IN-39
157881
p. 133

NASA Technical Memorandum 106004

High Temperature Composite Analyzer (HITCAN) Programmer's Manual

Version 1.0

J.J. Lackney
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio

and

P.L.N. Murthy and P.K. Gotsis
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

April 1993

(NASA-TM-106004) HIGH TEMPERATURE COMPOSITE ANALYZER (HITCAN) PROGRAMMER'S MANUAL. VERSION 1.0 (Sverdrup Technology) 133 p

N93-23056

Unclas

G3/39 0157881

NASA

TABLE OF CONTENTS

ABSTRACT

This manual describes the organization and flow of data and analysis modules in the computer code, HITCAN (HIgh Temperature Composite ANalyzer). HITCAN is a general purpose computer program for predicting nonlinear global structural and local stress-strain response of arbitrarily oriented, multilayered high temperature metal matrix composite structures. This manual describes the architecture of the HITCAN code, followed by the listing of subroutines and calling tree, data storage scheme, file system, and a dictionary of code terminology. The primary intention of the manual is to familiarize the user with some of the computer program related issues so as to facilitate maintenance/modification/updates of the HITCAN computer code.

CHAPTER 1

INTRODUCTION

HITCAN (High Temperature Composite ANalyzer) is a general purpose computer program for predicting global structural and local stress-strain response of high temperature metal matrix composite structures. This document describes the internal architecture of the code.

The HITCAN computer program is a combination of three computer programs developed in-house at the NASA Lewis Research Center (LeRC). They are:

- a dedicated mesh generator; COBSTRAN (Reference 1)
- a finite element structural analysis code; MHOST (Reference 2)
- a metal matrix composite analyzer; METCAN (Reference 3).

HITCAN itself is made up of approximately 112 subroutines consisting of 16,000 lines of coding. COBSTRAN, METCAN, and MHOST consist of approximately 7,000 10000, and 51000 lines, respectively. HITCAN is written in FORTRAN 77. At the present time HITCAN has been configured and executed on both the CRAY XMP and YMP computers at NASA LeRC.

This document describes:

- the control structure of the program;
- the data storage scheme and the memory allocation procedure;
- the file management facilities.

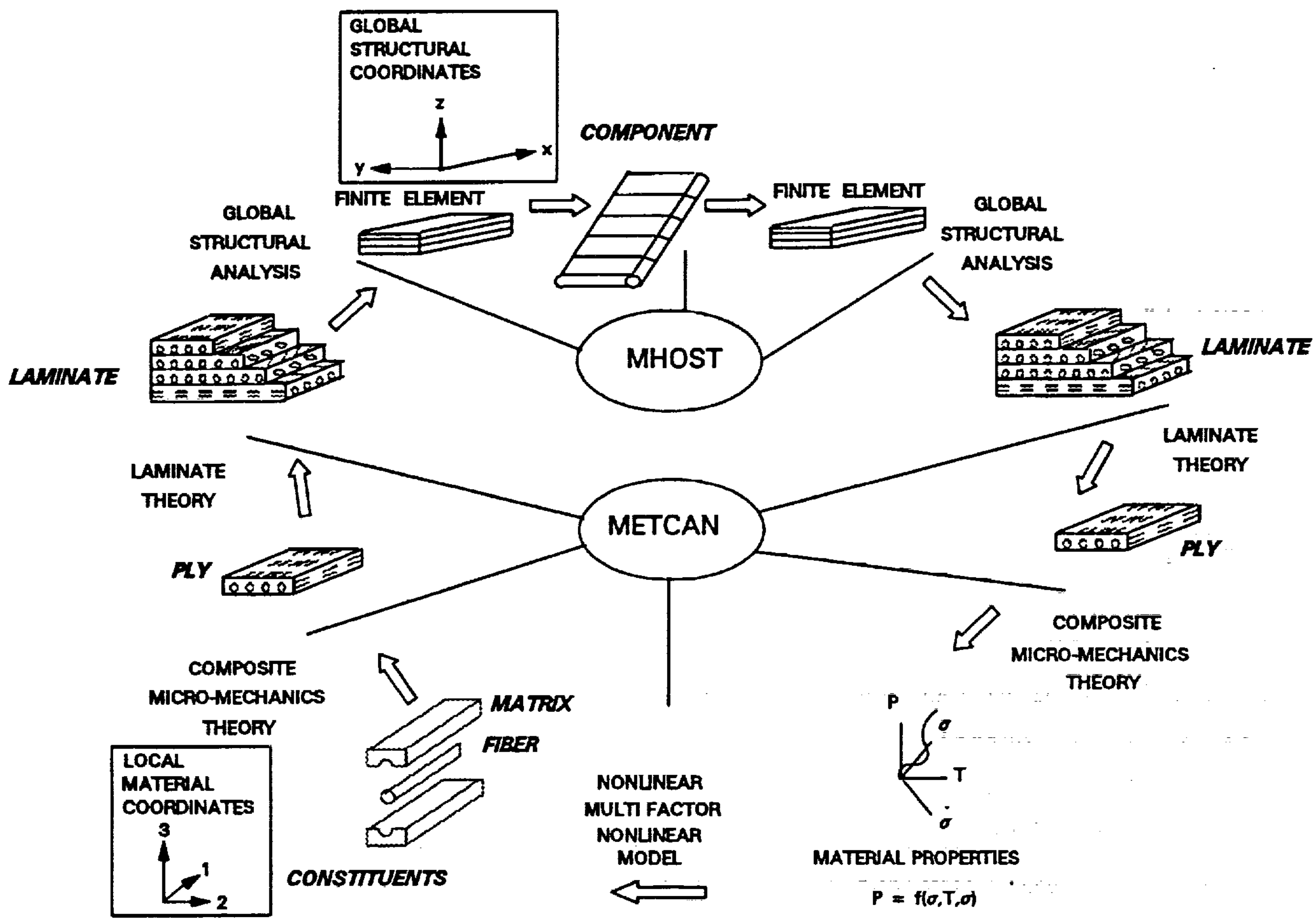
The architecture of the program is described in Chapter 2. The HITCAN calling tree and a brief description of each subroutine can be found in Chapter 3. The common blocks and the memory allocation procedure is reviewed in Chapter 4. The file system is described in Chapter 5. Finally, in Chapter 6, a dictionary of most of the variables used in HITCAN is provided.

CHAPTER 2

PROGRAM ARCHITECTURE

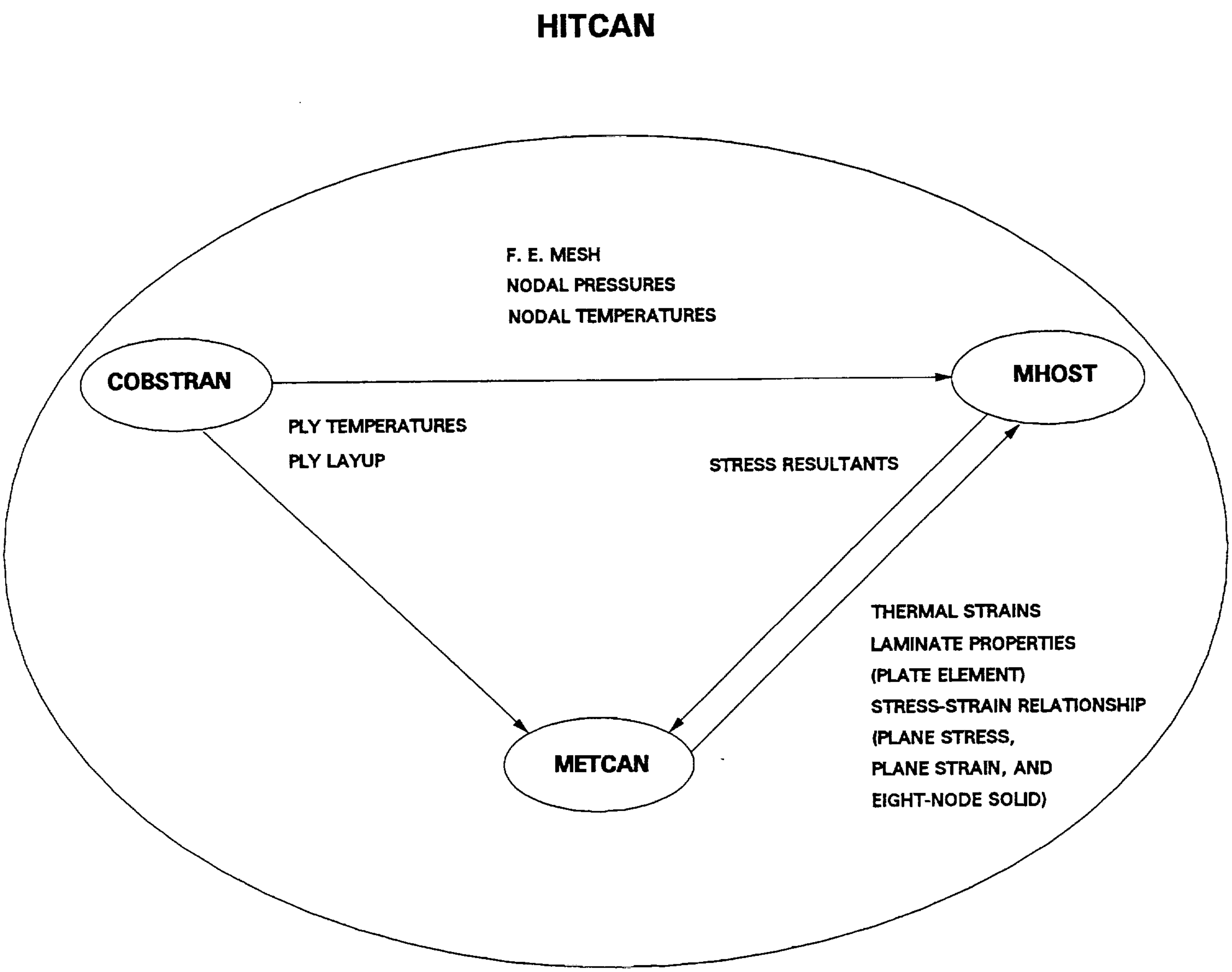
As mentioned previously, HITCAN is the synthesis of three in-house computer programs COBSTRAN, MHOST, and METCAN. Figure 2.1, illustrates what the task is of each program. As seen in the figure, COBSTRAN determines the finite element model, ply lay-up at each node, and the interpolated nodal loads. MHOST determines the structural response of the structure and METCAN calculates the laminate properties, ply stresses, and constituent stresses. Figure 2.2 shows the how the three programs fit together in HITCAN. As seen in this figure COBSTRAN needs to be called only once. MHOST and METCAN, which are part of an incremental nonlinear solution scheme, must be called repeatedly. In this scheme, METCAN and MHOST continually feed information back and forth, until convergence is reached.

To see the actual connections between COBSTRAN, MHOST, and METCAN, the architecture of HITCAN needs to be examined. Figure 2.3 shows the flow chart of the executive control module. This module initializes the program FORTRAN units, stores the user's input deck for later reading, and initializes the program options. The options selected are used to control program flow through the three second level executive subroutines HSOLID, HPLATE, and S3DSOL. These secondary executive subroutines initialize storage arrays, process the input files, generate the finite element model, and perform either an incremental static analysis or a dynamic analysis. The general flow diagram shown in Figure 2.4 corresponds to each one of the three second level executive subroutines. This can be seen in their flow charts (See Sections 2.1 to 2.3). Note that the module COBSTRAN is directly called from these subroutines. MHOST and METCAN are called indirectly from the second level subroutines via the subroutines NLINER and DYNMCH. Since the dynamic analysis is not complete, only the interaction between MHOST and METCAN in subroutine NLINER will be described. For a summary of subroutine NLINER, see Section 2.4.



HITCAN: AN INTEGRATED APPROACH FOR HIGH TEMPERATURE COMPOSITE STRUCTURAL ANALYSIS

FIGURE 2.1

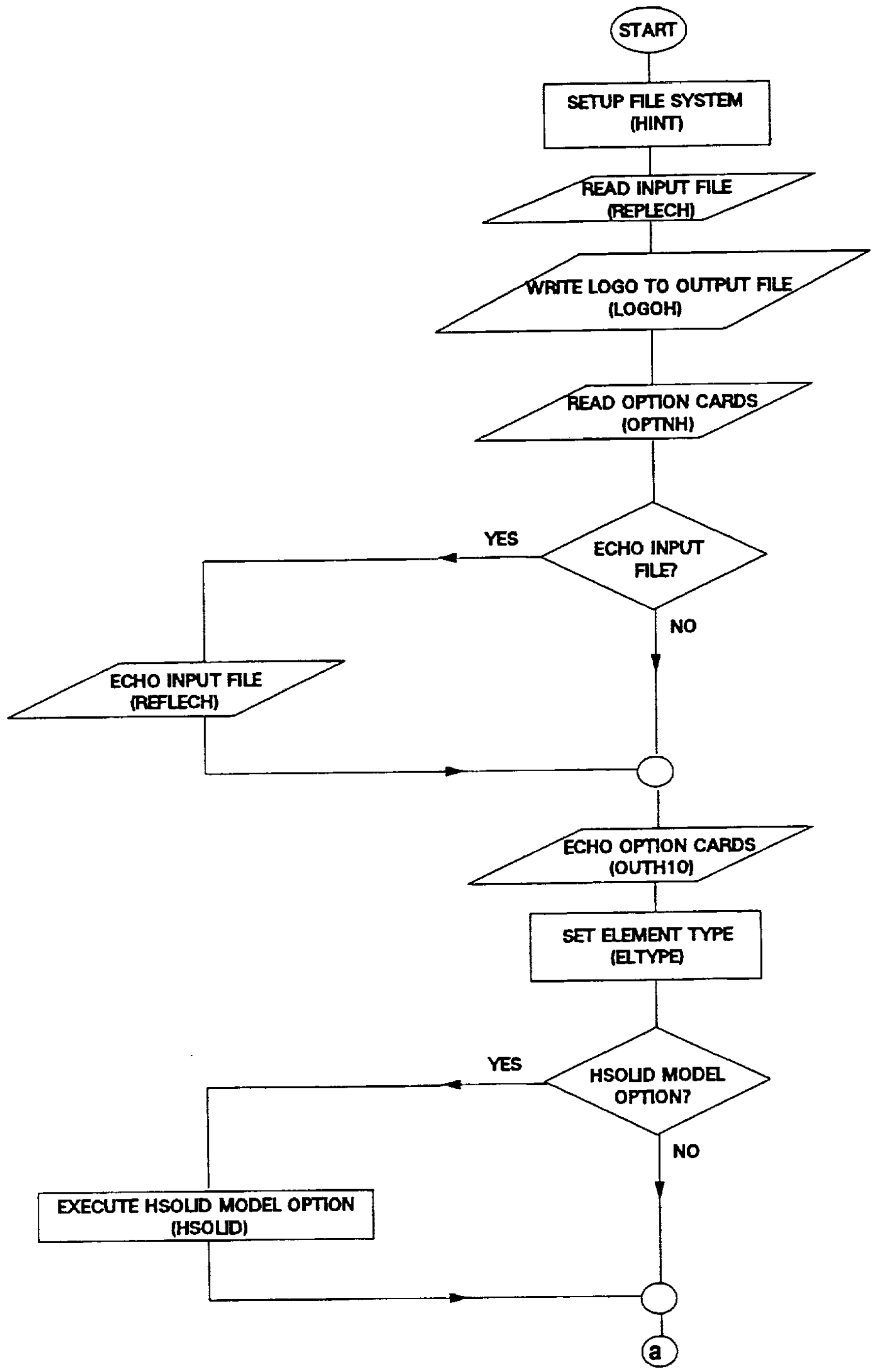


COBSTRAN, MHOST, AND METCAN SYNTHESIS

FIGURE 2.2

The secondary executive subroutine HSOLID controls the generation and the analysis of solid structures using plane stress, plane strain, or plate elements. This executive subroutine is described in Section 2.1. The function of the subroutine HPLATE is to control the generation and analysis of hollow structures using plate elements. A description of this subroutine is given in Section 2.2. The subroutine S3DSOL is used to control the generation and analysis of solid structures using eight-node solid elements. This subroutine is described in Section 2.3.

A dynamic core allocation strategy has been implemented in the HITCAN program. There are three storage arrays used in HITCAN. The array A is used for storage in METCAN, the array B is used in HITCAN, and the array IWORK is used as the storage array in MHOST. The variables MAXLEM, NASIZE, and ISIZE are the length of these arrays, respectively. These variables control the core memory requirements for the HITCAN program. The three arrays A, B, and IWORK and the variables MAXLEN, ISIZE, and NASIZE are set in the executive control module. If the number of words required to do an analysis exceeds the size of the array B, execution will stop and a message will be printed with the number of additional words needed. The number of words of allowed storage should be increased and the code re-compiled, so that the analysis can continue.

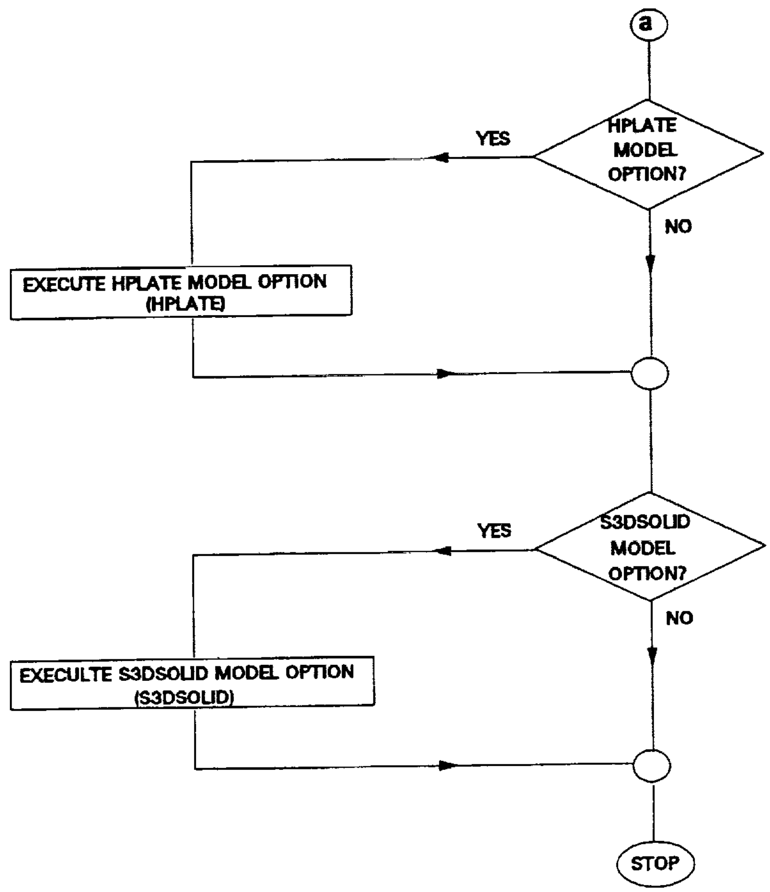


NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF EXECUTIVE MODEL

FIGURE 2.3

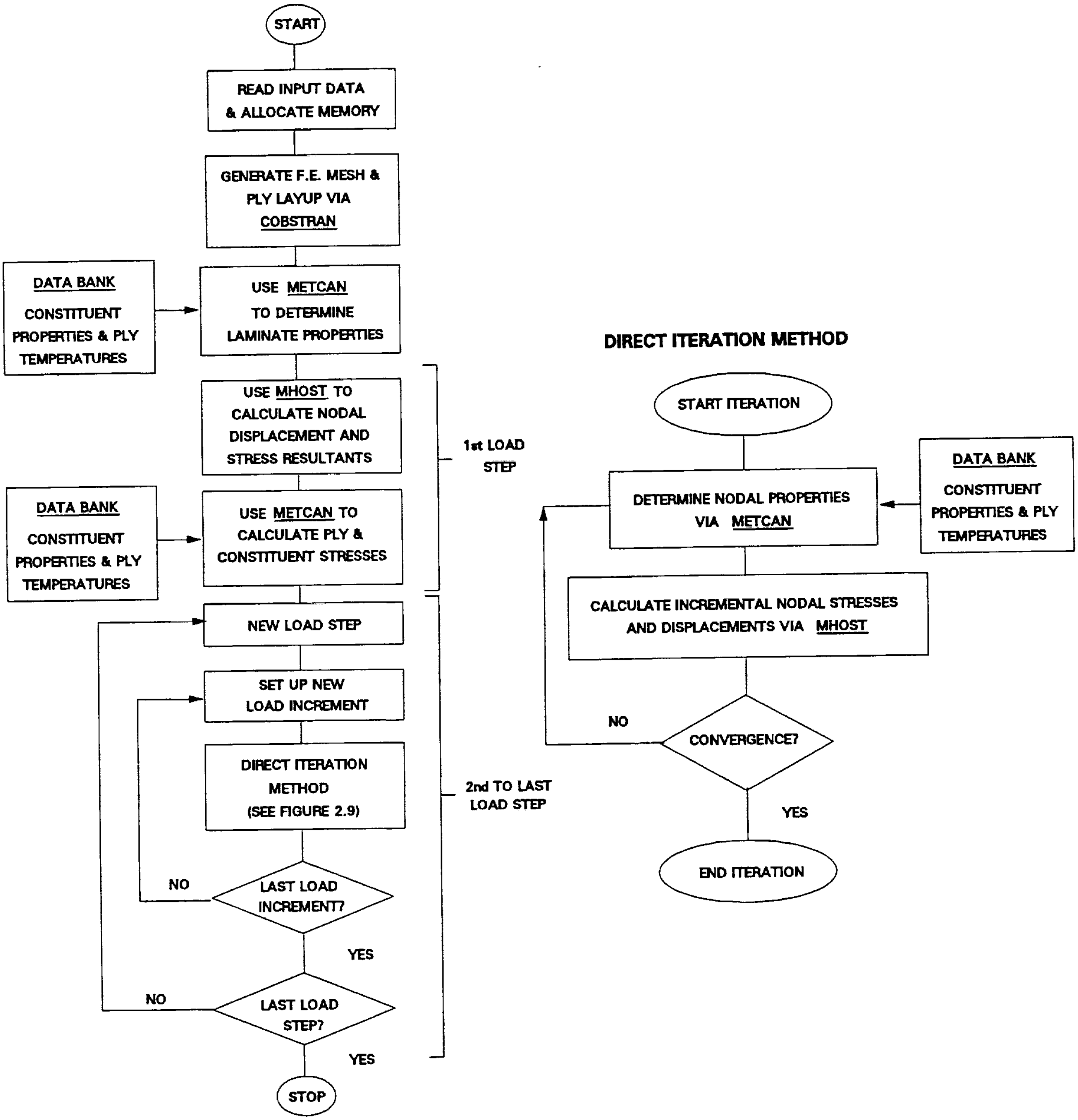
ARCH5



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF EXECUTIVE MODEL

FIGURE 2.3 (continued)



 GENERAL FLOW DIAGRAM OF THE HITCAN PROGRAM

FIGURE 2.4

ARCH7

2.1 SUBROUTINE HSOLID

This subroutine is a second level executive subroutine, which controls the generation and the analysis of solid structures using plane strain, plane stress, or plate elements. The general flow diagram of this subroutine is shown in Figure 2.3. This subroutine performs the following functions:

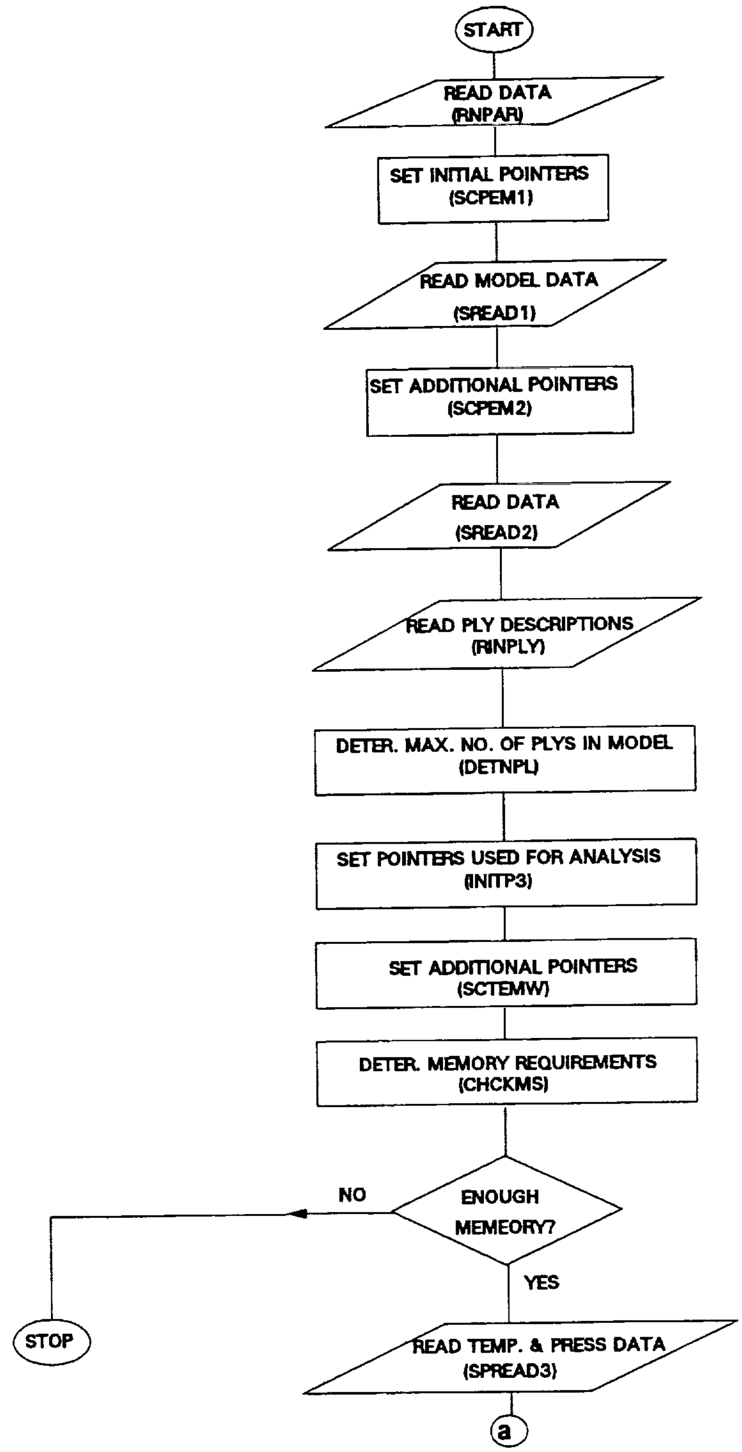
1. Initialization of the storage array B. This is rendered through the subroutines SCPEM1, SCPEM2, INITP3, and SCTEMW.

2. Processing of user input file. This function is done through the subroutines RINPAR, SREAD1, SREAD2, RINPLY, SREAD3, RINFOR, DYNMIN, RIBOUN, and OUTPTC.

3. Generation of the finite element model. This job is carried out in the subroutine PREPSO. This subroutine is a modified version of COBSTRAN, see Reference 4.

4. Incremental static or dynamic analysis. These two functions are performed in the subroutines NLINER and DYNMCH. A description of the subroutine NLINER can be found in Section 2.4.

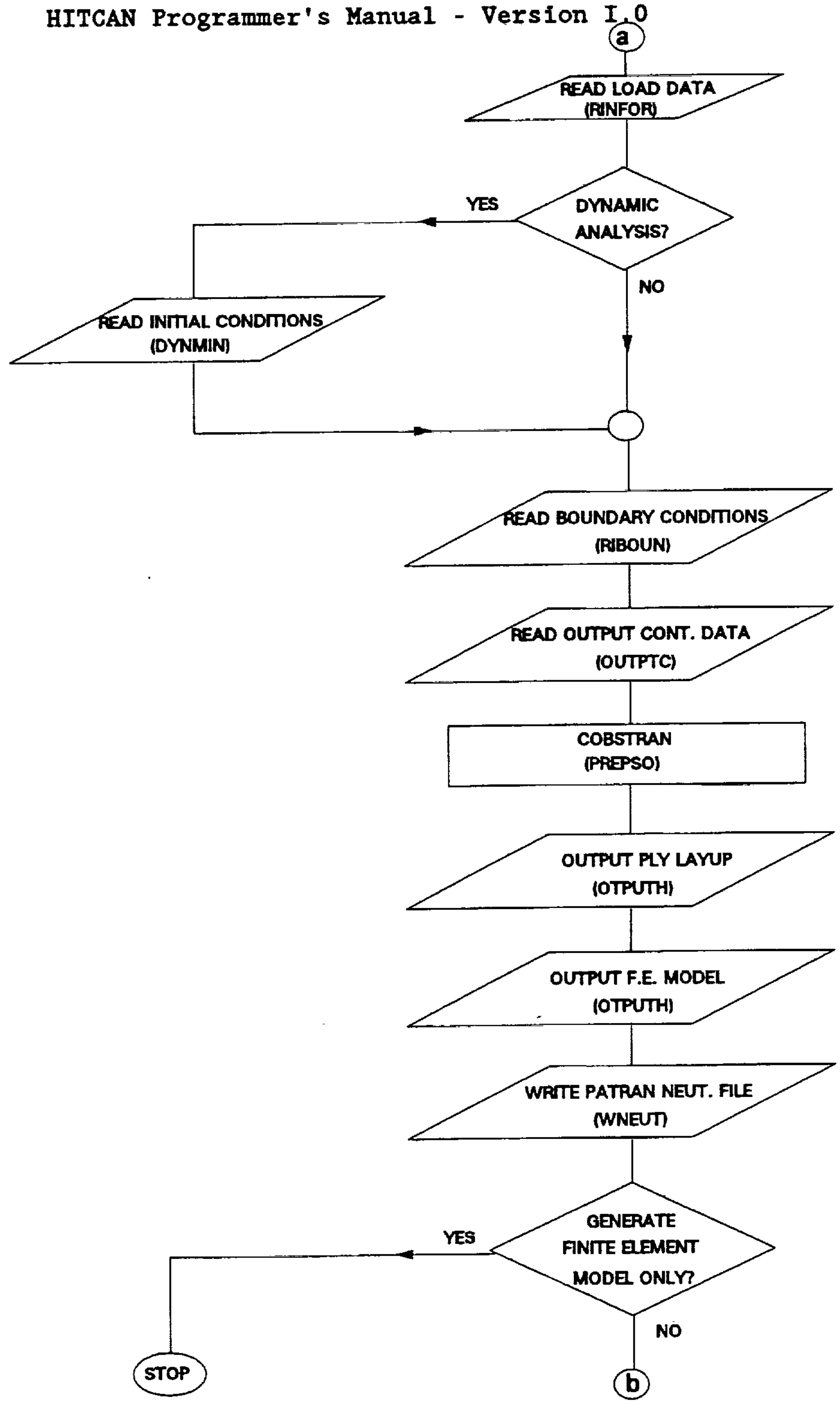
A detailed flow chart of this subroutine is shown in Figure 2.5.



