TRW
SYSTEMS GROUP

ANALYSIS OF FOLDING STRAINS ON COMPOSITE BLADDER STRUCTURES

Contract NAS 7-682 1 April 1970

Prepared for

NASA WESTERN OPERATIONS OFFICE
Pasadena, California

VOLUME II - PROGRAMMING MANUAL

TRW SYSTEMS GROUP
One Space Park
Redondo Beach, California

FOLDING STRAINS ANALYSIS PRØGRAM

(BLADER - TRW Program No. AS189)

by

James R. Taylor

Approved

S. Gurnik
Engineering Analysis
Programming Department

Abstract

The Folding Strains Analysis Program (also called BLADER, TRW Computer program AS189) performs a step-by-step large strain, elastoplastic static analysis of a two dimensional continuum. The program allows the user to perform a plane strain analysis using a rectangular coordinate system or to perform an axisymmetric strain analysis using a cylindrical coordinate system. The structure to be studied may consist of one or more homogeneous, isotropic materials whose plastic and elastic properties are temperature and strain rate independent. The structure may be loaded either by specifying certain nodes to be subjected to imposed displacements or by specifying certain sides of some of the finite elements to be subjected to pressure loadings. Boundary conditions which constrain specified nodes from movement in one or both coordinate directions must be imposed by the user.

One of the major features of the program provides for the analysis of folding bladder problems by the specification of a minimum amount of input.

The program employs a dynamic storage feature which allows the program to efficiently use the available computer memory space. Thus small problems require only a relatively small amount of memory while the ability of the program to handle large problems without programming modifications is limited only by the maximum amount of memory available.

The program is written in Fortran IV and was developed and checked out on the CDC 6500 computing system at TRW. With a minimal conversion effort the program should be capable of being compiled and executed on other computing systems.

TABLE OF CONTENTS

1.0 PROBLEM DESCRIPTION

The BLADER Program (TRW Program AS189) performs a step-by-step large strain, elastoplastic static analysis of a two-dimensional continuum. The user may perform a plane strain analysis using a rectangular coordinate system or perform an axisymmetric strain analysis using a cylindrical coordinate system. In either coordinate system the basic finite element used is a triangle with six nodal points, one at each corner and one on the midpoint of each side (fig. 1.1). A quadratic variation of displacement is assumed for each element, and is expressed in terms of two components of displacement at each node, giving a total of twelve degrees of freedom per element or 2N degrees of freedom for the system, where N = number of nodes.

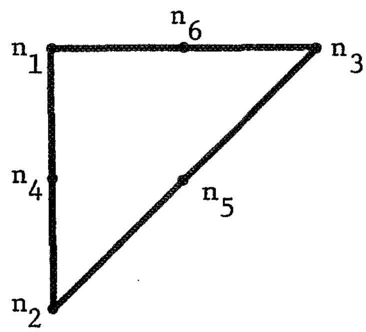


Figure 1.1 Basic Finite Element

The structure consists of one or more homogeneous, isotropic materials whose plastic and elastic properties (i.e., elastic modulus, Poisson's ratio, plastic/elastic modulus ratio, and initial yield point) are temperature and strain rate independent.

The structure may be loaded by three different methods. However, only one method of loading may be applied in a single run. The first method allows the user to require that specified sides of specified elements are to be loaded. Each side S_i loaded may be loaded with a total load P_i which is applied in N equal load steps, or the loading may be specified as a tabulated function of the load step. If the latter procedure is used, then all sides loaded must be loaded according to the same schedule. In either case some sides may be loaded with a positive load (compression) and other sides may be loaded with a negative load (tension).

The second method of loading allows the user to specify any number of imposed displacement constraints. In this method of loading, each designated node is moved by a specified amount in a specified direction and held in this position during the analysis.

The third method of loading applies only to axisymmetric problems and allows the user to load the structure by shrinking the secondary radius of curvature.

The user must specify boundary conditions which restrain specified nodes from motion in one or both coordinate directions. In addition the user may impose maximum displacement boundary conditions which restrain specified nodes from exceeding certain maximum displacements in specified directions.

The primary application of the program is to be in the analysis of folding bladders which are used in rocket propulsion systems. Accordingly one of the major options of the program called the Bladder Analysis Option allows the analysis of the stress-strain relationships in single folds or double folds of bladders. The specialized nature of such analyses makes it possible to specify the problem completely by a minimum of input quantities. Thus the user is required to specify only the most basic problem parameters such as the number of element layers, the dimensions and material properties of each layer, and the basic loading parameters such as the initial and final radii of curvature.

Another option available as part of the Bladder Analysis Option provides for the calculation of the fatigue life of each layer of elements using input tables of fatigue life versus plastic strain for the materials which make up the structure.

Any consistent set of units may be used by the program. That is, all measures of length should be in the same units, all measures of force should be in the same units, etc.

A more detailed discussion of the engineering aspects of the problem is presented in a companion document written by L. D. Hofmeister.

2.0 INPUT

2.1 Introduction

All data input to the BLADER program is specified on punched cards which are read using the Namelist feature of Fortran IV. Because of the dynamic storage feature used in BLADER, the input data is divided into two major categories. The first category of input data is the General Input Data, which consists of an input header, a convergence criterion, option flags, and parameters defining the problem dimensions. The second category of input is the Array Data, which consists of all input arrays other than the input header. The separation of the data into these two categories is a physical as well as a logical separation.

