CNTLFL -- INFO WHICH DESCRIBES ALL ELEMENTS
FUNCTIONS/ROUTINES CALLED--
RWNODS
HEADEL
RWELMS
READPS
PROFAC
EMINMX
LIBRARIES ACCESSED--
NONE
LOCAL VARIABLES--
NNODMX MAXIMUM NODE NAME
NNODMN MINIMUM NODE NAME
NUMNOD NUMBER OF NODES
NELMX MAXIMUM ELM NAME
NELMN MINIMUM ELM NAME
NUME NUMBER OF ELEMENTS
CRD(45) ARRARY OF COORDINATES FOR THE 15 POINTS IN A ELM
IFACPC PACKED TTBD FACE NUMBERS
IE(8) ARRAY OF CONNECTIVITY FOR ELM CORENER POINTS
AED(56) -- ELM ATTRIBUTE ARRAY

-- GE/TRANCITS
-- MKRGF
PAGE= 54
TITLE=

THIS ARRAY CONTAINS NE, CRD, IE, IFACPC, VOL

-- ELM NAME

VOL -- VOLUME OF THE ELM

XYZ(3) -- COORDINATE ARRAY FOR EACH NODE

SPECIAL COMMENTS ABOUT THIS ROUTINE

NOTE THE 15 POINTS FOR EACH ELM ARE 8 CORNERS, 6 MIDSIDES AND 1 CENTROID

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
SUBROUTINE PROFAC

DATE/WRITTEN BY-- 8/04/82  R J MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE READS THE PACKED THTD FACE NUMBER FROM THE END OF THE NEUTRAL ELEMENT FILE AND STORES THEM ON THE INTERNAL DATA BASE FILE.

THE INTERNAL DATA BASE FILE IS RANDOM FILE IFL4, THE ELEMENTAL ATTRIBUTE FILE.
NOTE THE PACKED FACE NUMBERS EXIST ON THE HEAT TRANSFER OUTPUT FILE.
IN RECORDS OF THE FORM- <ELMT #> <FACE INFO>

CALLING SEQUENCE-- CALL PROFAC

CALLING ARGUMENTS--
NAME ATTRIBUTES DEFINITION

FILES USED--
IFL4 -- RANDOM ELEMENT ATTRIBUTE FILE
INTITS -- HEAT TRANSFER OUTPUT FILE (INPUT TO TITAN)
IERF -- ERROR REMARK FILE

COMMONS USED--
AFIL -- FILECODE COMMON
ELMDAT -- ELEMENT ATTRIBUTE COMMON
SIZE -- SIZE OF PROBLEM VARIABLES IN THIS COMMON

FUNCTIONS/ROUTINES CALLED--
RWEILMS

LIBRARIES ACCESSED--

LOCAL VARIABLES--
NRED --- NUMBER OF NUMBERS READ FROM NEF FOR ONE SET OF ELEMENT FACE DEFINITIONS (LIMIT IS 200)
IFACPC --- PACKED FACE NUMBER
NCNT --- TOTAL NUMBER OF ELMS THAT HAVE FACES DEFINED

SPECIAL COMMENTS ABOUT THIS ROUTINE

HOST = H6000

END

-- GE/TRANCITS -- PROFAC PAGE= 56 133
SUBROUTINE SKIP (IFC, ISETS)

ROUTINE-- SKIP

DATE/WRITEEN BY-- 08/31/83  SR MARTIN

DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE ALLOWS THE CALLING ROUTINE TO SKIP PAST
UNNEEDED TIMESTEP AND TEMPERATURE DATA

CALLING ARGUMENTS --
NAME  ATTRIBUTES  DEFINITION
IFC    INPUT,INTEGER   FILE CODE
ISETS  INPUT,INTEGER   NUMBER OF SETS TO SKIP OVER

COMMONS USED --
CNTLFL
SIZE

FUNCTIONS/ROUTINES CALLED --

FILES USED --
FC  S

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --
DUMMY (CHARACTER) USED TO DUMMY READ LINES IN THE FILE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE HTFIO (IFC, DATA, ICNT, ISTAT)

DATE/Written By-- 09/13/83  SR MARTIN
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE READS INFO FROM THE HEAT TRANSFER INPUT FILE

CALLING ARGUMENTS --
IFC (INPUT/INTEGER) FILE CODE OF FILE TO READ FROM
DATA (OUTPUT/REAL) ARRAY OF DATA PASSED BACK
ICNT (INPUT/INTEGER) AMOUNT OF DATA IN 'DATA' ARRAY
ISTAT (OUTPUT/INTEGER) RETURN STATUS, VALUES ARE--
0-OK, 1-END OF FILE, 2-READ ERROR

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

IFC

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE RDCNTL

READ USER HEADER INFORMATION AND FILE COUNTERS

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION

COMMONS USED --
AFIL
CNTLF
CNTLM
SIZE

FUNCTIONS/ROUTINES CALLED --
HTFIO

FILES USED --
INTITS $ INOT6 $

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE TFLSET

ROUTINE-- TFLSET

DATE/WRITTEN BY-- 09/14/83 SR MARTIN

FUNCTION/PURPOSE--

THIS ROUTINE REWINDS THE HEAT TRANSFER FILE AND REPOSITIONS
THE FILE POINTER TO THE FIRST RECORD OF THE TIME/TEMP

INFORMATION

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

COMMONS USED --
AFIL
CNTLM

FUNCTIONS/ROUTINES CALLED --

FILES USED --
INTITS S

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
ROUTINE-- GENTMP

THIS ROUTINE IS THE MINI-EXEC USED TO CALL ALL ROUTINES ASSOCIATED WITH COMPUTING THE FINAL VALUES OF TEMPERATURE FOR EACH STRESS NODE.

IT CAN DEAL WITH BOTH FINITE DIFFERENCE AND FINITE ELEMENT HEAT TRANSFER CODES.

IF THE HEAT TRANSFER CODE IS FINITE DIFFERENCE, THE TEMPERATURES OF THE CORNERS OF THE HEAT TRANSFER ELEMENTS ARE FOUND VIA CALLS FROM THIS ROUTINE.

CALLING ARGUMENTS -- NONE

COMMONS USED --

MAFCOM
AFIL
MAFCOM
SIZE
MAFCOM
WORK
MAFCOM
CNTLM
MAFCOM
CNTLFL

FUNCTIONS/ROUTINES CALLED --

GETMPE
SETPFL
RDTEMP
CWCRNT
ETCORE
NTERC
STSTMP

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

-- HOST • H6000
HDST = H6000

END
SUBROUTINE STSTMP(TEMP,TIMVAL)

---

THIS ROUTINE READS THE COEFF INFO FOR EACH STRESS POINT
(VIA CALL RWSPCF) GETS THE CORNER TEMPS FROM THE TEMP ARRAY
AND COMPUTES THE TEMPERATURE OF THE STRESS POINT BY
SUMMING THE COEFFS TIMES THE CORNER TEMPS

CALLING ARGUMENTS --
NAME                  ATTRIBUTES       DEFINITION
TEMP                  (I)              ARRAY CONTAINING TEMPS OF HEAT
TIMVAL                (I)              TIME STEP VALUE

COMMONS USED --
MAFCOM
AFIL
MAFCOM
SIZE
SPDAT

FUNCTIONS/RUTINES CALLED --
RWSPCF
WRSTMP

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE WRSTMP(NAME, TEMP, TIMVAL, NDONE)