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE HSOLID

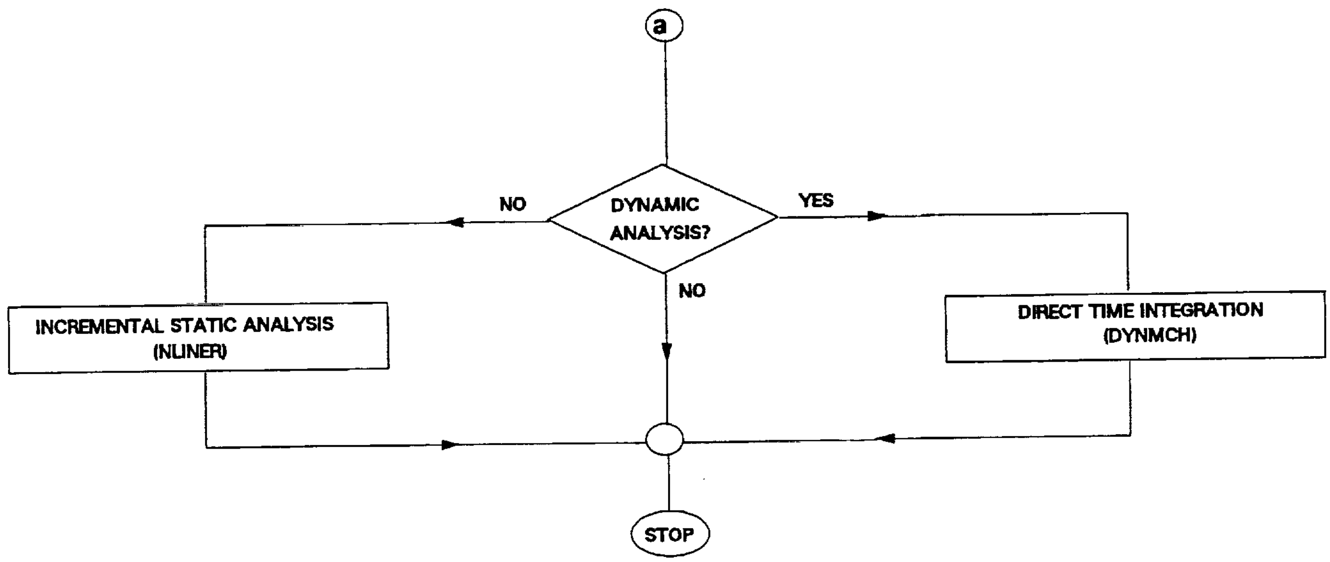
FIGURE 2.5



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE HSOLID

FIGURE 2.5 (continued)



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE HSOLID

FIGURE 2.5 (continued)

2.2 SUBROUTINE HPLATE

This subroutine is a second level executive program, which controls the generation and the analysis of hollow structures using a plate element. This subroutine follows the general flow diagram shown in Figure 2.3. The functions performed in this subroutine are:

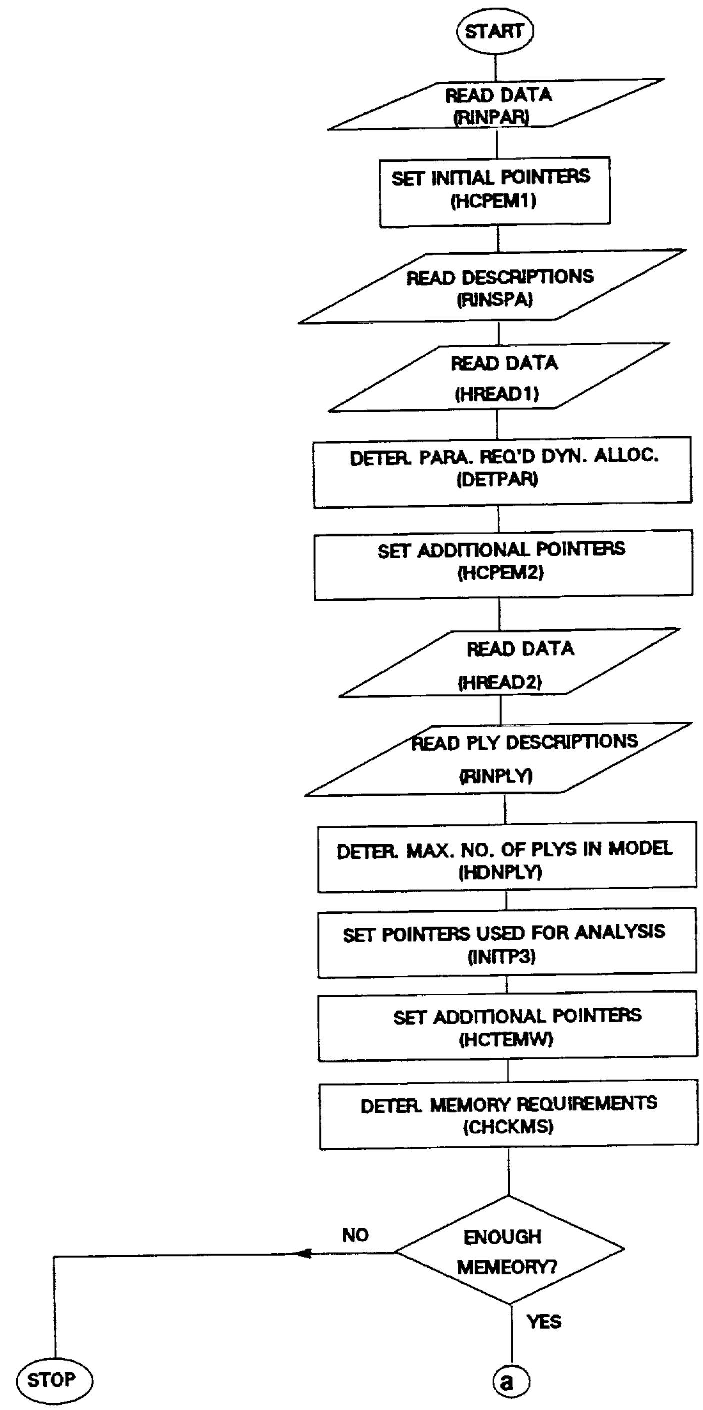
1. Initialization of the storage array B. This is rendered through the subroutines HCPEM1, HCPEM2, INITP3, and HCTEMW.

2. Processing of user input file. This function is done through the subroutines RINPAR, RINSPA, HREAD1, HREAD2, RINPLY, HREAD3, RINFOR, DYNMIN, RIBOUN, and OUTPTC.

3. Generation of the finite element model. This job is carried out in the subroutines PREPSH and PREPSP. These subroutines taken together are a modified version of COBSTRAN, see Reference 4.

4. Incremental static or dynamic analysis. These two functions are performed in the subroutines NLINER and DYNMCH. A description of the subroutine NLINER can be found in Section 2.4.

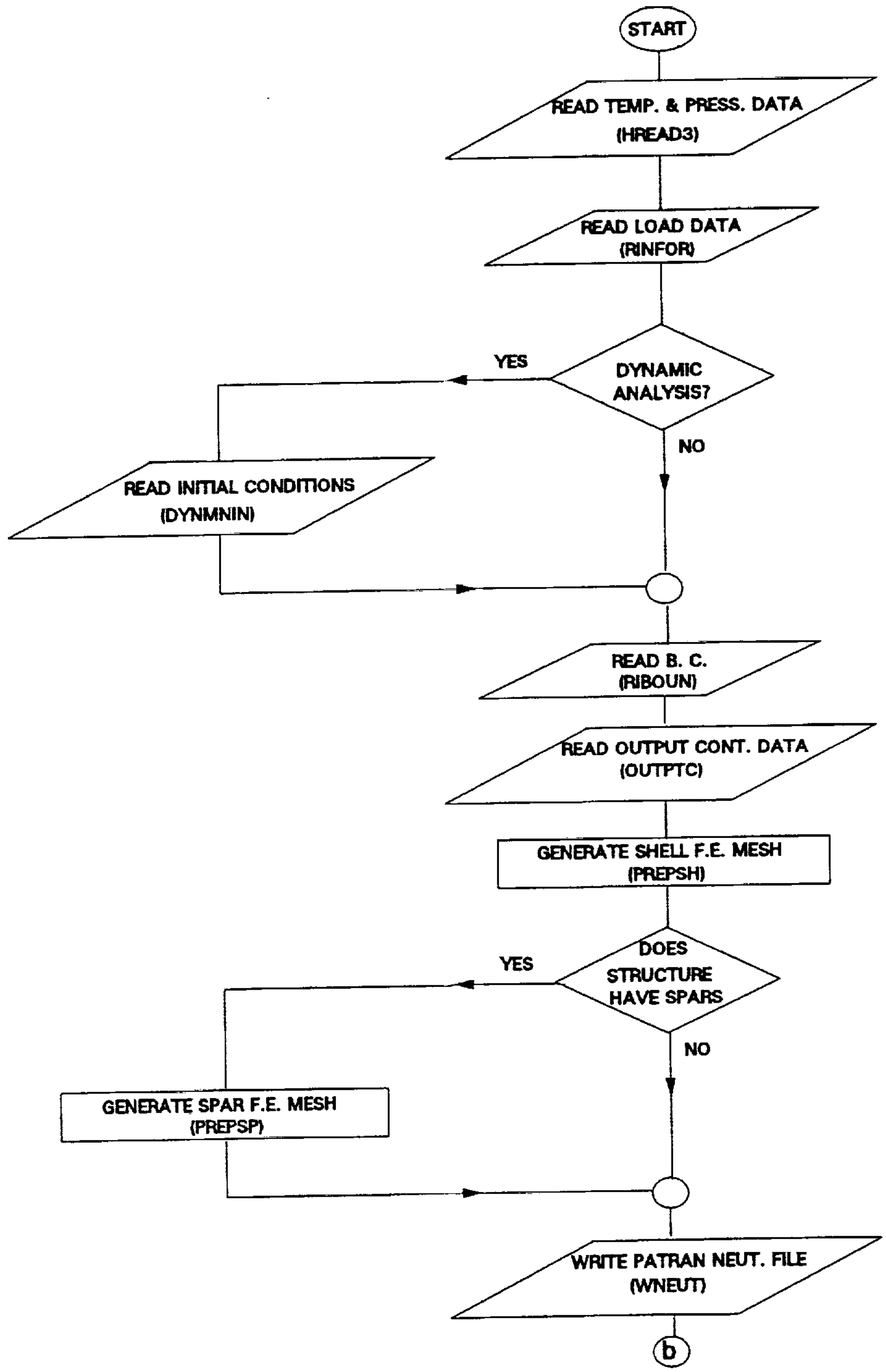
The flow chart of this subroutine is shown in Figure 2.6.



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE HPLATE

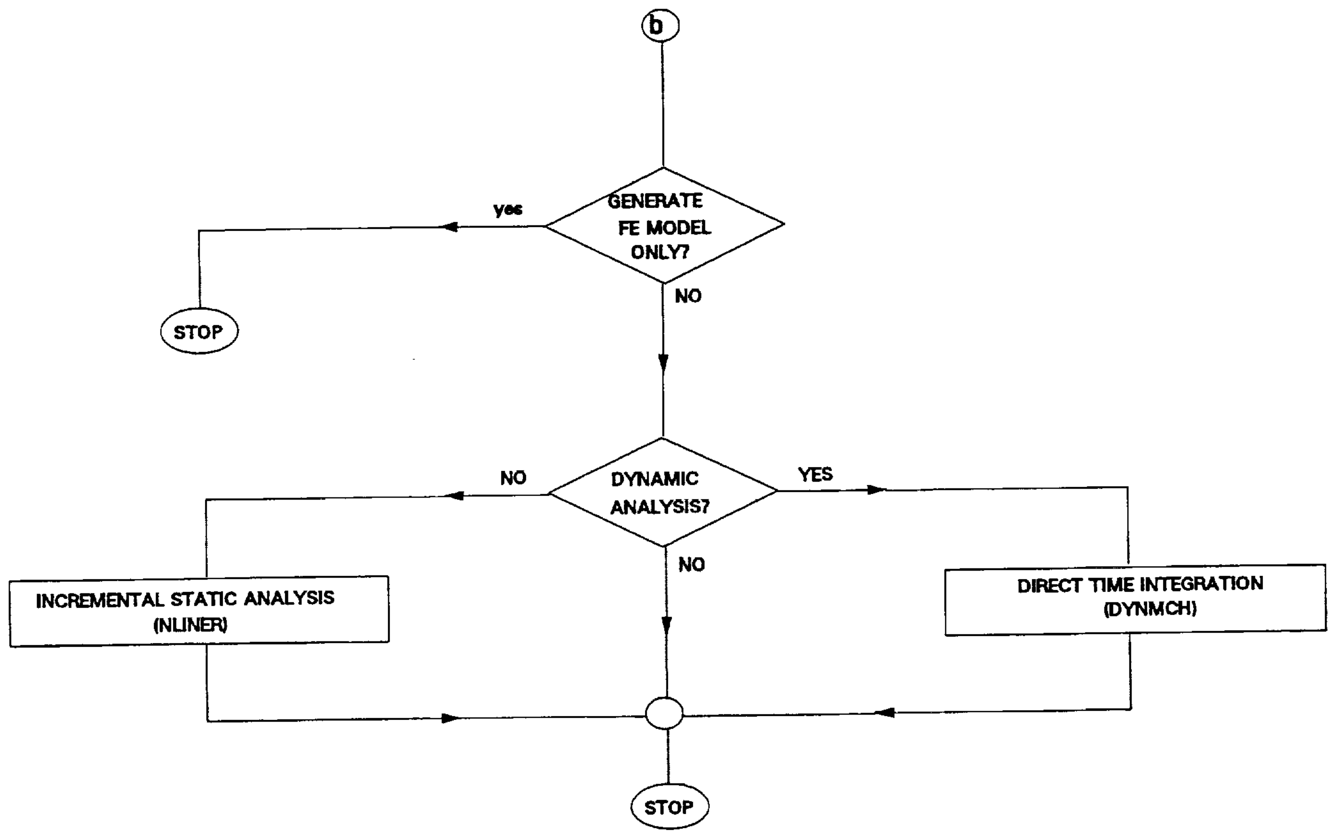
FIGURE 2.6



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE HPLATE

FIGURE 2.6 (continued)



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE HPLATE

FIGURE 2.6 (continued)

2.3 SUBROUTINE S3DSOL

This subroutine is a second level executive program, which controls the generation and the analysis of solid structures using the eight-node solid element. This subroutine follows the general flow diagram shown in Figure 2.3. The functions performed in this subroutine are:

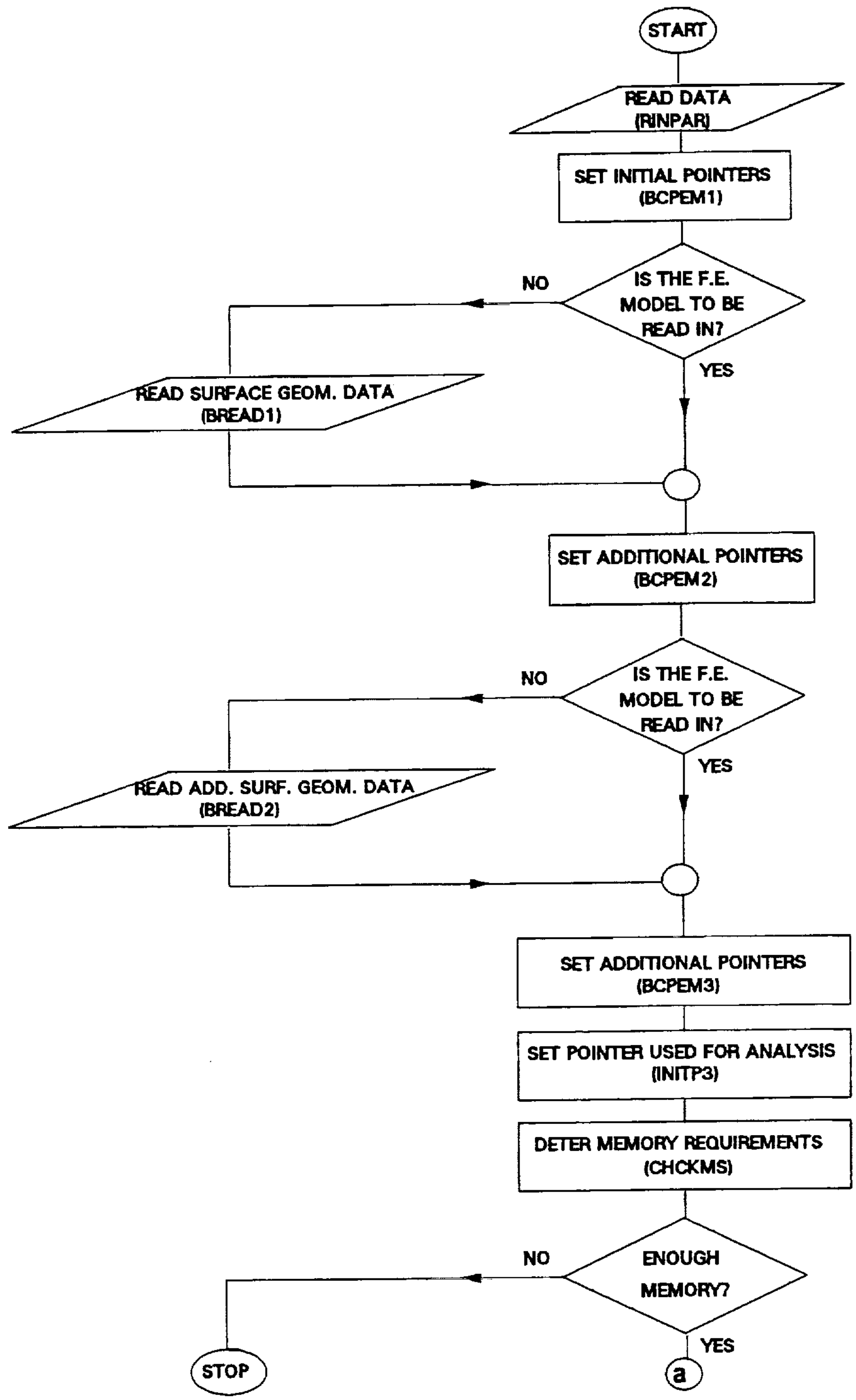
1. Initialization of the storage array B. This is rendered through the subroutines BCPEM1, BCPEM2, BCPEM3, and INITP3.

2. Processing of user input file. This function is done through the subroutines RINPAR, BREAD1, BREAD2, READML, BREAD3, RINPLY, RINLAY, BREAD4, RINFOR, DYNMIN, RIBOUN, and OUTPTC.

3. Generation of the finite element model. This job is carried out in the subroutines PCOBST, COBSTRAN, NASINT, and METINT. The subroutine PCOBST creates the input file to COBSTRAN, see Reference 4. The subroutines NASINT and METINT interpret the files from COBSTRAN.

4. Incremental static or dynamic analysis. These two functions are performed in the subroutines NLINER and DYNMCH. A description of the subroutine NLINER can be found in Section 2.4.

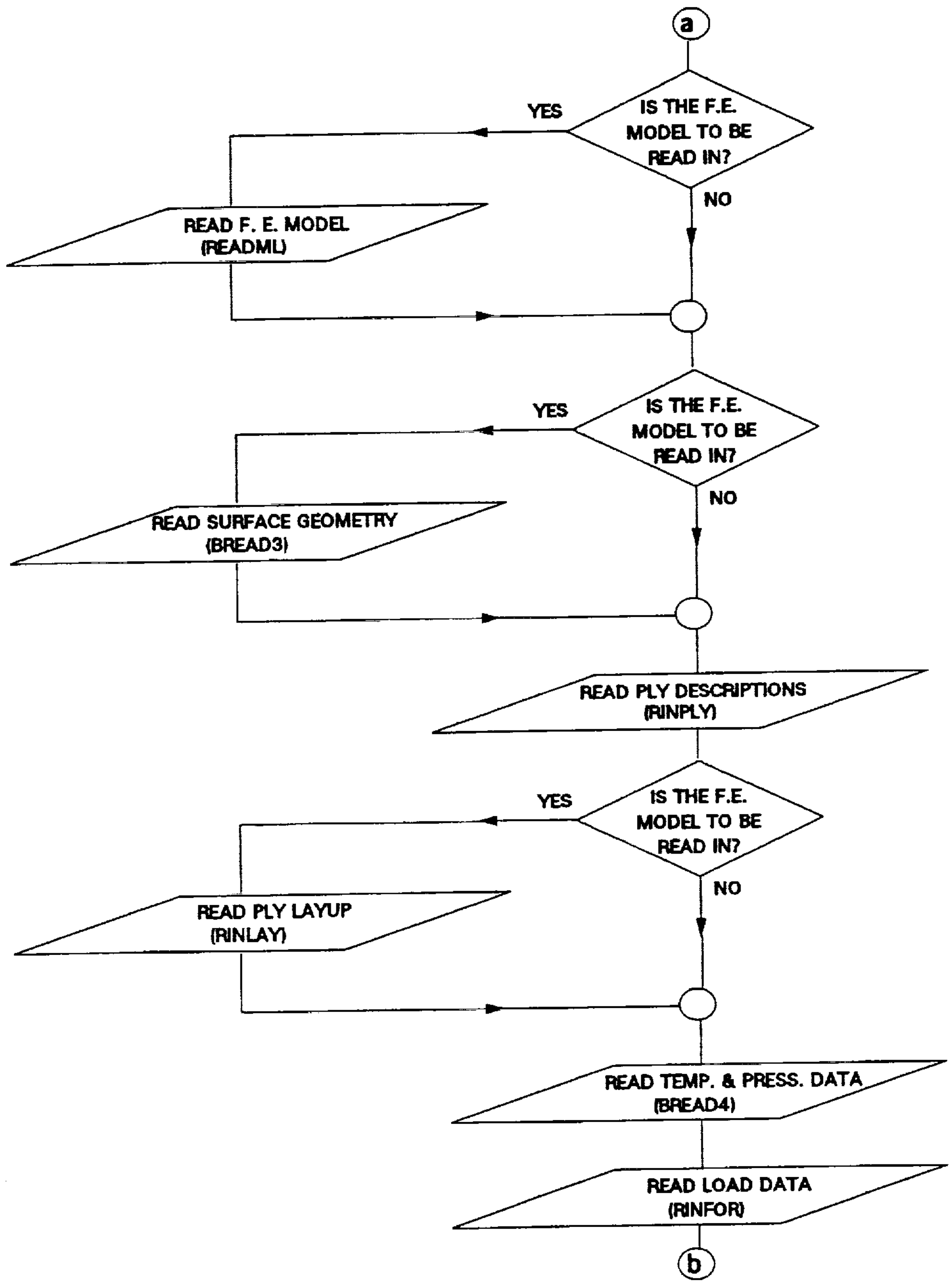
The flow chart of this subroutine is shown in Figure 2.7.



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE S3DSOL

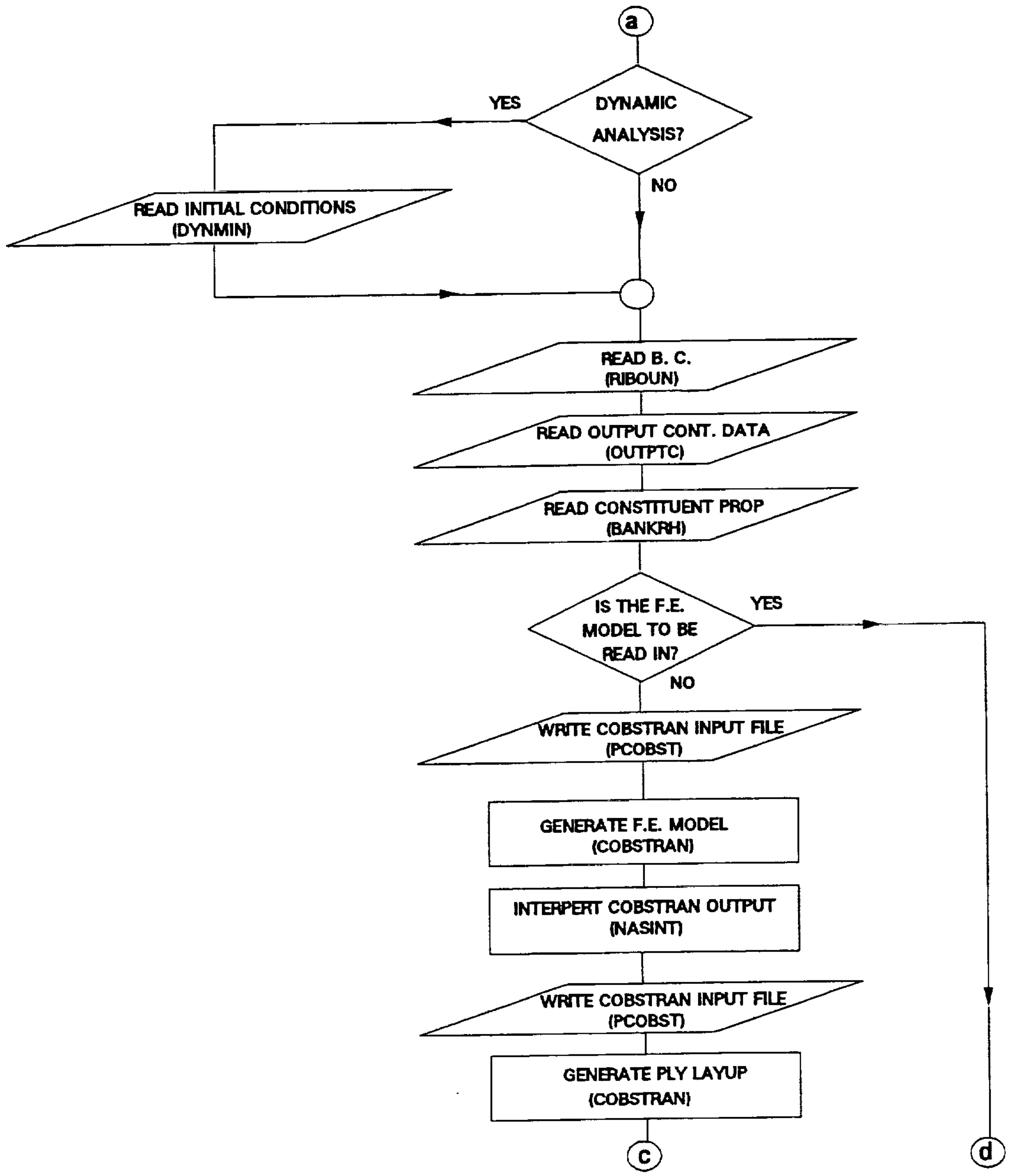
FIGURE 2.7



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE S3DSOL

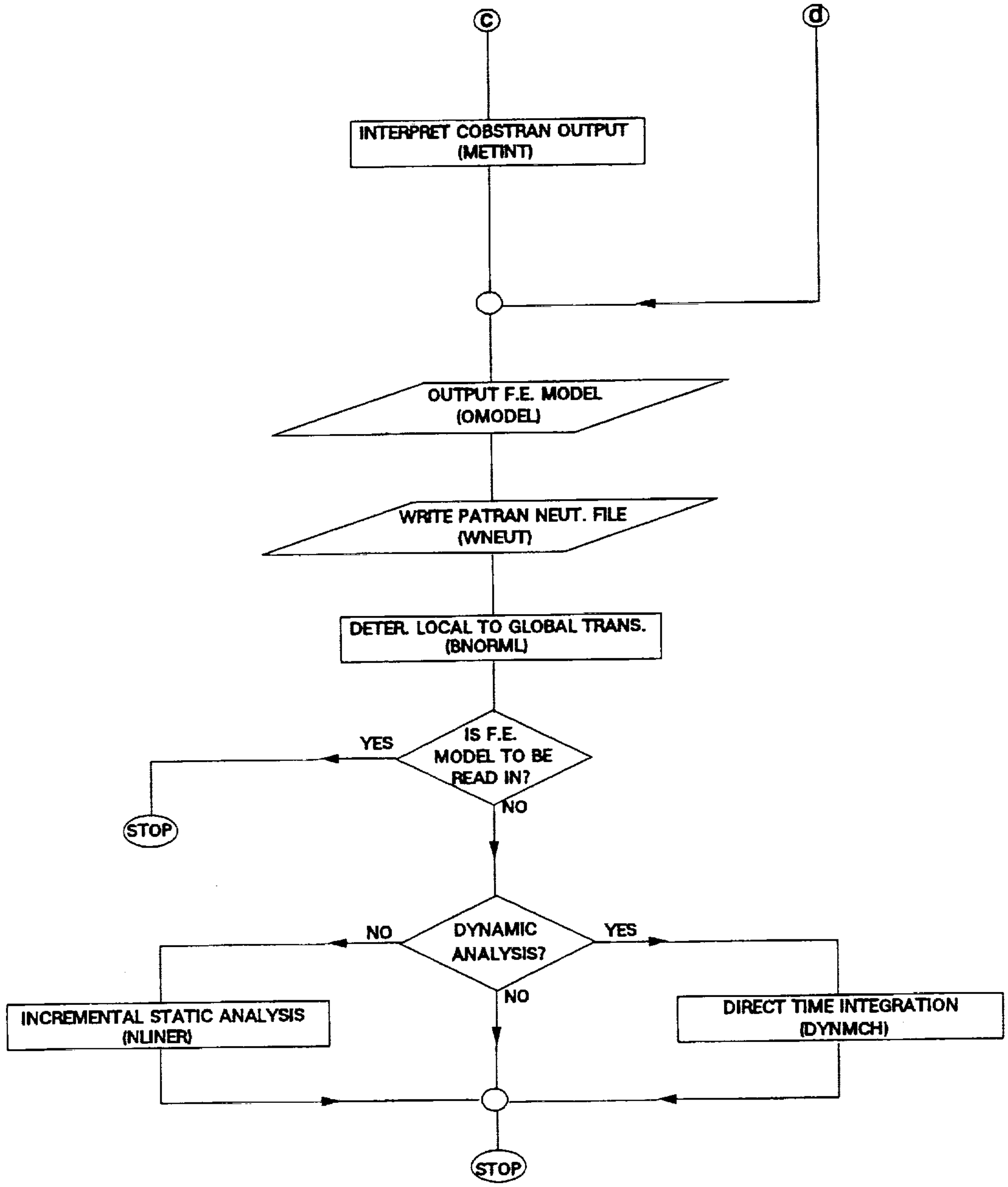
FIGURE 2.7 (continued)



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE S3DSOL

FIGURE 2.7 (continued)



NOTE: ( ) Denotes a Subroutine Name.

FLOW CHART OF SUBROUTINE S3DSOL

FIGURE 2.7 (continued)

2.4 SUBROUTINE NLINER

This subroutine controls the sequence of execution for the incremental static analysis. The scheme used in HITCAN is the incremental direct iteration method. The library modules MHOST and METCAN play an integral part in this method. Before examining the roles of MHOST and METCAN in the direct iteration method, the scheme itself needs to be described.

The solution scheme in a general form, assuming a single degree-of-freedom system, works as follows.

- Initially, the material properties are calculated based on the nodal temperatures and the nodal stresses obtained from the previous load increment.
- The incremental displacements are calculated from the equation

$$\Delta D^1 = [K^0]^{-1} \Delta F,$$

  where $[K^0]$ is the initial stiffness matrix and ΔF is the incremental load.
- Repetition of this process can be written as

$$\Delta D^n = [K^{n-1}]^{-1} \Delta F.$$

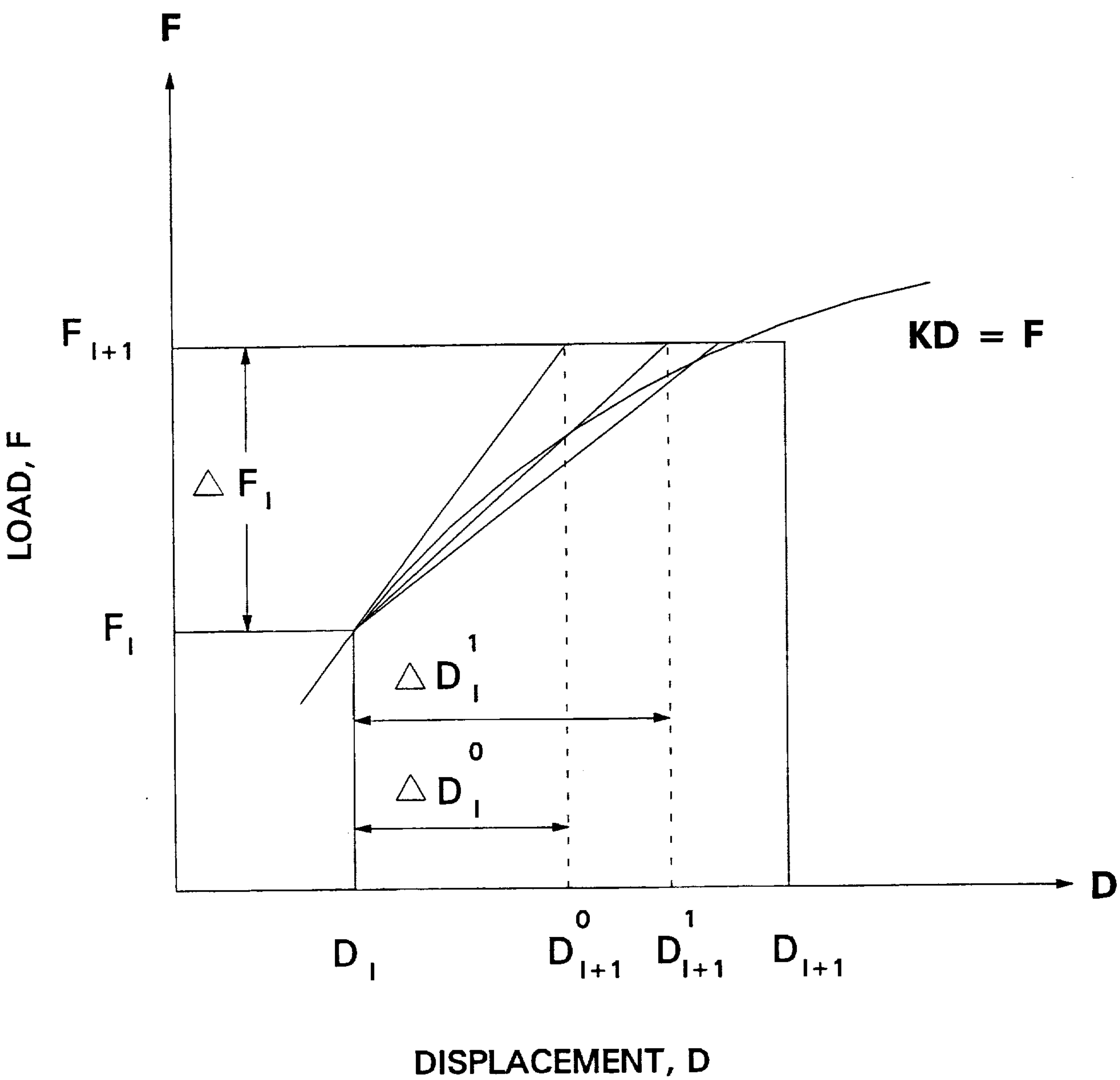
  Here the stiffness matrix $[K^{n-1}]$ is calculated based on the material properties of the previous iteration. The material properties are functions of the nodal temperatures and stresses.
- This process is terminated when the error becomes sufficiently small, i. e. ,

$$||\Delta D^n - \Delta D^{n-1}|| / ||\Delta D^{n-1} - \Delta D^{n-2}|| \leq \text{tolerance}.$$

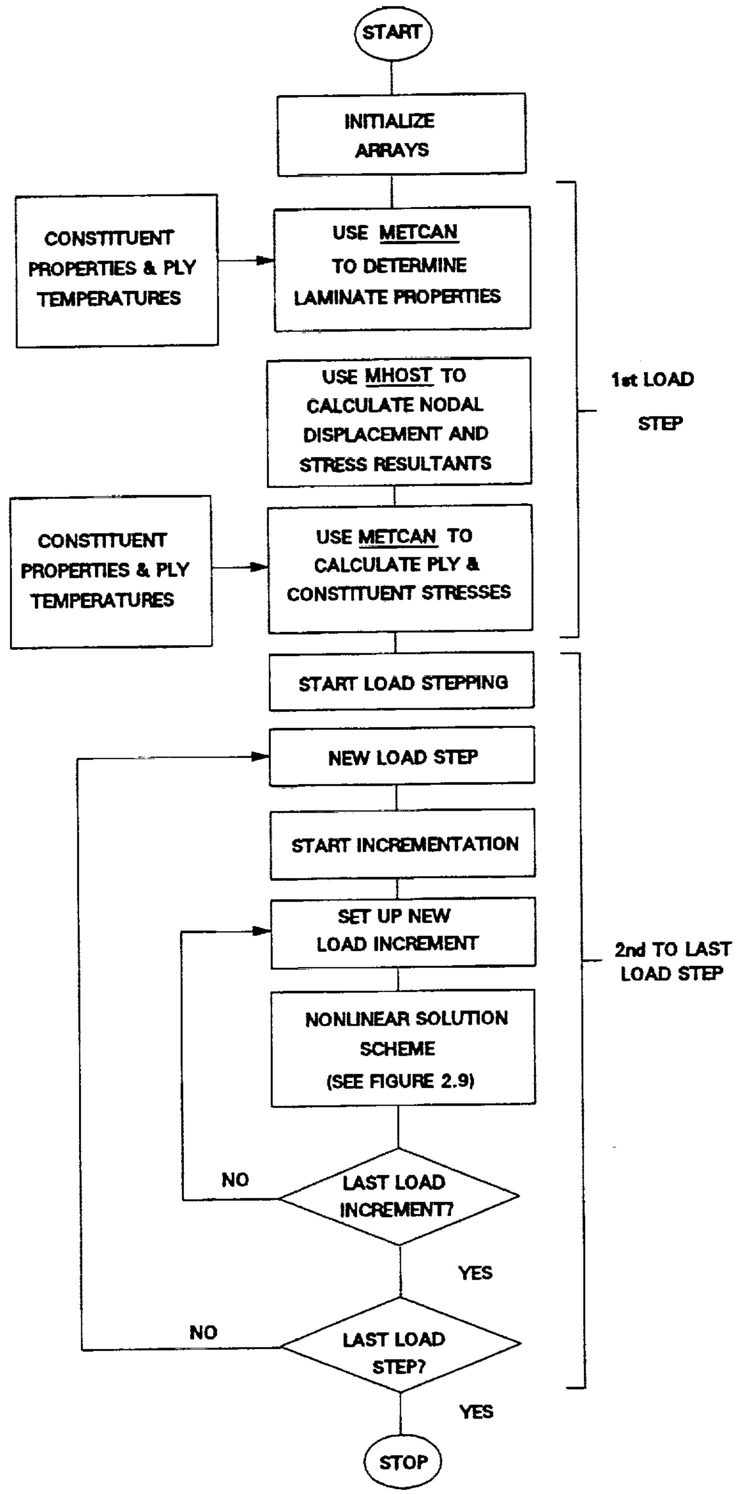
This process is shown graphically in Figure 2.8.