That is, the Namelist cards which specify the General Input Data for a single case are read in first, being initiated by the "Name card"

$AS189

and being terminated by a dollar sign ($). (See section 2.1.2, "Name cards, and section 2.1.5, "End-of-Data"). After the program determines the dimensions of the problem and requests the necessary memory for the arrays to be used, the Array Data is read in, being initiated by the "Name card"

$DATA

and being terminated by another dollar sign.

The data cards should be punched on a standard keypunch machine using the 026 character set. With the exception of a possible P in column 1 of the Name card (defined below), no characters should be punched in any columns other than columns 2-72 of a Namelist data card.

2.1.1 Keypunching Conventions

Due to the similar shapes of many letters, numbers, and symbols it is possible to confuse one character for another. In order to assist the keypuncher in discerning between the many similar characters, the following coding conventions should be observed:

ⓝ	indicates n blank columns		
Ø	alphabetic as distinguished from	O	(zero) numeric
I	alphabetic as distinguished from	1	(one) numeric
Ƶ	alphabetic as distinguished from	2	(two) numeric
U	alphabetic as distinguished from	4	(four) numeric
S	alphabetic as distinguished from	5	(five) numeric
G	alphabetic as distinguished from	6	(six) numeric
4	numeric as distinguished from	9	(nine) numeric
U	alphabetic as distinguished from	V	alphabetic
C	alphabetic as distinguished from	(	(left parenthesis)
T	alphabetic as distinguished from	+	(plus)

2.1.2 Name Cards

The first card of the General Input Data must have \$AS189 punched in columns 2-7. The first card of the Array Data must have \$DATA punched in columns 2-6. When executing on the CDC 6500 computing system at TRW, a P may be punched in column 1 of each Name card if it is desired to list the card images of the Namelist input. In order to maintain compatibility with other computing systems, the rest of the card should be blank.

2.1.3 Head Card

The user may specify alphanumeric information for labeling a given case by entering a card of the form

HEADER = nHbcd... ,

where $n(\leq 60)$ is the number of characters in the header and where bcd... is a string of n characters (including blanks). The terminating comma is required and must not be included in the character count.

Example

HEADER = 21H THIS IS AN EXAMPLE.,

2.1.4 Numeric Data Entries

Data cards other than header cards consist of one or more entries of the form

$$\alpha = b_1, b_2, b_3, \ldots , b_n,$$

where α is the name of a real (i.e., floating point) or integer variable and may have one, two, three, or no positive integer subscripts and where $b_1, b_2, b_3, \ldots , b_n$ are decimal constants separated by commas and are to be stored in n successive locations beginning with the location corresponding to the symbol α. With the possible exception of the last data card, a comma should always immediately follow the last value on a card. For example, a card punched in the format

ØMEGA = 1.3,MAS = 8000,I = 2,Z(5) = -1.8,3,6.15E-6,B = 1,2,3, would indicate the following:

1) The value 1.3 is to be entered for ØMEGA,
2) The value 8000 is to be entered for MASS,
3) The value 2 is to be entered for I,
4) The values -1.8, 3, and 6.15 x 10^{-6} are to be entered for Z(5), Z(6), and Z(7) respectively,
5) The values 1, 2, and 3 are to be entered for B(1), B(2), B(3) respectively.

It should be remarked that the user does not have to be concerned with the mode (i.e., integer of floating point) of decimal input quantities. The program will interpret the proper mode for all decimal input quantities. If no decimal point is input for floating point quantities, the program will assume a decimal point immediately following the rightmost digit (e.g., 10.0, 10., and 10 are all equivalent for floating point quantities. If decimal scaling is desired for floating point quantities, then the notation aEb may be used (where a is a decimal quantity with or without a decimal point and b is the desired scaling factor). Thus to enter 3.123 x 10^{-6} one could write 3.123E-6 or 3123E-9.

If a decimal point appears in the value entered for a quantity which is typed as an integer by the program, the program will truncate rather than round off the value to an integer. Thus the values 10, 10., 10.0, 10.4, and 10.9999 will all be stored as the value 10 if the mode of the variable is integer.

An additional feature of Namelist allows the user to specify a repetition factor to indicate the same value to be stored in several consecutive locations. Thus the notation n*v indicates n consecutive values of v.

2.1.5 End-of-Data

Each category of input data (i.e., the General Input Data and the Array Data) must be terminated by a $ immediately following the last item on the last card containing data entries for that category or by a $ in column 2 of a card immediately following the last card containing data entries. Any characters punched to the right of the $ are ignored. Hence the remainder of the end-of-data card may be used for comments.

2.1.6 Stacking Cases

The user may "stack cases" (i.e., request the program to execute more than one case) by placing the input decks for the second and subsequent cases in order behind the input deck for the first case. Because of the dynamic storage feature used in the BLADER program, the input data for each case must be complete. There is no carryover of data from one case to the next. Also because certain coefficients are built into the program by means of DATA statements, the user is not permitted to stack a plane strain analysis problem behind a axisymmetric strain analysis problem.

Example

$AS189	Name card for General Input Data.
——— ——— ——— ——— ———	Cards specifying General Input Data for case 1.
$END	End-of-data card
$DATA	Name card for Array Data
——— ——— ——— ———	Cards specifying Array Data for case 1
$END	End-of-data card
$AS189	
——— ——— ——— ———	General Input Data for case 2.