THIS ROUTINE WRITES THE STRESS POINT TEMPERATURES
TO THE OUTPUT FILE (IN NEUTRAL TEMPERATURE FILE FORM)

CALLING ARGUMENTS --
NAME (I) STRESS POINT LABEL
TEMP (I) TEMPERATURE OF STRESS POINT
TIMVAL (I) TIME VALUE OF STEP
NDONE (I) # OF NODES WRITTEN

COMMONS USED --
MAFCOM
AFIL
FUNCTIONS/ROUTINES CALLED -- NONE
FILES USED --
IOUTF S
LIBRARIES ACCESSED --
NONE
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END

-- GE/TRANCITS
-- WRSTMP
PAGE= 63
SUBROUTINE ETCORE(ETMP)

THIS ROUTINE TRANSFERS THE FINITE DIFFERENCE ELEMENTAL CORNER TEMPERATURES FROM SCRATCH FILE TO CORE

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
ETMP (O) ARRAY CONTAINING ELM CORNER TEMPS

COMMONS USED --
AFIL SIZE

FUNCTIONS/Routines CALLED --
FILREW

FILES USED --
ICRNTF S IERF S

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000
SUBROUTINE NTCORE(TEMP)

ROUTINE-- MAIN

FUNCTION/PURPOSE--
THIS ROUTINE ASSUMES THE FILE POINTER IS POINTING AT THE
FIRST NODE NUMBER, TEMPERATURE PAIR OF A
TIMESTEP IT READS IN EACH PAIR AND STORES THE
TEMPERATURE (FROM DATA(2)) INTO THE 'TEMP' ARRAY
IN POSITION <NODE NUMBER> (IE--DATA(1))
THE RESULT BEING THAT, FOR SAY NODE 12, THE TEMPERATURE
WILL BE IN RECORD 13 OF THE 'TEMP' ARRAY
UPON RETURN, THE FILE POINTER WILL POINT
TO THE NEXT TIMESTEP VALUE

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
TEMP REAL, OUTPUT ARRAY WHICH WILL CONTAIN
TEMPERATURE VALUES NOTE--
TEMP(I) WILL CONTAIN THE TEMPERATURE VALUE FOR NODE 'I'

COMMONS USED --
AFIL
SIZE

FUNCTIONS/ROUTINES CALLED --
HTFIO

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --
DATA (REAL, ARRAY) USED TO RETRIEVE THE NODE NUMBER,
TEMPERATURE PAIR FORM THE INPUT FILE
NODNUM (INTEGER) USED TO STORE THE NODE NUMBER
AS AS INTEGER AFTER RETRIEVING IT VIA THE
'DATA' ARRAY.

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE WRCRNT(TBAR, NODE)

ROUTINE -- WRCRNT

DATE/Written BY -- 09/29/83  RJ MAFFEO

DATE/REVISED BY --

FUNCTION/PURPOSE --

THIS ROUTINE WRITES THE FINITE DIFFERENCE CORNER TEMPS TO THE SCRATCH FILE.

CALLING ARGUMENTS --

NAME  ATTRIBUTES  DEFINITION

TBAR  (I)   VALUE OF TEMPERATURE

NODE  (I)   H T ELM CORNER NAME

COMMONS USED --

AFIL

FUNCTIONS/ROUTINES CALLED --

FILES USED --

ICRNTF  S

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE HTFPAS(IFC,NFP,IFILP)

DATE/WRITTEN BY-- 09/29/83  RJ MAFFEO

FUNCTION/PURPOSE--

THIS ROUTINE SKIPS PAST PORTIONS OF THE HEAT TRANSFER INPUT FILE IT CURRENTLY ONLY CAN SKIP PAST THE TIME-TEMPERATURE PORTION(IFILP=6)

DECLARATION

NAME ATTRIBUTES DEFINITION

IFC (I) FILE CODE OF FILE TO BE SKIPPED

NFP (I) # OF PARTITIONS TO SKIP

IFILP (I) PORTION OF FILE(DATA TYPE) TO BE SKIPPED

IFILP=1 THRU 5 NOT CURRENTLY USED

IFILP=6 SKIP TIME-TEMPERATURE RECORDS

COMMONS USED --

SIZE

FUNCTIONS/ROUTINES CALLED --

FILES USED --

IFC S

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE SORT(SEEDS,LAX,NDUM,IKEY)

ROUTINE-- SORT
DATE/Written by-- ??????? GE SORT ROUTINE
DATE/REVISED BY--
FUNCTION/PURPOSE--
THIS ROUTINE SORTS 1 DIMENSIONAL ARRAYS
CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
SEEDS (I/O) ARRAY TO BE SORTED
LAX (I) NUMBER OF ITEMS IN ARRAY
NDUM DUMMY DUMMY
IKEY DUMMY DUMMY
COMMONS USED -- NONE
FUNCTIONS/Routines CALLED -- NONE
FILES USED -- NONE
LIBRARIES ACCESSED --
NONE
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000
END
SUBROUTINE HTICON(IHTIC)

ROUTINE-- HTICON

DATE/WRITTEN BY-- 01/14/84 RJ MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE CONVERTS FROM THE HEAT TRANSFER OUTPUT FORM TO THE INTERNAL 3D TITAN HEAT TRANSFER INPUT FORM

VALID HEAT TRANSFER INPUT CODES ARE

0 --- INTERNAL 3D TITAN HEAT TRANSFER INPUT FORM
1 --- SINDA HEAT TRANSFER INPUT FORM
2 --- MARC HEAT TRANSFER INPUT FORM (FORMATTED)
3 --- MARC HEAT TRANSFER INPUT FORM (SEQUENTIAL BINARY)
4 --- THTD HEAT TRANSFER INPUT FORM

CALLING ARGUMENTS --
NAME       ATTRIBUTES       DEFINITION
IHTIC      (I)              HEAT TRANSFER INPUT CODE

COMMONS USED --
AFIL

FUNCTIONS/ROUTINES CALLED --
SINTIT
MARCRRD

FILES USED --
IERFS

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE MARCRD(IFTyp,NSTEPS)

ROUTINE-- MARCRD

DATE/WRITTEN BY-- 01/11/84  RD MCCLAIN

DATE/REVISED BY--

FUNCTION/PURPOSE--

READ MARC POST TAPE AND WRITE TRANSFER MODULE

INPUT FILE

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

IFTYP INPUT FILE TYPE OF MARC POST TAPE

NSTEPS INPUT NUMBER OF STEPS TO READ FROM

COMMONS USED --

AFIL -- FILE CODES

FUNCTIONS/ROUTINES CALLED --

READR READS DATA FROM POST TAPE

SKPREC SKIPS RECORDS ON POST TAPE

FILES USED --

TRANSFER MODULE INPUT FILE

MARC INPUT FILE (SEQ BIN OR FORMATTED)

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE READR(LFN, DATA, NDATA, CHAR, NWRDS, ITYPE, IFTYP, IEND)