A general flow diagram of the subroutine NLINER is shown in Figure 2.9. Figure 2.10 shows how MHOST and METCAN are incorporated into the solution scheme. At the beginning of each iteration METCAN, block 1, is called. Using the constituent properties, the ply stresses, the constituent stresses, and the new temperature, METCAN determines the new laminate properties for each node. This means METCAN must be called for each node. MHOST (block 2) using the laminate properties from METCAN and the nodal loads, determines the incremental nodal displacements and the incremental nodal stress resultants. The new incremental displacements are then compared with the previous incremental displacements to check for convergence. If the solution fails to converge, METCAN (block 1) determines new laminate properties. However, if convergence is achieved, the nodal displacements are added to the incremental displacements (block 3) and METCAN is then called again to determine the new incremental ply stresses and constituent stresses (block 4). These stresses are then added to the cumulative stresses in block 5. It should be pointed out that METCAN is called for each node so that the effects at the micromechanics level can be captured.

A detailed flow chart of this subroutine is shown in Figure 2.11.

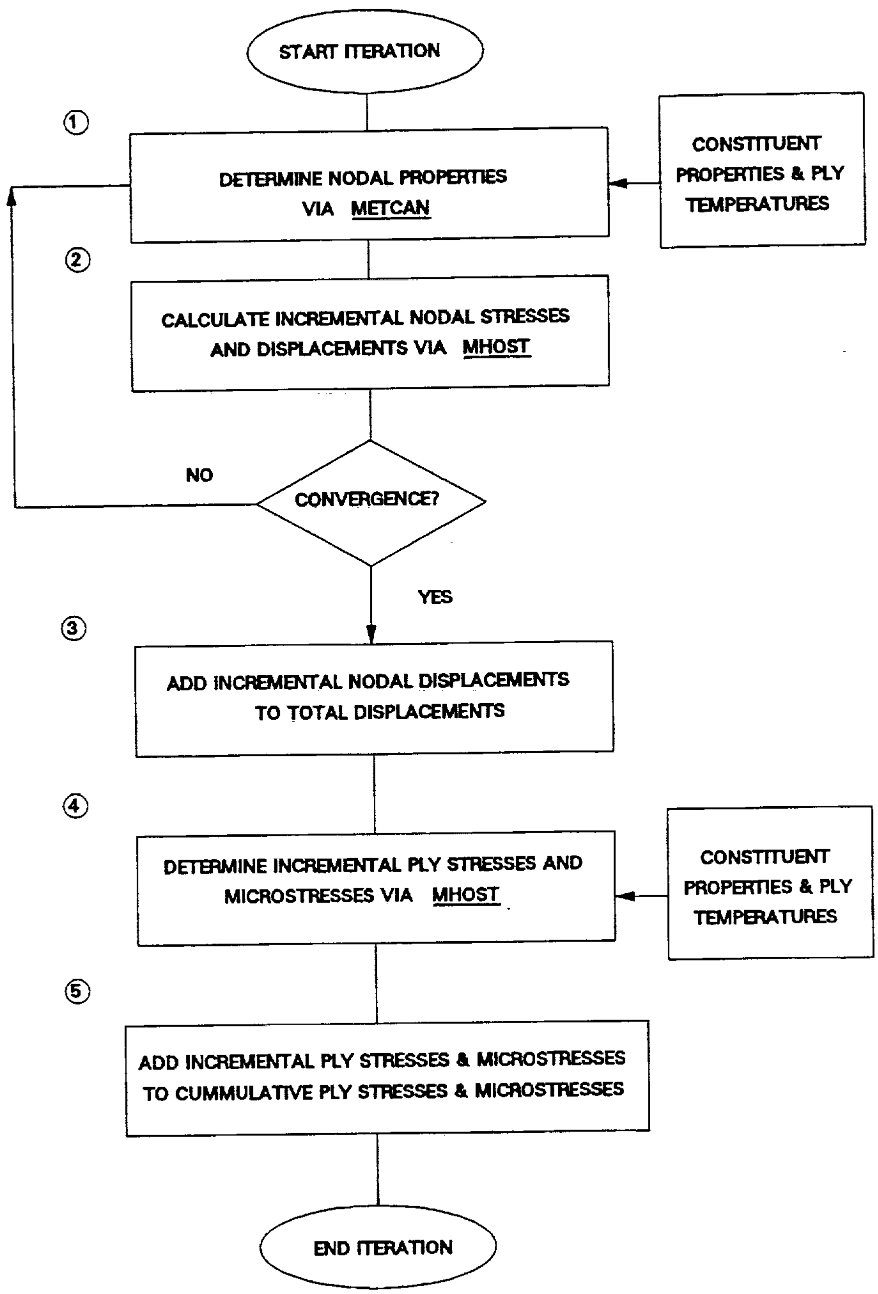


DIRECT ITERATION METHOD
FIGURE 2.8

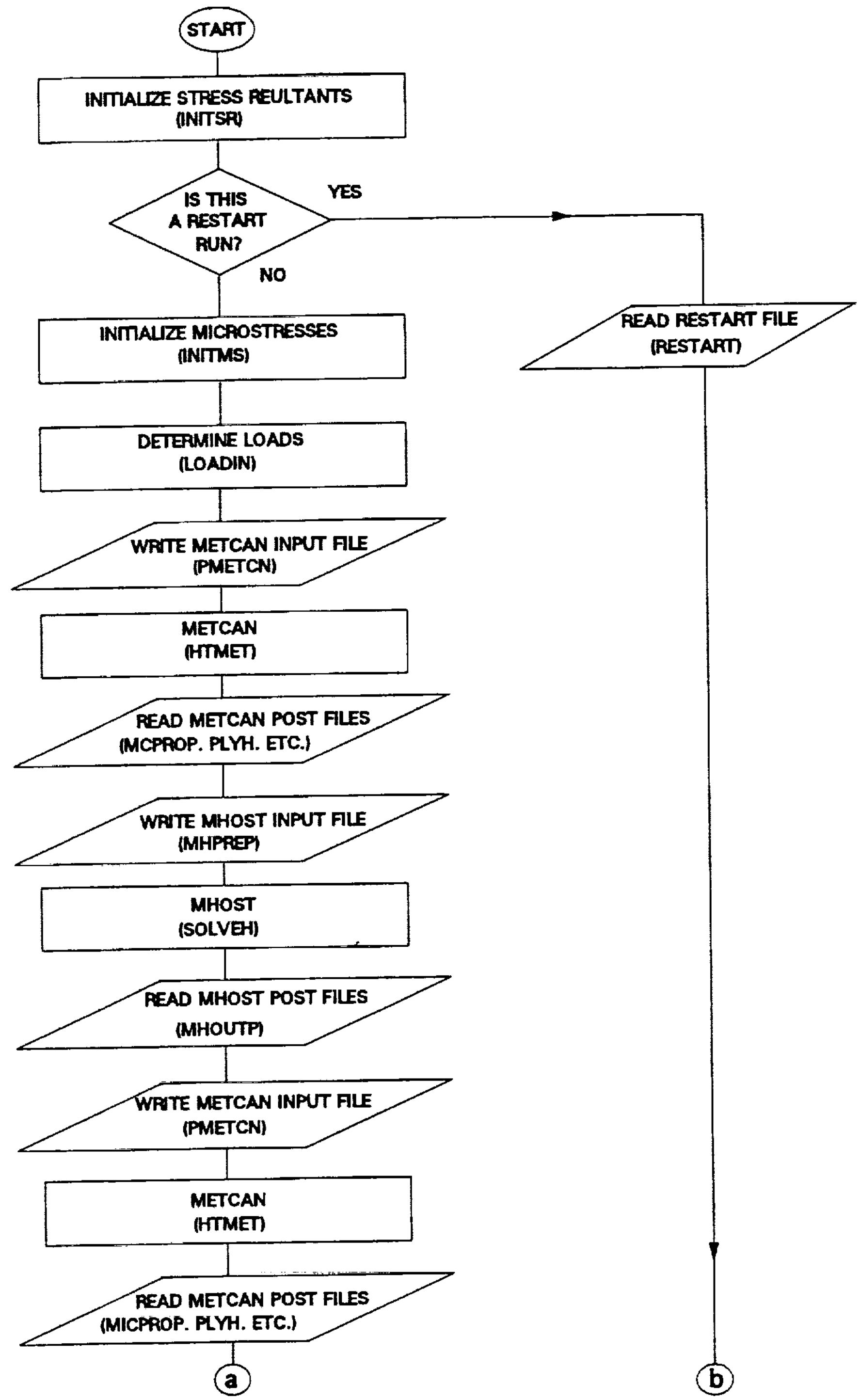


GENERAL FLOW DIAGRAM OF SUBROUTINE NLINER

FIGURE 2.9



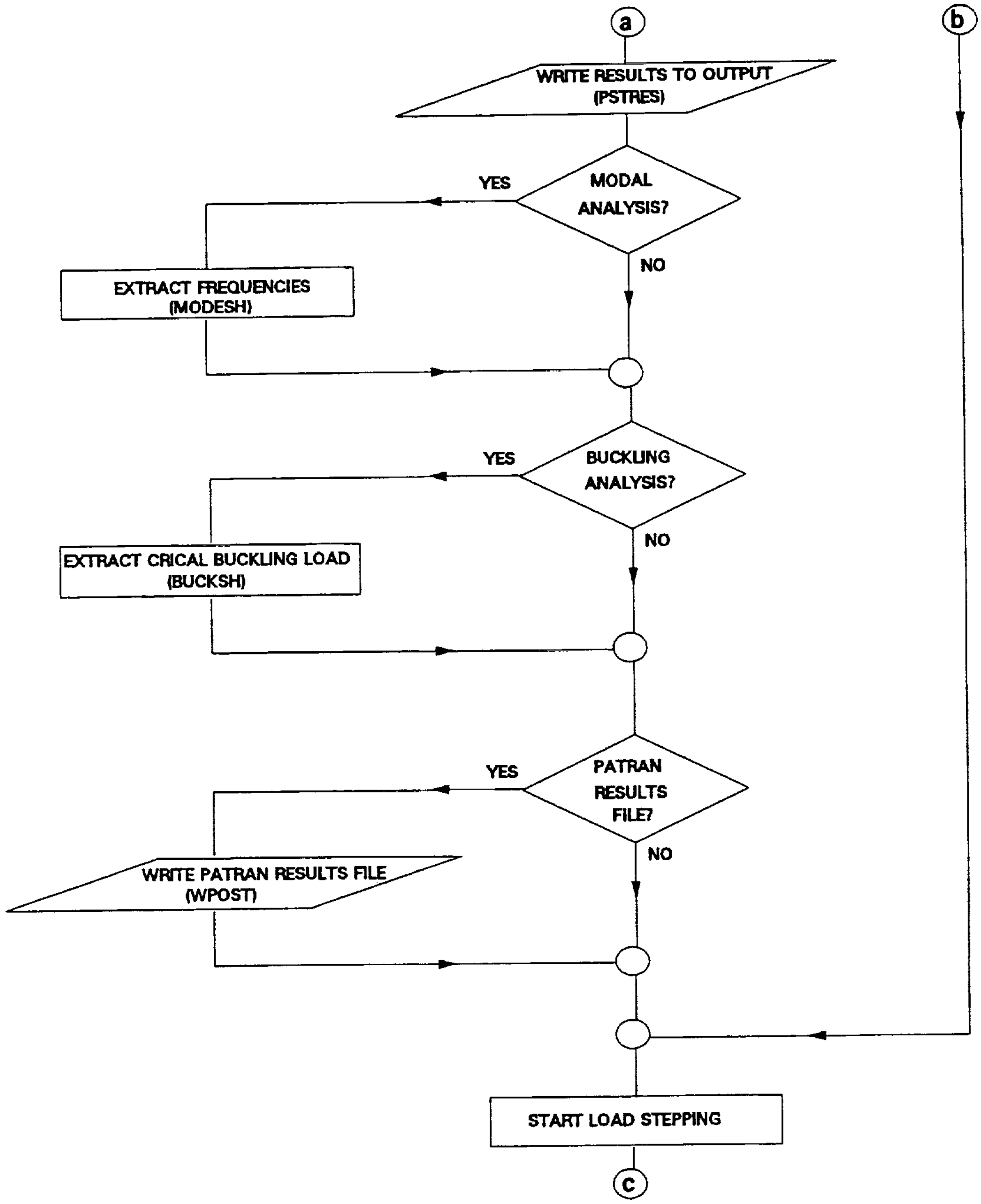
FLOW DIAGRAM OF ITERATIVE SOLUTION PROCEDURE
FIGURE 2.10



NOTE: ( ) Denotes a Subroutine Name.

SUBROUTINE NLINER

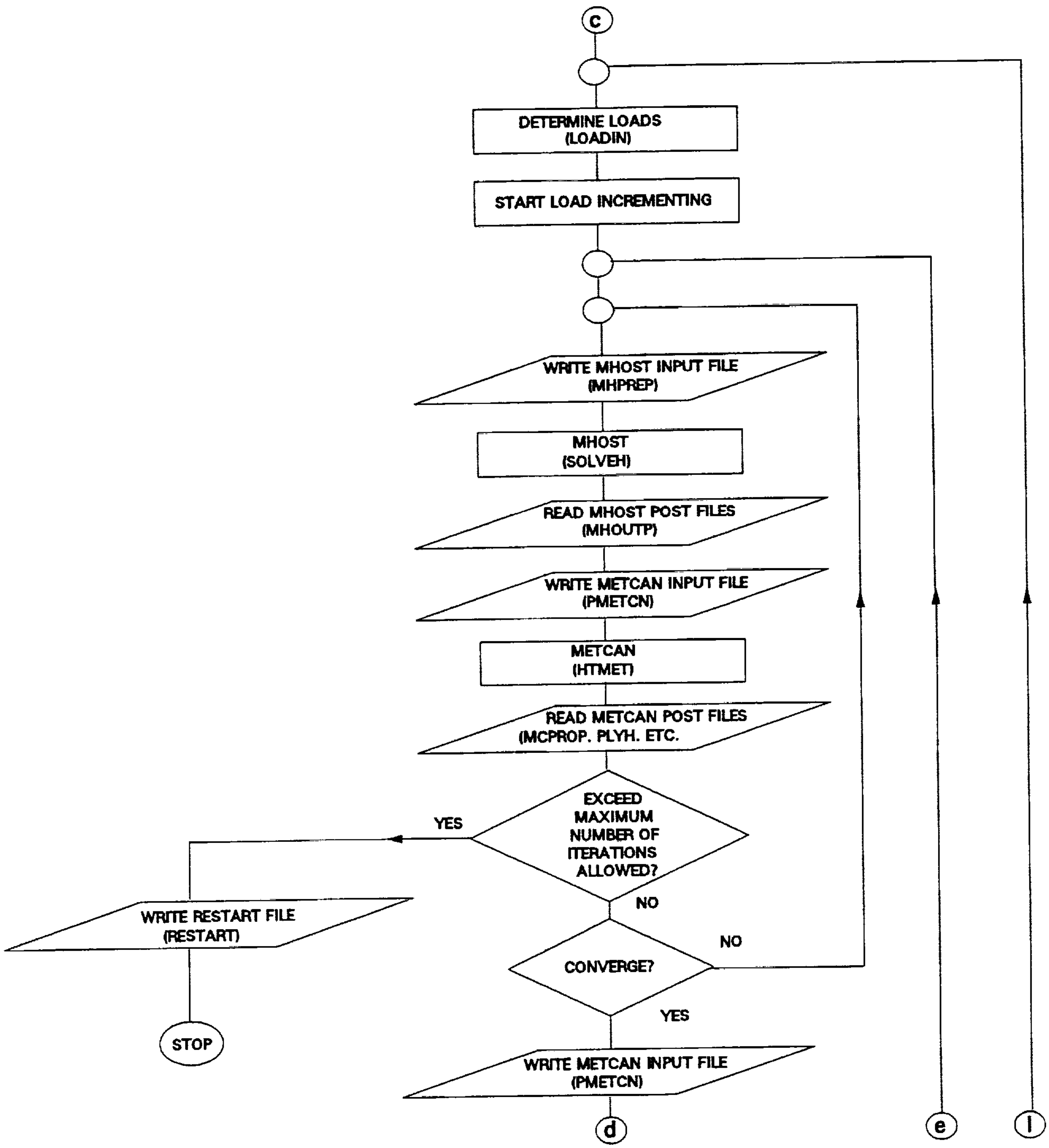
FIGURE 2.11



NOTE: ( ) Denotes a Subroutine Name.

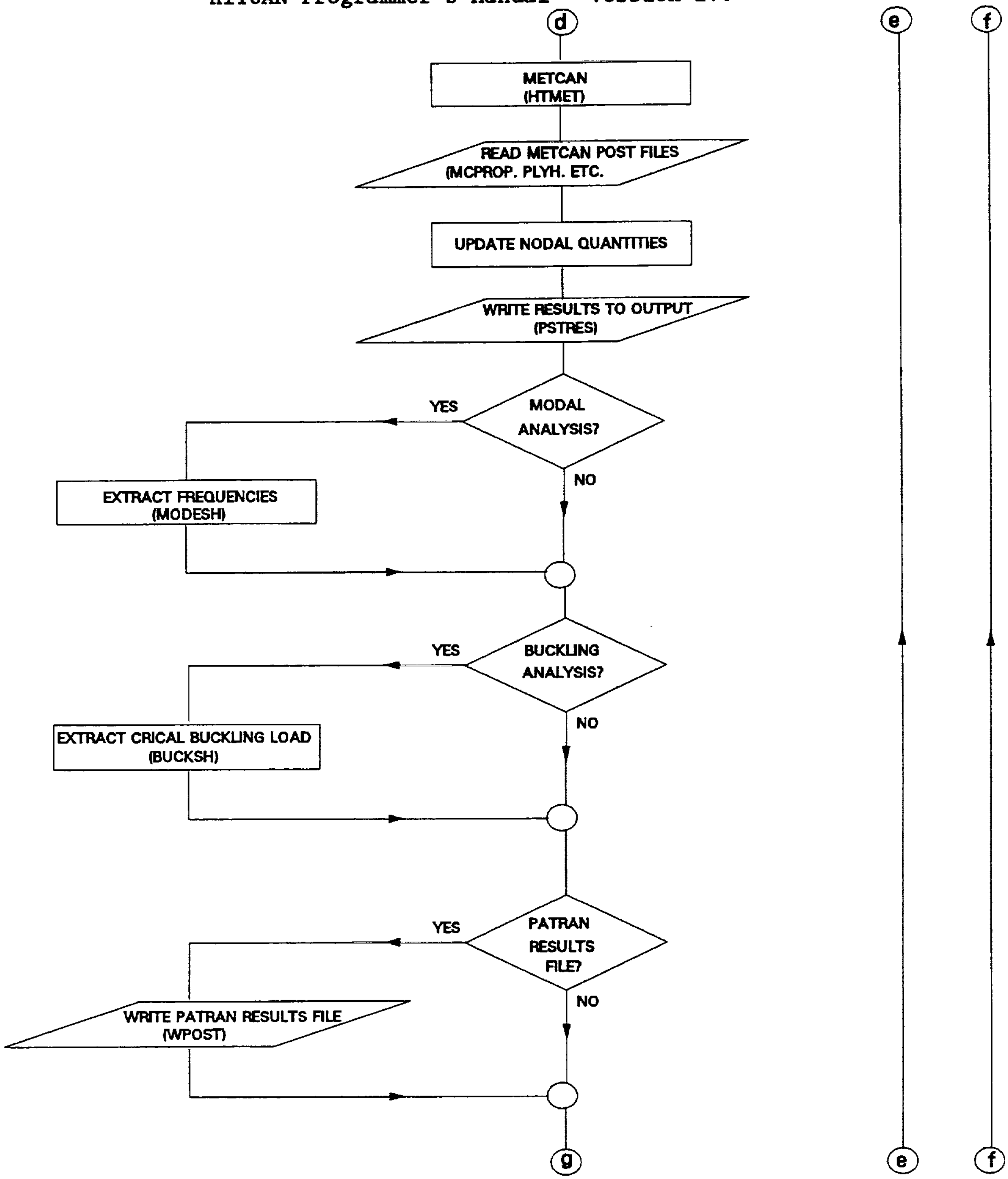
SUBROUTINE NLINER

FIGURE 2.11 (continued)



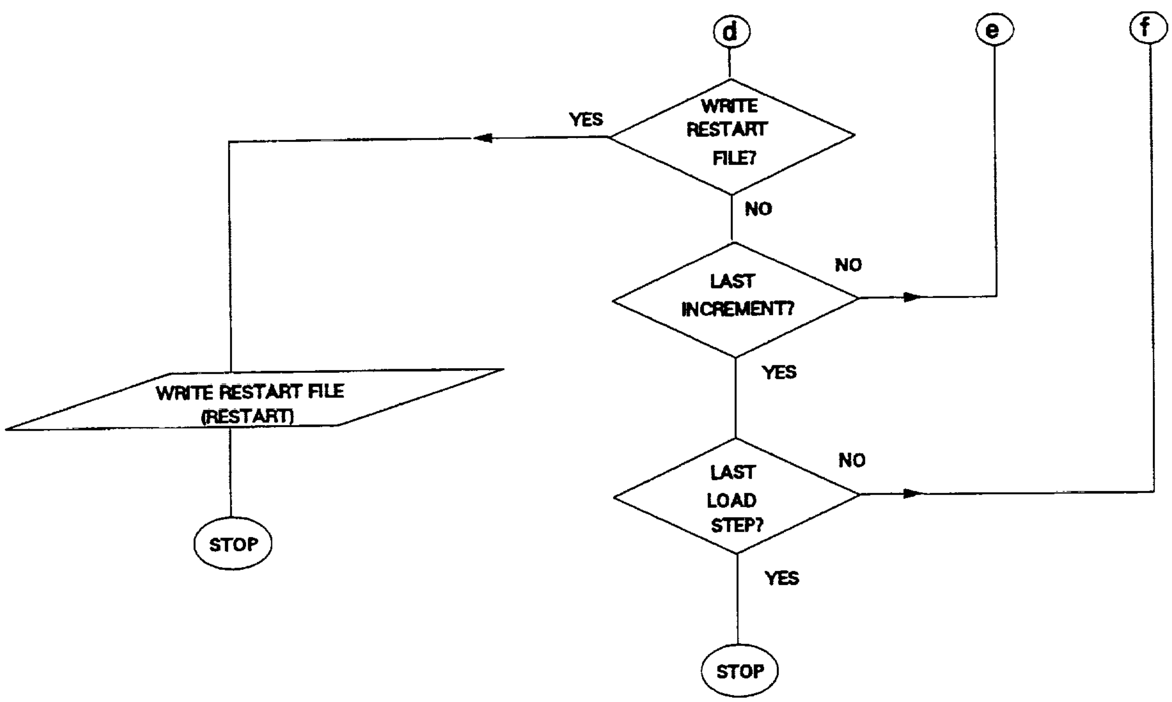
NOTE: ( ) Denotes a Subroutine Name.

SUBROUTINE NLINER
FIGURE 2.12 (continued)



NOTE: ( ) Denotes a Subroutine Name.

SUBROUTINE NLINER
FIGURE 2.11 (continued)



NOTE: ( ) Denotes a Subroutine Name.

SUBROUTINE NLINER

FIGURE 2.11 (continued)

CHAPTER 3

SUBROUTINE LISTING AND CALLING TREE

The HITCAN calling tree and brief descriptions of the subroutines included in the HITCAN code are given in this chapter. Figure 3.1 shows the HITCAN calling tree.

A brief description of each routine is given below. The subroutine names are sorted in alphabetical order. Note that almost all subroutines written for the HITCAN computer code are self-documented and further information can be obtained directly from the source listing.

<u>BANKRH</u>

Reads in the fiber, matrix, and interface properties. Called from the subroutines PREPSH, PREPSO, and S3DSOL.

<u>BCPEM1</u>

Sets the pointers NB1, NB2, and NB3. Called from the secondary executive subroutine S3DSOL.

<u>BCEPM2</u>

Sets the pointer NB4. Called from the secondary executive subroutine S3DSOL.

<u>BCPEM3</u>

Sets the pointers NB5, NB6, ..., NB39, NB40. Called from the secondary executive subroutine S3DSOL.

<u>BNORML</u>

Generates the nodal coordinate system and determines the transformation matrix from the global to local coordinate systems. Also computes the normals to the nodes. Called from the secondary executive subroutine S3DSOL.

BREAD1

Reads in the variables NXSPC, XH, and LSECT. Called from the secondary executive subroutine S3DSOL.

BREAD2

Reads in the array MSECT. Called from the secondary executive subroutine S3DSOL.

BREAD3

Reads in the array PIN. Called from the secondary executive subroutine S3DSOL.

BREAD4

Reads in the arrays TISTPS, TEPR, TPGNP, NPRESS, and PREVAL. Called from the secondary executive subroutine S3DSOL.

BUCKSH

Generates the input file for the MHOST module (SOLVEH), calls SOLVEH, and writes the eigenvalues and eigenvectors to the output file. Called from the subroutine NLINER.

CHCKMS

Determines the number of words required in HITCAN and compares this value to the size of the array B. Called from the secondary executive subroutines HSOLID, HPLATE, and S3DSOL.

CHECKD

Compares the new incremental nodal displacements with the previous incremental displacements using the norm. Called from the subroutine NLINER.

<u>DETNPL</u>

Determines the maximum nodal thickness in the model and calculates the maximum number of plies required to fill that thickness. Called from the secondary executive subroutine S3DSOL.

<u>DETPAR</u>

Determines several parameters that are rquired in allocating storage. Called from the secondary executive subroutine S3DSOL.

<u>DYNMIN</u>

Reads in the data for the dynamic analysis. This data consists of the arrays NDISPL, DISPIN, DISPI, NVELO, VELOIN, VELOI, NACCEL, ACELIN, ACCELI, NPERD, and PERDLD. Called from the secondary executive subroutines HSOLID, HPLATE, and S3DSOL.

<u>DYNMCH</u>

Performs the dynamic analysis. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

<u>FFREADH</u>

This subroutine is used to support the free-field input format when reading program option cards. Called from the subroutine OPTNH.

<u>E11CON</u>

Writes the element connectivities of the plane strain element to the MHOST input file. Called by the subroutine MHPREP.

<u>E75COND</u>

Writes the element connectivities of the plate element to the MHOST input file. Called by the subroutine MTPREP.

EL3CON

Writes the element connectivities of the plane stress element to the MHOST input file. Called by the subroutine MHPREP.

EL7CON

Writes the element connectivities of the eight node solid element plane strain to the MHOST input file. Called by the subroutine MHPREP.

EL75CON

Writes the element connectivities of the plane strain element to the MHOST input file. Called by the subroutine MHPREP.

EL3PROP

Writes the D-matrix to the MHOST input file for the plane stress element. Called by the subroutine MHPREP.

EL7PROP

Writes the D-matrix to the MHOST input file for the eight node solid element. Called by the subroutine MHPREP.

E11PROP

Writes the D-matrix to the MHOST input file for the plane stress element. Called by the subroutine MHPREP.

E75PROP

Writes the nodal laminate properties to the MHOST input file for the plate element. Called by the subroutine MHPREP.

ELTYPE

Sets various parameters based on the type of element used in the analysis. This subroutine is called by the executive module.

GETIME

A utility subroutine, whose function is to obtain the cumulative cpu time.

HCPEM1

Sets the pointers NH1, NH2, ..., NH32, NH33. Called from the secondary executive subroutine HPLATE.

HCPEM2

Sets the pointers NH34, NH35, ..., NH45, NH46. Called from the secondary executive subroutine HPLATE.

HCTEMW

Sets the pointers NH36T, NH37T, ..., NH53T, NH54T. Called from the secondary executive subroutine HPLATE.

HDNPLY

Determines the maximum wall thickness of the shell and calculates the number of plies required to fill the wall thickness. Called from the secondary executive subroutine HPLATE.

HINIT

Sets the FORTRAN unit numbers. Also opens various files. This subroutine is called by the executive module.

HPLATE

This subroutine is a second level executive program, which controls the generation and the analysis of hollow type structures using a plate element.

HREAD1

Reads in the array MWSECT. Called from the secondary executive subroutine HPLATE.

HREAD2

Reads in the array PIN. Called from the secondary executive subroutine HPLATE.

HREAD3

Reads in the arrays TISTPS, TEPR, NPRESS, NTNOD, TEPLY, and PREVAL. Called from the secondary executive subroutine HPLATE.

HSOLID

This subroutine is a second level executive program, which controls the generation and the analysis of solid type structures, using plane strain, plane stress, or plate elements.

HTMET

Calls the HITCAN library module METCAN. Called from the subroutine NLINER.

INITMS

Initializes the microstresses, microstress rates, microstrains, ply stresses, ply strains, and cumulative global displacements. Called from the subroutine NLINER.

INITP3

Sets the pointers ND42, ND43, ..., ND83, ND84. Called from the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

INITSR

Initializes nodal stress resultants. Called from the subroutine NLINER.

LOGOH

Writes to the output file the HITCAN logo. This subroutine is called by the executive module.

MHPREP

Writes the input file KMHOST into the module MHOST. Called from the subroutine NLINER.

MIDWAL

Moves the input points from the shell surface to the shell wall mid-thickness line. This subroutine is used in the HPLATE model option. Called by the subroutine WPROFIL.

MODESH

Generates the input file for the MHOST module (SOLVEH), calls SOLVEH, and writes the frequencies, eigenvalues, and eigenvectors to the output file KOUT. Called from the subroutine NLINER.

MPROP

Reads the microproperties calculated in METCAN from the post files KPOST2, KPOST3, KPOST4, and KPOST5. Sets the current material flags and writes the microproperties to the output file KOUT. Called from the subroutine NLINER.

MSTRES

Reads the ply stresses and microstresses calculated in METCAN from the post files KPOST6, KPOST7, KPOST8, and KPOST9 and adds these stresses to the cumulative stresses. Also, writes the microstresses to the output file KOUT. Called from the subroutine NLINER.

MTPREP

Writes the input file KMHOST into the module MHOST. Called from the subroutine DYNMCH.

NASINT

Interprets a NASTRAN bulk data deck (generated by COBSTRAN). This bulk data deck contains the finite element model. Called from the secondary executive subroutine S3DSOL.

NLINER

Performs the incremental static analysis. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

NULIN

A utility routine that zeros out an integer matrix.

NULREA

A utility routine that zeros out a real matrix.

OMODEL

Writes to the output file KOUT the element connectivities, nodal coordinates, and nodal temperatures. Called from the secondary executive module S3DSOL.

OPTNH

This subroutine reads the program option cards and sets all logical controls in accordance with user choices. Called from the executive control module.

OTPUTH

Writes to the output file KOUT, the ply lay-up at each node, the finite element model, load data, and nodal radii of curvature. Called from the executive control module.

<u>**OUTH10**</u>

Writes to the output file KOUT, the program option cards chosen. Called from the executive control module.