$END

$DATA

______
______ Array Data for
______ case 2.
______

$END

2.2 Summary Listing of Input Quantities

This section consists of a listing of the symbolic names of all possible input quantities for the BLADER program. For the convenience of the potential user, the quantities are divided into subsets according to the nature of their usage. First of all, the quantities are separated into the two major categories mentioned in the introduction (section 2.1): (1) General Input Data, and (2) Array Data. As mentioned in the introduction this is a physical as well as a logical separation. Each of the two major categories may be subdivided into several subsets according to logical usuage. However, this second level of subdivision is strictly a logical separation. No physical separation is required except that certain combinations of input quantities will never appear together because some quantities are defined for one particular option whereas other quantities are not defined, or at least are not input for that option. For example the array NP must be input for the standard strain analysis option, but will be calculated for the Bladder Analysis option. On the other hand the array DEPTH must be input for the Bladder Analysis option, but isn't even defined for the standard strain analysis option.

For convenience the nominal values of optional quantities are indicated by values or expressions enclosed by brackets, e.g., [0]. Also the dimensions of arrays are indicated by values or expressions enclosed by parentheses, e.g., (NUMEL).

2.2.1 General Input Data

A. Quantities which are always required. (At present there are no quantities in this particular category.)

B. Quantities which are optional regardless of the value of BLANAL

HEADER	[blanks]
RECCYL	[0]
NUMAT	[1]
NNDLIM	[0]
MAXST	[25]
ITER	[1 for double fold option; 10 otherwise]
CØNVER	[0.1 for double fold option; 10^{-4} otherwise]
IPRIN	[1]
IRESI	[0]
MAXBW	[50]
MAXBKT	[25000]
RESTRT	[0]
SAVETP	[0]
IRSIN	[1 on CDC 6500]
IRSØUT	[3 on CDC 6500]

C. Quantities which are required only for the Bladder Analysis option.

BLANAL
NLAYER
RFINAL

D. Quantities which are optional only for the Bladder Analysis option.

LWRAT	[2]	(single fold option only)
NXINT	[LWRAT*NLAYER]	(single fold option only)

XLNGTH [LWRAT*BLTHNS where $BLTHNS=\sum_{i=1}^{NLAYER} DEPTH(i)$]

(single fold option only)

XSTART [0] (single fold option only)

YSTART [0] (single fold option only)

NANGIN [18] (double fold option only)

RPTRAT $\begin{bmatrix} 1000 & \text{for single fold option,} \\ 10 & \text{for double fold option} \end{bmatrix}$

RSTRAT [100] double fold option only

RPFØLD [RPTRAT*BLTHNS]

RSFOLD [RSTRAT*BLTHNS] (double fold option only)

FLIFE [0]

NCYCF [0]

E. Quantities which are required only for the standard strain analysis option.

NUMEL

NUMNP

NUMCP

NUMBC

NLOAD }
NID } One quantity must be non-zero; the other must be zero.

F. Quantities which are optional for the standard strain analysis option.

NPD [0]

PRESS ignored if NPD≠0.

2.2.2 Array Data

A. Arrays which are always required

VEM (NMAT)

VXI (NMAT)

VXU (NMAT)

VYP (NMAT)

B. Arrays which are optional regardless of the value of BLANAL.

DLIM	(3,NNDLIM) - required if NNDLIM > 0.	
LSSAVE	(2, 20) - required if SAVETP≠0.	
XØRD	(NUMNP)	May be input only on
YØRD	(NUMNP)	a restart run.

C. Arrays which are required only for the Bladder Analysis option.

DEPTH (NLAYER)

D. Arrays which are optional only for the Bladder Analysis option.

MLAYER	(NLAYER) (required if NMAT > 1)	
CYCTØF	(40, NCYCF) — (used only if FLIFE≠0)	
CYCM	(NLAYER)	Used only if FLIFE≠0.
CYCZ	(NLAYER)	May be input or calculated.

E. Arrays which are required only for the standard strain analysis option.

NP	(6,NUMEL)	
CØRNER	(3,NUMCP)	
VTH	(NMAT)	
IBCØND	(2,NUMBC)	
DISPL	(3,NID) - required if NID > 0.	
LØADE	(NLØAD)	required if NLØAD > 0.
LØADS	(NLØAD)	

F. Arrays which are optional for the Standard Strain Analysis option.

DELTAP	(NDP) - required if NDP > 0.
MATEL	(NMAT) - required if NMAT > 0.
VPR	(NLØAD)

2.3 Definition of Input Symbols

2.3.1 General Input Quantities

1) HEADER - Input header (See section 2.1.3, "Header Card" for details)

2) BLANAL - Bladder Analysis option flag:

   ≠0 if Bladder Analysis option is desired;

   =0 if Standard Strain Analysis option is desired.

   [nominal value is 0]

3) RECCYL - Coordinate System option flag:

   =0 if rectangular coordinates are desired (single fold analysis if BLANAL≠0);

   ≠0 if cylindrical coordinates are desired (double fold analysis if BLANAL≠0).

   [nominal value is 0]

4) NMAT - Number of Materials:

   If NMAT >1, then the material code of each element must be entered in the array MATEL.