ROUTINE-- READR
DATE/WRITTEN BY-- 01/11/84  RD MCCLAIN
DATE/REVISED BY--
FUNCTION/PURPOSE--
READ FLOATING POINT, INTEGER, OR CHARACTER DATA FROM MARC POST TAPE
CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
LFN INPUT FILE CODE TO READ FROM
DATA OUTPUT FLOATING POINT ARRAY
NDATA OUTPUT INTEGER ARRAY
CHAR OUTPUT CHARACTER ARRAY
NWRDS INPUT NUMBER OF ITEMS TO READ
ITYPE INPUT TYPE OF DATA TO READ
1=LOAD FP DATA INTO ARRAY DATA
2=LOAD INTEGER DATA INTO ARRAY NDATA
3=LOAD CHAR DATA INTO ARRAY CHAR
IFTYP INPUT FILE TYPE
0=FORMATTED
1=SEQUENTIAL BINARY
COMMONS USED --
AFIL -- FILE CODES
FUNCTIONS/ROUTINES CALLED --
FILES USED --
LFN S FILE CODE TO BE READ
INDT6 S STANDARD OUTPUT FILE
LIBRARIES ACCESSED --
NONE
LOCAL VARIABLES --
SPECIAL COMMENTS ABOUT THIS ROUTINE
HOST = H6000
SPECIAL REMARKS/INSTRUCTIONS --
END
SUBROUTINE SKPREC(IFILE,NUMREC,IFTYP)

ROUTINE-- SKPREC

DATE/WRITTEN BY-- 01/11/84    RD MCCLAIN

DATE/REVISED BY--

FUNCTION/PURPOSE--

SKIP RECORDS ON A SEQUENTIAL FILE

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

IFILE    INPUT    FILE CODE TO SKIP RECORDS ON
NUMREC    INPUT    NUMBER OF RECORDS TO SKIP
IFTYP    INPUT    FILE TYPE OF MARC POST TAPE

O=FORMATTED, 1=BINARY

COMMONS USED --   NONE

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE SINTIT

FUNCTION/PURPOSE--
MAIN EXECUTIVE ROUTINE FOR CREATION OF SINDA-TO-3DTITAN INTERFACE FILE

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
-----NONE-----

COMMONS USED --
WORK - STORES INTERNAL ARRAY DATA

WORK TEMP ARRAY OF ELEMENTAL TEMPERATURES
IPKFAC CORRESPONDING TO 'ITELM'
IGELM ARRAY OF ELEMENTS WITH GEOMETRY
ITELM ARRAY OF ELEMENTS WITH TEMPERATURES

AFIL - FILE CODES

AFIL - FILE CODES

ELEMENTS DEFINITION

IHTIN SINDA TEMPERATURE OUTPUT FILE (INPUT)
ISINGF SINDA GEOMETRY FILE (INPUT)
INTITS INTERFACE FILE (OUTPUT)
IERF HARDCOPY FILE (OUTPUT)

ELEMENTS DEFINITION

NUMNOD NUMBER OF NODES (CORNER POINTS)
NUMEWG NUMBER OF ELEMENTS WITH GEOMETRY
NUMTIM NUMBER OF SOLUTION TIMES
NNPE NUMBER OF NODES PER ELEMENT
ECODE ELEMENT CODE
#1 - 2D MODEL (NNPE=4)
#11 - 3D MODEL (NNPE=8)
ACODE ANALYSIS CODE
#1 - FINITE DIFFERENCE ANALYSIS
NUMWT NUMBER OF ELEMENTS WITH TEMPERATURES
NUMSKP NUMBER OF RECORDS TO SKIP ON 'IHTIN'
NUMFUL NUMBER OF FULL RECORDS OF DATA
NUMREM NUMBER OF WORDS ON FINAL RECORD

FUNCTIONS/ROUTINES CALLED --
DATINT
PRCSOF
PRCGEO
WRTNTF

FILES USED --
NONE
TITLE=  
DATE =07/15/85  
PROGRAM IS CURRENTLY DIMENSIONED TO PERMIT A MAXIMUM OF
5000 ELEMENTS WITH GEOMETRY AND 5000 ELEMENTS WITH
TEMPERATURES

SPECIAL REMARKS/INSTRUCTIONS --
HOST = HIS

END
SUBROUTINE DATINT

ROUTINE -- DATINT

DATE/WRITTEN BY -- 01/11/84 RN PITTMAN

DATE/REVISED BY --

FUNCTION/PURPOSE --

THIS ROUTINE Initializes ALL OF THE GLOBAL COMMON VARIABLES

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

COMMONS USED --

WORK -- STORES INTERNAL ARRAY DATA

AFIL -- FILE CODES (MAIN)

KNTDAT -- COUNTER DATA (MAIN)

FUNCTIONS/ROUTINES CALLED --

NONE

FILES USED --

WORK

LOCAL VARIABLES --

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H1012

END OF SUBROUTINE 'DATINT'

END
SUBROUTINE PRC Sof

Routine -- PRC Sof

Calling Arguments --

Commons Used --
AFIL -- File Codes (Main)

Functions/Routines Called --
GETEWT
GETNUMT

Files Used --
NONE

Libraries Accessed --
NONE

Local Variables --
NONE

Special Comments About This Routine --
NONE

Special Remarks/Instructions --
HOST = H6000

End Of Subroutine 'PRC Sof'

END
SUBROUTINE GETEWT

1 READS THE SINDA ELEMENT NUMBERS FROM 'IHTIN' INTO 'ITELM'

2 READS FROM 'IHTIN' THE TOTAL NUMBER OF SINDA ELEMENTS

3 COUNTS THE NUMBER OF RECORDS PRECEEDING THE FIRST TEMPERATURE SOLUTION ON 'IHTIN' AND STORES THIS VALUE IN 'NUMSKP'.

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION

COMMONS USED --
WORK -- STORES INTERNAL ARRAY DATA
AFIL -- FILE CODES (MAIN)
KNDAT -- COUNTER DATA (MAIN)

FUNCTIONS/ROUTINES CALLED --
ERRPRT

FILES USED --
IHTIN (S) SINDA OUTPUT FILE (INPUT TO THIS ROUTINE)

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --
IPOS - POSITION OF THE BEGINNING OF EACH GROUP OF 10 ELEMENT NUMBERS IN THE 'ITELM' ARRAY

***NOTE*** IN THIS ROUTINE, THE GLOBAL VARIABLES 'NUMFUL' AND 'NUMREM' ARE REDEFINED AS FOLLOWS-
NUMFUL - NUMBER OF 'FULL' 10-WORD RECORDS OF SINDA ELEMENT NUMBERS CONTAINED ON 'IHTIN'
NUMREM - NUMBER OF ELEMENT NUMBERS REMAINING ON FINAL RECORD OF ELEMENT NUMBERS

SPECIAL COMMENTS ABOUT THIS ROUTINE --

SPECIAL REMARKS/INSTRUCTIONS --
HOST = HG000

END OF SUBROUTINE 'GETEWT'
END
FUNCTION/PURPOSE--

THIS ROUTINE COUNTS THE TOTAL NUMBER OF SOLUTION BLOCKS
CONTAINED IN THE 'SINDA OUTPUT FILE' AND STORES THIS VALUE
IN THE GLOBAL VARIABLE 'NUMTIM'

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
-----NONE-----

COMMONS USED --
WORK --STORES INTERNAL ARRAY DATA
AFIL -- FILE CODES (MAIN)
KNTDAT -- COUNTER DATA (MAIN)

FUNCTIONS/ROUTINES CALLED --
ERRPRT

FILES USED --
IHTIN (S) SINDA OUTPUT FILE (INPUT TO THIS PROGRAM)