<u>**OUTPTC**</u>

Reads in the arrays NPRT, NPRTS, NPRTP, NPPLY, TIMEPN, TIMEMS, and TIMEMB. These arrays control the output and also determine when a PATRAN results file is to be created, a modal analysis is performed, and buckling analysis is done. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

<u>**PCOBST**</u>

Generates the input file KCBST, for COBSTRAN. Called from the secondary executive subroutine S3DSOL.

<u>**PLYH**</u>

Calculates the ply properties PLH(1), PLH(2), ..., PLH(13), PLH(14). Called from the subroutine NLINER.

<u>**PLPROP**</u>

Reads the ply properties from the METCAN post file KPOST5. Also, determines the compliance matrix and the inverse of the compliance matrix. Called from the subroutine NLINER.

<u>**PLYTEM**</u>

Sets the ply temperatures. Called from the subroutines PREPSH, PREPSO, and PREPSP.

<u>**PMETCN**</u>

Generates the input files KMETCN and KDBANK for the module METCAN. Called from the subroutine NLINER.

PREPSH

Cobstran executive control module. Called from the secondary executive subroutine HPLATE.

PREPSO

Cobstran executive control module. Called from the secondary executive subroutine HSOLID.

PREPSP

Generates that finite element associated with the spars. Called from the secondary executive subroutine HPLATE.

PROPH

Determines the axial, bending, and coupled stiffness matrices. Called from the subroutine NLINER.

PRTEPS

Writes the microstrains to the output file KOUT. This subroutine is not used at the present time.

PRTMPR

Writes the current constituent properties to the output file KOUT. Called from the subroutine MPROP.

PRTMS

Writes the microstresses to the output file KOUT. Called from the subroutine MSTRES.

PRTSR

Writes the microstress rates to the output file KOUT. Not used at the present time.

PSTRES

Write the ply stresses, global displacements, and global reactions to the output file KOUT. Called from the subroutine NLINER.

PSTRSS

Write the ply stresses, global displacements, and global reactions to the output file KOUT. Called from the subroutine DYNMCH.

READML

Reads in the nodal coordinates and element connectivities, the arrays KEI and GNP, from the input file KREAD. Also, determines the nodes above and below each node. Called from the secondary executive subroutine S3DSOL.

REFLECH

Temporarily store the user input file in the file KREFL and print the user input file, if requested by the user. Called from the executive control module.

RESTART

Read/write the restart file KSTART. Called from the subroutine NLINER.

RIBOUN

Reads in the boundary conditions and the nodal transformation data from the input file KREAD, this includes the arrays NTRANS, TRANG, and NBOUND. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

RINFOR

Reads in the nodal forces and centrifugal loading data from the input file KREAD, i. e. , the variables GRIDP1, GRIDP2, ANGVEL, NCFNOD, NCFDIR, and NCFVAL. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

RINLAY

Reads in the ply lay-up at each node. This includes the arrays NPNOD and MPLY. Called by the secondary executive subroutine S3DSOL.

RINPAR

Reads in the parameter data from the input file KREAD. This data consists of the variables NDES, NSECT, IGRD, IU, JU, MESH, XBEG, XEND, NXSPAR, NYSPC, MAXNP, NETOT, LSECT, NIPL, NOSC, NEYY, NETT, MAXPLY, LMAX, NTISTP, NMECHC, NTHERC, LINC, MSTART, MITER, TOL, NEIGV, NSUBD, INCREG, MHITER, RESID, DAMPMS, DAMPST, NPLSET, INCDYN, NCFOR, NPRES, NTEMP, NOPERD, NBC, NTR, NDIS, NVEL, NACC, and PINTER. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

RINPLY

Reads in ply data, i. e., the arrays PERT, CODEH, MPLY, and NPLY from the input file KREAD. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

RINSPA

Reads in the spar description data from the input file KREAD. This data consists of the arrays NSPDES, NXSPC, XH, and SY. Called by the secondary executive subroutine HPLATE.

S3DSOL

This subroutine is a second level executive program, which controls the generation and the analysis of solid type structures, using the eight node solid element.

SCPEM1

Sets the pointers NM1 and NM2. Called from the secondary executive subroutine HSOLID.

<u>SCPEM2</u>

Sets the pointers NM3, NM4, ..., NM60, NM61. Called from the secondary executive subroutine HSOLID.

<u>SCTEMW</u>

Sets the pointers NMT61, NMT62, ..., NMT72, NMT73. Called from the secondary executive subroutine HSOLID.

<u>SERCH</u>

Determines the maximum nodal displacement in the PATRAN results file KNODAL. Called from the subroutine WPOST.

<u>SHELLT</u>

Determines the transformation matrix from the local nodal coordinate system to the global coordinate system. This subroutine is used only for the plate element. Called from the secondary executive subroutines HPLATE and HSOLID.

<u>SOLVEH</u>

Calls the HITCAN library module MHOST. Called from the subroutine NLINER.

<u>SREAD1</u>

Reads in the array MSECT from the input file KREAD. Called from the secondary executive subroutine HSOLID.

<u>SREAD2</u>

Reads in the arrays NXDIV and PIN from the input file KREAD. Called from the secondary executive subroutine HSOLID.

SREAD3

Reads in the arrays TISTPS, TEPR, NTNOD, TEPLY, NPRES, and PREVAL from the input file KREAD. Called from the secondary executive subroutine HSOLID.

TANLOC

Establishes a set of three orthigonal unit vectors V1, V2, and V3, such that V3 lies along the given vector X and V2 is in the plane defined by the vectors X and Y. Called from the subroutine NLINER.

TBGRDH

Calculates the nodal coordinates, nodal temperatures, nodal pressure loading, and the wall thickness at each node from the user input. This subroutine is used in the HSOLID model option. Called from the subroutine PREPSO. This subroutine is a modified version of the subroutine TBGRD, which is found in the library module COBSTRAN (see Reference 1).

TBG01H

Divides a solid structure along the X-axis and Y-axis into the appropriate divisions. Called from the subroutine TBGRDH. This subroutine is a modified version of the subroutine TBG01, which is found in the library module COBSTRAN (see Reference 1).

TBPLY

Determines the number of plies required at each node for the HSOLID model option. Called from the subroutine PREPSO. This subroutine is a modified version of the subroutine TBGRD, which is found in the library module COBSTRAN (see Reference 1).

WDIS

Writes the global displacements to the PATRAN nodal results file KNODAL. Called from the subroutine WPOST.

WINDH

Generates the element connectivities for the HPLATE model option. Called from the subroutine PREPSH. This subroutine is a modified version of the subroutine WIND, which is found in the library module COBSTRAN (see Reference 1).

WNOD

Writes the stress resultants to the PATRAN nodal results file KNODAL. Called from the subroutine WPOST. This subroutine is not used at the present time.

WOUTH

Writes to the output file KOUT the element connectivities, nodal coordinates, etc. for the HPLATE model option. Called from the subroutines PREPSH and PREPSP. This subroutine is a modified version of the subroutine WINOUT, which is found in the library module COBSTRAN (see Reference 1).

WPLYH

Determines the ply lay-up at each node for the HPLATE model option. Called from the subroutines PREPSH and PREPSP. This subroutine is a modified version of the subroutine WINOUT, which is found in the library module COBSTRAN (see Reference 1).

WPOST

Controls the generation of the PATRAN results files KNODAL and KTRAN. Called from the subroutine NLINER.

WPUTH

Determines the nodal coordinates, nodal temperatures, and nodal pressures for the HPLATE model option. Called from the subroutine PREPSH. This subroutine is a modified version of the subroutine WINPUT, which is found in the library module COBSTRAN (see Reference 1).

WNEUT

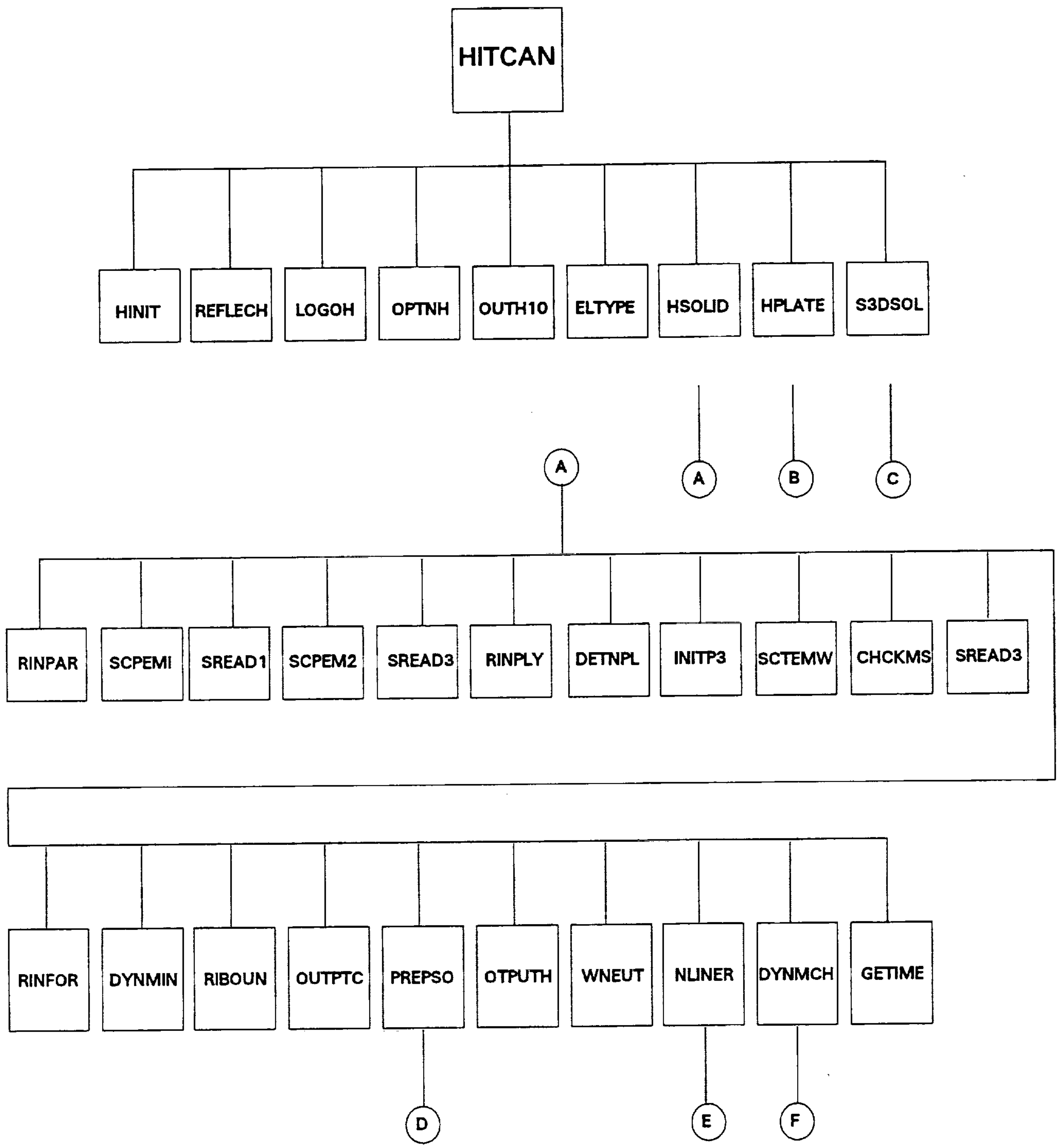
Writes a PATRAN neutral file. Called by the secondary executive subroutines HPLATE, HSOLID, and S3DSOL.

WPROFIL

Moves the grid points from the surface of the structure to the mid-plane of the shell. This subroutine is used in the HPLATE model option. Called from the subroutine PREPSH. This subroutine is a modified version of the subroutine MIDWAL, which is found in the library module COBSTRAN (see Reference 1).

WSPARH

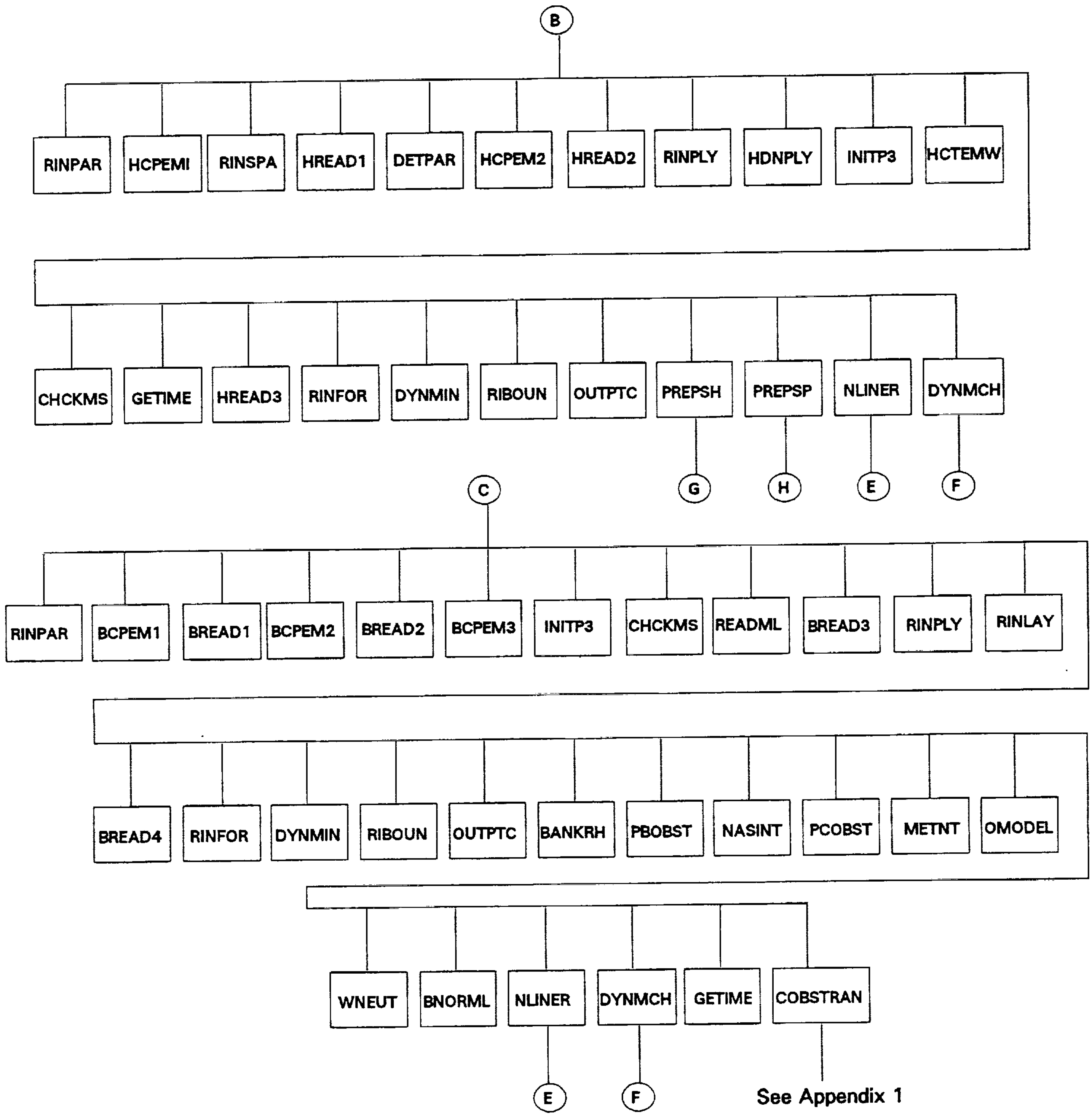
Determines the element connectivities for the spars. Also, finds the corresponding shell node for each spar node. Called from the subroutine PREPSP. This subroutine is a modified version of the subroutine WSPAR, which is found in the library module COBSTRAN (see Reference 1).



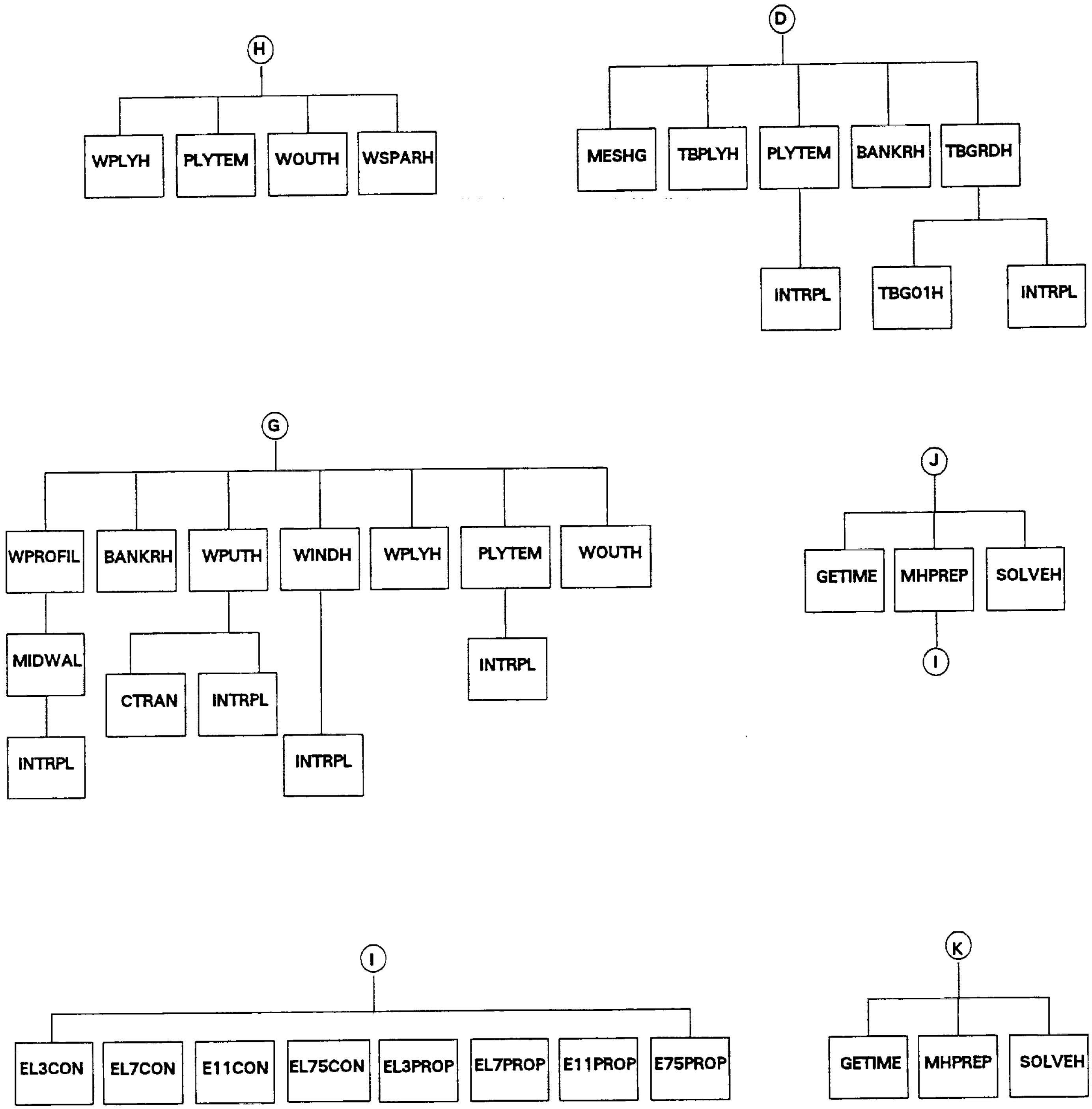
HITCAN CALLING TREE

FIGURE 3.1

LIST18

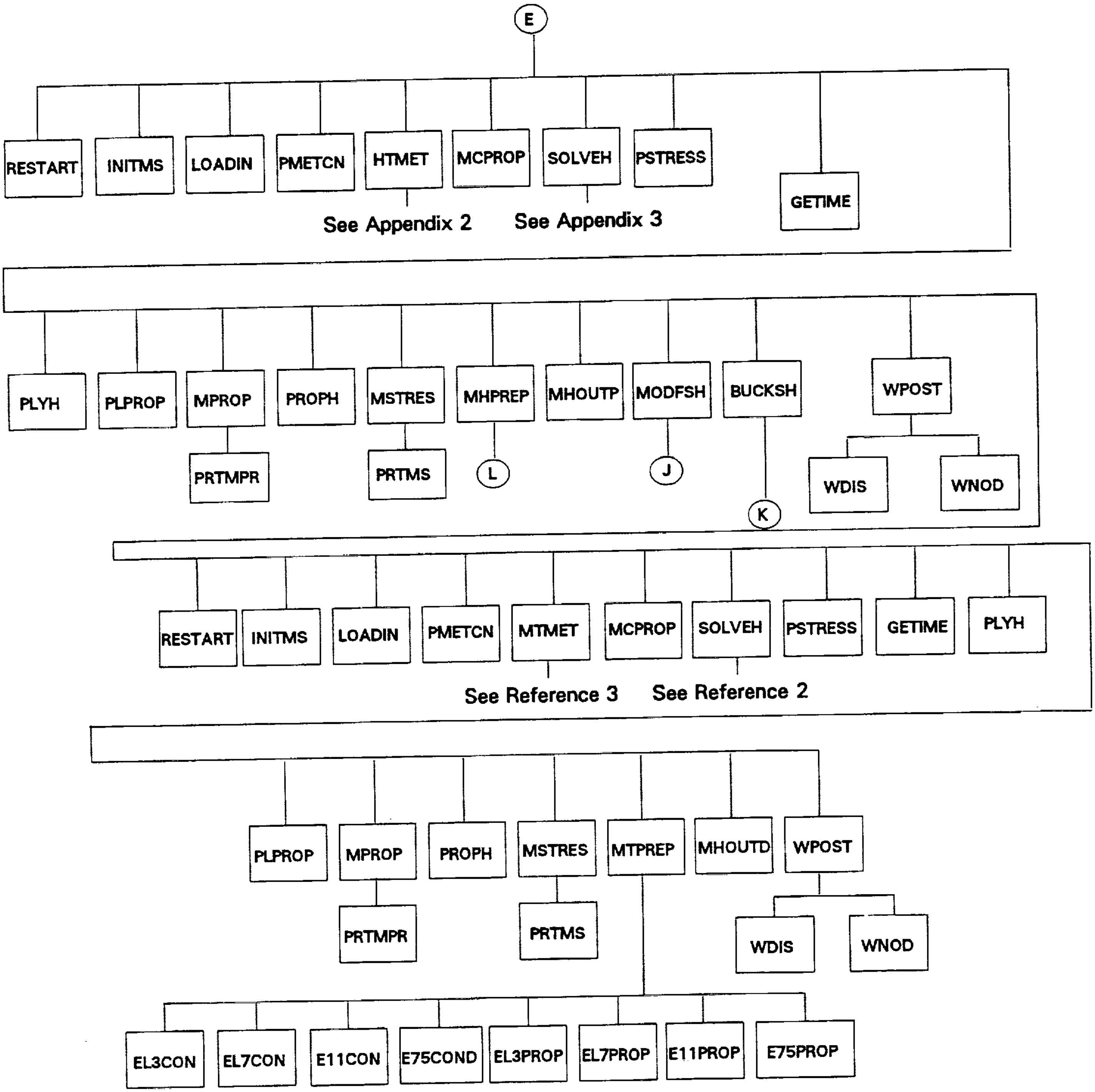


HITCAN CALLING TREE
FIGURE 3.1 (continued)



HITCAN CALLING TREE

FIGURE 3.1 (continued)



HITCAN CALLING TREE
FIGURE 3.1 (continued)

LIST21

CHAPTER 4

DATA STORAGE SCHEME

A dynamic core allocation strategy has been implemented in the HITCAN program. The array B is used as the working storage. The length of this array is the variable NASIZE. The array B is used to store only the information used directly in HITCAN. METCAN and MHOST have their own dynamic storage areas. In METCAN the array A is used as the working storage, MAXLEN is the length of this array. IWORK is the working storage array for MHOST and ISIZE is its length. The three arrays A, B, and IWORK and the variables MAXLEN, ISIZE, and NASIZE are set in the main program of HITCAN. If the number of words required for an analysis exceeds the size of the array B, execution will stop and a message will be printed with the number of additional words needed. For a description of the dynamic core allocation scheme employed in MHOST, the programmer is referred to Reference 5.

In HITCAN the core allocation is carried out in the subroutines HCPEM1, HCPEM2, HCTEMW, SCPEM1, SCPEM2, SCTEMW, BCPEM1, BCPEM2, BCPEM3, and INITP3. The first three of these subroutines are used for the HPLATE model option, the next three subroutines are used in the HSOLID model option, the next three are used in the S3DSOLID model option and the READ IN THE MODEL option. The last subroutine INITP3 is used by each of the model options.

In Sections 4.1 through 4.3, the pointers used in the dynamic allocation and the common blocks they reside in are listed and described. In Section 4.4 the common blocks used in HITCAN, which do not contain pointers are also listed and described.

4.1 POINTERS USED IN THE HPLATE MODEL OPTION

For the HPLATE model option the variables used as pointers and the common blocks in which they are located are listed below. The following variables are initialized in the subroutine HCPEM1 and are located in the common block POINT.

```
 COMMON   /POINT/   NH1,  NH2,  NH3,  NH4,  NH5,  NH6,  NH7,
1                   NH8,  NH9,  NH10, NH11, NH12, NH13, NH14,
2                   NH15, NH16, NH17, NH18, NH19, NH20, NH21,
3                   NH22, NH23, NH24, NH25, NH26, NH27, NH28,
4                   NH29, NH30, NH31, NH32, NH33, NH34, NH35,
5                   NH36, NH37, NH38, NH39, NH40, NH41, NH42,
6                   NH43, NH44, NH45, NH46, NH47, NH48, NH49,
7                   NH50, NH51, NH52, NH53, NH54, NH55, NH56,
8                   NH57, NH58, NH59, NH60, NH61, NH62, NH63
```

NH1 Points to the array NSPDES storing the spar descriptions.

NH2 Points to the array NXSPC storing the number of elements between output sections.

NH3 Points to array X, which contains the x-coordinate of the output cross sections.

NH4 Points to the array storing spar information, SY.

NH5 Pointer to the array (PERT) containing ply information.

NH6 Pointer to the ply data array CODEH.

NH7 Points to the array MPLY, which stores the ply lay-up for one half the thickness of a symmetrical ply lay-up.

NH8 Points to the array TISTPS, which contains the time at each time step.

NH9 Points to the array storing the boundary conditions, NBOUND.

NH10 Pointer to the array containing the initial fiber properties (FPI).

NH11 Pointer to the array storing the initial matrix properties (MPI).

NH12 Pointer to the array storing the initial interface properties (DPI).

NH13 Pointer to the array storing the initial fiber stresses (FSIGI).

NH14 Pointer to the array containing the initial matrix stresses (MSIGI).

NH15 Pointer to the array containing the initial interface stresses (DSIGI).

NH16 Pointer to the array storing the initial fiber stress rates (FDOTI).

NH17 Pointer to the array storing the initial matrix stress rates (MDOTI).

NH18 Pointer to the array storing the initial interface stress rates (DDOTI).

NH19 Pointer to the array containing the fiber exponents (FTVCI).

NH20 Pointer to the array storing the matrix exponents (MTVCI).

NH21 Pointer to the array storing the interface exponents (DTVCI).

NH22 Points to the array LSP, which stores the spar location on the top and bottom surfaces.

NH23 Points to the array NF, which stores the number of fibers per bundle for each ply.

NH24 Points to the scratch array NT.

NH25 Points to the array (DF) storing the fiber diameter for each different ply.

NH26 Points to the scratch array NPD.

NH27 Points to the array containing pressure data, NPRESS.

NH28 Points to the array ANGVEL, which contains the angular velocity for each time step.

NH29 Points to the array NCFNOD, which contains concentrated force data.

NH30 Points to the array NCFDIR, which contains concentrated force data.

NH31 Points to the array CFVAL, which contains concentrated force data.

NH32 Not used.

NH33 Points to the array MWSCT, which contains the number of input points for each cross section.

The following variables are initialized in the subroutine HCPEM2 and are located in the common block POINT.

NH34 Points to the array (PIN) storing surface geometry.

NH35 Points to the scratch array NPCE.

NH36 Points to the array (TEPR) storing temperature and pressure data.

NH37 Points to the array PREVAL, which contains edge load intensity for each time step.

NH38 Points to the scratch array NCORDL.

NH39 Points to the scratch array RAD.

NH40 Points to the array NTRANS, which stores data for nodal coordinate transformation.

NH41 Pointer to the array TRANG, which stores data for nodal coordinate transformation.

NH42 Points to the array (DISPI) containing the initial nodal displacements.

NH43 Points to the array (VELOI) containing the initial nodal velocities.

NH44 Points to the array (ACCELI) containing the initial nodal accelerations.

NH45 Points to the array (NPERD) containing periodic load data.

NH46 Points to the array (PERDLD) containing periodic load data.

The following variables are initialized in the subroutine HCTEMW and are located in the common block POINTT.

```
 COMMON    /POINTT/ NH23T,      NH24T,      NH25T,      NH26T,
1                   NH27T,      NH28T,      NH29T,      NH30T,
2                   NH31T,      NH32T,      NH33T,      NH34T,
3                   NH35T,      NH36T,      NH37T,      NH38T,
4                   NH39T,      NH40T,      NH41T,      NH42T,
5                   NH43T,      NH44T,      NH45T,      NH46T,
6                   NH47T,      NH48T,      NH49T,      NH50T,
7                   NH51T,      NH52T,      NH53T,      NH54T
```

NH36T Points to the scratch array TTAB.

NH37T Points to the scratch array TPTT.

NH38T Points to the scratch array XINT.

NH39T Points to the scratch array ZINT.

NH40T Points to the scratch array VDOT.

NH41T Points to the array ANG containing the ply orientation angle.

NH42T Points to the scratch array XR.

NH43T Points to the scratch array YS.

NH44T Points to the scratch array YINT.

NH45T Points to the scratch array TPTTAB.

NH46T Points to the scratch array TP.

NH47T Points to the scratch array TPPP.

NH48T Points to the scratch array TJJ.

NH49T Points to the scratch array TIN.

NH50T Points to the scratch array TPTIN.

NH51T Pointer to the array NTNOD. This array contains temperature data information.

NH52T Pointer to the array TEPLY. This array contains temperature data information.

NH53T Points to the scratch array TINT.

NH54T Points to the scratch array PPTS.

All of the model options use the pointers listed below. These variables are initialized in the subroutine INITP3 and are located in the common block IPOINT.

```
 COMMON   /IPOINT/ ND1,  ND2,  ND3,  ND4,  ND5,  ND6,  ND7,
1                  ND8,  ND9,  ND10, ND11, ND12, ND13, ND14,
2                  ND15, ND16, ND17, ND18, ND19, ND20, ND21,
3                  ND22, ND23, ND24, ND25, ND26, ND27, ND28,
4                  ND29, ND30, ND31, ND32, ND33, ND34, ND35,
5                  ND36, ND37, ND38, ND39, ND40, ND41, ND42,
6                  ND43, ND44, ND45, ND46, ND47, ND48, ND49,
7                  ND50, ND51, ND52, ND53, ND54, ND55, ND56,
8                  ND57, ND58, ND59, ND60, ND61, ND62, ND63,
9                  ND64, ND65, ND66, ND67, ND68, ND69, ND70,
+                  ND71, ND72, ND73, ND74, ND75, ND76, ND77,
1                  ND78, ND79, ND80, ND81, ND82, ND83, ND84
```

ND42 Pointer to the array (KAD) storing the shell node numbers of the duplicate nodes.

ND43 Pointer to the array (KEI) storing the element connectivities.

ND44 Pointer to the array (GNP) storing the nodal coordinates.

ND45 Pointer to the array (TPGNP) storing the nodal temperatures and pressures.

ND46 Pointer to the array (THHF) storing the nodal half thicknesses.

ND47 Pointer to the array (NLEP) containing the number of plies at each node.

ND48 Pointer to the array (MDES) storing the ply lay-up.

ND49 Pointer to the array (THERMO) storing the ply temperatures.

ND50 Pointer to the axial stiffness matrix (AS).

ND51 Pointer to the coupling stiffness matrix (CS).

ND52 Pointer to the bending stiffness matrix (BS).

ND53 Pointer to the coefficients of thermal expansion.

ND54 Pointer to array storing the ply density (DEN).

ND55 Pointer to the array TLAY. This array contains the incremental nodal temperatures.

ND56 Pointer to the array PDIFI, which contains the incremental nodal pressures on the lower surface.

ND57 Pointer to the array PDIFO, which contains the incremental nodal pressures on the upper surface.

ND58 Pointer to the total global displacements (DISP).

ND59 Pointer to the incremental global displacements (DISVH).

ND60 Pointer to the incremental stress resultants (MSH).

ND61 Pointer to the incremental stress resultants (NSH).

ND62 Pointer to the incremental stress resultants (QSH).

ND63 Pointer to the incremental nodal reactions (REACF).

ND64 Points to the incremental nodal loads COFORC.

ND65 Pointer to the array (PLH) containing the ply properties.

ND66 Pointer to the array (TOLD) containing the previous ply temperatures.

ND67 Pointer to the scratch array TEMP.

ND68 Pointer to the array (TNEW) containing the current ply temperatures.

ND69 Pointer to the array containing the eigenvalues (EIGNVA).

ND70 Pointer to the array storing the eigenvectors (EIGNVE).

ND71 Pointer to the scratch array TEMPER.

ND72 Pointer to the array (PC), which contains the equivalent nodal composite properties.

ND73 Pointer to the array STRMOM. This array either contains the nodal stresses or stress resultants.

ND74 Pointer to the array PRESUR. This array contains the incremental pressure loading.

ND75 Points to the array (ETRAN). This array contains nodal transformation matrix from global to local coordinates.

ND76 Pointer to the array (THRSTN), which contains the thermal strain calculated in METCAN.