   [nominal value is 1]

5) NNDLIM - Number of limiting displacement entries:

   If NNDLIM >0, then the limiting displacements are specified in the array DLIM.

   [nominal value is 0]

6) MAXST - Number of load steps to be used.

   [nominal value is 25]

7) ITER - Maximum of iterations to be used for each load step.

   $\left[\text{nominal value is} \begin{cases} 1 & \text{for double fold option} \\ 10 & \text{otherwise.} \end{cases}\right]$

8) CØNVER Convergence criterion:

$$\text{If} \quad \frac{\left[\sum_{i=1}^{N}\left(DF(i)\right)^2\right]^{1/2}}{\left[\sum_{i=1}^{N}\left(F(i)\right)^2\right]^{1/2}} \leqslant \quad C\emptyset NVER,$$

the solution is considered to have converged.

N = 2*NUMNP,

DF(i) = residual load for i^{th} degree of freedom,

F(i) = accumulated load for i^{th} degree of freedom.

$$\left[\text{nominal value} = \begin{Bmatrix} 0.1 & \text{for double fold option} \\ 10^{-4} & \text{otherwise} \end{Bmatrix}\right]$$

9) IPRIN - Detailed print interval:

The point strains and stresses will be printed out every IPRIN-th load step starting with load step IPRIN.

[nominal value is 1]

10) IRESI - Load residual option flag:

If IRESI≠0, then at the start of each load step except the first, the incremental load vector DF will be set equal to the residual load vector from the previous load step.

If IRESI=0, then at the start of each load step, the incremental load vector DF will be cleared to zero.

[nominal value is 0]

11) MAXBW - Maximum band width: i.e., 2* (maximum difference in node numbers of any single element) + 2.

For the Bladder Analysis option MAXBW must be ≥ 8* NLAYER + 10.

[nominal value is 50]

12) MAXBT - Maximum size of the Dynamic Storage array. See section 6.2 "Dynamic Storage Array" and section 5.6 "Memory Requirements" for information describing how to estimate the size of the Dynamic Storage array.

[nominal value is 25000]

13) RESTRT - Restart dump number:

= n>0 if the current run is to be restart from dump n of the input restart tape;

= 0 otherwise

[nominal value is 0]

14) SAVETP - Restart flag:

$\neq 0$ if is is desired to save data periodically on a "save tape" for a possible restart in a subsequent run;

= 0 otherwise.

If SAVETP $\neq 0$, then the frequency at which the data is saved on the output restart tape must be specified by the array LSSAVE.

[nominal value is 0]

15) IRSIN : Internal file name for input restart tape.

[nominal value is 1 for CDC 6500;
see section 6.6.4 of this report]

16) IRSØUT : Internal file name for output restart tape.

[nominal value is 3 for CDC 6500;
see section 6.6.4 of this report]

17) NLAYER - Number of element layers for the Bladder Analysis option.

[no nominal value; must be specified]

18) RFINAL - R_f = final inner folding radius.

[no nominal value; must be specified]

19) LWRAT - Ratio of the total length in the x-direction to the total bladder thickness and/or the ratio of the number of intervals in the x-direction to the number of element layers (for the single fold bladder analysis option). See NXINT and XLNGTH below.

[nominal value is 2]

20) NXINT - Number of intervals in the x-direction (for the single fold option).

[nominal value is LWRAT*NLAYER]

21) XLNGTH - Total length in the x-direction (for the single fold option).

[nominal value is LWRAT*BLTHNS where

$$\text{BLTHNS} = \sum_{i=1}^{\text{NLAYER}} \text{DEPTH}(i)$$

is the total "depth" of the bladder area]

22) XSTART - x_o = x-coordinate of the left edge of the rectangular area (for the single fold option).

[nominal value is 0]

23) YSTART - y_o = y-coordinate of bottom edge of the rectangular area (for the single fold option).

[nominal value is 0]

24) NANGIN - Number of angular intervals (for the double fold option).

[nominal value is 18 corresponding to angular spacing of 10°]

25) RPTRAT - $\frac{R_p}{t}$ = ratio of primary fold radius R_p to total bladder thickness t, where

$$t = \sum_{i=1}^{\text{NLAYER}} \text{DEPTH}(i).$$

[nominal value is 1000 for the single fold option and 10 for the double fold option]

26) RSTRAT - $\frac{R_s}{t}$ = ratio of secondary fold R_s to total bladder thickness t (for the double fold option).

[nominal value is 100]

27) RPFØLD - R_p = initial radius of primary fold

(inner fold radius for double fold;
radius to median line for single fold.)

[nominal value is RPTRAT*BLTHNS where BLTHNS = t =
total bladder thickness]

28) RSFØLD - R_s = inner radius of secondary fold (for double fold option).

[nominal value is RSTRAT*BLTHNS]

29) FLIFE - Fatigue life option flag.

$\neq$ 0 if the fatigue life is to be calculated and printed out;

= 0 otherwise.

[nominal value is zero]

30) NCYCF - Maximum number of pairs of entries ε_i and N_i in the CYCTØF table for any material, where

ε_i = plastic strain,

N_i = corresponding value of the fatigue life.

[nominal value is 0;
must not exceed 20]

31) NUMEL - Number of elements.

32) NUMNP - Number of nodal points.

33) NUMCP - Number of corner points.