LIBRARIES ACCESSED --

LOCAL VARIABLES --
Dummy - VARIABLE USED IN DUMMY READS TO PASS THROUGH EACH
SOLUTION BLOCK TO THE NEXT
KNT - CURRENT COUNT OF THE NUMBER OF SOLUTION BLOCKS
ENCOUNTERED ON 'IHTIN'

***NOTE***
IN THIS ROUTINE, THE GLOBAL VARIABLES 'NUMFUL' AND 'NUMREM'
ARE REDEFINED AS FOLLOWS-
NUMFUL - NUMBER OF 'FULL' 6-WORD RECORDS OF TEMPERATURE
DATA IN EACH SOLUTION BLOCK ON 'IHTIN'
NUMREM - NUMBER OF TEMPERATURES ON FINAL RECORD OF EACH
SOLUTION BLOCK

THESE DEFINITIONS ARE NOT ALTERED FURTHER IN THIS PROGRAM

SPECIAL COMMENTS ABOUT THIS ROUTINE --

EACH TEMPERATURE SOLUTION BLOCK ON 'IHTIN' IS PASSED THROUGH
WITH DUMMY READS UNTIL THE END OF FILE IS ENCOUNTERED
THE SOLUTION COUNTER, 'KNT', IS INCREMENTED BY ONE AT THE
BEGINNING OF EACH BLOCK

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END OF SUBROUTINE 'GTNUMT'

END
SUBROUTINE PRCGEO

--- ROUTINE -- PRCGEO ---

DATE/Written BY-- 01/11/84 RN PITTMAN

DATE/REVISED BY--

FUNCTION/PURPOSE--
1. Reads the values contained in the 'DIRECTORY' record of
the 'SINDA GEOMETRY' file into the global common values
'NUMNOD', 'NUMEWG', and 'NNPE'.
2. Sets the value for 'ECODE' based upon the value of 'NNPE'.

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
--------------- ------- ------

COMMONS USED --
AFIL -- FILE CODES (MAIN)
KNTDAT -- CCOUNTER DATA (MAIN)

FUNCTIONS/Routines CALLED --
ERRPRT

FILES USED --
ISINGF (S) SINDA GEOMETRY FILE (INPUT)

LIBRARIES ACCESSED --

SPECIAL COMMENTS ABOUT THIS ROUTINE --
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

--- END OF SUBROUTINE 'PRCGEO' ---
SUBROUTINE WRTNTF

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

DATE/WRITTEN BY-- 01/11/84 RN PITTMAN
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE WRITES THE TITLE (RECORDS 1-4) AND DIRECTORY
(RECORD 5) RECORDS OF THE 'INTERFACE FILE'.

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
-----NONE-----

COMMONS USED --
WORK -- STORES INTERNAL ARRAY DATA
AFIL -- FILE CODES (MAIN)
KNTDAT -- COUNTER DATA (MAIN)

FUNCTIONS/ROUTINES CALLED --
ERRPRT

FILES USED --
INTITS (S) INTERFACE FILE (OUTPUT)
ISINGF (S) SINDA GEOMETRY FILE (INPUT)

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --
TITLE -- 80-CHARACTER TITLE CONTAINED ON RECORD 1 OF 'ISINGF'
USRTTL -- 80-CHARACTER ARRAY USED TO 'DUMMY-OUT' RECORDS 1-3 OF 'INTITS'

SPECIAL COMMENTS ABOUT THIS ROUTINE --

SPECIAL REMARKS/INSTRUCTIONS --
HOST = HGO00

END OF SUBROUTINE 'WRTDIR'

END
SUBROUTINE WRTGEO

DATE/WRITTEN BY-- 01/11/84  RN PITTMAN
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE WRITES THE GEOMETRICAL DATA (COORDINATE, CONNECTIVITY, AND FACES) ON THE INTERFACE FILE

CALLING ARGUMENTS--
NAME  ATTRIBUTES  DEFINITION
-----  --------  -------

COMMONS USED--
WORK  --STORES INTERNAL ARRAY DATA
AFIL -- FILE CODES (MAIN)
KNTDAT -- COUNTER DATA (MAIN)

FUNCTIONS/ROUTINES CALLED--
ERRPRT

FILES USED--
ISINGF  (S)  SINDA GEOMETRY FILE (INPUT)
INTITS  (S)  INTERFACE FILE (OUTPUT)

LIBRARIES ACCESSED--
NONE

LOCAL VARIABLES--
DUMMY  - USED IN 'DUMMY READ' TO POSITION 'ISINGF' AT BEGINNING OF COORDINATE DATA
N  - ARRAY OF NODE NUMBERS DEFINING CONNECTIVITY OF 'IGELM(I)'
NODE  - CURRENT NODE (CORNER POINT) NUMBER
X  - X-COORDINATE OF 'NODE'
Y  - Y-COORDINATE OF 'NODE'
Z  - Z-COORDINATE OF 'NODE'

SPECIAL COMMENTS ABOUT THIS ROUTINE--

SPECIAL REMARKS/INSTRUCTIONS--
HOST = H6000

END OF SUBROUTINE 'WRTGEO'

END
SUBROUTINE WRTEMP

DATE/WRITTEN BY-- 01/11/84  RN PITTMAN
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE WRITES THE TIME AND TEMPERATURE DATA ON THE
'INTERFACE FILE'

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
-----NONE-----

COMMONS USED --
WORK --STORES INTERNAL ARRAY DATA
AFIL -- FILE CODES (MAIN)
KNTDAT -- COUNTER DATA (MAIN)

FUNCTIONS/ROUTINES CALLED --
ERRPRT

FILES USED --
IHTIN (S)  SINDA OUTPUT FILE (INPUT)
INTITS (S)  INTERFACE FILE (OUTPUT)

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --
KNT - POSITION IN THE 'TEMP' ARRAY OF THE NEXT GROUP OF
SIX TEMPERATURES TO BE WRITTEN TO 'INTITS'
TIME - CURRENT VALUE OF TIME READ FROM 'IHTIN' FOR THIS
TEMPERATURE BLOCK

SPECIAL COMMENTS ABOUT THIS ROUTINE --
NONE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END OF SUBROUTINE 'WRTEMP'

END
SUBROUTINE ERRPRT (IROUT)

Routines-- ERRPRT

DATE/Written By-- 01/12/84 RN PITTMAN

FUNCTION/PURPOSE--

THIS ROUTINE PROCESSES ALL FATAL ERROR CONDITIONS BY
PRINTING AN EXPLANATION OF THE ERROR ON THE HARDCOPY
OUTPUT FILE AND TERMINATING PROGRAM EXECUTION

CALLING ARGUMENTS --

IROUT INTEGER/INPUT ROUTING VARIABLE FOR APPROPRIATE
ERROR MESSAGE

COMMONS USED --

AFIL -- FILE CODES (MAIN)

FUNCTIONS/ROUTINES CALLED --

NONE

FILES USED --

IERF ($) HARDCOPY FILE (OUTPUT)

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

NONE

SPECIAL COMMENTS ABOUT THIS ROUTINE --

NONE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE TEMFOR(ISPOC)

ROUTINE-- TEMFOR

DATE/WRITTEN BY-- 01/14/84  RJ MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE CALLS ALL THE REQUIRED ROUTINES TO FORMAT THE
OUTPUT TEMPERATURES INTO FORMED NEEDED FOR THE STRESS ANALYSIS
PROGRAM