ND77 Pointer to the array DISPD. This array contains the initial displacements.

ND78 Pointer to the array VELOD. This array contains the initial velocities.

ND79 Pointer to the array ACCELD. This array contains the initial accelerations.

ND80 Points to the array TOTFOR.

ND81 Points to the array TOTRED.

4.2 POINTERS USED IN THE HSOLID MODEL OPTION

For the HSOLID model option the variables used as pointers and the common blocks in which they are located are listed below. The following variables are initialized in the subroutine SCPEM1 and are located in the common block POINTS.

```
 COMMON    /POINTS/ NM1,  NM2,  NM3,  NM4,  NM5,  NM6,  NM7,
1                   NM8,  NM9,  NM10, NM11, NM12, NM13, NM14,
2                   NM15, NM16, NM17, NM18, NM19, NM20, NM21,
3                   NM22, NM23, NM24, NM25, NM26, NM27, NM28,
4                   NM29, NM30, NM31, NM32, NM33, NM34, NM35,
5                   NM36, NM37, NM38, NM39, NM40, NM41, NM42,
6                   NM43, NM44, NM45, NM46, NM47, NM48, NM49,
7                   NM50, NM51, NM52, NM53, NM54, NM55, NM56,
8                   NM57, NM58, NM59, NM60, NM61, NM62, NM63
```

NM1 Pointer to the array (MSECT) storing the number of input points for each cross-section.

NM2 Pointer to the array PIN. This array contains the coordinates of the input points, which describe the surface geometry.

The following variables are initialized in the subroutine SCPEM2 and are located in the common block POINTS.

NM3 Pointer to the scratch array NCOOR.

NM4 Pointer to the array (PERT) containing ply information.

NM5 Pointer to the ply data array CODEH.

NM6 Pointer to the array NXDIV, which contains the number of elements between output sections.

NM7 Pointer to the scratch array THWAL.

NM8 Pointer to the array MPLY. This array stores one half of the ply order for a symmetric ply lay-up.

NM9 Pointer to the array NPLY. This array stores one half of the ply order for an unsymmetrical ply lay-up.

NM10 Pointer to the scratch array NPCE.

NM11 Pointer to the array containing the initial fiber properties (FPI).

NM12 Pointer to the array storing the initial matrix properties (MPI).

NM13 Pointer to the array storing the initial interface properties (DPI).

NM14 Pointer to the array storing the initial fiber stresses (FSIGI).

NM15 Pointer to the array containing the initial matrix stresses (MSIGI).

NM16 Pointer to the array containing the initial interface stresses (DSIGI).

NM17 Pointer to the array storing the initial fiber stress rates (FDOTI).

NM18 Pointer to the array storing the initial matrix stress rates (MDOTI).

NM19 Pointer to the array storing the initial interface stress rates (DDOTI).

NM20 Pointer to the array containing the fiber exponents (FTVCI).

NM21 Pointer to the array storing the matrix exponents (MTVCI).

NM22 Pointer to the array storing the interface exponents (DTVCI).

NM23 Pointer to the array (NF) containing the number of fibers per bundle for each ply.

NM24 Pointer to the scratch array NT.

NM25 Pointer to the array (DF), which stores the fiber diameters.

NM26 Pointer to the scratch array NPD.

NM27 Not used.

NM28 Pointer to the array storing the time at each time step (TISTPS).

NM29 Pointer to the rotational speed (ANGVEL).

NM30 Pointer to the location of the concentrated forces (NCFNOD).

NM31 Pointer to the array containing the direction of the concentrated forces (NCFDIR).

NM32 Pointer to the value of the concentrated forces (CFVAL).

NM33 Not used.

NM34 Pointer to the array NBOUND. This array stores the boundary conditions.

NM35 Pointer to the array storing edge load data (NPRESS).

NM36 Pointer to the array PREVAL. This array stores the intensity of the edge loading at each time step.

NM37 Pointer to the array storing nodal transformation data (NTRANS).

NM38 Pointer to the array TRANG. This array stores the angle of rotation.

NM39 Pointer to the array storing the initial displacements (DISPI) for direct time integration.

NM40 Pointer to the array storing the initial velocities (VELOI) for direct time integration.

NM41 Pointer to the array storing the initial accelerations (ACCELI) for direct time integration.

NM42 Pointer to the array (NPERD) containing periodic load data.

NM43 Pointer to the array (PERDLD) containing periodic load data.

NM55 Pointer to the scratch array NCORDL.

NM56 Pointer to the scratch array RAD.

NM57 Pointer to the scratch array XINT.

NM58 Pointer to the scratch array XOUT.

NM59 Pointer to the scratch array YINT.

NM60 Pointer to the scratch array RX.

NM61 Pointer to the scratch array KDES.

The following variables are initialized in the subroutine HCTEMW and are located in the common block POINTT.

```
COMMON   /PINTTS/ NMT61,    NMT62,    NMT63,    NMT64,
1                 NMT65,    NMT66,    NMT67,    NMT68,
2                 NMT69,    NMT70,    NMT71,    NMT72,
3                 NMT73,    NMT74,    NMT75,    NMT76
```

NMT61 Points to the scratch array TEPR.

NMT62 Points to the scratch array TIN.

NMT63 Points to the scratch array TP.

NMT64 Points to the scratch array TJ.

NMT65 Points to the scratch array TT.

NMT66 Points to the scratch array VDOT.

NMT67 Points to the scratch array V2DOT.

NMT68 Points to the scratch array YOUT.

NMT69 Points to the scratch array RY.

NMT70 Points to the array NTNOD. This array contains temperature data information.

NMT71 Points to the array TEPLY. This array contains temperature data information.

NMT72 Points to the scratch array TINT.

NMT73 Pointer to the scratch array PPTS.

4.3 POINTERS USED IN THE S3DSOLID MODEL OPTION

For the S3DSOLID model option and the READ IN THE MODEL option the variables used as pointers and the common blocks in which they are located are listed below. The following variables are initialized in the subroutine BCPEM1 and are located in the common block POINTB.

```
COMMON    /POINTB/  NB1,      NB2,      NB3,      NB4,      NB5,
1                   NB6,      NB7,      NB8,      NB9,      NB10,
2                   NB11,     NB12,     NB13,     NB14,     NB15,
3                   NB16,     NB17,     NB18,     NB19,     NB20,
4                   NB21,     NB22,     NB23,     NB24,     NB25,
5                   NB26,     NB27,     NB28,     NB29,     NB30,
6                   NB31,     NB32,     NB33,     NB34,     NB35,
7                   NB36,     NB37,     NB38,     NB39,     NB40,
8                   NB41,     NB42,     NB43,     NB44,     NB45,
9                   NB46,     NB47,     NB48,     NB49,     NB50,
+                   NB51,     NB52,     NB53,     NB54,     NB55
```

NB1 Pointer to the array (NXSPC) storing the number of elements between output sections.

NB2 Pointer to the array PIN. This array contains the x-ordinate of each output section.

NB3 Pointer to the array (LSECT) storing the number of sections for each input plane.

NB4 Points to the array MSECT. This array stores the number of input points for each section.

NB5 Pointer to the array PIN. This array contains the coordinates of the input points, which describe the surface geometry.

NB6 Points to the array (TEPR) containing temperature data.

NB7 Pointer to the array (PERT) containing ply information.

NB8 Pointer to the ply data array CODEH.

NB9 Pointer to the array MPLY. This array stores one half of the ply order for a symmetric ply lay-up.

NB10 Pointer to the array NPLY. This array stores one half of the ply order for an unsymmetrical ply lay-up.

NB11 Pointer to the array storing the time at each time step (TISTPS).

NB12 Pointer to the array NBOUND. This array stores the boundary conditions.

NB13 Pointer to the array containing the initial fiber properties (FPI).

NB14 Pointer to the array storing the initial matrix properties (MPI).

NB15 Pointer to the array storing the initial interface properties (DPI).

NB16 Pointer to the array storing the initial fiber stresses (FSIGI).

NB17 Pointer to the array containing the initial matrix stresses (MSIGI).

NB18 Pointer to the array containing the initial interface stresses (DSIGI).

NB19 Pointer to the array storing the initial fiber stress rates (FDOTI).

NB20 Pointer to the array storing the initial matrix stress rates (MDOTI).

NB21 Pointer to the array storing the initial interface stress rates (DDOTI).

NB22 Pointer to the array containing the fiber exponents (FTVCI).

NB23 Pointer to the array storing the matrix exponents (MTVCI).

NB24 Pointer to the array storing the interface exponents (DTVCI).

NB25 Pointer to the array (NF) containing the number of fibers per bundle for each ply.

NB26 Pointer to the array (DF), which stores the fiber diameters.

NB27 Pointer to the scratch array NPD.

NB28 Pointer to the rotational speed (ANGVEL).

NB29 Pointer to the location of the concentrated forces (NCFNOD).

NB30 Pointer to the array containing the direction of the concentrated forces (NCFDIR).

NB31 Pointer to the value of the concentrated forces (CFVAL).

NB32 Pointer to the array storing edge load data (NPRESS).

NB33 Pointer to the array PREVAL. This array stores the intensity of the edge loading at each time step.

NB34 Pointer to the array storing nodal transformation data (NTRANS).

NB35 Pointer to the array TRANG. This array stores the angle of rotation.

NB36 Pointer to the array storing the initial displacements (DISPI) for direct time integration.

NB37 Pointer to the array storing the initial velocities (VELOI) for direct time integration.

NB38 Pointer to the array storing the initial accelerations (ACCELI) for direct time integration.

NB39 Pointer to the array (NPERD) containing periodic load data.

NB40 Pointer to the array (PERDLD) containing periodic load data.

4.4 COMMON BLOCKS USED IN HITCAN

The following additional common blocks are used in HITCAN. After a listing of each named common block the variables contained in the common block are described, where appropriate. Note that if a variable in a common block is no longer used in the program this variable is not defined.

The common blocks HUNITS and CMETCN contain the variable names of the Fortran units used in HITCAN. These variables and their assigned unit numbers are described in Chapter 6.

```
 COMMON  /HUNITS/   KDES1,    KKEI,    KBULK,   KREAD,   KOUT,
1                   KCBST,    KCBMT,   KPROP,   KCBFE,   KDBANK,
2                   KTRAN,    KPOST,   KMHFOR,  KREFL,   KPLY,
3                   KPROPI,   KSTR,    KLAM,    KSIG,    KNODAL,
4                   KDISP,    KMSTR,   KTRANF,  KCBOUT,  KTEMP,
5                   KMHOST,   KSTART,  KPOSTF,  KPAT,    KEIGEN
 COMMON  /CMETCN/   KPOST0,   KPOST1,  KPOST2,  KPOST3,  KPOST4,
1                   KPOST5,   KPOST6,  KPOST7,  KPOST8,  KPOST9,
2                   KMETCN
```

The common block BOLH contains the logical variable names of program option cards.

```
 COMMON   /BOLH/   BIDE,     DYNA,     PRTOUT,  WINDML,  SOLID,
1                  SUPER,    UNSY,     PROFIL,  PROT,    PREP,
2                  POST,     MODES,    HOLLOW,  DISPL,   DAMP,
3                  PLYORD,   ACCEL,    TEM,     VELO,    PRES,
4                  PERIOD,   PATRAN,   MHOST,   EZRD,    ECHO,
5                  RANG,     ANGULA,   CFORCE,  PANEL,   LINEAR,
6                  SPARS,    RESTAR,   HEAT,    MODEL,   FABRIC,
7                  CYLIND,   BUCK,     INTER,   BMODEL,  EL3,
8                  EL7,      EL10,     EL11,    EL75,    TRANS,
9                  MODLIN
```

DYNA Flag for dynamic analysis using direct time integration.

WINDML Flag for the HPLATE model option.

SOLID Flag for the HSOLID model option.

UNSY Flag for an unsymmetrical ply lay-up.

PROFIL Flag for the PROFILE program option card.

MODES Flag for modal analysis.

DISPL Flag for inputing initial displacements.

DAMP Flag for damping.

PLYORD Flag for symmetrical ply order.

ACCEL Flag for inputing initial accelerations.

TEM Flag for temperature loading.

VELOL Flag for inputing initial velocities.

PRES Flag for pressure loading.

PERIOD	Flag for periodic loading.
ECHO	Flag for echoing input file.
ANGULA	Flag for centrifugal loading.
CFORCE	Flag for nodal loads.
PANEL	Flag for initiating the PANEL option.
SPARS	Flag for generating spar elements.
RESTAR	Flag for initiating a restart run.
MODEL	Flag for creating a PATRAN neutral file.
FABRIC	Flag for initiating an analysis with fabrication.
BUCK	Flag for buckling analysis.
INTER	Flag for setting an interface.
BMODEL	Flag for the H3DSOLID model option.
EL3	Flag for plane stress element.
EL7	Flag for 3-D solid element.
EL11	Flag for plane strain element.
EL75	Flag for plate element.
TRANS	Flag for inputing nodal transformations.
MODLIN	Flag for reading in finite element model.

The common block INTPH contains the reference temperature.

```
 COMMON   /INTPH/   TREF,      BFT,      BFC,      BCS,      FDC,
1                   FDK
```

TREF Reference temperature

The common block LOADD contains the load data.

```
 COMMON   /LOADD/  NMECHC,   NTHERC,  LINC,    MSTART,
1                  MITER,    NTISTP,  NCFOR,   TOL,
2                  NEIGV,    NSUBD,   INCREG,  NBC,
3                  MHITER,   RESID,   AESID,   RERMSQ,
4                  AERMSQ,   NPRES,   NTEMP,   NTR,
5                  NBCON,    NTCON,   NUMPLY,  NOPERD,
6                  DAMPMS,   DAMPST,  NDIS,    NVEL,
7                  NACC,     INCDYN
```

NMECHC Number of mechanical cycles.

NTHERC Number of thermal cycles.

LINC Number of load increments between load steps.

MSTART Number of increments to be accumulated before a restart file is written.

MITER Maximum number of allowable iterations for global convergence.

NTISTP Number of load steps.

NCFOR Number of nodal loads.

TOL Global tolerence.

NEIGV Number of eigenvalues to be extracted.

NSUBD Number of subspace dimensions.

INCREG Number of increments between eigenvalue extraction.

NBC Number of boundary conditions.

MHITER Allowable number of iterations in MHOST.

RESID Allowable residual error in MHOST.

NPRES Number of edge loads.

NTEMP Number of temperature data sets.

NTR Number of transformation data sets.

NBCON Scratch variable for boundary conditions.

NTCON Scratch variable for nodal transformations.

NUMPLY Number of plies.

NOPER Number of periodic loading data sets.

DAMPMS Damping coefficient applied to the mass matrix.

DAMPST Damping coefficient applied to the stiffness matrix.

NDIS Number of initial displacement data sets.

NVEL Number of initial velocity data sets.

NACC Number of initial acceleration data sets.

INCDYN Number of accumulated load increments before material updating.

The common block ETABLE contains the element data.

```
 COMMON  /ETABLE/   NELCRD,    NELNFR,    NELNOD,    NELSTR,
1                   NELCHR,    NELCMP,    NDI
```

NELCRD Number of coordinates per node.

NELNFR Number of degrees-of-freedom per node.

NELNOD Number of nodes per element.

NELSTR Number of generalized stresses.

NELCHR Number of element material properties

NELCMP Number of stress and/or strain components per node.

NDI Number of direct stress components per node.

The common blocks HINDS, SOLDYN, and VALUEH contain model data. Note that the common block HINDS is used in the HPLATE model option and the common block SOLDYN is used in the HSOLID model option. The common block VALUEH is used in both.

```
 COMMON    /HINDS/   ALPHA(3,3), NGRID,      NELT,      NPAST,
1                    THMAX,      NXSECT,     NSPAR,     NXSPAR,
2                    NYSPC,      NYPTS,      NNQ,       NNX,
3                    LSECT(2),   NPTOT,      NETOT,     HOLD(10),
4                    MAXPLY,     MNNX,       MNNSECT,   MMWSCT,
5                    MNXSPC1,    MNXSPC,     MKNT,      NKNT,
6                    NLEPM,      MNSPDES,    MK,        MMSCAN,
7                    SPAMTH,     PLYMTH,     MNOPLY,    NPTOT1,
8                    MAXNP
 COMMON    /SOLDYN/  SPMIN,      SPMAX,      NGRD,      MNOPLY,
1                    LKDES,      LMAX,       MAXPLY,    PLYMTH
 COMMON    /VALUEH/  NP,         NE,         NDES,      IMAT,
1                    KSMF,       IU,         JU,        XBEG,
2                    XEND,       NMODE(3),   NOPLY,     NSECT,
3                    MESH,       NGROUP,     NCOREG,    IGRD
```

NELT Scratch variable.

NPAST Scratch variable.

NXSECT Number of input cross sections.

NSPAR Number of spars.

NXSPAR Number of cross sections containing spars.

NYSPC Number of elements along the Y-axis.

NYPTS Number of nodes along the Y-axis.

NNQ NYSPC + 2

NNX NXSCET + 1

LSECT Number of input sections on the bottom and top surfaces.

NPTOT Number of nodes in the shell portion of a f. e. model generated in the HPLATE model option.

NETOT Total number of elements in f. e. model.

MAXPLY Maximum number of plies.

MNNX The maximum of NNX and NXSPAR.

MNNSECT The maximum of LSCET(1) and LSECT(2).

MMWSCT The maximum value in the array MWSCT.

MNXSPC1 MNXSPC + 1

MNXSPC Maximum value in the array NXSPC plus 1.

MNSPDES Maximum value in the array NSPDES.

SPAMTH Minimum spar thickness.

PLYMTH Minimum ply thickness.

MNOPLY Maximum number of plies required at any one node.

MAXNP Total number of nodes.

SPMIN Minimum ply thickness.

SPMAX Maximum wall thickness.

LMAX Number of plies specified for one-half of an unsymmetrical ply lay-up.

NP Number of nodes.

NE Number of elements.

NDES Number of ply descriptions.

IU Number of nodes along X-axis.

JU Number of nodes along Y-axis.

XBEG Initial x coordinate of structure.

XEND Final x coordinate of structure.

NSECT Number of cross sections.

IGRD Grid point input options.

The common block PRINH contains the output control information.

```
 COMMON   /PRINH/   ID(37,5),       IOPT(48,30),    NPRT(2,10),
1                   NPRTS(2,10),  NPRTP(2,10),    NPPLY(2,10),
                  TIMEPN(10),   TIMEMS(10),     TIMEMB(10)
```

ID Array containing the header for the 5 title lines.

IOPT Array storing the program options.

NPRT Displacement output sets.

NPRTS Ply stress output sets.

NPRTP Ply property output sets.

NPPLY Ply property output sets.

TIMEPN Array containing times at which PATRAN result files are to written.

TIMEMS Array containing times at which eigenvalues are to be extracted.

TIMEMB Array containing times at which a critical buckling load is to be extracted.

The common block LCOUNT contains the following variable.

```
 COMMON    /LCOUNT/  LCN1,        LCN2,        LCN3,        LCN4,
1                    LCN5,        LCN6,        LCN7,        LCN8
```

LCN1 Beginning of the temporary storage.

CHAPTER 5

FILE SYSTEM

The following is a summary of the FORTRAN I/O units used in HITCAN. The FORTRAN units are assigned in the subroutine HINIT and reside in the common blocks HUNITS and CMETCN. The unit numbers of these files and the variables they contain are described below.

Note that additional files, not listed here, are used in METCAN, COBSTRAN, and MHOST. These files are assigned in the subroutines BLOCK, CINIT, and BLOCKM. The subroutine BLOCK resides in the METCAN library module, CINIT is in the COBSTRAN library module, and the BLOCKM resides in the MHOST library module. These files are integral to the library modules in which they are used and are not described here. For further information on these files see References 4 and 5.

Code	FORTRAN Unit No.	File Contents
KREAD	5	Input (read) unit
KOUT	6	Output (write) unit
KPAT	18	PATRAN neutral file containing the finite element model
KPOSTF	19	Postprocessing file from MHOST
KSTR	21	Contains fiber, matrix, and interface microstresses (FSIGNN, MSIGNN, and DSIGNN) and microstress rates (FDOTN, MDOTN, and DDOTN)
KLAM	22	Contains flags for material failure LAGF, FLAGM, and FLAGD)

Code	FORTRAN Unit No.	File Contents
KSIG	24	Contains cumulative ply stresses and ply strains (SIGLC and EPSPLC)
KMHOST	27	Input file for MHOST from HITCAN
KPOST2	32	METCAN output file. Contains the current time and fiber material properties (TIME and PFN)
KPOST3	33	METCAN output file. Contains the current time and matrix material properties (TIME and PMN)
KPOST4	34	METCAN output file. Contains the current time and interface material properties (TIME and PDN)
KPOST5	35	METCAN output file. Contains the current time and ply properties (TIME and PPN)
KPOST6	36	METCAN output file. Contains the incremental fiber microstresses (FS1GN)
KPOST7	37	METCAN output file. Contains the incremental matrix microstresses (MSIGN)
KPOST8	38	METCAN output file. Contains the incremental interface microstresses (DSIGN)
KPOST9	39	METCAN output file. Contains the time, incremental ply stresses, incremental ply strains, incremental composite stresses, and incremental composite strains (TIME, STRNS1, STRNS2, STRNS3, LSIGN, EPSPL, and CSTRES)
KPOST0	40	METCAN output file. Contains the current composite properties
KBULK	48	Scratch file for HITCAN
KREFL	49	For input dataset reflection
KCBST	51	Input file for COBSTRAN
KCBFE	54	Post file from COBSTRAN containing the finite element model of a solid structure, using the eight node solid element
KMETCN	67	Input file for METCAN

Code	FORTRAN Unit No.	File Contents
KSTART	68	HITCAN output file containing restart information
KDBANK	70	Databank unit.
KCBFE	75	Post file from COBSTRAN containing the ply lay-up at each node. Used for a solid structure modeled with the eight node solid element
KNODAL	76	PATRAN results file containing the nodal displacements
KTRAN	77	PATRAN results file containing the nodal stresses
KDISP	78	MHOST output file containing the incremental displacements (DISP) and reactions (REACF)
KTRANF	89	MHOST output file containing the global to local transformation matrix
KMHFOR	93	MHOST output file containing the incremental stress resultants (NSB, MSB, and QSB)
KCBOUT	98	Output file from COBSTRAN

CHAPTER 6

HITCAN DICTIONARY

ACCEL	LOGI	Flag for inputing initial accelerations
ACCELI	REAL	Array containing the initial nodal accelerations
ACELIN	REAL	Array containing the initial nodal accelerations
AESID	REAL	Maximum allowable absolute error in residuals
AERMSQ	REAL	Maximum allowable relative error in the root mean square of the strain energy
ALPH(1,I)	REAL	Coefficient of thermal expansion in the x-direction at node I
ALPH(2,I)	REAL	Coefficient of thermal expansion in the y-direction at node I
ALPH(3,I)	REAL	Coefficient of thermal expansion in the z-direction at node I
ALPHA	REAL	Matrix of direction angles between local coordinates and global coordinates
ANG	REAL	Ply orientation angle
ANGULA	LOGI	Flag for centrifugal loading
ANGVEL	REAL	Array containing the angular velocity at each time step
AS	REAL	Axial stiffness matrix
ANVDIF	REAL	Incremental angular velocity
BS	REAL	Bending stiffness matrix
BMODEL	LOGI	Flag for the H3DSOLID model option
BUCK	LOGI	Flag for buckling analysis

CFORCE	LOGI	Flag for nodal loads
CODES(1)	CHAR	Type of ply fiber
CODES(2)	CHAR	Type of ply matrix
CODES(3)	REAL	Ply thickness
CODES(4)	REAL	Void volume ratio
CODES(S)	REAL	Fiber volume ratio
CODES(6)	REAL	Ply orientation angle (degrees)
CONV	LOGI	Flag for global convergence
COFORC	REAL	Array containing the incremental loads
CS	REAL	Coupling stiffness matrix
CSTRES	REAL	Incremental composite stresses
DAMP	LOGI	Flag for damping
DAMPMS	REAL	Coefficient for the mass matrix for Rayleigh damping
DAMPST	REAL	Coefficient for the stiffness matrix for Rayleigh damping
DDOTO	REAL	Array of initial interface microstress rates
DDOTN	REAL	Array of incremental interface microstress rates
DDOTNN	REAL	Array of cumulative interface microstress rates
DDOTI	REAL	Array of interface microstress rates read from the databank
DEN	REAL	Array of nodal densities
DFH	REAL	Array of fiber bundle diameters
DISP	REAL	Array of nodal displacements
DISPI	REAL	Array containing the initial nodal displacements
DISPIN	REAL	Array containing the initial nodal displacements

DISPL	LOGI	Flag for inputing initial displacements
DISVH	REAL	Array of incremental displacements
DPI	REAL	Array of interface properties read from the databank
DSIGO	REAL	Array of initial interface microstresses
DSIGN	REAL	Array of incremental interface microstresses
DSIGNN	REAL	Array of cumulative interface microstresses
DSIGI	REAL	Array of interface microstresses read from the databank
DTI	REAL	Incremental elapsed cpu time
DTVCI	REAL	Array of interface exponents read from the databank
DYNA	LOGI	Flag for dynamic analysis using direct time integration
ECHO	LOGI	Flag for echoing input file
EL3	LOGI	Flag for plane stress element
EL7	LOGI	Flag for 3-D solid element
EL11	LOGI	Flag for plane strain element
EL75	LOGI	Flag for plate element
EPSDC	REAL	Cumulative interface strain
EPSFC	REAL	Cumulative fiber strain
EPSLC	REAL	Cumulative ply strains
EPSMC	REAL	Cumulative matrix strain
EPSPL	REAL	Incremental ply strains
FABRIC	LOGI	Flag for initiating an analysis with fabrication
FPI	REAL	Array of fiber properties read from the databank
FDOTO	REAL	Array of initial fiber microstress rates
FDOTI	REAL	Array of fiber microstress rates read from the databank

FDOTN	REAL	Array of incremental fiber microstress rates
FDOTNN	REAL	Array of cumulative initial fiber microstress rates
FLAGD	REAL	Array of flags for interface property failure
FLAGF	REAL	Array of flags for fiber property failure
FLAGM	REAL	Array of flags for matrix property failure
FSIGO	REAL	Array of initial fiber microstresses
FSIGI	REAL	Array of fiber microstresses read from the databank
FSIGN	REAL	Array of incremental fiber microstresses
FSIGNN	REAL	Array of cumulative fiber microstresses
FTVCI	REAL	Array of fiber exponents read from the databank
GNP	REAL	Array of nodal coordinates and thicknesses
GRIDP1	REAL	First point of the vector required to define the axis of rotation
GRIDP2	REAL	Second point of the vector required to define the axis of rotation
ID	CHAR	Array of five title lines
IGRD	INT	Grid point input options
INCDYN	INT	Number of increments per load increment in the direct time integration
INCR	INT	Current increment
INCREG	INT	Increment number at which eigenvalues are to extracted
INTER	LOGI	Flag for setting an interface
IOPT	INT	Array storing the program options.
IRAMP	INT	Current ramp
IU	INT	Number of nodes along x-axis

JU	INT	Number of nodes along y-axis
KAD	IN	Used to store shell grid point numbers corresponding to the spar internal nodes
KEI	INT	Array of element connectivities
KF	REAL	Apparent fiber volume ratio
KFB	REAL	Actual fiber volume ratio
KM	REAL	Apparent matrix volume ratio
KMB	REAL	Actual matrix volume ratio
KV	REAL	Void volume ratio
L	INT	Number of points in interpolation function
LINEAR	LOGI	Flag for a linear analysis
LCN1	INT	Beginning of the temporary storage.
LDALOW	INT	Number of allowable load steps in METCAN
LDITER	INT	Number of iterations performed in METCAN
LINC	INT	Number of load increments
LMAX	INT	Number of plies specified for opposite half of an unsymmetrical lay-up
LSECT(1)	INT	Number of input cross sections on the top surface (HPLATE model option)
LSECT(2)	INT	Number of input cross sections on the bottom surface (HPLATE model option)
LSECT	INT	Number of input sections of each input plane (S3DSOLID model option)
LSIGN	REAL	Incremental ply stresses
LSMAX	INT	Maximum value in the array LSECT
LSP	INT	Location of the spar on the top and bottom surfaces from the left end

MAXNP	INT	Number of nodal points including both shell and spar nodes
MAXPLY	INT	Number of plies specified for half-thickness at point of maximum thickness
MDES	INT	Array containing ply lay-up for each node
MDOTO	REAL	Array of initial matrix microstress rates
MDOTI	REAL	Array of matrix microstress rates read from the databank
MDOTN	REAL	Array of incremental initial matrix microstress rates
MDOTNN	REAL	Array of cumulative initial matrix microstress rates
MESH	INT	Modeling technique. No longer applicable
MHITER	INT	Maximum number of iterations allowed in MHOST
MITER	INT	Maximum number of iterations per load increment
MMAX	INT	Maximum value in the array MSECT
MMWSCT	INT	The maximum value in the array MWSCT
MNNX	INT	The maximum of NNX and NXSPAR
MNNSECT	INT	The maximum of LSCET(1) and LSECT(2)
MNSPDES	INT	Maximum value in the array NSPDES
MNXSPC	INT	Maximum value in the array NXSPC plus 1
MNXSPC1	INT	MNXSPC + 1
MNOPLY	INT	Maximum number of plies at a node
MODEL	LOGI	Flag for creating a PATRAN neutral file
MODES	LOGI	Flag for modal analysis
MODLIN	LOGI	Flag for reading in finite element model
MSH	REAL	Applied moments at nodal points
MPLY	INT	Ply order using ply designation numbers for one half of the thickness starting at the bottom surface

MPI	REAL	Array of matrix properties read from the databank
MSIGO	REAL	Array of initial matrix microstresses
MSIGI	REAL	Array of matrix microstresses read from the databank
MSIGN	REAL	Array of incremental matrix microstresses
MSIGNN	REAL	Array of cumulative matrix microstresses
MSECT	INT	Number of input points for each cross section
MSTART	INT	Write a restart file after this many load increments
MTVCI	REAL	Array of matrix exponents read from the databank
MWSCT(J,1)	INT	Number of grid points input on the top surface for each cross section J
MWSCT(J,2)	INT	Number of grid points input on the bottom surface for each cross section J
N	INT	Number of points for interpolation
NACC	INT	Number of initial acceleration data sets
NACCEL(1)	INT	Beginning node in an initial acceleration data set
NACCEL(2)	INT	Ending node in an initial acceleration data set
NACCEL(3)	INT	Increment
NBC	INT	Number of boundary conditions
NBOUND(1)	INT	Beginning node in a boundary data set
NBOUND(2)	INT	Ending node in a boundary data set
NBOUND(3)	INT	Increment
NBOUND(4)	INT	Degree-of-freedom which is fixed
NCFOR	INT	Number of concentrated nodal forces
NCFDIR	INT	Direction of the concentrated nodal force
NCFNOD	INT	Array of nodes having a concentrated nodal force

NCFVAL	REAL	Magnitude of the concentrated force at each time step
NDES	INT	Number of ply designations
NDI	INT	Number of direct stress components per node
NDIS	INT	Number of initial displacement data sets
NDISPL(1)	INT	Beginning node in an initial displacement data set
NDISPL(2)	INT	Ending node in an initial displacement data set
NDISPL(3)	INT	Increment
NE	INT	Number of elements
NEIGV	INT	Number of modes to be extracted
NELCHR	INT	Number of element material properties
NELCMP	INT	Number of stress and/or strain components per node
NELCRD	INT	Number of coordinates per node
NELNFR	INT	Number of degrees-of-freedom per node
NELNOD	INT	Number of nodes per element
NELSTR	INT	Number of generalized stresses per node
NESPAR	INT	Number of spar elements
NETOT	INT	Total number of elements
NETT structure	INT	Number of elements through the thickness of a solid modeled by brick elements
NEXX	INT	Number of elements along the X-axis of a solid structure modeled by brick elements
NEYY	INT	Number of elements along the Y-axis of a solid structure modeled by brick elements
NF	INT	Array containing the number of fibers per bundle for each ply
NIPL	INT	Number of input planes

NLEP	INT	Array containing the number of plies at each node
NLS	INT	Number of load steps in METCAN
NMECHC	INT	Number of mechanical cycles
NNQ	INT	NYSPC + 2
NNX	INT	NXSECT + 1
NOPERD	INT	Number of periodic loadings
NORAMP	INT	NTISTP - 1
NOSC	INT	Number of output sections
NOSC1	INT	NOSC + 1
NPERD(1)	INT	Beginning node in a periodic loading data set
NPERD(2)	INT	Ending node in a periodic loading data set
NPERD(3)	INT	Increment
NPERD(4)	INT	Loading type
NPERD(5)	INT	Degree-of-freedom the load is applied to
NP	INT	Number of nodes
NPAST	INT	Last node of each SHELL/SPAR cycle
NPLSET	INT	Number of data sets describing the ply lay-up
NPLY	INT	Ply order using ply designation numbers for one half of the thickness starting at the top surface, except this order is for the opposite half of an unsymmetrical lay-up
NPNOD(1)	INT	Beginning node in a ply lay-up data set
NPNOD(2)	INT	Ending node in a ply lay-up data set
NPNOD(3)	INT	Increment
NPNOD(4)	INT	Number of plies in the lay-up
NPRT(1,I)	INT	Initial node for displacement output for set