34) NUMBC - Number of boundary condition entries in IBCØND array.

35) NLØAD - Number of elements which are pressure loaded:

a) If BLANAL = 0, then a non-zero value must be specified for either NLØAD or NID (but not both).

b) If NLØAD > 0, then NLØAD specifies the number of entries in each of the arrays LØADE and LØADS (i.e., the number of elements which are pressure loaded).

c) If RECCYL = 0, then NLOAD must be ≥ 0.

d) If RECCYL $\neq$ 0, then a negative value may be entered for NLØAD. If NLØAD = -n (where $n > 0$), then n is the index of the node which is used as a reference for changing the radius of curvature of the secondary fold.

[nominal value is 0]

36) NID - Number of imposed displacement constraints.

a) See NLØAD above for compatibility requirements.

b) If NID > 0, then NID specifies the number of imposed displacement specifications in the array DISPL.

[nominal value is 0]

37) NDP - Number of pairs of entries in the DELTAP array.

[nominal value is 0]

38) PRESS - Total pressure to be loaded:

a) Ignored if NDP $\neq$ 0, or if BLANAL $\neq$ 0.

b) If BLANAL = 0, if NLØAD > 0, and if NDP = 0, then each side S_i loaded will be loaded in MAXST equal load steps each of magnitude

$$\Delta P_i = \frac{P_i}{MAXST.}$$

If PRESS is entered, then P_i = PRESS for each side loaded. If PRESS is not entered then P_i = VPR(i).

c) If BLANAL = 0 and if NLØAD = $-n < 0$, then the structure will be loaded by changing the radius of curvature of node n. The change in the radius will be given by the table DELTAP if NDP > 0 or by PRESS if NDP = 0.

2.3.2 Array Data

With the exception of the array LSSAVE, which is described below, all of the input arrays are stored in the Dynamic Storage Array and are described in detail in section 6.2.4 of this document. Section 6.2.3 lists all of the arrays in alphabetical order for easy reference.

LSSAVE - 2*10 array in which LSSAVE(1,i) specifies N_i and LSSAVE(2,i) specifies n_i, which are used to control the frequency at which dumps of the program data are taken for a possible restart in a subsequent run (provided that SAVETP $\neq$ 0).

Dumps of the restart data are made on load step n_1 and every n_1 load steps until load step N_1 is reached. In general dumps of the restart data are made every n_i load steps for load steps between N_{i-1} and N_i.

A maximum of ten pairs of values (N_i, n_i) may be entered.

NI

Page of

Priority

NAMELIST INPUT FORM

1

Keypunched by

Verified by

General Input Data for BLADER (AS189) 80 COMMENTS

P	$AS189	[P in column 1⟹print card images (TRW only)]
	HEADER=60H	[followed by 60 characters including blanks]
	RECCYL=	[1 for cylindrical coordinates; no entry otherwise]
	NMAT=	[number of materials; no entry⟹1]
	NNDLIM=	[number of limiting displacements]
	MAXST=	[number of load steps desired]
	ITER=	[max. no. of iterations]
	CØNVER=	[convergence criterion]
	IPRIN=	[detailed print interval]
	IRESI=	[residuals option flag]
	MAXBW=	[maximum allowable band width]
	MAXBKT=	[max. size of dynamic storage array]
	RESTRT=	[n to restart from dump n]
	SAVETP=	[1 to save data for a restart]
	BLANAL=	[1 for bladder analysis; no entry otherwise]
	NLAYER=	[no. of element layers]
	RFINAL=	[final folding radius]
	FLIFE=	[1 if fatigue life calcualtions are desired]

[Additional input quantities are described in section 2.2.1]

$END

NL

Date ________ Page ____ of ____

Name ________ Priority ________

Problem No. ________ Keypunched by ________

NAMELIST INPUT FORM

1

No. of Cards ________ Verified by ________

2 TITLE: Array Data for BLADER (AS189) 80 COMMENTS

P $DATA [P in column 1 ⟹ Print card images (TRW only)]

VEM= [elastic modulus for each material]

VXI= [Plastic/elastic modulus ratio for each material]

VXU= [Poisson's ratio for each material]

VYP= [yield point for each material]

DLIM= [limiting displacement specifications]

DEPTH= [depth of each element layer]

MLAYER= [material code for each layer]

NL $END

2.5 Restart Procedure

2.5.1 Introduction

An option available in the BLADER program allows a problem which has been interrupted for some reason in one computer run to be continued in a subsequent computer run. In order to restart an interrupted problem the user must have instructed the program to save the necessary data on a restart tape during the initial run. He does this by appropriate entries in the Namelist input of the initial run and by requesting the restart tape to be saved by having the appropriate control cards inserted in the deck. Then in the restart run the user must request the saved restart tape to be mounted again by having appropriate control cards inserted in the deck, and must enter RESTART ≠ 0 in the Namelist input.

2.5.2 Initial Run

1) Enter SAVETP = 1 in the General Input Data.
2) Enter the LSSAVE array to specify the frequency of the restart dumps. See section 2.3.2 for a description of the array LSSAVE.
3) Insert the appropriate control cards to have the restart tape saved.

2.5.3 Restart Run

1) Enter RESTRT = n, where n is the number of the dump corresponding to the load step from which the user wishes to continue. The dump number is printed out during the initial run each time a dump of the restart data is taken.
2) Insert the appropriate control cards to have the previously generated restart tape mounted.