VALID CODES ARE

0 --- NEUTRAL TEMPERATURE OUTPUT FILE ONLY

1 --- NASTRAN FORMATTED TEMPERATURES

2,3 --- MARC FORMATTED TEMPERATURES

CALLING ARGUMENTS --

NAME     ATTRIBUTES   DEFINITION

ISPOC (I) STRESS PROGRAM OUTPUT CODE

COMMONS USED --

AFIL

SIZE

FUNCTIONS/ROUTINES CALLED --

TPNAST

MARCTO

FILES USED --

IERF S

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE TPNAST(NUMNOD, NUMTMP)

ROUTINE-- TPNAST

DATE/Written by-- 01/14/84  RD MCCLAIN
DATE/REVISED BY--

FUNCTION/PURPOSE--

WRITE NASTRAN BULK DATA TEMPERATURE CARDS

CALLING ARGUMENTS --

NAME  ATTRIBUTES  DEFINITION
NUMNOD  INPUT  NUMBER OF NODES ON NEUT TEMP FILE
NUMTMP  INPUT  NUMBER OF TEMPERATURE SETS

COMMONS USED --

AFIL  --  FILE CODES

FUNCTIONS/ROUTINES CALLED --

FILES USED --

IOUTF  S  NEUTRAL OUTPUT FILE
ITEMP  S  NASTRAN BULK DATA FILE

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS--

HOST = H6000

END
SUBROUTINE I3DSF(CMNMX, CRD, XP, YP, ZP, NSP, 
ROUTINE-- I3DSF
DATE/WRITTEN BY-- 11/28/83 R J MAFFEO
DATE/REVISED BY--
FUNCTION/PURPOSE--
THIS ROUTINE COMPUTES THE VALUES OF THE LOCAL COORDINATES
OF A STRESS POINT WRT THE VERTICES OF THE 8-NODED HEAT
TRANSFER ELEMENT. IF THE STRESS POINT LIES IN THE ELM
IT COMPUTES THE 8 WEIGHTING COEFFICIENTS FOR THIS POINT
IF THE POINT LIES OUTSIDE THE ELM IT COMPUTES THE SURFACE
COORDINATES OF THE POINT PROJECTED IN THE LOCAL COORD SYSTEM
AND COMPUTES THE WEIGHTING COEFFS BASED ON THESE SURFACE
COORDINATES IT ALSO COMPUTES THE DISTANCE THE POINT IS AWAY
FROM THE SURFACE
CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
CMNMX (I) ARRAY WITH MIN AND MAX WINDOW COORDS
CRD (I) ARRAY WITH ELM VERTEX COORDS
XP (I) STRESS POINT COORDS
YP (I) STRESS POINT COORDS
ZP (I) STRESS POINT NAME
NSP (I) STRESS POINT NAME
XLOC (O) LOCAL COORDS OF STRESS POINT
YLOC (O) LOCAL COORDS OF STRESS POINT
ZLOC (O) LOCAL COORDS OF STRESS POINT
SF (O) WEIGHTING COEFFICIENTS
ICONV (O) CONVERGENCE PARAMETER
IPAS (I) SEARCH TYPE PARAMETER
DIST (O) DISTANCE FROM OUTSIDE POINT TO SURFACE
XSUR (O) SURFACE COORDINATES
YSUR (O) SURFACE COORDINATES
ZSUR (O) SURFACE COORDINATES
COMMONS USED --
AFIL
FUNCTIONS/ROUTINES CALLED --
REAARY
SHAPFV
CHKPT
JACBCK
SFDIST
FILES USED --
LIBRARIES ACCESSED --
GE/TRANCITS
I3DSF
LOCAL VARIABLES --

IPINT --- ACCURACY INDICATOR
0 -- STRESS POINT NOT RECALCULATED ACCURATELY
1 -- STRESS POINT RECALCULATED ACCURATELY

IJC --- JACOBIAN CHECK PARAMETER
0 -- JACOBIAN NOT POSITIVE
1 -- JACOBIAN IS POSITIVE

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000
SUBROUTINE REAARY(X,Y,Z,XP,YP,ZP,SF,ICLOS,ITRYC,DTOCN,XL,YL,ZL)

ROUTINE-- REAARY

DATE/WRITTEN BY-- 11/28/83  R J MAFFEO
DATE/REVISED BY--
FUNCTION/PURPOSE--

THIS ROUTINE REARRANGES THE CONNECTIVITY COORDS AND
THE SHAPE FUNCTION ARRAYS IN AN ATTEMPT TO GET AN
ORDER THAT WILL ENHANCE CONVERGENCE OF THE INVERSE
SHAPE FUNCTION ROUTINE(I3DSF). IT ALSO COMPUTES THE
DISTANCE FROM THE STRESS POINT TO THE NEAREST ELM
CORNER OPTIONALY IT WILL "UN-REARRANGE" THE
SHAPE FUNCTION ARRAY AND THE LOCAL COORDS

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

X (I/O) >> CONNECTIVITY COORD ARRAYS
Y (I/O) >>>
Z (I/O) >>
XP (I) >>
YP (I) >>> COORDS OF STRESS POINTS
ZP (I) >>
SF (I/O) SHAPE FUNCTION ARRAY
ICLOS (I/O) VALUE OF NEW JOINT
ITRYC (I) CONTROL ARGUMENT

-1 -- REARRANGE SHAPE FUNCTION ARRAY
0 -- PICK JOINT AS JOINT NEAREST TO STRESS POINT

DTOCN (O) N>0 -- USE VALUE OF N AS JOINT
DISTANCE TO NEAREST CORNER

XL (I/O) >>> VALUES OF LOCAL COORDINATES
YL (I/O) >>>>
ZL (I/O) >>>>

COMONS USED --

FILES USED --

FILES USED --

FUNCTIONS/ROUTINES CALLED --

SORT2

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE
NOTE THE IJ ARRAY IS USED TO RELATE THE REARRANGED
CONNECTIVITY ARRAY TO THE JOINT PICKED AS THE I JOINT

END

--- REAARY PAGE= 89
SUBROUTINE SORT2(SEEDS, FOLLO, LAX)

DATE/Written BY-- ????????? GE DOUBLE SORT

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE Sorts TWO 1 DIMENSIONAL ARRAYS

CALLING ARGUMENTS--

NAME | ATTRIBUTES | DEFINITION
--- | --- | ---
SEEDS | (I/O) PRIMARY ARRAY TO BE SORTED
FOLLO | (I/O) ARRAY TO BE SORTED LIKE THE PRIMARY
LAX | (I) NUMBER OF ITEMS IN ARRAYS

COMMONS USED -- NONE

FUNCTIONS/Routines CALLED -- NONE

FILES USED -- NONE

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
**SUBROUTINE SFDIST(SF,DIST,XNC,YNC,ZNC,XSUR,YSUR,ZSUR,XP,YP,ZP,**

**FUNCTION/PURPOSE--**

THIS ROUTINE COMPUTES THE DISTANCE FROM STRESS POINTS THAT ARE OUTSIDE A HEAT TRANSFER ELM TO THE SURFACE OF THE HEAT TRANSFER ELM. IT ALSO COMPUTES THE SURFACE COORDINATES OF THE POINT ON THE SURFACE THAT IS THE 'TRUNCATING' POINT OF THE LOCAL COORDINATES TO 1 OR -1.