NPRT(2,I)	INT	Final node for displacement output for set I
NPRTS(1,I)	INT	Initial node for ply stress output for set I
NPRTS(2,I)	INT	Final node for ply stress output for set I
NPRTP(1,I)	INT	Initial node for ply property output for set
NPRTP(2,I)	INT	Final node for ply property output for set I
NPPLY(1,I)	INT	Initial ply for ply property output for set I
NPPLY(2,I)	INT	Final ply for ply property output for set I
NPRES	INT	Number of edge loads
NPRESS(1)	INT	Beginning element number
NPRESS(2)	INT	Ending element number
NPRESS(3)	INT	Surface with pressure loading
NPPLY(2,I)	INT	Final ply for ply property output for set I
NPTOT	INT	Number of shell elements
NSECT	INT	Number of cross sections
NSH	REAL	Applied membrane loads at nodal points
NSPAR	INT	Number of spars
NSPDES	INT	Spar composite designation
NSUBD	INT	Number of subspace dimensions for eigenvalue extraction
NTEMP	INT	Number of temperature data sets
NTHERC	INT	Number of thermal cycles
NTISTP	INT	Number of time steps
NTNOD(1)	INT	Beginning element number
NTNOD(2)	INT	Ending element number
NTNOD(3)	INT	Increment

NTNOD(4)	INT	Number of plies in this data set
NTR	INT	Number of coordinate transformation data sets
NTRANS(1)	INT	Beginning node in a transformation data set
NTRANS(2)	INT	Ending node in a transformation data set
NTRANS(3)	INT	Increment
NTRANS(4)	INT	Axis about which the nodal coordinates are to be rotated
NUMPLY	INT	Number of plies
NVEL	INT	Number of initial velocity data sets
NVELO(1)	INT	Beginning node in an initial velocity data set
NVELO(2)	INT	Ending node in an initial velocity data set
NVELO(3)	INT	Increment
NWORDT	INT	Total number of words required in HITCAN for in core storage
NXDIV	INT	Array containing the number of elements between cross sections
NXSECT	INT	Number of cross sections
NXSPAR	INT	Number of cross sections containing spars
NXSPC	INT	Number of elements between cross sections
NYPTS	INT	Number of nodal points along Y-axis
NYSPC	INT	Number of elements along Y-axis
PANEL	LOGI	Flag for initiating the PANEL option
PC(1)	REAL	Composite density
PC(2) through PC(11)	REAL	Composite compliance matrix C(1,1),C(1,2),C(1,3),C(2,2),C(2,3), C(3,3),C(4,4),C(5,5),C(6,6)
PC(12)	REAL	Composite thermal expansion coefficient in 11 direction

PC(13)	REAL	Composite thermal expansion coefficient in 22 direction
PC(14)	REAL	Composite thermal expansion coefficient in 33 direction
PC(15)	REAL	Composite thermal conductivity in 11 direction
PC(16)	REAL	Composite thermal conductivity in 22 direction
PC(17)	REAL	Composite thermal conductivity in 33 direction
PC(19)	REAL	Composite modulus in 11 direction
PC(20)	REAL	Composite modulus in 22 direction
PC(21)	REAL	Composite modulus in 33 direction
PC(22)	REAL	Composite shear modulus in 23 direction
PC(23)	REAL	Composite shear modulus in 31 direction
PC(24)	REAL	Composite shear modulus in 12 direction
PC(25)	REAL	Composite poisson's ratio in 12 direction
PC(26)	REAL	Composite poisson's ratio in 21 direction
PC(27)	REAL	Composite poisson's ratio in 13 direction
PC(28)	REAL	Composite poisson's ratio in 31 direction
PC(29)	REAL	Composite poisson's ratio in 23 direction
PC(30)	REAL	Composite poisson's ratio in 32 direction
PDIFU	REAL	Incremental pressure loading on the upper surface
PDIFL	REAL	Incremental pressure loading on the lower surface
PDN	REAL	Current interface material properties
PERDLD(1)	REAL	Period of the load
PERDLD(2)	REAL	Intensity of the load
PERIOD	LOGI	Flag for periodic loading
PERT(1)	REAL	Initial thickness (percent of thickness at each grid point)

PERT(2)	REAL	Final thickness (percent of thickness at each grid point)
PERT(3)	REAL	Initial x-coordinate (percent length)
PERT(4)	REAL	Final x-coordinate (percent length)
PERT(5)	REAL	Initial y-coordinate (percent width)
PERT(6)	REAL	Final y-coordinate (percent width)
PIN(1)	REAL	X coordinate
PIN(2)	REAL	Y coordinate
PIN(3)	REAL	ZU (HSOLID model option) Z coordinate (HPLATE & S3DSOLID model options)
PIN(4)	REAL	Z coordinate (HPLATE model options) Thickness (HPLATE &S3DSOLID model options)
PIN(5)	REAL	ZL
PIN(6)	REAL	Thickness
PINTER	REAL	Thickness of interface as a percent of fiber diameter
PFN	REAL	Current fiber material properties
PL	REAL	Lower nodal pressure
PLH(1)	REAL	Ply void content
PLH(2)	REAL	Ply apparent fiber content
PLH(3)	REAL	Ply actual fiber content
PLH(4)	REAL	Ply apparent matrix content
PLH(5)	REAL	Ply actual matrix content
PLH(6)	REAL	Ply weight density
PLH(7)	REAL	Ply thickness
PLH(8)		Not used
PLH(9)		Not used

PLH(10)	REAL	Distance from bottom of composite to ply centroid
PLH(11)	REAL	Distance from reference plane to ply centroid
PLH(12)	REAL	Angle form structural axis to composite material axis
PLH(13)	REAL	Angle from ply material axis to composite material axis
PLH(14)	REAL	Angle from ply material axis to composite structural axis
PLH(15) through PLH(23)	REAL	Ply stress-strain relations SC11,SC12,SC13,SC22,SC23,SC33,SC44,SC55,SC66
PLH(24)	REAL	Ply coeff. of thermal expansion in 11 direction
PLH(25)	REAL	Ply coeff. of thermal expansion in 22 direction
PLH(26)	REAL	Ply coeff. of thermal expansion in 33 direction
PLH(27)	REAL	Ply heat conductivity in 11 direction
PLH(28)	REAL	Ply heat conductivity in 22 direction
PLH(29)	REAL	Ply heat conductivity in 33 direction
PLH(30)	REAL	Ply heat capacity
PLH(31)	REAL	Ply modulus in 11 direction
PLH(32)	REAL	Ply modulus in 22 direction
PLH(33)	REAL	Ply modulus in 33 direction
PLH(34)	REAL	Ply shear modulus in 23 direction
PLH(35,I)	REAL	Ply shear modulus in 13 direction
PLH(36,I)	REAL	Ply shear modulus in 12 direction
PLH(37) through PLH(42)	REAL	Ply Poisson's ratios NU12,NU21,NU13,NU31,NU23,NU32
PLH(43) through PLH(49)		Not used

PLH(50)	REAL	Ply temperature
PLH(51) through PLH(71)		Not used
PLH(72)	REAL	Ply tensile strength in 11 direction
PLH(73)	REAL	Ply compressive strength in 11 direction
PLH(74)	REAL	Ply tensile strength in 22 direction
PLH(75)	REAL	Ply compressive strength in 22 direction
PLH(76)	REAL	Ply tensile strength in 33 direction
PLH(77)	REAL	Ply compressive strength in 33 direction
PLH(78)	REAL	Ply shear strength in 12 direction
PLH(79)	REAL	Ply shear strength in 23 direction
PLH(80)	REAL	Ply shear strength in 13 direction
PLYMTH	REAL	Minimum ply thickness.
PLYORD	LOGI	Flag for symmetrical ply order
PMN	REAL	Current matrix material properties
PPN	REAL	Current ply material properties from METCAN
PRES	LOGI	Flag for pressure loading
PRESUR	REAL	Incremental pressure loading
PREVAL	REAL	Magnitude of the pressure loading as a function of time
PROFIL	LOGI	Flag for the PROFILE program option card
PU	REAL	Upper nodal pressure
REACF	REAL	Incremental nodal reactions
RERMSQ	REAL	Maximum allowable relative error in the root mean square of the displacement
RESID	REAL	Maximum allowable relative error in the residuals

RESTAR	LOGI	Flag for initiating a restart run
QSH	REAL	Applied shear loads at nodal points
SIGLC	REAL	Cumulative ply stresses
SOLID	LOGI	Flag for the HSOLID model option
SPARS	LOGI	Flag for generating spar elements
SPAMTH	REAL	Minimum spar thickness
SPMIN	REAL	Minimum ply thickness
SPMAX	REAL	Maximum wall thickness
SY(1,I)	REAL	Spar y coordinate from left edge
SY(2,I)	REAL	Spar thickness
TB	REAL	Wall thickness
TEM	LOGI	Flag for temperature loading
TEPR(1,I)	REAL	Temperature on the upper surface
TEPR(2,I)	REAL	Temperature on the lower surface
TEPR(3,I)	REAL	Pressure on the upper surface
TEPR(4,I)	REAL	Pressure on either the lower or inner surface
TEPLY	REAL	Array containing the temperature at each at each node
TIMEE	REAL	Current time
TIMEPN	REAL	Array containing times at which PATRAN plot files are to written
TIMEMS	REAL	Array containing times at which eigenvalues are to be extracted
TIMEMB	REAL	Times at which a critical buckling load is to be extracted
TIMES(1)	REAL	Previous time
TIMES(2)	REAL	Current time

THERMO	REAL	Array containing the temperature of each ply at a given node and a given time step
THRSTN	REAL	Array containing the mid-plane strains and the rotations due to thermal loading
THHF	REAL	Half-thickness at each node
TLAY	REAL	Incremental nodal temperature
TNEW	REAL	Current ply temperature
TOL	REAL	Maximum allowable global error on the displacements
TOLD	REAL	Ply temperature at the previous load increment
TOLMET	REAL	Allowable tolerence in METCAN on local convergence
TOTREA	REAL	Cumulative nodal reactions
TPGNP(1,I)	REAL	Temperature on the upper surface of node I
TPGNP(2,I)	REAL	Temperature on the lower surface of node I
TPGNP(3,I)	REAL	Pressure on the upper surface of node I
TPGNP(4,I)	REAL	Pressure on the lower surface of node I
TRANG	REAL	Angle about which the axis is to be rotated
TRANS	LOGI	Flag for inputing nodal transformations
TT1	REAL	Current elapsed cpu time
TT2	REAL	Previous elapsed cpu time
U	REAL	Array of x values for interpolation
UNSY	LOGI	Flag for an unsymmetrical ply lay-up
V	REAL	Array of y values for interpolation
VELOL	LOGI	Flag for inputing initial velocities
VELOI	REAL	Array containing the initial nodal velocity
VELOIN	REAL	Array containing the initial nodal velocity

VDOT	REAL	Array of first derivative of interpolation
WHAT	CHAR	Four character name
WINDML	LOGI	Flag for the HPLATE model option
X	REAL	X coordinate of cross section
X	REAL	Array containing x coordinates of cross sections
XBEG	REAL	Initial x coordinate
XEND	REAL	Final x coordinate
XINT	REAL	Dummy array for interpolation
XOUT	REAL	Dummy array for interpolation
XX	REAL	Array for which the order is reversed
Y	REAL	Array containing y coordinates
YINT	REAL	Dummy array for interpolation
YOUT	REAL	Dummy array for interpolation
YY	REAL	Array for which order is reversed
Z	REAL	Z mid-thickness coordinate
ZB	REAL	Distance from bottom of composite to ply centroid
ZGC	REAL	Distance from reference plane to ply centroid
ZL	REAL	Z lower coordinate
ZU	REAL	Z upper coordinate

REFERENCES

1. Aiello, R. A., "Composite Blade Structural Analyzer (COBSTRAN) User's Manual", NASA, 1989.

2. Murthy, P. L. N. and Hopkins D. A.,"Metal Matrix Composite Analyzer User's Guide", NASA, 1988.

3. Nakazawa, S., "The MHOST Finite Element Program: 3-D Inelastic Analysis Methods for Hot Section Components", Volume II, NASA, 1989

4. Aiello, R. A., "Composite Blade Structural Analyzer (COBSTRAN) Theoretical/Programmer's Manual", NASA, 1988.

5. Nakazawa, S., "The MHOST Finite Element Program: 3-D Inelastic Analysis Methods for Hot Section Components", Volume III, NASA, 1989

APPENDIX A
EXECUTIVE MODULE

```
C ----------------------------------------------------------------------
C
C    H I T C A N      E X E C U T I V E      M O D U L E
C
C
C    METCAN:     A ........... STORAGE ARRAY
C                MAXLEN ...... SIZE OF ARRAY A
C    HITCAN:     B............ STORAGE ARRAY
C                NASIZE ...... SIZE OF ARRAY B
C    MHOST:      IWORK ....... STORAGE ARRAY
C                ISIZE ....... SIZE OF ARRAY IWORK
C
C ----------------------------------------------------------------------
      PROGRAM            HITCAN
      LOGICAL            PLYORD,  ACCEL,   TEM,     PRES,    VELO,
     1                   PROT,    PREP,    POST,    MODES,   PRTOUT,
     2                   SOLID,   SUPER,   WINDML,  DYNA,    SPARS,
     3                   BIDE,    UNSY,    PROFIL,  HOLLOW,  DISPL,
     4                   DAMP,    PERIOD,  PATRAN,  MHOST,   EZRD,
     5                   ECHO,    RANG,    ANGULA,  CFORCE,  PANEL,
     6                   LINEAR,  RESTAR,  HEAT,    MODEL,   FABRIC,
     7                   CYLIND,  BUCK,    INTER,   BMODEL,  EL3,
     8                   EL7,     EL10,    EL11,    EL75,    TRANS,
     9                   MODLIN
      COMMON   /HUNITS/  KDES1,   KKEI,    KBULK,   KREAD,   KOUT,
     1                   KCBST,   KCBMT,   KPROP,   KCBFE,   KDBANK,
     2                   KTRAN,   KPOST,   KMHFOR,  KREFL,   KPLY,
     3                   KPROPI,  KSTR,    KLAM,    KSIG,    KNODAL,
     4                   KDISP,   KMSTR,   KTRANF,  KCBOUT,  KTEMP,
     5                   KMHOST,  KSTART,  KPOSTF,  KPAT,    KEIGEN
      COMMON   /BOLH/    BIDE,    DYNA,    PRTOUT,  WINDML,  SOLID,
     1                   SUPER,   UNSY,    PROFIL,  PROT,    PREP,
     2                   POST,    MODES,   HOLLOW,  DISPL,   DAMP,
     3                   PLYORD,  ACCEL,   TEM,     VELO,    PRES,
     4                   PERIOD,  PATRAN,  MHOST,   EZRD,    ECHO,
     5                   RANG,    ANGULA,  CFORCE,  PANEL,   LINEAR,
     6                   SPARS,   RESTAR,  HEAT,    MODEL,   FABRIC,
     7                   CYLIND,  BUCK,    INTER,   BMODEL,  EL3,
     8                   EL7,     EL10,    EL11,    EL75,    TRANS,
     9                   MODLIN
      DIMENSION          B ( 400000 )
      DIMENSION          IWORK ( 300000 )
      DIMENSION          A ( 8000 )
      DATA     NASIZE    / 400000 /
      DATA      ISIZE    / 300000 /
      DATA     MAXLEN    / 8000 /
```

```
C
      CALL HINIT
      CALL REFLECH (1,KREAD,KREFL,KOUT)
      CALL LOGOH (KOUT)
      CALL OPTNH (KBULK,KOUT,KREAD)
      IF (ECHO)      CALL REFLECH (2,KREAD,KREFL,KOUT)
      CALL OUTH10 (KOUT)
      CALL ELTYPE (KOUT,EL3,EL7,EL10,EL11,EL75)
      IF (SOLID)     CALL HSOLID (A,MAXLEN,IWORK,ISIZE,B,NASIZE)
      IF (WINDML)    CALL HPLATE (A,MAXLEN,IWORK,ISIZE,B,NASIZE)
      IF (BMODEL)    CALL S3DSOL (A,MAXLEN,IWORK,ISIZE,B,NASIZE)
C
      STOP
      END
```

APPENDIX B

SUBROUTINE HSOLID

```
C ------------------------------------------------------------------
C
C     SUBROUTINE HSOLID
C
C     PURPOSE      THIS SUBROUTINE CONTROLS THE GENERATION OF A SOLID
C                  (THRU-THE-THICKNESS) MODEL. IT CALLS THE APPROPRIATE
C                  SUBROUTINES TO READ IN THE DATA, TO SETUP THE DYNAMIC
C                  ALLOCATION, TO GENERATE THE FINITE ELEMENT MODEL, AND
C                  PERFORM THE STRUCTURAL ANALYSIS.
C
C ------------------------------------------------------------------
      SUBROUTINE HSOLID (A,MAXLEN,IWORK,ISIZE,B,NASIZE)
      LOGICAL             PLYORD,     ACCEL,      TEM,        PRES,
     1                    VELO,       PROT,       PREP,       POST,
     2                    MODES,      PRTOUT,     SOLID,      SUPER,
     3                    WINDML,     DYNA,       SPARS,      BIDE,
     4                    UNSY,       PROFIL,     HOLLOW,     DISPL,
     5                    DAMP,       PERIOD,     PATRAN,     MHOST,
     6                    EZRD,       ECHO,       RANG,       ANGULA,
     7                    CFORCE,     PANEL,      LINEAR,     RESTAR,
     8                    HEAT,       MODEL,      FABRIC,     CYLIND,
     9                    BUCK,       INTER,      BMODEL,     EL3,
     +                    EL7,        EL10,       EL11,       EL75,
     1                    TRANS,      MODLIN
      COMMON    /HUNITS/  KDES1,      KKEI,       KBULK,      KREAD,
     1                    KOUT,       KCBST,      KCBMT,      KPROP,
     2                    KCBFE,      KDBANK,     KTRAN,      KPOST,
     3                    KMHFOR,     KREFL,      KPLY,       KPROPI,
     4                    KSTR,       KLAM,       KSIG,       KNODAL,
     5                    KDISP,      KMSTR,      KTRANF,     KCBOUT,
     6                    KTEMP,      KMHOST,     KSTART,     KPOSTF,
     7                    KPAT,       KEIGEN
      COMMON    /BOLH/    BIDE,       DYNA,       PRTOUT,     WINDML,
     1                    SOLID,      SUPER,      UNSY,       PROFIL,
     2                    PROT,       PREP,       POST,       MODES,
     3                    HOLLOW,     DISPL,      DAMP,       PLYORD,
     4                    ACCEL,      TEM,        VELO,       PRES,
     5                    PERIOD,     PATRAN,     MHOST,      EZRD,
     6                    ECHO,       RANG,       ANGULA,     CFORCE,
     7                    PANEL,      LINEAR,     SPARS,      RESTAR,
     8                    HEAT,       MODEL,      FABRIC,     CYLIND,
     9                    BUCK,       INTER,      BMODEL,     EL3,
     +                    EL7,        EL10,       EL11,       EL75,
     1                    TRANS,      MODLIN
```

```
 COMMON   /POINTS/  NM1,  NM2,  NM3,  NM4,  NM5,  NM6,  NM7,
1                   NM8,  NM9,  NM10, NM11, NM12, NM13, NM14,
2                   NM15, NM16, NM17, NM18, NM19, NM20, NM21,
3                   NM22, NM23, NM24, NM25, NM26, NM27, NM28,
4                   NM29, NM30, NM31, NM32, NM33, NM34, NM35,
5                   NM36, NM37, NM38, NM39, NM40, NM41, NM42,
6                   NM43, NM44, NM45, NM46, NM47, NM48, NM49,
7                   NM50, NM51, NM52, NM53, NM54, NM55, NM56,
8                   NM57, NM58, NM59, NM60, NM61, NM62, NM63
 COMMON   /PINTTS/  NMT61,      NMT62,      NMT63,      NMT64,
1                   NMT65,      NMT66,      NMT67,      NMT68,
2                   NMT69,      NMT70,      NMT71,      NMT72,
3                   NMT73,      NMT74,      NMT75,      NMT76
 COMMON   /IPOINT/  ND1,  ND2,  ND3,  ND4,  ND5,  ND6,  ND7,
1                   ND8,  ND9,  ND10, ND11, ND12, ND13, ND14,
2                   ND15, ND16, ND17, ND18, ND19, ND20, ND21,
3                   ND22, ND23, ND24, ND25, ND26, ND27, ND28,
4                   ND29, ND30, ND31, ND32, ND33, ND34, ND35,
5                   ND36, ND37, ND38, ND39, ND40, ND41, ND42,
6                   ND43, ND44, ND45, ND46, ND47, ND48, ND49,
7                   ND50, ND51, ND52, ND53, ND54, ND55, ND56,
8                   ND57, ND58, ND59, ND60, ND61, ND62, ND63,
9                   ND64, ND65, ND66, ND67, ND68, ND69, ND70,
+                   ND71, ND72, ND73, ND74, ND75, ND76, ND77,
1                   ND78, ND79, ND80, ND81, ND82, ND83, ND84
 COMMON   /VALUEH/  NP,         NE,         NDES,       IMAT,
1                   KSMF,       IU,         JU,         XBEG,
2                   XEND,       NMODE(3),   NOPLY,      NSECT,
3                   MESH,       NGROUP,     NCOREG,     IGRD
 COMMON   /SOLDYN/  SPMIN,      SPMAX,      NGRD,       MNOPLY,
1                   LKDES,      LMAX,       MAXPLY,     PLYMTH
 COMMON   /LOADD/   NMECHC,     NTHERC,     LINC,       MSTART,
1                   MITER,      NTISTP,     NCFOR,      TOL,
2                   NEIGV,      NSUBD,      INCREG,     NBC,
3                   MHITER,     RESID,      AESID,      RERMSQ,
4                   AERMSQ,     NPRES,      NTEMP,      NTR,
5                   NBCON,      NTCON,      NUMPLY,     NOPERD,
6                   DAMPMS,     DAMPST,     NDIS,       NVEL,
7                   NACC,       INCDYN
 COMMON   /LCOUNT/  LCN1,       LCN2,       LCN3,       LCN4,
1                   LCN5,       LCN6,       LCN7,       LCN8
 DIMENSION          GRIDP1(3),  GRIDP2(3),  HOLD(10)
 DIMENSION          IWORK ( 1 )
 DIMENSION          A ( 1 )
 DIMENSION          B ( 1 )
```

```
C
      TT1 = 0.0
      CALL GETIME (TT1,TT2,DTI)
C
C     READ IN PARAMETER DATA
C
      CALL RINPAR (KREAD,KOUT,NDES,NSECT,IGRD,IU,JU,MESH,XBEG,XEND,
     1             IMAT,NXSPAR,NSPAR,NYSPC,ALPHA,LSECT,NIPL,NOSC,NEYY,
     2             NETT,MAXNP,NETOT,MAXPLY,LMAX,NPLSET,PINTER)
C
C     SET THE INITIAL POINTERS FOR DYNAMIC ALLOCATION
C
      CALL SCPEM1 (NSECT)
C
C     READ IN THE FIRST SET OF MODEL DATA
C
      CALL SREAD1 (KREAD,NSECT,B(NM1))
C
C     SET ADDITIONIAL POINTERS FOR DYNAMIC ALLOCATION
C
      CALL SCPEM2 (UNSY,HOLLOW,PLYORD,TEM,PRES,ANGULA,CFORCE,DISPL,
     1             VELO,ACCEL,NGRD,LMAX,MAXPLY,NPM,B(NM1),MMSECT,N,N1,
     2             N2,N3,N4,NSTART)
C
C     READ IN SURFACE GEOMETRY
C
      CALL SREAD2 (EL3,EL10,EL11,EL75,KREAD,KOUT,N3,B(NM1),B(NM2),
     1             B(NM6))
C
C     READ IN THE DESCRIPTION OF THE PLIES AND THE PLY LAY-UP
C
      CALL RINPLY (PLYORD,UNSY,EL3,EL10,EL11,EL75,KREAD,KOUT,NDES,LMAX,
     1             MAXPLY,B(NM4),B(NM5),B(NM8),B(NM9))
C
C     DETERMINE THE MAXIMUM NO. OF PLIES IN THE MODEL
C
      CALL DETNPL (PLYORD,UNSY,N3,B(NM1),B(NM2),B(NM5))
C
C     SET POINTERS REQUIRED FOR ANALYSIS
C
      CALL INITP3 (KOUT,CFORCE,TRANS,MODES,SUPER,DYNA,BUCK,NSTART,NWORD,
     1             NWORDT,NP,NE,MNOPLY)
C
C     SET POINTERS FOR THE TEMPORARY WORDS
C
      CALL SCTEMW (KOUT,NP,N,N1,N2,N3,N4,NWORDT)
```

```
C
C      CHECK FOR ENOUGH MEMORY. IF EXCEEDS NASIZE SET IN THE MAIN
C      PROGRAM, TERMINATE THE RUN.
C
       CALL CHCKMS (KOUT,NASIZE,NSTART,NWORD,NWORDT,JSSTOP)
       IF (JSSTOP .EQ. 1) GO TO 10
C
C      READ IN TEMPERATURE AND PRESSURE DATA
C
       CALL SREAD3 (TEM,PRES,EL3,EL7,EL10,EL11,EL75,KREAD,KOUT,N3,N4,
      1             B(NM1),B(NM28),B(NM35),B(NM36),B(NMT61),B(NMT70),
      2             B(NMT71))
C
C      READ IN ROTATIONAL VELOCITY AND CONCENTRATED FORCE DATA
C
       CALL RINFOR (ANGULA,CFORCE,KREAD,KOUT,GRIDP1,GRIDP2,B(NM28),
      1             B(NM29),B(NM30),B(NM31),B(NM32))
C
C      READ IN DATA FOR DYNMIC ANALYSIS
C
       IF (DYNA) CALL DYNMIN (EL3,EL7,EL10,EL11,EL75,DISPL,VELO,ACCEL,
      1                       PERIOD,KREAD,KOUT,NP,B(NM39),B(NM40),
      2                       B(NM41),B(NM42),B(NM43))
C
C      READ IN BOUNDARY CONDITIONS
C
       CALL RIBOUN (TRANS,KREAD,KOUT,NBC,NTR,NBCON,NTCON,B(NM34),
      1             B(NM37),B(NM38))
C
C      READ IN OUTPUT CONTROL DATA
C
       CALL OUTPTC (MODES,BUCK,KREAD,KOUT)
       CALL GETIME (TT1,TT2,DTI)
       DT1 = DTI
```

```
C
C     CALL THE SUBROUTINE PREPSO TO DETERMINE THE SHELL MODEL AND
C     READ IN THE INITIAL MICROMECHANICS MATERIAL PROPERTIES
C
      CALL PREPSO (PLYORD,UNSY,HOLLOW,TEM,PRES,KOUT,KDBANK,KPROPI,KDES1,
     1             KGNP,NTISTP,NTEMP,NUMPLY,N,N2,N3,N4,MMSECT,B(NM1),
     2             B(NM2),B(NM3),B(NM4),B(NM5),B(NM6),B(NM7),B(NM8),
     3             B(NM9),B(NM10),B(NM11),B(NM12),B(NM13),B(NM14),
     4             B(NM15),B(NM16),B(NM17),B(NM18),B(NM19),B(NM20),
     5             B(NM21),B(NM22),B(NM23),B(NM25),B(NM26),B(NM28),
     6             B(NM55),B(NM56),B(NM57),B(NM58),B(NM59),B(NM60),
     7             B(NM61),B(ND43),B(ND44),B(ND45),B(ND46),B(ND47),
     8             B(ND48),B(ND49),B(NMT61),B(NMT62),B(NMT63),B(NMT64),
     9             B(NMT65),B(NMT66),B(NMT67),B(NMT68),B(NMT69),
     +             B(NMT70),B(NMT71),B(NMT72),B(NMT73))
C
C     WRITE TO OUTPUT FILE THE PLY STACKUP AT EACH NODE
C
      CALL OTPUTH (3,SOLID,WINDML,BMODEL,TEM,PRES,EL3,EL7,EL10,EL11,
     1             EL75,KBULK,KOUT,MNOPLY,NSECT,NSPAR,NXSPAR,NPTOT,
     2             NETOT,B(NM28),B(ND43),B(ND44),B(ND45),B(ND46),
     3             B(ND47),B(NM55),B(NM56),B(ND48),B(ND42))
C
C     WRITE TO OUTPUT FILE THE FINITE ELEMENT MODEL
C
      CALL OTPUTH (7,SOLID,WINDML,BMODEL,TEM,PRES,EL3,EL7,EL10,EL11,
     1             EL75,KBULK,KOUT,MNOPLY,NSECT,NSPAR,NXSPAR,NPTOT,
     2             NETOT,B(NM28),B(ND43),B(ND44),B(ND45),B(ND46),
     3             B(ND47),B(NM55),B(NM56),B(ND48),B(ND42))
C
C     CALL THE SUBROUTINE WNEUT TO GENERATE A PATRAN NEUTRAL FILE
C
      CALL WNEUT (KPAT,4,NP,NE,B(ND43),B(ND44))
      CALL GETIME (TT1,TT2,DTI)
      DT2 = DTI
      IF (MODEL) GO TO 10
```

```
C
C     CALL THE SUBROUTINE NLINER TO PERFORM THE STATIC ANALYSIS
C
      IF (.NOT. DYNA) THEN
      CALL NLINER (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,NP,NP,NE,
     1             MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,B(NM5),B(NM11),
     2             B(NM12),B(NM13),B(NM14),B(NM15),B(NM16),B(NM17),
     3             B(NM18),B(NM19),B(NM20),B(NM21),B(NM22),B(NM23),
     4             B(NM25),B(NM28),B(NM29),B(NM30),B(NM31),B(NM32),
     5             B(NM34),B(NM35),B(NM36),B(NM37),B(NM38),B(ND42),
     6             B(ND43),B(ND44),B(ND45),B(ND46),B(ND47),B(ND48),
     7             B(ND49),B(ND50),B(ND51),B(ND52),B(ND53),B(ND54),
     8             B(ND55),B(ND56),B(ND57),B(ND58),B(ND59),B(ND60),
     9             B(ND61),B(ND62),B(ND63),B(ND64),B(ND65),B(ND66),
     +             B(ND67),B(ND68),B(ND69),B(ND70),B(ND71),B(ND72),
     1             B(ND73),B(ND74),B(ND75),B(ND76))
C
C     CALL THE SUBROUTINE DYNMCH TO PERFORM THE DYNAMIC ANALYSIS
C
      ELSE
      CALL DYNMCH (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,NP,NP,NE,
     1             MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,B(NM5),B(NM11),
     2             B(NM12),B(NM13),B(NM14),B(NM15),B(NM16),B(NM17),
     3             B(NM18),B(NM19),B(NM20),B(NM21),B(NM22),B(NM23),
     4             B(NM25),B(NM28),B(NM29),B(NM30),B(NM31),B(NM32),
     5             B(NM34),B(NM35),B(NM36),B(NM37),B(NM38),B(NM39),
     6             B(NM40),B(NM41),B(NM42),B(NM43),B(ND42),B(ND43),
     7             B(ND44),B(ND45),B(ND46),B(ND47),B(ND48),B(ND49),
     8             B(ND50),B(ND51),B(ND52),B(ND53),B(ND54),B(ND55),
     9             B(ND56),B(ND57),B(ND58),B(ND59),B(ND60),B(ND61),
     +             B(ND62),B(ND63),B(ND64),B(ND65),B(ND66),B(ND67),
     1             B(ND68),B(ND69),B(ND70),B(ND71),B(ND72),B(ND73),
     2             B(ND74),B(ND75),B(ND76),B(ND77),B(ND78),B(ND79),
     3             B(ND80),B(ND81))
      END IF
C
      CALL GETIME (TT1,TT2,DTI)
      WRITE (KOUT,20) DT1, DT2, TT1
   10 CONTINUE
C
   20 FORMAT (/' TIME REQUIRED TO : ',
     1        'READ IN DATA              ',18X,F10.3,' SEC.',/,20X,
     2        'DO PREPROCESSING          ',18X,F10.3,' SEC.',/,20X,
     3        'TOTAL TIME FOR THIS RUN   ',18X,F10.3,' SEC.')
C
      RETURN
      END
```