2.5.4 Modifying Data on a Restart Run

On a restart run the user is allowed to modify the values of some input quantities through the Namelist input. The original intention of the programmer was to allow the user as much freedom as possible in modifying the problem on a restart run, but several programming considerations (the most important of which is the Dynamic Storage feature) made it necessary to impose some restrictions on the quantities which may be modified on a restart run. Thus major modifications such as changing coordinate systems or changing the dimensions of the problem are not allowed. In most cases these restrictions do not impose unreasonable limitations on the user. It is unlikely that the user would want to change coordinate systems, or to change the dimensions of the problem in the middle of a given problem.

Below, an attempt has been made to indicate those quantities which may be safely modified on a restart run and those quantities which may not be modified.

1) Quantities which may be modified on any restart run:

HEADER

ITER

CØNVER

IPRIN

IRESI

RESTRT

SAVETP

VEM

VXI

VXU

VYP

DLIM (however it is not possible to remove a constraint once a node has reached its limiting displacement.)

LSSAVE

XØRD, YØRD — The arrays XØRD and YØRD are not input on an initial run, but the coordinates of any node may be modified on a restart run.

2) Additional quantities which may be modified only if BLANAL $\neq$ 0:

CYCM

CYCZ

3) Additional quantities which may be modified only if BLANAL = 0:

PRESS : The value of PRESS may be modified, but the method of loading may not be changed. Thus if PRESS was not entered on the initial run, it must not be entered on a restart run.

MAXST

VTH

LØADE

LØADS

DELTAP

MATEL

VPR

4) Quantities which must never be modified:

RECCYL

BLANAL

NLAYER

RFINAL

LWRAT

NXINT

XLNGTH

XSTART

YSTART

NANGIN

RPTRAT

RSTRAT

RPFØLD

RSFØLD

FLIFE

NCYCF

NUMCP

DEPTH

MLAYER

CYCTØF

CØRNER

DISPL

5) Quantities which might be modified with some risk.

A word of explanation is appropriate here. It is recommended that none of the quantities listed below be modified in a restart run. Changing the problem dimensions on a restart is at best hazardous. At worst it could be disasterous. However, in case someone feels particularly adventuresome we list these quantities in the hope that the program won't completely blow its mind if some user desires to change the problem dimensions. One final point is that because of the Dynamic Storage feature used in this program it definitely is <u>not possible to decrease any problem dimensions.</u>

NMAT

NNDLIM

MAXBW

MAXBKT

NUMEL

NUMNP

NUMBC

NLØAD
NID
NDP — May not change method of loading

NP

IBCØND — The boundary conditions are cummulative. It may be possible (but not advisable) to add boundary conditions on a restart run, but existing boundary conditions may not be deleted.

3.0 OUTPUT

There are two basic types of printed output: the printout produced by the operating system and the printout produced by the program itself. The system generated printout consists of such things as the load map, the listing of Namelist card images, the data or program dumps, and the dayfile, and is more or less not under the control of the program. The program generated printout consists of everything printed directly under the control of the program by means of Fortran WRITE and FORMAT statements. The two different types of output may appear interspersed throughout the output listing and may even appear on the same page.

The main reason for distinguishing between the two different types of printout is that while the program generated output is more or less independent of the computing system which the program was run, the system generated output is very much dependent on the computing system. For example, the feature which allows the listing of the Namelist card images may be unavailable on some computing systems.

Below we shall list and, in some cases discuss, the various categories of printout that may be produced. The system generated printout will be indicated by placing "(system)" after each appropriate category. In addition, comments enclosed in parentheses will be used to indicate the conditions under which certain categories of printout will be produced. The categories will be listed in the approximate order in which they may be produced.

1) Load map: (system) - see section 5.4 for a load map produced on the CDC 6500 at TRW.

2) Listing of card images for the General Input Data: (system).

3) Input Parameters: This category consists of the values of parameters such as the number of elements, the maximum number of load steps, etc., which may be input or, in some cases, generated by the program.

4) Listing of card images for the Array Data: (system).

5) Bladder Analysis Parameters: (BLANAL $\neq$ 0) -
This category consists of such quantities as number of element layers, total length, final fold radius, etc.

6) Nodal coordinates: (BLANAL $\neq$ 0) -
The x and y coordinates for the single fold option or the r and z coordinates for the double fold option are printed out for each nodal point.

7) Fatigue life as a function of plastic strain for each material: (BLANAL $\neq$ 0 and FLIFE $\neq$ 0).

8) Plastic strain coefficients: (BLANAL $\neq$ 0 and FLIFE $\neq$ 0)
M and z are printed out for each material.

9) Material and geometric properties: The elastic modulus, Poisson's ratio, etc. are printed out for each material.

10) Variable material specification: (NMAT < 1) -
The material code is print out for each element.

11) Corner point array: (BLANAL = 0) -
The coordinates (x and y or r and z) of each corner point are printed out.

12) Element array: The node numbers n_1, n_2, ..., n_6 for each element are printed out under the headers I, J, K, L, M, and N respectively.

13) Boundary conditions: (BLANAL = 0) -
The nodal index and the constraint code are printed out for each boundary condition under the headings PØINT and NFIX respectively. See the description of the input array IBCØND (item 40) in section 6.2.4 of this document for details.