**CALLING ARGUMENTS --**

<table>
<thead>
<tr>
<th>NAME</th>
<th>Attributes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>(I)</td>
<td>SHAPE FUNCTION ARRAY</td>
</tr>
<tr>
<td>DIST</td>
<td>(O)</td>
<td>COMPUTED DISTANCE</td>
</tr>
<tr>
<td>XNC</td>
<td>(I)</td>
<td>LOCAL VALUES OF STRESS POINT</td>
</tr>
<tr>
<td>YNC</td>
<td>(I)</td>
<td>LOCAL VALUES OF STRESS POINT</td>
</tr>
<tr>
<td>ZNC</td>
<td>(I)</td>
<td>LOCAL VALUES OF STRESS POINT</td>
</tr>
<tr>
<td>XSUR</td>
<td>(O)</td>
<td>COORDS OF SURFACE POINT</td>
</tr>
<tr>
<td>YSUR</td>
<td>(O)</td>
<td>COORDS OF SURFACE POINT</td>
</tr>
<tr>
<td>ZSUR</td>
<td>(O)</td>
<td>COORDS OF SURFACE POINT</td>
</tr>
<tr>
<td>XP</td>
<td>(I)</td>
<td>COORDS OF STRESS POINT</td>
</tr>
<tr>
<td>YP</td>
<td>(I)</td>
<td>COORDS OF STRESS POINT</td>
</tr>
<tr>
<td>ZP</td>
<td>(I)</td>
<td>COORDS OF STRESS POINT</td>
</tr>
<tr>
<td>NNPE</td>
<td>(I)</td>
<td>NUMBER OF NODES PER ELM</td>
</tr>
<tr>
<td>CRD</td>
<td>(I)</td>
<td>COORDS OF VERTICES OF ELM</td>
</tr>
</tbody>
</table>

**COMMONS USED --**

**FUNCTIONS/ROUTINES CALLED --**

SHAPFV

**FILES USED --**

**LIBRARIES ACCESSED --**

NONE

**LOCAL VARIABLES --**

**SPECIAL COMMENTS ABOUT THIS ROUTINE**

**SPECIAL REMARKS/INSTRUCTIONS --**

HOST = H6000
SUBROUTINE CHKPT(SF,X,Y,Z,XS,YS,ZS,IPOINT)

ROUTINE-- CHKPT

DATE/WRITTEN BY-- 11/28/83  R J. MAFFEO

DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE RECOMPUTES THE STRESS POINT COORDS BASED
ON THE VALUES OF THE LOCAL COORDINATES TO CHECK FOR

ACCURACY

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

SF (I) ARRAY OF SHAPE FUNCTIONS

X (I) ARRAY OF ELM CORNER COORDS

Z (I) ORIGINAL STRESS POINT COORDS

XS (I) 

YS (I) 

ZS (I) 

IPOINT (0) ACCURACY INDICATOR

0 -- DID NOT REPRODUCE COORD

1 -- DID REPRODUCE COORD

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END

--- GE/TRANCITS

--- CHKPT PAGE= 92

---
SUBROUTINE PDOFSF(TETA,YETA,ZETA,IGP,IND,PD)

FUNCTION/PURPOSE--

THIS SUB COMPUTES THE DERIVATIVES OF THE
SHAPE FUNCTIONS WRt THE LOCAL SYSTEM

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
--- --- ----
TETA (I) > LOCAL COORDINATES
YETA (I) >> GAUSS POINT COUNTER
ZETA (I) >
IGP (I) 2D/3D INDICATOR
IND (I)  IND=0 >> 3D
IND=1 THRU 6 >> 2D IND*FACE NUMBER
PD (O)  ARRAY OF DERIVATIVES

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

END
SUBROUTINE FNDJAC(X,Y,Z,P3D,XJAC)

ROUTINE-- FNDJAC

DATE/WRITTEN BY-- 11/28/83 R.J MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE COMPUTES THE VALUE OF THE DETERMINANT OF
THE JACOBIAN EVALUATED AT THE STRESS POINT

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
X (I) >>
Y (I) >>> ARRAY OF ELM CORNER COORDS
Z (I) >>
P3D (I) ARRAY WITH PARTIAL DERIVATIVES OF
LOCAL COORD WRT THE GLOBAL COORD
XJAC (O) VALUE OF DETERMINANT OF JACOBIAN

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE SHAPFV(SF,XNC,YNC,ZNC)

DATE/WRITTEN BY-- 11/28/83   R J MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--
THIS ROUTINE FINDS THE VALUES OF THE SHAPE FUNCTIONS
FOR THE 8-NODED ISOPARAMETRIC ELM BASED ON THE VALUES
OF THE LOCAL COORDINATES

CALLING ARGUMENTS --
NAME       ATTRIBUTES   DEFINITION
SF         (O)          ARRAY WITH SHAPE FUNCTIONS
XNC        (I) >>
YNC        (I) >>> LOCAL COORDS
ZNC        (I) >>

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END

END
SUBROUTINE JACBCK(XL, YL, ZL, X, Y, Z, XJAC, IJACB)

DATE/WRITTEN BY-- 11/28/83  R.U. MAFFEO
DATE/REVISED BY--

THIS ROUTINE CALLS THE REQUIRED ROUTINES TO CHECK THE VALUE OF THE DETERMINANT OF THE JACOBIAN AT THE STRESS POINT

CALLING ARGUMENTS --
NAME  ATTRIBUTES  DEFINITION
XL (I) >>  LOCAL COORDS
YL (I) >>>  LOCAL COORDS
ZL (I) >>  LOCAL COORDS
X (I) >>  LOCAL COORDS
Y (I) >>>  LOCAL COORDS
Z (I) >>  LOCAL COORDS
XJAC (D) VALUE OF DETERMINANT OF JACOBIAN
IJACB (D) INTERGER VARIABLE
O -- JACOBIAN NOT POSITIVE
1 -- JACOBIAN IS POSITIVE

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --
PDOFSF
FNDJAC
FILES USED --

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE
SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE RWSPCF(IREC,B,IRW)

ROUTINE-- RWSPCF

DATE/Written By-- 04/07/83  RJ Maffeo

FUNCTION/PURPOSE--

This routine reads and writes data to the
random stress point coeff file.

This file contains the following info per record:
- Stress node name (1)
- Stress node name (1)
- Heat tran elm containing this stress point (1)
- Connectivity of the heat tran elm (8)
- Weighting coeff for this stress node (8)

CALLING ARGUMENTS --

NAME  ATTRIBUTES  DEFINITION
IREC   (I)       FILE RECORD NUMBER
B      (I/O)     DATA ARRAY
IRW    (I)       READ/WRITE OPTION

COMMONS USED --
AFIL

FUNCTIONS/ROUTINES CALLED --
IRSPCF

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

HOST = H6000

SPECIAL REMARKS/INSTRUCTIONS --
106190C SUBROUTINE ELTRAN
106210C ROUTINE-- ELTRAN
106230C DATE/WRITTEN BY-- 01/27/84 RJ MAFFEO
106250C DATE/REVISED BY--
106270C FUNCTION/PURPOSE--
106280C Validate the values input for IECODE and IACODE then assign
106290C the values to NNPE and NTPE depending on the values of
106300C IECODE and IACODE
106310C
106320C CALLING ARGUMENTS --
106330C NAME ATTRIBUTES DEFINITION
106340C AFIL
106350C CNTLFL
106360C
106370C COMMONS USED --
106380C
106390C AFIL
106400C CNTLFL
106410C
106420C FUNCTIONS/Routines Called --
106430C
106440C FILES USED --
106450C
106460C IERF S
106470C
106480C LIBRARIES ACCESSED --
106490C
106500C
106510C LOCAL VARIABLES --
106520C
106530C SPECIAL COMMENTS ABOUT THIS ROUTINE
106540C
106550C SPECIAL REMARKS/INSTRUCTIONS --
106560C
106570C
106580C
106590C
106600C
106610C
106620C
106630C
106640C
106650C
106660C
106670C
106680C
106690C
106700C
106710C
106720C END
107330