APPENDIX C

SUBROUTINE HPLATE

```
C --------------------------------------------------------------------
C
C     SUBROUTINE HPLATE
C
C     PURPOSE    THIS SUBROUTINE CONTROLS THE GENERATION OF BUILT-UP
C                STRUCTURES. IT CALLS THE APPROPRIATE SUBROUTINES TO
C                READ IN THE DATA, TO SETUP THE DYNAMIC ALLOCATION,
C                TO GENERATE THE FINITE ELEMENT MODEL, AND PERFORM THE
C                STRUCTURAL ANALYSIS.
C
C --------------------------------------------------------------------
      SUBROUTINE HPLATE (A,MAXLEN,IWORK,ISIZE,B,NASIZE)
      LOGICAL             PLYORD,     ACCEL,      TEM,        PRES,
     1                    VELO,       PROT,       PREP,       POST,
     2                    MODES,      PRTOUT,     SOLID,      SUPER,
     3                    WINDML,     DYNA,       SPARS,      BIDE,
     4                    UNSY,       PROFIL,     HOLLOW,     DISPL,
     5                    DAMP,       PERIOD,     PATRAN,     MHOST,
     6                    EZRD,       ECHO,       RANG,       ANGULA,
     7                    CFORCE,     PANEL,      LINEAR,     RESTAR,
     8                    HEAT,       MODEL,      FABRIC,     CYLIND,
     9                    BUCK,       INTER,      BMODEL,     EL3,
     +                    EL7,        EL10,       EL11,       EL75,
     1                    TRANS,      MODLIN
      COMMON    /HUNITS/  KDES1,      KKEI,       KBULK,      KREAD,
     1                    KOUT,       KCBST,      KCBMT,      KPROP,
     2                    KCBFE,      KDBANK,     KTRAN,      KPOST,
     3                    KMHFOR,     KREFL,      KPLY,       KPROPI,
     4                    KSTR,       KLAM,       KSIG,       KNODAL,
     5                    KDISP,      KMSTR,      KTRANF,     KCBOUT,
     6                    KTEMP,      KMHOST,     KSTART,     KPOSTF,
     7                    KPAT,       KEIGEN
      COMMON    /BOLH/    BIDE,       DYNA,       PRTOUT,     WINDML,
     1                    SOLID,      SUPER,      UNSY,       PROFIL,
     2                    PROT,       PREP,       POST,       MODES,
     3                    HOLLOW,     DISPL,      DAMP,       PLYORD,
     4                    ACCEL,      TEM,        VELO,       PRES,
     5                    PERIOD,     PATRAN,     MHOST,      EZRD,
     6                    ECHO,       RANG,       ANGULA,     CFORCE,
     7                    PANEL,      LINEAR,     SPARS,      RESTAR,
     8                    HEAT,       MODEL,      FABRIC,     CYLIND,
     9                    BUCK,       INTER,      BMODEL,     EL3,
     +                    EL7,        EL10,       EL11,       EL75,
     1                    TRANS,      MODLIN
```

```
      COMMON    /LCOUNT/  LCN1,          LCN2,          LCN3,          LCN4,
     1                    LCN5,          LCN6,          LCN7,          LCN8
      COMMON    /LOADD/   NMECHC,        NTHERC,        LINC,          MSTART,
     1                    MITER,         NTISTP,        NCFOR,         TOL,
     2                    NEIGV,         NSUBD,         INCREG,        NBC,
     3                    MHITER,        RESID,         AESID,         RERMSQ,
     4                    AERMSQ,        NPRES,         NTEMP,         NTR,
     5                    NBCON,         NTCON,         NUMPLY,        NOPERD,
     6                    DAMPMS,        DAMPST,        NDIS,          NVEL,
     7                    NACC,          INCDYN
      DIMENSION           NPLY(1),       GRIDP1(3),     GRIDP2(3)
      DIMENSION           IWORK ( 1 )
      DIMENSION           A ( 1 )
      DIMENSION           B ( 1 )
C
      TT1 = 0.0
      MESH = 4
      CALL GETIME (TT1,TT2,DTI)
C
C     READ IN PARAMETER DATA
C
      CALL RINPAR (KREAD,KOUT,NDES,NXSECT,IGRD,IU,JU,MESH,XBEG,XEND,
     1             IMAT,NXSPAR,NSPAR,NYSPC,ALPHA,LSECT,NIPL,NOSC,NEYY,
     2             NETT,MAXNP,NETOT,MAXPLY,LMAX,NPLSET,PINTER)
C
C     SET THE INITIAL POINTERS FOR DYNAMIC ALLOCATION
C
      CALL HCPEM1 (PLYORD,ANGULA,CFORCE,PRES,PANEL,KOUT,MPLY,NPM)
C
C     READ IN SPAR DESCRIPTIONS
C
      CALL RINSPA (KREAD,KOUT,NSPAR,NNX,B(NH1),B(NH2),B(NH3),B(NH4))
C
C     READ IN FIRST SET OF MODEL DATA
C
      CALL HREAD1 (KREAD,KOUT,LSECT,MNNSECT,B(NH33))
C
C     DETERMINE SEVERAL PARAMETERS REQUIRED FOR DYNAMIC ALLOCATION
C
      CALL DETPAR (PLYORD,PANEL,KOUT,B(NH1),B(NH2),B(NH4),B(NH5),B(NH6),
     1             B(NH22),B(NH33))
```

```
C
C     SET ADDITIONAL POINTERS FOR DYNAMIC ALLOCATION
C
      CALL HCPEM2 (CFORCE,PRES,TRANS,NSTART)
C
C     READ IN SURFACE GEOMETRY
C
      CALL HREAD2 (KREAD,KOUT,B(NH33),B(NH34))
C
C     READ IN THE DESCRIPTION OF THE PLIES AND THE PLY LAY-UP
C
      CALL RINPLY (PLYORD,UNSY,EL3,EL10,EL11,EL75,KREAD,KOUT,NDES,LMAX,
     1             MAXPLY,B(NH5),B(NH6),B(NH7),NPLY)
C
C     DETERMINE THE MAXIMUM NUMBER OF PLIES AT A NODE
C
      CALL HDNPLY (B(NH4),B(NH6),B(NH33),B(NH34))
C
C     SET POINTERS REQUIRED FOR THE ANALYSIS
C
      CALL INITP3 (KOUT,CFORCE,TRANS,MODES,SUPER,DYNA,BUCK,NSTART,NWORD,
     1             NWORDT,MAXNP,NETOT,MNOPLY)
C
C     SET POINTERS FOR THE TEMPORARY WORDS
C
      CALL HCTEMW (CFORCE,TRANS,NWORDT)
C
C     CHECK FOR ENOUGH MEMORY. IF EXCEEDS NASIZE SET IN THE MAIN
C     PROGRAM, TERMINATE THE RUN.
C
      CALL CHCKMS (KOUT,NASIZE,NSTART,NWORD,NWORDT,JSSTOP)
      IF (JSSTOP .EQ. 1) GO TO 5
C
C     READ IN LOAD DATA AND BOUNDARY CONDITIONS
C
      CALL HREAD3 (PROFIL,PLYORD,TEM,PRES,KREAD,KOUT,B(NH8),B(NH27),
     1             B(NH33),B(NH36),B(NH37),B(NH51T),B(NH52T))
C
C     READ CONCENTRATED FORCES & ROTATIONAL SPEED
C
      CALL RINFOR (ANGULA,CFORCE,KREAD,KOUT,GRIDP1,GRIDP2,B(NH8),
     1             B(NH28),B(NH29),B(NH30),B(NH31))
```

```
C
C     READ IN DATA FOR DYNMIC ANALYSIS
C
      IF (DYNA) CALL DYNMIN (EL3,EL7,EL10,EL11,EL75,DISPL,VELO,ACCEL,
     1                       PERIOD,KREAD,KOUT,MAXNP,B(NH42),B(NH43),
     2                       B(NH44),B(NH45),B(NH46))
C
C     READ IN BOUNDARY CONDITIONS
C
      CALL RIBOUN (TRANS,KREAD,KOUT,NBC,NTR,NBCON,NTCON,B(NH9),
     1             B(NH41),B(NH42))
C
C     READ IN OUTPUT CONTROL DATA
C
      CALL OUTPTC (MODES,BUCK,KREAD,KOUT)
      CALL GETIME (TT1,TT2,DTI)
      DT1 = DTI
C
C     CALL THE SUBROUTINE PREPSH TO DETERMINE THE SHELL MODEL
C
      CALL PREPSH (SPARS,PLYORD,PRES,TEM,PROFIL,PANEL,CYLIND,B(NH1),
     1             B(NH2),B(NH3),B(NH4),B(NH5),B(NH6),B(NH7),B(NH8),
     2             B(NH9),B(NH10),B(NH11),B(NH12),B(NH13),B(NH14),
     3             B(NH15),B(NH16),B(NH17),B(NH18),B(NH19),B(NH20),
     4             B(NH21),B(NH22),B(NH23),B(NH25),B(NH26),B(NH33),
     5             B(NH34),B(NH35),B(NH36),B(ND42),B(ND43),B(ND44),
     6             B(ND45),B(ND46),B(ND47),B(ND48),B(ND49),B(NH36T),
     7             B(NH37T),B(NH38T),B(NH39T),B(NH40T),B(NH41T),
     8             B(NH42T),B(NH43T),B(NH44T),B(NH45T),B(NH46T),
     9             B(NH47T),B(NH48T),B(NH49T),B(NH50T),B(NH51T),
     +             B(NH52T),B(NH53T),B(NH54T))
C
C     CALL THE SUBROUTINE PREPSP TO DETERMINE THE SPAR MODEL
C
      IF (NSPAR .NE. 0) THEN
      CALL PREPSP (SPARS,PLYORD,PRES,TEM,PROT,PANEL,KOUT,B(NH1),B(NH2),
     1             B(NH3),B(NH4),B(NH5),B(NH6),B(NH7),B(NH8),B(NH22),
     2             B(NH23),B(NH25),B(NH26),B(ND42),B(ND43),B(ND44),
     3             B(ND45),B(ND46),B(ND47),B(ND48),B(ND49),B(NH51T),
     4             B(NH52T),B(NH53T),B(NH54T))
      END IF
C
C     CALL THE SUBROUTINE WNEUT TO GENERATE A PATRAN NEUTRAL FILE
C
      CALL WNEUT (KPAT,4,MAXNP,NETOT,B(ND43),B(ND44))
```

```
      CALL GETIME (TT1,TT2,DTI)
      DT2 = DTI
      IF (MODEL) GO TO 5
C
C     CALL THE SUBROUTINE NLINER TO PERFORM THE ANALYSIS
C
      IF (.NOT. DYNA) THEN
      CALL NLINER (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,MAXNP,NPTOT,
     1             NETOT,MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,B(NH6),
     2             B(NH10),B(NH11),B(NH12),B(NH13),B(NH14),B(NH15),
     3             B(NH16),B(NH17),B(NH18),B(NH19),B(NH20),B(NH21),
     4             B(NH23),B(NH25),B(NH8),B(NH28),B(NH29),B(NH30),
     5             B(NH31),B(NH9),B(NH27),B(NH37),B(NH41),B(NH42),
     6             B(ND42),B(ND43),B(ND44),B(ND45),B(ND46),B(ND47),
     7             B(ND48),B(ND49),B(ND50),B(ND51),B(ND52),B(ND53),
     8             B(ND54),B(ND55),B(ND56),B(ND57),B(ND58),B(ND59),
     9             B(ND60),B(ND61),B(ND62),B(ND63),B(ND64),B(ND65),
     +             B(ND66),B(ND67),B(ND68),B(ND69),B(ND70),B(ND71),
     1             B(ND72),B(ND73),B(ND74),B(ND75),B(ND76))
C
C     CALL THE SUBROUTINE DYNMCH TO PERFORM THE DYNAMIC ANALYSIS
C
      ELSE
      CALL DYNMCH (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,MAXNP,NPTOT,
     1             NETOT,MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,B(NH6),
     2             B(NH10),B(NH11),B(NH12),B(NH13),B(NH14),B(NH15),
     3             B(NH16),B(NH17),B(NH18),B(NH19),B(NH20),B(NH21),
     4             B(NH23),B(NH25),B(NH8),B(NH28),B(NH29),B(NH30),
     5             B(NH31),B(NH9),B(NH27),B(NH37),B(NH40),B(NH41),
     6             B(NH42),B(NH43),B(NH44),B(NH45),B(NH46),B(ND42),
     7             B(ND43),B(ND44),B(ND45),B(ND46),B(ND47),B(ND48),
     8             B(ND49),B(ND50),B(ND51),B(ND52),B(ND53),B(ND54),
     9             B(ND55),B(ND56),B(ND57),B(ND58),B(ND59),B(ND60),
     +             B(ND61),B(ND62),B(ND63),B(ND64),B(ND65),B(ND66),
     1             B(ND67),B(ND68),B(ND69),B(ND70),B(ND71),B(ND72),
     2             B(ND73),B(ND74),B(ND75),B(ND76),B(ND77),B(ND78),
     3             B(ND79),B(ND80),B(ND81))
```

```
      END IF
C
      CALL GETIME (TT1,TT2,DTI)
      WRITE (KOUT,20) DT1, DT2, TT1
    5 CONTINUE
C
   20 FORMAT (/' TIME REQUIRED TO : ',
     1        'READ IN DATA                 ',18X,F10.3,' SEC.',/,20X,
     2        'DO PREPROCESSING             ',18X,F10.3,' SEC.',/,20X,
     3        'TOTAL TIME FOR THIS RUN      ',18X,F10.3,' SEC.')
C
      RETURN
      END
```

APPENDIX D

SUBROUTINE S3DSOL

```
C -----------------------------------------------------------------
C
C     SUBROUTINE S3DSOL
C
C     PURPOSE    THIS SUBROUTINE CONTROLS THE GENERATION OF STRUCTURES
C                USING SOLID ELEMENTS. IT CALLS THE APPROPRIATE SUBR.
C                TO READ IN THE DATA, TO SETUP THE DYNAMIC ALLOCATION,
C                TO GENERATE THE FINITE ELEMENT MODEL, AND PERFORM THE
C                STRUCTURAL ANALYSIS.
C
C -----------------------------------------------------------------
      SUBROUTINE S3DSOL (A,MAXLEN,IWORK,ISIZE,B,NASIZE)
      LOGICAL             PLYORD,  ACCEL,   TEM,     PRES,    VELO,
     1                    PROT,    PREP,    POST,    MODES,   PRTOUT,
     2                    SOLID,   SUPER,   WINDML,  DYNA,    SPARS,
     3                    BIDE,    UNSY,    PROFIL,  HOLLOW,  DISPL,
     4                    DAMP,    PERIOD,  PATRAN,  MHOST,   EZRD,
     5                    ECHO,    RANG,    ANGULA,  CFORCE,  PANEL,
     6                    LINEAR,  RESTAR,  HEAT,    MODEL,   FABRIC,
     7                    CYLIND,  BUCK,    INTER,   BMODEL,  EL3,
     8                    EL7,     EL10,    EL11,    EL75,    TRANS,
     9                    MODLIN
      COMMON    /HUNITS/  KDES1,   KKEI,    KBULK,   KREAD,   KOUT,
     1                    KCBST,   KCBMT,   KPROP,   KCBFE,   KDBANK,
     2                    KTRAN,   KPOST,   KMHFOR,  KREFL,   KPLY,
     3                    KPROPI,  KSTR,    KLAM,    KSIG,    KNODAL,
     4                    KDISP,   KMSTR,   KTRANF,  KCBOUT,  KTEMP,
     5                    KMHOST,  KSTART,  KPOSTF,  KPAT,    KEIGEN
      COMMON    /BOLH/    BIDE,    DYNA,    PRTOUT,  WINDML,  SOLID,
     1                    SUPER,   UNSY,    PROFIL,  PROT,    PREP,
     2                    POST,    MODES,   HOLLOW,  DISPL,   DAMP,
     3                    PLYORD,  ACCEL,   TEM,     VELO,    PRES,
     4                    PERIOD,  PATRAN,  MHOST,   EZRD,    ECHO,
     5                    RANG,    ANGULA,  CFORCE,  PANEL,   LINEAR,
     6                    SPARS,   RESTAR,  HEAT,    MODEL,   FABRIC,
     7                    CYLIND,  BUCK,    INTER,   BMODEL,  EL3,
     8                    EL7,     EL10,    EL11,    EL75,    TRANS,
     9                    MODLIN
      COMMON    /INTPH/   TREF,    BFT,     BFC,     BCS,     FDC,
     1                    FDK
      COMMON    /LOADD/   NMECHC,  NTHERC,  LINC,    MSTART,  MITER,
     1                    NTISTP,  NCFOR,   TOL,     NEIGV,   NSUBD,
     2                    INCREG,  NBC,     MHITER,  RESID,   AESID,
     3                    RERMSQ,  AERMSQ,  NPRES,   NTEMP,   NTR,
     4                    NBCON,   NTCON,   NUMPLY,  NOPERD,  DAMPMS,
     5                    DAMPST,  NDIS,    NVEL,    NACC,    INCDYN
```

```
      COMMON    /POINTB/  NB1,    NB2,    NB3,    NB4,    NB5,
     1                    NB6,    NB7,    NB8,    NB9,    NB10,
     2                    NB11,   NB12,   NB13,   NB14,   NB15,
     3                    NB16,   NB17,   NB18,   NB19,   NB20,
     4                    NB21,   NB22,   NB23,   NB24,   NB25,
     5                    NB26,   NB27,   NB28,   NB29,   NB30,
     6                    NB31,   NB32,   NB33,   NB34,   NB35,
     7                    NB36,   NB37,   NB38,   NB39,   NB40,
     8                    NB41,   NB42,   NB43,   NB44,   NB45,
     9                    NB46,   NB47,   NB48,   NB49,   NB50,
     +                    NB51,   NB52,   NB53,   NB54,   NB55
      COMMON    /IPOINT/  ND1,    ND2,    ND3,    ND4,    ND5,
     1                    ND6,    ND7,    ND8,    ND9,    ND10,
     2                    ND11,   ND12,   ND13,   ND14,   ND15,
     3                    ND16,   ND17,   ND18,   ND19,   ND20,
     4                    ND21,   ND22,   ND23,   ND24,   ND25,
     5                    ND26,   ND27,   ND28,   ND29,   ND30,
     6                    ND31,   ND32,   ND33,   ND34,   ND35,
     7                    ND36,   ND37,   ND38,   ND39,   ND40,
     8                    ND41,   ND42,   ND43,   ND44,   ND45,
     9                    ND46,   ND47,   ND48,   ND49,   ND50,
     +                    ND51,   ND52,   ND53,   ND54,   ND55,
     1                    ND56,   ND57,   ND58,   ND59,   ND60,
     2                    ND61,   ND62,   ND63,   ND64,   ND65,
     3                    ND66,   ND67,   ND68,   ND69,   ND70,
     4                    ND71,   ND72,   ND73,   ND74,   ND75,
     5                    ND76,   ND77,   ND78,   ND79,   ND80,
     6                    ND81,   ND82,   ND83,   ND84
      DIMENSION           GRIDP1(3),      GRIDP2(3)
      DIMENSION           IWORK ( 1 )
      DIMENSION           A ( 1 )
      DIMENSION           B ( 1 )
C
C     READ IN PARAMETER DATA
C
      TT1 = 0.0
      CALL GETIME (TT1,TT2,DTI)
      CALL RINPAR (KREAD,KOUT,NDES,NSECT,IGRD,IU,JU,MESH,XBEG,XEND,
     1             IMAT,NXSPAR,NSPAR,NYSPC,ALPHA,LSECT,NIPL,NOSC,NEYY,
     2             NETT,MAXNP,NETOT,MAXPLY,LMAX,NPLSET,PINTER)
C
C     SET THE INITIAL POINTERS FOR THE DYNAMIC ALLOCATION
C
      CALL BCPEM1 (NOSC,NIPL,NOSC1)
```

```
C
C     READ IN THE FIRST SET OF DATA DESCRIBING THE SURFACE GEOMETRY
C
      IF (.NOT. MODLIN) CALL BREAD1 (KREAD,KOUT,NIPL,NOSC,NOSC1,B(NB1),
     1                               B(NB2),B(NB3))
C
C     SET ADDITIONIAL POINTERS FOR DYNAMIC ALLOCATION
C
      CALL BCPEM2 (MODLIN,NOSC,NIPL,NOSC1,LSMAX,NEXX,NEYY,NETT,NPXX,
     1             NPYY,NPTT,MAXNP,NETOT,B(NB1),B(NB3))
C
C     READ IN ADDITIONIAL DATA DESCRIBING THE SURFACE GEOMETRY
C
      IF (.NOT. MODLIN) CALL BREAD2 (KREAD,KOUT,NIPL,LSMAX,NOSC1,MMAX,
     1                               B(NB3),B(NB4))
C
C     SET ADDITIONIAL POINTERS FOR DYNAMIC ALLOCATION
C
      CALL BCPEM3 (PLYORD,UNSY,CFORCE,ANGULA,PRES,TRANS,NDES,NOSC,NIPL,
     1             MAXPLY,LMAX,NCFOR,NTISTP,NPRES,NBC,NOSC1,LSMAX,MMAX,
     2             MAXNP,NETOT,MNOPLY,NTR,NOPERD,NWORDT,NSTART,B(NB3),
     3             B(NB4))
C
C     SET POINTERS REQUIRED FOR ANALYSIS
C
      CALL INITP3 (KOUT,CFORCE,TRANS,MODES,SUPER,DYNA,BUCK,NSTART,NWORD,
     1             NWORDT,MAXNP,NETOT,MNOPLY)
C
C     CHECK FOR ENOUGH MEMORY. IF EXCEEDS NASIZE SET IN THE MAIN
C     PROGRAM, TERMINATE THE RUN.
C
      CALL CHCKMS (KOUT,NASIZE,NSTART,NWORD,NWORDT,JSSTOP)
      IF (JSSTOP .EQ. 1) GO TO 5
C
C     READ IN FINITE ELEMENT MODEL
C
      IF (MODLIN) CALL READML (KREAD,MAXNP,NETOT,B(ND43),B(ND44))
C
C     READ IN SURFACE GEOMETRY
C
      IF (.NOT. MODLIN) CALL BREAD3 (KREAD,KOUT,NIPL,LSMAX,MMAX,B(NB3),
     1                               B(NB4),B(NB5))
```

```
C
C     READ IN THE DESCRIPTION OF THE PLIES
C
      CALL RINPLY (PLYORD,UNSY,EL3,EL10,EL11,EL75,KREAD,KOUT,NDES,LMAX,
     1             MAXPLY,B(NB7),B(NB8),B(NB9),B(NB10))
C
C     READ IN THE PLY LAY-UP
C
      IF (MODLIN) CALL RINLAY (KREAD,KOUT,NPLSET,MNOPLY,MAXNP,B(NB8),
     1                         B(NB9),B(ND46),B(ND47),B(ND48))
C
C     READ IN TEMPERATURE AND PRESSURE DATA
C
      CALL BREAD4 (CFORCE,PRES,TEM,KREAD,KOUT,NIPL,MAXNP,B(NB6),B(NB11),
     1             B(NB32),B(NB33),B(ND45))
C
C     READ CONCENTRATED FORCES & ROTATIONAL SPEED
C
      CALL RINFOR (ANGULA,CFORCE,KREAD,KOUT,GRIDP1,GRIDP2,B(NB11),
     1             B(NB28),B(NB29),B(NB30),B(NB31))
C
C     READ IN DATA FOR DYNMIC ANALYSIS
C
      IF (DYNA) CALL DYNMIN (EL3,EL7,EL10,EL11,EL75,DISPL,VELO,ACCEL,
     1                       PERIOD,KREAD,KOUT,MAXNP,B(NB36),B(NB37),
     2                       B(NB38),B(NB39),B(NB40))
C
C     READ IN BOUNDARY CONDITIONS AND TRANSFORMATIONS
C
      CALL RIBOUN (TRANS,KREAD,KOUT,NBC,NTR,NBCON,NTCON,B(NB12),B(NB34),
     1             B(NB35))
C
C     READ IN OUTPUT CONTROL DATA
C
      CALL OUTPTC (MODES,BUCK,KREAD,KOUT)
C
C     READ IN THE DATABANK
C
      CALL BANKRH (KDBANK,KOUT,NDES,B(NB8),B(NB26),B(NB25),B(NB27),
     1             B(NB13),B(NB14),B(NB15),B(NB16),B(NB17),B(NB18),
     2             B(NB19),B(NB20),B(NB21),B(NB22),B(NB23),B(NB24),
     3             TREF,BFT,BFC,BCS,FDC,FDK)
      CALL GETIME (TT1,TT2,DTI)
      DT1 = DTI
```

```
C
C     CREATE THE INPUT DECK FOR COBSTRAN ---- GENERATE FE MODEL
C
      IF (.NOT. MODLIN) THEN
      MTCODE = 0
      CALL PCOBST (PLYORD,UNSY,KREAD,KOUT,KCBST,MTCODE,NDES,NIPL,NOSC,
     1            NOSC1,LSMAX,MMAX,NEYY,NETT,MAXPLY,LMAX,B(NB1),
     2            B(NB2),B(NB3),B(NB4),B(NB5),B(NB7),B(NB8),B(NB9),
     3            B(NB10))
C
C     GENERATE THE FINITE ELEMENT MODEL
C
      CALL COBSTRAN
C
C     INTERPRET NASTRAN BULKDATA
C
      CALL NASINT (KCBFE,KOUT,NEXX,NEYY,NETT,NPXX,NPYY,MAXNP,NETOT,
     1            B(ND43),B(ND44))
C
C     CREATE THE INPUT DECK FOR COBSTRAN ---- CREATE PLY LAY-UP
C
      MTCODE = 1
      CALL PCOBST (PLYORD,UNSY,KREAD,KOUT,KCBST,MTCODE,NDES,NIPL,NOSC,
     1            NOSC1,LSMAX,MMAX,NEYY,NETT,MAXPLY,LMAX,B(NB1),
     2            B(NB2),B(NB3),B(NB4),B(NB5),B(NB7),B(NB8),B(NB9),
     3            B(NB10))
C
C     GENERATE THE PLY LAY-UP
C
      CALL COBSTRAN
C
C     INTERPRET BULKDATA ---- TO GET PLY LAY-UP
C
      CALL METINT (KCBMT,KOUT,NEXX,NEYY,NPXX,NPYY,MAXNP,MNOPLY,B(NB8),
     1            B(ND44),B(ND46),B(ND47),B(ND48))
      END IF
C
C     OUTPUT MODEL DATA
C
      CALL OMODEL (TEM,PRES,KOUT,MAXNP,NETOT,MNOPLY,NTISTP,B(NB11),
     1            B(ND43),B(ND44),B(ND45),B(ND47),B(ND48))
C
C     CALL THE SUBROUTINE WNEUT TO GENERATE A PATRAN NEUTRAL FILE
C
      CALL WNEUT (KPAT,8,MAXNP,NETOT,B(ND43),B(ND44))
```

```
C
C     DETERMINE NODAL TRANSFORMATION MATRIX
C
      CALL BNORML (KOUT,MAXNP,NETOT,B(ND43),B(ND44),B(ND75))
      CALL GETIME (TT1,TT2,DTI)
      DT2 = DTI
      IF (MODEL) GO TO 5
C
C     CALL THE SUBROUTINE NLINER TO DO THE STATIC ANALYSIS
C
      IF (.NOT. DYNA) THEN
      CALL NLINER (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,MAXNP,MAXNP,
     1             NETOT,MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,B(NB8),
     2             B(NB13),B(NB14),B(NB15),B(NB16),B(NB17),B(NB18),
     3             B(NB19),B(NB20),B(NB21),B(NB22),B(NB23),B(NB24),
     4             B(NB25),B(NB26),B(NB11),B(NB28),B(NB29),B(NB30),
     5             B(NB31),B(NB12),B(NB32),B(NB33),B(NB34),B(NB35),
     6             B(ND42),B(ND43),B(ND44),B(ND45),B(ND46),B(ND47),
     7             B(ND48),B(ND49),B(ND50),B(ND51),B(ND52),B(ND53),
     8             B(ND54),B(ND55),B(ND56),B(ND57),B(ND58),B(ND59),
     9             B(ND60),B(ND61),B(ND62),B(ND63),B(ND64),B(ND65),
     +             B(ND66),B(ND67),B(ND68),B(ND69),B(ND70),B(ND71),
     1             B(ND72),B(ND73),B(ND74),B(ND75),B(ND76))
C
C     CALL THE SUBROUTINE DYNMCH TO PERFORM THE DYNAMIC ANALYSIS
C
      ELSE
      CALL DYNMCH (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,MAXNP,NPTOT,
     1             NETOT,MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,B(NB8),
     2             B(NB13),B(NB14),B(NB15),B(NB16),B(NB17),B(NB18),
     3             B(NB19),B(NB20),B(NB21),B(NB22),B(NB23),B(NB24),
     4             B(NB25),B(NB26),B(NB11),B(NB28),B(NB29),B(NB30),
     5             B(NB31),B(NB12),B(NB32),B(NB33),B(NB34),B(NB35),
     6             B(NB36),B(NB37),B(NB38),B(NB39),B(NB40),B(ND42),
     7             B(ND43),B(ND44),B(ND45),B(ND46),B(ND47),B(ND48),
     8             B(ND49),B(ND50),B(ND51),B(ND52),B(ND53),B(ND54),
     9             B(ND55),B(ND56),B(ND57),B(ND58),B(ND59),B(ND60),
     +             B(ND61),B(ND62),B(ND63),B(ND64),B(ND65),B(ND66),
     1             B(ND67),B(ND68),B(ND69),B(ND70),B(ND71),B(ND72),
     2             B(ND73),B(ND74),B(ND75),B(ND76),B(ND77),B(ND78),
     3             B(ND79),B(ND80),B(ND81))
      END IF
```

```
      CALL GETIME (TT1,TT2,DTI)
      WRITE (KOUT,20) DT1, DT2, TT1
C
    5 CONTINUE
C
   20 FORMAT (/' TIME REQUIRED TO : ',
     1        'READ IN DATA                ',18X,F10.3,' SEC.',/,20X,
     2        'DO PREPROCESSING            ',18X,F10.3,' SEC.',/,20X,
     3        'TOTAL TIME FOR THIS RUN     ',18X,F10.3,' SEC.')
C
      RETURN
      END
```