14) Nodal displacement limits: (NNDLIM > 0) -
The nodal index, the direction code, and the magnitude of the displacement are printed out for each limiting displacement constraint. See the description of DLIM (item 30) in section 6.2.4.

15) Pressure loading specifications: (NLØAD > 0) -
The element index and the side code for each pressure loading condition are printed out. If, in addition, neither of NDP and PRESS is specified, then the pressure loading for each side loaded is also printed out. See the descriptions of LØADE, LØADS, and VPR in section 6.2.4 of this document.

16) Imposed displacements: (BLANAL = 0) -
The nodal index, the direction code and the magnitude of each imposed displacement constraint are printed out. See the description of DISPL (item 39) in section 6.2.4.

17) The band width of the stiffness matrix is printed out opposite the heading BAND WIDTH.

18) The total length of the dynamic storage array is printed out as a decimal integer and as an octal integer.

19) Load step and iteration information:
   a) Load step number.
   b) Load step size(s) $(\Delta P)_i$ and total load(s) P_i:
      i) If NLØAD > 0 and if either NDP > 0 or PRESS was specified in the Namelist input, then $(\Delta P)_i$ is the current value of the incremental pressure loading and P_i is the accumated pressure load. The two values $(\Delta P)_i$ and P_i apply to all sides loaded.
      ii) If NLØAD > 0 and neither NDP nor PRESS was specified, then $(\Delta P)_i$ is the incremental pressure load for the ith side loaded and P_i is the corresponding accumulated pressure loading. See the descriptions of LØADE, LØADS, DPR and TPR in section 6.2.4 of this document.
      iii) If NLØAD = - n < 0, then $(\Delta P)_i$ is the current value of the incremental change in primary fold radius and P_i is the accumulated change in fold radius (i.e. $r - r_p$) of node n.

19) Load step and iteration information: (continued)

iv) If NLØAD = 0 (hence NID $\neq$ 0) and if BLANAL = 0, then $(\Delta P)_i$ and P_i have no significance.

v) If BLANAL $\neq$ 0 and RECCYL = 0 (single fold bladder analysis), then $(\Delta P)_i = 0$ has no significance, but P_i specifies the current value of the fold radius (measured from the origin to the median line).

vi) If BLANAL $\neq$ 0 and RECCYL $\neq$ 0 (double fold bladder analysis), then $(\Delta P)_i$ is the incremental change in the secondary inner fold radius and IPRESS is the accumulated change in the secondary inner fold radius.

c) Force norm:

$$\|F\| = \left[\sum_{i=1}^{numnp} F_{xi}^2 + \sum_{i=1}^{numnp} F_{yi}^2 \right]^{1/2}$$

is printed out for each load step.

d) DF norm

$$\|DF\| = \left[\sum_{i=1}^{numnp} (\Delta F_{xi})^2 + \sum_{i=1}^{numnp} (\Delta F_{yi})^2 \right]$$

e) DF/F norm ratio

For each iteration the DF norm $\|DF\|$ and the ratio $\frac{\|DF\|}{\|F\|}$ are printed out.

20) Total force and the total moments: (RECCYL $\neq$ 0) -

For each load step, the total force f_θ and the total moments M_r and M_z are printed out.

21) Fold radius: (NLØAD < 0) -

If NLØAD = $- n < 0$, then the current value of the primary fold radius $R = r_n$ is printed out for each load step.

22) Radius of curvature: (single fold analysis) -
For each load step, the inner radius of curvature at the centerline ($x = 0$) is printed out.

23) Nodal point forces and displacements -
The x and y (for rectangular coordinates) or the r and z (for cylindrical coordinates) components of displacement, accumulated loads, and residual loads are printed out for each nodal point for each load step.

24) Estimated fatigue life (FLIFE and BLANAL $\neq 0$) -
The estimated fatigue life and the maximum equivalent plastic strain, and the index of the node at which the maximum equivalent plastic strain occurs is printed out for each element layer for each load step.

25) Point strains -
At load step intervals determined by the input parameter IPRIN, the elastic, plastic and total strains are printed out for each nodal point. If RECCYL = 0, then the following components of strain are printed out for each type of strain:

$$\varepsilon_x, \quad \varepsilon_y, \quad \gamma_{xy}$$

If RECCYL $\neq 0$, then the following components of strain are printed out for each type of strain

$$\varepsilon_r, \quad \varepsilon_z, \quad \varepsilon_\theta, \quad \gamma_{rz}$$

26) Point stresses -
At load step intervals determined by the input parameter IPRIN, the nodal coordinates (x and y or r and z), the point stresses (σ_x, σ_y, σ_{xy}, σ_z, and $\bar{\sigma}$ if RECCYL = 0, or σ_r, σ_z, σ_θ, τ_{rz}, and $\bar{\sigma}$ if RECCYL $\neq 0$), and the accumulated plastic work and the incremental plastic work (DWP) are printed out for each nodal point.

27) Dump of LBUCKT array and Dynamic Storage Array (System) - If the solution fails to converge, the program calls the library routine PDUMP to dump out the LBUCKT array, which specifies the D.S.A. pointers, and the Dynamic Storage Array. See section 6.2 for a description of LBUCKT and the D.S.A.