-- GE/TRANCITS -- ELTRAN PAGE= 98

175
SUBROUTINE INITCM

DATE/WRITTEN BY-- 01/31/84  RJ MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE Initializing the CRNDEF COMMON
COMMON IS USED IN CALCRD AND I3DSF ROUTINES

CALLING ARGUMENTS --

COMMONS USED --
CRNDEF

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE MARCTD(ISPOC, NUMSP, NUMSOL)

DATE/WRITTEN BY-- 03/08/84  RJ MAFFEO

FUNCTION/PURPOSE--

THIS ROUTINE CALLS ALL REQUIRED ROUTINES TO COMPUTE
THE MARC 8-NODED ELM GAUSS POINT TEMPERATURES(OR THE
ELM CENTROIDAL TEMPERATURE)

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION
ISPOC (I) DESIRED ELM LOCATION
ISPOC=2 --- CENTROIDAL TEMPERATURE
ISPOC=3 --- 2X2X2 GAUSS POINT TEMPERATURES
NUMSP (I) NUMBER OF STRESS MODEL NODAL POINTS
NUMSOL (I) NUMBER OF TRANSIENT SOLUTIONS

COMMONS USED --
AFIL
WORK

FUNCTIONS/ROUTINES CALLED --
GPCOEF
GPTEMP

FILES USED --
ISNEF  S
IOUTF  S
ITEMP  S
IERF  S

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE GPCOEF(GPF)

ROUTINE-- GPCOEF

DATE/WRITTEN BY-- 03/07/84 RJ MAFFEO
DATE/REVISED BY--

FUNCTION/PURPOSE--

THIS ROUTINE COMPUTES AND STORES THE WEIGHTING
COEFFS ASSOCIATED WITH THE 2X2X2 GAUSS POINTS
USED FOR THE MARC SOLID ELM(ELM TYPE 43)

CALLING ARGUMENTS --
NAME ATTRIBUTES DEFINITION
GPF (O) ARRAY WITH GAUSS PT COEFF

COMMONS USED --

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --
NONE

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --
HOST = H6000

END
SUBROUTINE GPTEMP(TA,IE,GPF,ISPOC,TV)

FUNCTION/PURPOSE--

THIS ROUTINE COMPUTES THE VALUE OF THE ELM TEMPERATURE FOR THE 8-NODED 3D MARC STRESS ELM. IT WILL COMPUTE TEMPS AT THE 8 2X2X2 GAUSS POINTS OR AT THE ELM CENTROID.

CALLING ARGUMENTS --

NAME ATTRIBUTES DEFINITION

TA (I) NODAL TEMPERATURE ARRAY
IE (I) ELM CONNECTIVITY ARRAY
GPF (I) GAUSS PT COEFF ARRAY
ISPOC (I) TEMPERATURE LOCATION PARAMETER
TV (O) ELEMENTAL TEMPERATURE ARRAY

COMMONS USED --

NODAL TEMPERATURE ARRAY
ELM CONNECTIVITY ARRAY
GAUSS PT COEFF ARRAY
TEMPERATURE LOCATION PARAMETER
ELEMENTAL TEMPERATURE ARRAY
TV(1)=CENTROIDAL TEMPERATURE

FUNCTIONS/ROUTINES CALLED --

FILES USED --

LIBRARIES ACCESSED --

LOCAL VARIABLES --

SPECIAL COMMENTS ABOUT THIS ROUTINE

SPECIAL REMARKS/INSTRUCTIONS --

HOST = H6000

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.
THIS IS CALLING STACK STARTING FROM $$$ MAI

---------
MAIN
---------

\**INITCM**
**********
001

**RFILCR**
**********
002

**RANSIZ**
**********
003

**ZERORF**
**********
004

**FTIME**
**********
005

**HTICON**
**********
006

**SINTIT**
**********
<table>
<thead>
<tr>
<th>TITLE=</th>
<th>DATE =07/15/85</th>
<th>TIME = 23 58</th>
<th>PAGE= 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORNCC</td>
<td>CD 3</td>
<td>CNTLGM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 101</td>
<td>RWELEMS</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 102</td>
<td>SORI</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 103</td>
<td>CCNVEC</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 105</td>
<td>GETCRD</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 106</td>
<td>CINV</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 108</td>
<td>RWCCF</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>SU 5 109</td>
<td>PUTWFI</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CORNCC</td>
<td>WR 8</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CCNVEC</td>
<td>SU 1</td>
<td>CCNVEC</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CCNVEC</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CCNVEC</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CCNVEC</td>
<td>WR 8</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETCRD</td>
<td>SU 1</td>
<td>GETCRD</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETCRD</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETCRD</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETCRD</td>
<td>WR 8</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>PUTWFI</td>
<td>SU 1</td>
<td>PUTWFI</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>PUTWFI</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>PUTWFI</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RWCCF</td>
<td>SU 1</td>
<td>RWCCF</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RWCCF</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RWCCF</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RWCCF</td>
<td>RD 7</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RWCCF</td>
<td>WR 8</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>SU 1</td>
<td>GETMPE</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>CD 3</td>
<td>CNTLFL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>CD 3</td>
<td>SIZE</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>SU 5 101</td>
<td>HTFID</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>GETMPE</td>
<td>SU 5 103</td>
<td>MPERW</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>MPERW</td>
<td>WR 8</td>
<td>MPERW</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>SETFLP</td>
<td>SU 1</td>
<td>SETFLP</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>SETFLP</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>SETFLP</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>SETFLP</td>
<td>SU 5 101</td>
<td>TFLSET</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>SETFLP</td>
<td>SU 5 102</td>
<td>HTFPAS</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>SU 1</td>
<td>RDTEMP</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>CD 3</td>
<td>SIZE</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>CD 3</td>
<td>CNTLFL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>SU 5 101</td>
<td>HTFIO</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>RDTEMP</td>
<td>WR 8</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 1</td>
<td>CWCRNT</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 5 101</td>
<td>FILEREW</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 5 102</td>
<td>RWCCF</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 5 103</td>
<td>UNPAKF</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 5 104</td>
<td>MPERW</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 5 105</td>
<td>FCRNTP</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 5 106</td>
<td>MAXTDF</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>CWCRNT</td>
<td>SU 1 107</td>
<td>WRCRNT</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>UNPAKF</td>
<td>SU 1</td>
<td>UNPAKF</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>UNPAKF</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>UNPAKF</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>UNPAKF</td>
<td>WR 8</td>
<td>YES</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>FCRNTP</td>
<td>SU 1</td>
<td>FCRNTP</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>FCRNTP</td>
<td>CD 3</td>
<td>MAFCOM</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>FCRNTP</td>
<td>CD 3</td>
<td>AFIL</td>
<td>NEWSRC</td>
</tr>
<tr>
<td>MAXTDF</td>
<td>SU 1</td>
<td>MAXTDF</td>
<td>NEWSRC</td>
</tr>
</tbody>
</table>