APPENDIX E
SUBROUTINE NLINER

```
C ----------------------------------------------------------------
C
C     SUBROUTINE NLINER
C
C     PURPOSE    PERFORMS THE INCREMENTAL STATIC ANALYSIS
C
C ----------------------------------------------------------------
      SUBROUTINE NLINER (A,MAXLEN,IWORK,ISIZE,NSTART,IMAT,MESH,MAXNP,
     1                   NPTOT,NETOT,MNOPLY,HOLD,GRIDP1,GRIDP2,PINTER,
     2                   CODEH,FPI,MPI,DPI,FSIGI,MSIGI,DSIGI,FDOTI,
     3                   MDOTI,DDOTI,FTVCI,MTVCI,DTVCI,NF,DFH,TISTPS,
     4                   ANGVEL,NCFNOD,NCFDIR,CFVAL,NBOUND,NPRESS,
     5                   PREVAL,NTRANS,TRANG,KAD,KEI,GNP,TPGNP,THHF,
     6                   NLEP,MDES,THERMO,AS,CS,BS,ALPH,DEN,TLAY,PDIFO,
     7                   PDIFI,DISP,DISVH,MSH,NSH,QSH,REACF,COFORC,PLH,
     8                   TOLD,TEMP,TNEW,EIGNVA,EIGNVE,TEMPER,PC,STRMOM,
     9                   PRESUR,ETRAN,THSTRN)
      REAL               MTVCI,      MDOTI,    MSIGI,    MPI,      NSH,
     1                   MSH,        NS,       MS
      LOGICAL            PLYORD,     ACCEL,    TEM,      PRES,     VELO,
     1                   PROT,       PREP,     POST,     MODES,    PRTOUT,
     2                   SOLID,      SUPER,    WINDML,   DYNA,     SPARS,
     3                   BIDE,       UNSY,     PROFIL,   HOLLOW,   DISPL,
     4                   DAMP,       PERIOD,   PATRAN,   MHOST,    EZRD,
     5                   ECHO,       RANG,     ANGULA,   CFORCE,   PANEL,
     6                   LINEAR,     RESTAR,   HEAT,     MODEL,    FABRIC,
     7                   CYLIND,     BUCK,     INTER,    BMODEL,   EL3,
     8                   EL7,        EL10,     EL11,     EL75,     TRANS,
     9                   MODLIN
      LOGICAL            CONV
      COMMON   /HUNITS/  KDES1,      KKEI,     KBULK,    KREAD,    KOUT,
     1                   KCBST,      KCBMT,    KPROP,    KCBFE,    KDBANK,
     2                   KTRAN,      KPOST,    KMHFOR,   KREFL,    KPLY,
     3                   KPROPI,     KSTR,     KLAM,     KSIG,     KNODAL,
     4                   KDISP,      KMSTR,    KTRANF,   KCBOUT,   KTEMP,
     5                   KMHOST,     KSTART,   KPOSTF,   KPAT,     KEIGEN
      COMMON   /CMETCN/  KPOST0,     KPOST1,   KPOST2,   KPOST3,   KPOST4,
     1                   KPOST5,     KPOST6,   KPOST7,   KPOST8,   KPOST9,
     2                   KMETCN
```

```
      COMMON    /BOLH/    BIDE,        DYNA,      PRTOUT,   WINDML,   SOLID,
     1                    SUPER,       UNSY,      PROFIL,   PROT,     PREP,
     2                    POST,        MODES,     HOLLOW,   DISPL,    DAMP,
     3                    PLYORD,      ACCEL,     TEM,      VELO,     PRES,
     4                    PERIOD,      PATRAN,    MHOST,    EZRD,     ECHO,
     5                    RANG,        ANGULA,    CFORCE,   PANEL,    LINEAR,
     6                    SPARS,       RESTAR,    HEAT,     MODEL,    FABRIC,
     7                    CYLIND,      BUCK,      INTER,    BMODEL,   EL3,
     8                    EL7,         EL10,      EL11,     EL75,     TRANS,
     9                    MODLIN
      COMMON    /INTPH/   TREF,        BFT,       BFC,      BCS,      FDC,
     1                    FDK
      COMMON    /LOADD/   NMECHC,      NTHERC,    LINC,     MSTART,
     1                    MITER,       NTISTP,    NCFOR,    TOL,
     2                    NEIGV,       NSUBD,     INCREG,   NBC,
     3                    MHITER,      RESID,     AESID,    RERMSQ,
     4                    AERMSQ,      NPRES,     NTEMP,    NTR,
     5                    NBCON,       NTCON,     NUMPLY,   NOPERD,
     6                    DAMPMS,      DAMPST,    NDIS,     NVEL,
     7                    NACC,        INCDYN
      COMMON    /ETABLE/  NELCRD,      NELNFR,    NELNOD,   NELSTR,
     1                    NELCHR,      NELCMP,    NDI
      DIMENSION           CODEH(6,1),   TISTPS(1),       TIMES(2),
     1                    FPI(22,1),    MPI(22,1),       DPI(22,1),
     2                    FSIGI(6,1),   MSIGI(14,1),     DSIGI(10,1),
     3                    FDOTI(6,1),   MDOTI(14,1),     DDOTI(10,1),
     4                    FTVCI(31,1),  MTVCI(31,1),     DTVCI(31,1),
     5                    NF(1),        DFH(1),          COFORC(1),
     6                    ANGVEL(1),    NCFNOD(1),       NCFDIR(1),
     7                    CFVAL(NCFOR,1),                PDIFI(1)
      DIMENSION           THHF(1),      NLEP(1),         CS(NELSTR,1),
     1                    DEN(1),       TLAY(5,1),       PDIFO(1),
     2                    GRIDP1(3),    GRIDP2(3),       NSH(3,1),
     3                    QSH(2,1),     KAD(1),          BS(NELSTR,1),
     4                    TRANG(1),     NTRANS(4,1),     ETRAN(3,3,1),
     5                    DISP(NELNFR,1),                REACF(NELNFR,1),
     6                    DISVH(NELNFR,1),               MSH(3,1),
     7                    AS(NELSTR,1),                  GNP(12,1)
      DIMENSION           TOLD(MNOPLY,1),                TEMP(MNOPLY,1),
     1                    TNEW(MNOPLY,1),                NBOUND(4,1),
     2                    TEMPER(1),                     PC(62,1),
     3                    STRMOM(NELSTR,1),              PRESUR(1),
     4                    NPRESS(3,1),                   PREVAL(NPRES,1)
```

```
      DIMENSION           KEI(9,1),     TPGNP(7,MAXNP,1),
     1                    PLH(80,1),    THERMO(MNOPLY,MAXNP,1),
     2                    ALPH(3,1),    MDES(MNOPLY,1)
      DIMENSION           EIGNVA(1),    EIGNVE(NELNFR,MAXNP,1)
      DIMENSION           THSTRN(6,1)
      DIMENSION           IWORK ( 1 )
      DIMENSION           A ( 1 )
C
      DERORO = 0.0
      TOLMET = 0.05
      LDALOW = 10
      LSALOW = 5
      ICONVE = 0
      KCODE = 0
      ICODE = 1
      MMODE = 0
      MBUCK = 0
      IF (MODES) MMODE = 1
      IF (BUCK) MBUCK = 1
      MODES = .FALSE.
      BUCK = .FALSE.
      CALL GETIME (TT1,TT2,DTI)
C --------
C     SUBROUTINE INITSR INITIALIZE THE STRESS RESULTANTS
C
      CALL INITSR (MAXNP,MSH,NSH,QSH,SIGZ,PDIFO,PDIFI)
C --------
C     IF THIS IS A RESTART RUN, READ DATA FROM FILE KREAD
C
      IF (.NOT. RESTAR) GO TO 5000
      JCODE = 2
      CALL RESTART (JCODE,KREAD,KSTART,KLAM,KSIG,KSTR,NMC,NORAMP,IRAMP,
     1              INCR,ACCTIM,NLEP,DISP,TOTREA,MAXNP,NELNFR)
      CALL GETIME (TT1,TT2,DTI)
      ACCT = ACCTIM + DTI
      WRITE (KOUT,2018) DTI, ACCT
      ACCT = ACCT - DTI
      GO TO 6000
 5000 CONTINUE
C --------
C     SUBROUTINE INITMS INITIALIZE THE MICRO STRESSES, ETC.
C
      CALL INITMS (MAXNP,KSTR,KSIG,KLAM,NLEP,NELNFR,DISP,TOTREA)
```

```
C --------
C     DETERMINE THE THERMAL & MECHANICAL LOADS IN THE SUBROUTINE LOADIN
C
      CALL LOADIN (PRES,TEM,ANGULA,CFORCE,0,MAXNP,1,0,NCFOR,NPRES,CFVAL,
     1            COFORC,TPGNP,PDIFO,PDIFI,TREF,TLAY,ANGVEL,ANVDIF,
     2            NTISTP,PREVAL,PRESUR)
      TIMES(1) = TISTPS(1)
      TIMES(2) = TISTPS(1)
      TIMEE = TISTPS(1)
C --------
C     ITER. OVER EACH NODE & RUN THE MAT'L PROP. MODULE  HTMET (METCAN).
C     SUB. PMETCN SETS UP THE INPUT FILE FOR METCAN. SUB. PLY DETER.
C     THE PLY PROP. ARRAY PLH(1, ),...,PLH(14, ). THE SUB. PLPROP READS
C     KPOST5 & SETS UP THE PLY PROP. PLH(15, ),...,PLH(80, ) FOR
C     EACH PLY. CALL THE SUB. PROP TO OBTAIN THE NODAL PROPERTIES.
C
      JREC = 1
      DO 30 IK = 1,MAXNP
C     PRINT *,'NODE # ',IK
      NL = NLEP(IK)
      HTHICK = THHF(IK)
      DO 35 IL = 1,NL
C     PRINT *,'PLY # ',IL
      TOLD(IL,IK) = TREF
      TNEW(IL,IK) = TREF
      TEMP(IL,IK) = 0.0
      IF (TEM) THEN
        IF (EL3 .OR. EL7 .OR. EL11) THEN
        TOLD(IL,IK) = TPGNP(6,IK,1)
        TNEW(IL,IK) = TPGNP(6,IK,1)
        TEMP(IL,IK) = TPGNP(6,IK,1) - TREF
        ELSE
CJOE    TOLD(IL,IK) = TREF
        TOLD(IL,IK) = 0.0
        IF (FABRIC) TOLD(IL,IK) = THERMO(IL,IK,1)
        TNEW(IL,IK) = THERMO(IL,IK,1)
        TEMP(IL,IK) = THERMO(IL,IK,1) - TREF
        END IF
       END IF
   35 CONTINUE
```

```
      PO = DABS(PDIFO(IK))
      PI = DABS(PDIFI(IK))
      NLS = 2
      CALL PMETCN (IK,IREC,JREC,INTER,KSTR,KOUT,KMETCN,KDBANK,TIMES,NL,
     1             TREF,NDES,MAXNP,MNOPLY,PO,PI,DFH,NF,CODEH,MDES,TOLD,
     2             TNEW,MSH,NSH,QSH,FPI,MPI,DPI,FSIGI,MSIGI,DSIGI,FDOTI,
     3             MDOTI,DDOTI,FTVCI,MTVCI,DTVCI,TEMPER,NMECHC,NTHERC,
     4             PINTER,NLS)
      CALL HTMET (A,MAXLEN,LDITER,LDALOW,TOLMET)
      IF (EL3 .OR. EL7 .OR. EL11) THEN
      CALL MCPROP (IK,KOUT,KPOST0,PC)
      ELSE IF (EL75) THEN
      CALL MCPROP (IK,KOUT,KPOST0,PC)
      CALL PLYH (IK,KOUT,NLEP,THHF,CODEH,DFH,MDES,MNOPLY,MAXNP,PLH,FPI,
     1           MPI,ANGLG)
      CALL PLPROP (IK,KOUT,KPOST5,NL,MNOPLY,TEMP,PLH,THSTRN)
      CALL PROPH (IK,KOUT,KPROP,IMAT,HTHICK,NL,PLH,AS,CS,BS,ALPH,DEN)
      END IF
C     IREC = JREC
C     CALL MPROP (IK,IREC,KOUT,KLAM,KPOST2,KPOST3,KPOST4,KPOST5,NL,
C    1            MNOPLY,TNEW)
      JREC = JREC + NL
   30 CONTINUE
      KCODE = 1
C --------
      CALL GETIME (TT1,TT2,DTI)
      DT1 = DTI
      IF (FABRIC) GO TO 6000
C --------
C     SUB. MHPREP SETS UP THE INPUT FILE KMHOST FOR THE MODULE SOLVEH
C     (MHOST). THE SUB. SOLVEH DETERMINES THE GLOBAL SOLUTION. THE
C     STRESS RESULT. & DISP. ARE OBTAINED FROM THE OUTPUT FILES KMHFOR
C     & KDISP.
C
      CALL MHPREP (SPARS,TEM,PRES,ANGULA,CFORCE,MODES,BUCK,EL3,EL7,EL10,
     1             EL11,EL75,KOUT,KREAD,KMHOST,MESH,NETOT,NPTOT,MAXNP,
     2             NBOUND,GNP,KEI,GRIDP1,GRIDP2,ANVDIF,KAD,NCFNOD,
     3             NCFDIR,COFORC,TLAY,PDIFO,PDIFI,AS,CS,BS,ALPH,DEN,PC,
     4             NPRESS,PRESUR,NTRANS,TRANG,THSTRN)
```

```
      CALL SOLVEH (ISIZE,IWORK)
      CALL MHOUTP (EL3,EL7,EL10,EL11,EL75,KOUT,KMHFOR,KDISP,MAXNP,GNP,
     1            THHF,MSH,NSH,QSH,SIGZ,DISVH,REACF,STRMOM,ETRAN)
C --------
      CALL GETIME (TT1,TT2,DTI)
      DT2 = DTI
      CALL BAN1 (KOUT,ITER,INCR,NMC,IRAMP,1,TIMEE)
C --------
C     ITERATE OVER EACH NODE TO SET THE FLAGS FOR FIBER, MATRIX, &
C     INTERPHASE PROP. FAILURE (SUB. MPROP). ADD THE CURRENT STRESSES TO
C     THE INITIAL STRESSES IN THE SUB. MSTRES.
C
      JREC = 1
      DO 120 IK = 1,MAXNP
C     PRINT *,'NODE = ',IK
      NL = NLEP(IK)
      HTHICK = THHF(IK)
      PO = DABS(PDIFO(IK))
      PI = DABS(PDIFI(IK))
      DO 119 IL = 1,NL
      TOLD(IL,IK) = TREF
      IF (TEM) THEN
      IF (EL3 .OR. EL7 .OR. EL11) TOLD(IL,IK) = TPGNP(6,IK,1)
      TOLD(IL,IK) = 0.0
CJOE  IF (.NOT. EL7) TOLD(IL,IK) = THERMO(IL,IK,1)
      END IF
  119 CONTINUE
      NLS = 2
  121 CONTINUE
      CALL PMETCN (IK,IREC,JREC,INTER,KSTR,KOUT,KMETCN,KDBANK,TIMES,NL,
     1            TREF,NDES,MAXNP,MNOPLY,PO,PI,DFH,NF,CODEH,MDES,TOLD,
     2            TNEW,MSH,NSH,QSH,FPI,MPI,DPI,FSIGI,MSIGI,DSIGI,FDOTI,
     3            MDOTI,DDOTI,FTVCI,MTVCI,DTVCI,TEMPER,NMECHC,NTHERC,
     4            PINTER,NLS)
      CALL HTMET (A,MAXLEN,LDITER,LDALOW,TOLMET)
      IREC = JREC

C     IF (LDITER .LE. LDALOW) WRITE (KOUT,4010) IK, LDITER, NLS
      IF (LDITER .LE. LDALOW) GO TO 122
      WRITE (KOUT,4000) IK, LDITER, NLS
      NLS = NLS + 1
      IF (NLS .GT. LSALOW) GO TO 9000
      GO TO 121
  122 CONTINUE
      IF (EL3 .OR. EL7 .OR. EL11) THEN
      CALL MCPROP (IK,KOUT,KPOSTO,PC)
      ELSE IF (EL75) THEN
      CALL PLYH (IK,KOUT,NLEP,THHF,CODEH,DFH,MDES,MNOPLY,MAXNP,PLH,FPI,
     1          MPI,ANGLG)
      CALL PLPROP (IK,KOUT,KPOST5,NL,MNOPLY,TEMP,PLH,THSTRN)
      CALL PROPH (IK,KOUT,KPROP,IMAT,HTHICK,NL,PLH,AS,CS,BS,ALPH,DEN)
      END IF
      IREC = JREC
      CALL MPROP (IK,IREC,KOUT,KLAM,KPOST2,KPOST3,KPOST4,KPOST5,NL,
     1            MNOPLY,TNEW)
      IREC = JREC
```

```
      CALL MSTRES (IK,IREC,KOUT,KSTR,KSIG,KLAM,KPOST6,KPOST7,KPOST8,
     1             KPOST9,NL)
      JREC = JREC + NL
  120 CONTINUE
C --------
C     TOTAL DISPLACEMENTS AND PLY STRESSES AFTER THE FIRST LOAD STEP
C
      DO 130 IK = 1,MAXNP
      DO 130 J = 1,NELNFR
      DISP(J,IK) = DISP(J,IK) + DISVH(J,IK)
  130 CONTINUE
      CALL PSTRES (KOUT,KSIG,EL3,EL7,EL10,EL11,EL75,NLEP,ITER,INCR,
     1             NMC,IRAMP,1,1,TIMEE,DISP,TOTREA,MAXNP)
C --------
C     PERFORM MODAL ANALYSIS
C
      IF (MMODE .GT. 0) THEN
      CALL MODESH (IWORK,ISIZE,SPARS,TEM,PRES,ANGULA,CFORCE,MODES,BUCK,
     1             EL3,EL7,EL10,EL11,EL75,KOUT,KREAD,KMHOST,KEIGEN,
     2             TIMEE,INCR,MESH,NETOT,NPTOT,MAXNP,GNP,KEI,NBOUND,
     3             GRIDP1,GRIDP2,ANVDIF,KAD,NCFNOD,NCFDIR,COFORC,TLAY,
     4             PDIFO,PDIFI,AS,CS,BS,ALPH,DEN,EIGNVA,EIGNVE,PC,
     5             NPRESS,PRESUR,NTRANS,TRANG,THSTRN)
      END IF

C --------
C     PERFORM BUCKLING ANALYSIS
C
      IF (MBUCK .GT. 0) THEN
      CALL BUCKSH (IWORK,ISIZE,SPARS,TEM,PRES,ANGULA,CFORCE,MODES,BUCK,
     1             EL3,EL7,EL10,EL11,EL75,KOUT,KREAD,KMHOST,KEIGEN,
     1             TIMEE,INCR,MESH,NETOT,NPTOT,MAXNP,GNP,KEI,NBOUND,
     2             GRIDP1,GRIDP2,ANVDIF,KAD,NCFNOD,NCFDIR,COFORC,TLAY,
     3             PDIFO,PDIFI,AS,CS,BS,ALPH,DEN,EIGNVA,EIGNVE,PC,
     4             NPRESS,PRESUR,NTRANS,TRANG,THSTRN)
      END IF
C --------
C     GENERATE A PATRAN NEUTRAL FILE
C
      CALL WPOST (KNODAL,KTRAN,MAXNP,DISP,NSH,MSH,QSH,TIMEE)
      CALL GETIME (TT1,TT2,DTI)
      DT3 = DT1 + DT2 + DTI
      WRITE (KOUT,3000) DT1, DT2, DT3
 6000 CONTINUE
      IF (.NOT. RESTAR) KCODE = 1
      IF (.NOT. RESTAR) ACCTIM = 0.0
      MST = 0
C --------
C     LOOP OVER EACH MECHANICAL & THERMAL CYCLE
C
CJOE  DO 1000 NMC = 1,NMECHC
      DO 1000 NMC = 1,1
      NORAMP = NTISTP - 1
      IF (.NOT. RESTAR) MRAMP = 1
      IF (RESTAR .AND. INCR .LT. LINC) MRAMP = IRAMP
      IF (RESTAR .AND. INCR .EQ. LINC) MRAMP = IRAMP + 1
```

```
C --------
C     LOOP OVER NUMBER OF LOADING RAMPS (NORAMP)
C
      DO 200 IRAMP = MRAMP,NORAMP
      NSTPI = IRAMP
      NSTPI1 = IRAMP + 1
      TIMEIN = TISTPS(NSTPI)
      TIMEI1 = TISTPS(NSTPI1)
      DTIME = (TIMEI1 - TIMEIN)/FLOAT(LINC)
      CALL LOADIN (PRES,TEM,ANGULA,CFORCE,LINC,MAXNP,NSTPI1,NSTPI,NCFOR,
     1             NPRES,CFVAL,COFORC,TPGNP,PDIFO,PDIFI,0.0,TLAY,ANGVEL,
     2             ANVDIF,NTISTP,PREVAL,PRESUR)

C --------
C     INCREMENT OVER EACH RAMP ACCORDING TO NO. OF LOAD INCR. (LINC)
C
      IF (.NOT. RESTAR) MINCR = 1
      IF (RESTAR) MINCR = INCR + 1
      IF (RESTAR .AND. MINCR .GT. LINC) MINCR = 1
      DO 500 INCR = MINCR,LINC
      RLINC = FLOAT(LINC)
      RINCR = FLOAT(INCR)
      MST = MST + 1
      CALL GETIME (TT1,TT2,DTI)
      DT7 = DTI
      TTIME1 = TT1
      TIMEE = TIMEIN + RINCR*DTIME
      TIMES(1) = TIMEIN + FLOAT(INCR-1)*DTIME
      TIMES(2) = TIMEE
C --------
C     ITERATE OVER EACH LOAD INCREMENT (INCR)
C
      ITER = 0
  600 CONTINUE
      DTIM = 0
      CALL GETIME (TT1,TT2,DTI)
      ITER = ITER + 1
C --------
C     IF NO. OF ITERATIONS ( ITER ) EXCEEDS MITER WRITE A RESTART FILE
C
      IF (ITER .LE. MITER) GO TO 7000
      WRITE (KOUT,3100) IRAMP, INCR
      IF (INCR .EQ. 1 .AND. IRAMP .EQ. 1) GO TO 7100
      IF (INCR .EQ. 1) IINCR = LINC
      IF (INCR .GT. 1) IINCR = INCR - 1
      IIRAMP = IRAMP
      IF (INCR .EQ. 1 .AND. IRAMP .GT. 1) IIRAMP = IRAMP - 1
      WRITE (KOUT,3200) IIRAMP, IINCR
      GO TO 9000
 7100 CONTINUE
      WRITE (KOUT,3300)
      GO TO 1000
 7000 CONTINUE

C --------
      IF (KCODE .LT. 1) GO TO 7050
      CALL MHPREP (SPARS,TEM,PRES,ANGULA,CFORCE,MODES,BUCK,EL3,EL7,EL10,
```

```
     1              EL11,EL75,KOUT,KREAD,KMHOST,MESH,NETOT,NPTOT,MAXNP,
     2              NBOUND,GNP,KEI,GRIDP1,GRIDP2,ANVDIF,KAD,NCFNOD,
     3              NCFDIR,COFORC,TLAY,PDIFO,PDIFI,AS,CS,BS,ALPH,DEN,PC,
     4              NPRESS,PRESUR,NTRANS,TRANG,THSTRN)
      CALL SOLVEH (ISIZE,IWORK)
      CALL MHOUTP (EL3,EL7,EL10,EL11,EL75,KOUT,KMHFOR,KDISP,MAXNP,GNP,
     1              THHF,MSH,NSH,QSH,SIGZ,DISVH,REACF,STRMOM,ETRAN)
      CALL GETIME (TT1,TT2,DTI)
      DT9 = DTI
 7050 CONTINUE
C --------
      JREC = 1
      DO 245 IK = 1,MAXNP
      HTHICK = THHF(IK)
      NL = NLEP(IK)
      DO 265 IL = 1,NL
      TOLD(IL,IK) = TREF
      TNEW(IL,IK) = TREF
      TEMP(IL,IK) = 0.0
      IF (TEM) THEN
        IF (EL3 .OR. EL7 .OR. EL11) THEN
        DTEMP = (TPGNP(6,IK,NSTPI1) - TPGNP(6,IK,NSTPI))/RLINC
        TOLD(IL,IK) = TPGNP(6,IK,NSTPI) + (RINCR - 1.)*DTEMP
        TNEW(IL,IK) = TPGNP(6,IK,NSTPI) + RINCR*DTEMP
        TEMP(IL,IK) = (TPGNP(6,IK,NSTPI) - TPGNP(6,IK,NSTPI1))/RLINC
        ELSE
        DTEMP = (THERMO(IL,IK,NSTPI1) - THERMO(IL,IK,NSTPI))/RLINC
        TOLD(IL,IK) = THERMO(IL,IK,NSTPI) + (RINCR - 1.)*DTEMP
        TNEW(IL,IK) = THERMO(IL,IK,NSTPI) + RINCR*DTEMP
        TEMP(IL,IK) = (THERMO(IL,IK,NSTPI) - THERMO(IL,IK,NSTPI1))/RLINC
        END IF
      END IF
  265 CONTINUE
      PO = DABS(PDIFO(IK))
      PI = DABS(PDIFI(IK))
      NLS = 2
  221 CONTINUE

      CALL PMETCN (IK,IREC,JREC,INTER,KSTR,KOUT,KMETCN,KDBANK,TIMES,NL,
     1              TREF,NDES,MAXNP,MNOPLY,PO,PI,DFH,NF,CODEH,MDES,TOLD,
     2              TNEW,MSH,NSH,QSH,FPI,MPI,DPI,FSIGI,MSIGI,DSIGI,FDOTI,
     3              MDOTI,DDOTI,FTVCI,MTVCI,DTVCI,TEMPER,NMECHC,NTHERC,
     4              PINTER,NLS)
      CALL HTMET (A,MAXLEN,LDITER,LDALOW,TOLMET)
      IREC = JREC
C     IF (LDITER .LE. LDALOW) WRITE (KOUT,4010) IK, LDITER, NLS
      IF (LDITER .LE. LDALOW) GO TO 222
      WRITE (KOUT,4000) IK, LDITER, NLS
      NLS = NLS + 1
      IF (NLS .GT. LSALOW) GO TO 9000
      GO TO 221
  222 CONTINUE
      CALL GETIME (TT1,TT2,DTI)
      DTIM = DTIM + DTI
      IF (EL3 .OR. EL7 .OR. EL11) THEN
      CALL MCPROP (IK,KOUT,KPOSTO,PC)
      ELSE IF (EL75) THEN
```

```
      CALL PLYH (IK,KOUT,NLEP,THHF,CODEH,DFH,MDES,MNOPLY,MAXNP,PLH,FPI,
     1           MPI,ANGLG)
      CALL PLPROP (IK,KOUT,KPOST5,NL,MNOPLY,TEMP,PLH,THSTRN)
      CALL PROPH (IK,KOUT,KPROP,IMAT,HTHICK,NL,PLH,AS,CS,BS,ALPH,DEN)
      END IF
      JREC = JREC + NL
  245 CONTINUE
      CALL PSTRES (KOUT,KSIG,EL3,EL7,EL10,EL11,EL75,NLEP,ITER,INCR,
     1             NMC,IRAMP,5,5,TIMEE,DISVH,REACF,MAXNP)
C --------
C     CHECK FOR CONVERGENCE
C
      CALL CHECKD (CONV,LINEAR,KOUT,KCODE,TOL,NMC,IRAMP,INCR,ITER,NPTOT,
     1             DISVH,DERORO,NELNFR)
      KCODE = 1
      CALL GETIME (TT1,TT2,DTI)
      DT11 = DT9 + DT10 + DTIM
      WRITE (KOUT,2007) DT9,DTIM,DT11
      IF (CONV) GO TO 280
      GO TO 600
  280 CONTINUE

C --------
C     IF CONVERGED, DETER. NEW FIBER, MATRIX, INTERPHASE, AND STRENGTHS
C
      JREC = 1
      DO 400 IK = 1,MAXNP
      HTHICK = THHF(IK)
      NL = NLEP(IK)
      DO 365 IL = 1,NL
      TOLD(IL,IK) = TREF
      TNEW(IL,IK) = TREF
      TEMP(IL,IK) = 0.0
      IF (TEM) THEN
        IF (EL3 .OR. EL7 .OR. EL11) THEN
        DTEMP = (TPGNP(6,IK,NSTPI1) - TPGNP(6,IK,NSTPI))/RLINC
        TOLD(IL,IK) = TPGNP(6,IK,NSTPI) + (RINCR - 1.)*DTEMP
        TNEW(IL,IK) = TPGNP(6,IK,NSTPI) + RINCR*DTEMP
        TEMP(IL,IK) = (TPGNP(6,IK,NSTPI) - TPGNP(6,IK,NSTPI1))/RLINC
        ELSE
        DTEMP = (THERMO(IL,IK,NSTPI1) - THERMO(IL,IK,NSTPI))/RLINC
        TOLD(IL,IK) = THERMO(IL,IK,NSTPI) + (RINCR - 1.)*DTEMP
        TNEW(IL,IK) = THERMO(IL,IK,NSTPI) + RINCR*DTEMP
        TEMP(IL,IK) = (THERMO(IL,IK,NSTPI) - THERMO(IL,IK,NSTPI1))/RLINC
        END IF
      END IF
  365 CONTINUE
      PO = DABS(PDIFO(IK))
      PI = DABS(PDIFI(IK))
      NLS = 2
  421 CONTINUE
      CALL PMETCN (IK,IREC,JREC,INTER,KSTR,KOUT,KMETCN,KDBANK,TIMES,NL,
     1             TREF,NDES,MAXNP,MNOPLY,PO,PI,DFH,NF,CODEH,MDES,TOLD,
     2             TNEW,MSH,NSH,QSH,FPI,MPI,DPI,FSIGI,MSIGI,DSIGI,FDOTI,
     3             MDOTI,DDOTI,FTVCI,MTVCI,DTVCI,TEMPER,NMECHC,NTHERC,
     4             PINTER,NLS)
```

```
      CALL HTMET (A,MAXLEN,LDITER,LDALOW,TOLMET)
      IREC = JREC
C     IF (LDITER .LE. LDALOW) WRITE (KOUT,4010) IK, LDITER, NLS
      IF (LDITER .LE. LDALOW) GO TO 422
      WRITE (KOUT,4000) IK, LDITER, NLS
      NLS = NLS + 1
      IF (NLS .GT. LSALOW) GO TO 9000

      GO TO 421
  422 CONTINUE
      CALL MPROP (IK,IREC,KOUT,KLAM,KPOST2,KPOST3,KPOST4,KPOST5,NL,
     1            MNOPLY,TNEW)
      IREC = JREC
      CALL MSTRES (IK,IREC,KOUT,KSTR,KSIG,KLAM,KPOST6,KPOST7,KPOST8,
     1             KPOST9,NL)
      JREC = JREC + NL
  400 CONTINUE
C --------
C     UPDATE THE TOTAL DISPL. (DISP) WITH INCR. DISPL. (DISVH)
C
      DO 415 IK = 1,MAXNP
      DO 415 J = 1,NELNFR
      DISP(J,IK) = DISP(J,IK) + DISVH(J,IK)
  415 CONTINUE
      CALL PSTRES (KOUT,KSIG,EL3,EL7,EL10,EL11,EL75,NLEP,ITER,INCR,
     1             NMC,IRAMP,3,3,TIMEE,DISP,TOTREA,MAXNP)
      CALL GETIME (TT1,TT2,DTI)
      DTI = TT1 - TTIME1
      WRITE (KOUT,2040) DTI
      ACCTIM = TT2
C --------
C     PERFORM MODAL ANALYSIS
C
      CALL MODESH (IWORK,ISIZE,SPARS,TEM,PRES,ANGULA,CFORCE,MODES,BUCK,
     1             EL3,EL7,EL10,EL11,EL75,KOUT,KREAD,KMHOST,KEIGEN,
     2             TIMEE,INCR,MESH,NETOT,NPTOT,MAXNP,GNP,KEI,NBOUND,
     3             GRIDP1,GRIDP2,ANVDIF,KAD,NCFNOD,NCFDIR,COFORC,TLAY,
     4             PDIFO,PDIFI,AS,CS,BS,ALPH,DEN,EIGNVA,EIGNVE,PC,
     5             NPRESS,PRESUR,NTRANS,TRANG,THSTRN)
C --------
C     PERFORM BUCKLING ANALYSIS
C
      CALL BUCKSH (IWORK,ISIZE,SPARS,TEM,PRES,ANGULA,CFORCE,MODES,BUCK,
     1             EL3,EL7,EL10,EL11,EL75,KOUT,KREAD,KMHOST,KEIGEN,
     1             TIMEE,INCR,MESH,NETOT,NPTOT,MAXNP,GNP,KEI,NBOUND,
     2             GRIDP1,GRIDP2,ANVDIF,KAD,NCFNOD,NCFDIR,COFORC,TLAY,
     3             PDIFO,PDIFI,AS,CS,BS,ALPH,DEN,EIGNVA,EIGNVE,PC,
     4             NPRESS,PRESUR,NTRANS,TRANG,THSTRN)
C --------
C     GENERATE A PATRAN NEUTRAL FILE
C
      CALL WPOST (KNODAL,KTRAN,MAXNP,DISP,NSH,MSH,QSH,TIMEE)
C
      IF (MST .LT. MSTART) GO TO 500
      IIRAMP = IRAMP
      IINCR = INCR
```

```
C --------
C     WRITE DATA TO RESTART FILE KREAD
C
 9000 CONTINUE
      JCODE = 1
      ACCTIM = ACCTIM + ACCT
      CALL RESTART (JCODE,KREAD,KSTART,KLAM,KSIG,KSTR,NMC,NORAMP,IRAMP,
     1              INCR,ACCTIM,NLEP,DISP,TOTREA,MAXNP,NELNFR)
      GO TO 1000
C --------
      JCODE = 1
      ACCTIM = ACCTIM + ACCT
      CALL RESTART (JCODE,KREAD,KSTART,KLAM,KSIG,KSTR,NMC,NORAMP,IRAMP,
     1              INCR,ACCTIM,NLEP,DISP,TOTREA,MAXNP,NELNFR)
  500 CONTINUE
  200 CONTINUE
C     CALL PSTRES (KOUT,KSIG,EL3,EL7,EL10,EL11,EL75,NLEP,ITER,INCR,
C    1             NMC,IRAMP,3,4,TIMEE,DISP,TOTREA,MAXNP)
 1000 CONTINUE
      CALL GETIME (TT1,TT2,DTI)
      ACCTIM = ACCTIM + DTI
      WRITE (KOUT,2150) ACCTIM
C
 2007 FORMAT (/' TIME REQUIRED TO : CALCULATE THE STRESS ',
     1        'RESULTANTS IN MHOST',6X,F10.3,' SEC.',/,20X,'DETERMINE',
     2        ' PLY AND NODAL PROPERTIES      ',6X,F10.3,' SEC.',/,20X,
     3        'PERFORM THIS ITERATION         ',14X,F10.3,' SEC.')
 2018 FORMAT (/' TIME REQUIRED TO : READ IN DATA FROM ',
     1        'RESTART FILE                 ',F10.3,' SEC.',/,
     2        20X,'TOTAL TIME REQUIRED FOR THE ANALYSIS SO FAR',
     3        3X,F10.3,' SEC.')
 2040 FORMAT (/' TIME REQUIRED TO EVALUATE THIS LOAD INCREMENT',20X,
     1        F10.3,' SEC.')
 2150 FORMAT (/' TIME REQUIRED TO CARRY OUT THE ANALYSIS      ',20X,
     1        F10.3,' SEC.')
 2160 FORMAT (/' TIME REQUIRED TO : ',
     1        'PERFORM FIRST LOAD STEP      ',17X,F10.3,' SEC.',/,20X,
     2        'DO THE NONLINEAR ANALYSIS    ',17X,F10.3,' SEC.',/,20X,
     3        'CARRY OUT THE ENTIRE ANALYSIS',17X,F10.3,' SEC.')
 3000 FORMAT (/' TIME REQUIRED TO : ',
     1        'LOOP THRU METCAN             ',17X,F10.3,' SEC.',/,20X,
     1        'DETER. THE DISPL. IN MHOST   ',17X,F10.3,' SEC.',/,20X,
     2        'ANALYZE THE FIRST LOAD STEP',17X,F10.3,' SEC.')
 3100 FORMAT (/' THE ANALYSIS HAS FAILED TO CONVERGE FOR RAMP NO.',1X,
     1        I3,' AND LOAD INCREMENT NO.',1X,I3)
 3200 FORMAT (/' A RESTART FILE HAS BEEN CREATED FOR RAMP NO.',5X,I3,
     1        ' AND LOAD INCREMENT NO.',1X,I3)
 3300 FORMAT (/' A RESTART FILE WAS NOT CREATED')
 4000 FORMAT (' NUMBER OF ITERATIONS FOR NODE # ',I4,' IS ',I2,' WITH ',
     1        I2,' LOAD STEPS THAT METCAN FAILED TO CONVERGE ')
 4010 FORMAT (' NUMBER OF ITERATIONS REQUIRED FOR NODE # ',I4,' IS ',
     1        I2,' WITH ',I2,' LOAD STEPS TO HAVE CONVERGENCE ')
C
      RETURN
      END
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 1993

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
High Temperature Composite Analyzer (HITCAN) Programmer's Manual
Version 1.0

5. FUNDING NUMBERS
WU–505–63–5B

6. AUTHOR(S)
J.J. Lackney, P.L.N. Murthy, and P.K. Gotsis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sverdrup Technology, Inc.
Lewis Research Center Group
2001 Aerospace Parkway
Brook Park, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER
E–7547

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM–106004

11. SUPPLEMENTARY NOTES
J.J. Lackney, Sverdrup Technology, Inc., Lewis Research Center Group, 2001 Aerospace Parkway, Brook Park, Ohio 44142 (work funded by NASA Grant 3–25266); P.L.N. Murthy and P.K. Gotsis, Lewis Research Center, Cleveland, Ohio. Responsible person, P.L.N. Murthy, (216) 433–3332.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 39

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This manual describes the organization and flow of data and analysis modules in the computer code, HITCAN (HIgh Temperature Composite ANalyzer). HITCAN is a general purpose computer program for predicting nonlinear global structural and local stress-strain response of arbitrarily oriented, multilayered high temperature metal matrix composite structures. This manual describes the architecture of the HITCAN code, followed by the listing of subroutines and calling tree, data storage scheme, file system, and a dictionary of code terminology. The primary intention of the manual is to familiarize the user with some of the computer program related issues so as to facilitate maintenance/modification/updates of the HITCAN computer code.

14. SUBJECT TERMS
Metal matrix composite structures; Fiber composites; Composite mechanics; Laminate analysis; Micro stresses; Micromechanics; High temperature; Thermal properties; Material non-linearities; 2D and 3D nonlinear analysis; Finite elements; Buckling; Stress analysis; Vibrations

15. NUMBER OF PAGES
132

16. PRICE CODE
A07

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102