--- GE/Z09 --- PAGE= 3
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 GETMP NEWSRC 100054
GENTMP SU 5 103 SETFMP NEWSRC 100054
GENTMP SU 5 104 RDTMP NEWSRC 100054
GENTMP SU 5 105 CMECMT NEWSRC 100054
GENTMP SU 5 106 ETCORE NEWSRC 100054
GENTMP SU 5 107 NTCORE NEWSRC 100054
GENTMP SU 5 108 STSTMP NEWSRC 100054
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 CO 3 WRSTMP 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
HTFPA SU 1 HTFPA NEWSRC 100060
HTFPA CO 3 MAFCOM M NEWSRC 100060
HTFPA CO 3 SIZE C NEWSRC 100060
HTFPA 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 SINIT NEWSRC 100062
<table>
<thead>
<tr>
<th>TITLE=</th>
<th>DATE =07/15/85</th>
<th>TIME = 23 58</th>
<th>PAGE= 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTICON SU 5 102 MARCRD</td>
<td>NEWSRC 100062</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTICON WR 8 YES</td>
<td>NEWSRC 100062</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARCRD SU 1 MARCRD</td>
<td>NEWSRC 100063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARCRD CO 3 AFIL C</td>
<td>NEWSRC 100063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARCRD SU 5 101 READR</td>
<td>NEWSRC 100063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARCRD SU 5 103 SKPREC</td>
<td>NEWSRC 100063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARCRD WR 8 YES</td>
<td>NEWSRC 100063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READR SU 1 READR</td>
<td>NEWSRC 100064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READR CO 3 AFIL C</td>
<td>NEWSRC 100064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READR RD 7 YES</td>
<td>NEWSRC 100064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READR WR 8 YES</td>
<td>NEWSRC 100064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKPREC SU 1 SKPREC</td>
<td>NEWSRC 100065</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKPREC CO 3 AFIL C</td>
<td>NEWSRC 100065</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKPREC RD 7 YES</td>
<td>NEWSRC 100065</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKPREC WR 8 YES</td>
<td>NEWSRC 100065</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT SU 1 SINTIT</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT CO 3 MAFCOM M</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT CO 3 WORK C</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT CO 3 MAFCOM M</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT CO 3 AFIL C</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT CO 3 KNTDAT C</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT SU 5 101 DATINT</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT SU 5 102 PRCSOF</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT SU 5 103 PRCGEO</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTIT SU 5 104 WRTNTF</td>
<td>NEWSRC 100066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATINT SU 1 DATINT</td>
<td>NEWSRC 100067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATINT CO 3 MAFCOM M</td>
<td>NEWSRC 100067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATINT CO 3 WORK C</td>
<td>NEWSRC 100067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATINT CO 3 MAFCOM M</td>
<td>NEWSRC 100067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATINT CO 3 AFIL C</td>
<td>NEWSRC 100067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATINT CO 3 KNTDAT C</td>
<td>NEWSRC 100067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCSOF SU 1 PRCSOF</td>
<td>NEWSRC 100068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCSOF CO 3 MAFCOM M</td>
<td>NEWSRC 100068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCSOF CO 3 AFIL C</td>
<td>NEWSRC 100068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCSOF SU 5 101 GETEWT</td>
<td>NEWSRC 100068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCSOF SU 5 102 GTNUTM</td>
<td>NEWSRC 100068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT SU 1 GETEWT</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT CO 3 MAFCOM M</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT CO 3 WORK C</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT CO 3 MAFCOM M</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT CO 3 AFIL C</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT CO 3 KNTDAT C</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT SU 5 101 ERRPRT</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETEWT RD 7 YES</td>
<td>NEWSRC 100069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM SU 1 GTNUTM</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM CO 3 MAFCOM M</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM CO 3 WORK C</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM CO 3 MAFCOM M</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM CO 3 AFIL C</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM CO 3 KNTDAT C</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM SU 5 101 ERRPRT</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTNUTM RD 7 YES</td>
<td>NEWSRC 100070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCGEO SU 1 PRCGEO</td>
<td>NEWSRC 100071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCGEO CO 3 MAFCOM M</td>
<td>NEWSRC 100071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCGEO CO 3 AFIL C</td>
<td>NEWSRC 100071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCGEO CO 3 KNTDAT C</td>
<td>NEWSRC 100071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCGEO SU 5 101 ERRPRT</td>
<td>NEWSRC 100071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRCGEO RD 7 YES</td>
<td>NEWSRC 100071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF SU 1 WRTNTF</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF CO 3 MAFCOM M</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF CO 3 WORK C</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF CO 3 MAFCOM M</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF CO 3 AFIL C</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF CO 3 KNTDAT C</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF SU 5 101 WRTDIR</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF SU 5 102 WRTGEO</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRTNTF SU 5 103 WRTTEMP</td>
<td>NEWSRC 100072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>CO</td>
<td>Variable</td>
<td>Source</td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>RWSPEF</td>
<td>3</td>
<td>MAFCOM</td>
<td>M</td>
</tr>
<tr>
<td>RWSPEF</td>
<td>3</td>
<td>AFIL</td>
<td>C</td>
</tr>
<tr>
<td>RWSPEF</td>
<td>7</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>RWSPEF</td>
<td>8</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>ELTRAN</td>
<td>1</td>
<td>ELTRAN</td>
<td></td>
</tr>
<tr>
<td>ELTRAN</td>
<td>3</td>
<td>MAFCOM</td>
<td>M</td>
</tr>
<tr>
<td>ELTRAN</td>
<td>3</td>
<td>AFIL</td>
<td>C</td>
</tr>
<tr>
<td>ELTRAN</td>
<td>3</td>
<td>MAFCOM</td>
<td>M</td>
</tr>
<tr>
<td>ELTRAN</td>
<td>3</td>
<td>CNILFL</td>
<td>C</td>
</tr>
<tr>
<td>ELTRAN</td>
<td>8</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>INITCM</td>
<td>1</td>
<td>INITCM</td>
<td></td>
</tr>
<tr>
<td>INITCM</td>
<td>3</td>
<td>CRNDEF</td>
<td>C</td>
</tr>
<tr>
<td>MARCTO</td>
<td>1</td>
<td>MARCTO</td>
<td></td>
</tr>
<tr>
<td>MARCTO</td>
<td>3</td>
<td>MAFCOM</td>
<td>M</td>
</tr>
<tr>
<td>MARCTO</td>
<td>3</td>
<td>AFIL</td>
<td>C</td>
</tr>
<tr>
<td>MARCTO</td>
<td>3</td>
<td>MAFCOM</td>
<td>M</td>
</tr>
<tr>
<td>MARCTO</td>
<td>3</td>
<td>WORK</td>
<td>C</td>
</tr>
<tr>
<td>MARCTO</td>
<td>5</td>
<td>101 GPCOEF</td>
<td></td>
</tr>
<tr>
<td>MARCTO</td>
<td>5</td>
<td>102 GPTMP</td>
<td></td>
</tr>
<tr>
<td>MARCTO</td>
<td>7</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>MARCTO</td>
<td>8</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>GPCOEFF</td>
<td>1</td>
<td>GPCOEFF</td>
<td></td>
</tr>
<tr>
<td>GPTMP</td>
<td>1</td>
<td>GPTMP</td>
<td></td>
</tr>
</tbody>
</table>
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