4.0 SAMPLE CASE

4.1 Description of Sample Problem

In this section, a simple single fold aluminum bladder problem is presented in order to illustrate some of the program features. The bladder finite element model consisted of six elements, as shown in Figure 4-1. The material properties are:

$$E = 200 \text{ KSI}$$
$$\mu = 0.33$$
$$X = .167$$
$$\sigma_y = 10 \text{ KSI}$$

The other pertinent input quantities are given in the next sections.

4.2 Load Sheets for Sample Problem

The load sheet for the example problem is shown on the next page.

4.3 Printout for Sample Problem

Following the load sheet is the printout for the single fold problem of Figure 4-1. For brevity, only load steps 1 and 20 (the first and last steps) are shown.

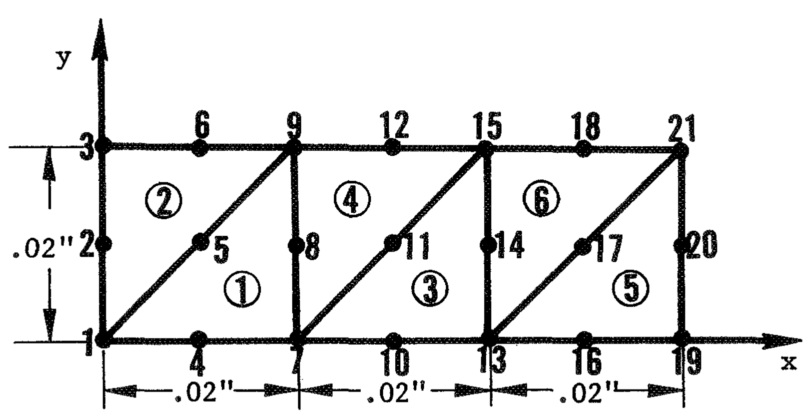


Figure 4-1: Sample Single Fold

Date

Page of

Name

Priority

Problem No.

NAMELIST INPUT FORM

Keypunched by

No. of Cards

1

Verified by

TITLE: 2 80 COMMENTS

P
```
$AS189
CØNVER = 1.0E-03,
HEADER= 17HSINGLE①FØLD①MØDEL,
BLANAL= 1, FLIFE =1, NCYCF= 2,
NLAYER = 1,
LWRAT = 3,
RFINAL= 0.05,
MAXST = 20,
$END
$DATA
VEM= 200., VXI = 0.33, VXU = 0.167, VYP = 10.,
DEPTH = 0.02, CYCTØF = .28, 10,
.008, 10000,
$END
```

NL

SYSTEMS 2554 REV. 8-65

LARGE STRAIN, ELASTOPLASTIC FINITE ELEMENT ANALYSIS

PROGRAM AS189 (BLADER)

```
$AS189
CONVER=1.0E-03,
HEADER=17HSINGLE FOLD MODEL,
BLANAL=1,FLIFE=1,NCYCF=2,
NLAYER=1,
LWRAT=3,
RFINAL=0.05,
MAXST=20,
$END
```

SINGLE FOLD MODEL

PLANE STRAIN ANALYSIS

NO. OF ELEMENTS	6
NO. OF CORNER POINTS	8
NO. OF NODAL POINTS	21
NO. OF BOUNDARY CONDITIONS	3
NO. OF ELEMENTS LOADED	0
NO. OF IMPOSED DISPLACEMENTS	6
NUMBER OF MATERIALS	1
NUMBER OF LOAD INCREMENT ENTRIES	0
MAXIMUM NUMBER OF STEPS	20
MAXIMUM BAND WIDTH	50
NO. OF DISPL. LIMITS	0
MAX. NO. OF ITERATIONS	10
NO. OF STEPS PER DETAILED PRINT	1
IRESI FLAG	0
CONVERGENCE CRITERION	1.0000E-03
TOTAL PRESSURE LOADING	-0.

NO. OF ELEMENT LAYERS	1
LENGTH TO WIDTH RATIO	3
NO. OF ANGULAR INTERVALS	0
NO. OF X-INTERVALS	3
TOTAL LENGTH	6.00000E-02
PRIMARY FOLD RADIUS	2.00000E+01
SECONDARY FOLD RADIUS	2.00000E+01
FINAL FOLDING RADIUS	5.00000E-02

NODAL COORDINATES

POINT	X-ORD	Y-ORD
1	0.00000	0.00000
2	0.00000	.01000
3	0.00000	.02000
4	.01000	0.00000
5	.01000	.01000
6	.01000	.02000
7	.02000	0.00000
8	.02000	.01000
9	.02000	.02000
10	.03000	0.00000
11	.03000	.01000
12	.03000	.02000
13	.04000	0.00000
14	.04000	.01000
15	.04000	.02000
16	.05000	0.00000
17	.05000	.01000
18	.05000	.02000
19	.06000	0.00000
20	.06000	.01000
21	.06000	.02000

PLASTIC STRAIN	NO. OF CYCLES TO FAILURE	FOR MATERIAL 1
2.80000E-01	10	
8.00000E-03	10000	

PLASTIC STRAIN COEFFICIENTS

MATERIAL	M	Z
1	9.15899E-01	-5.14689E-01

MATERIAL AND GEOM. PROPERTIES

ELASTIC MODULUS	200.00
POISSONS RATIO	.330
INITIAL THICKNESS	1.000
PLASTIC/ELASTIC MODULUS RATIO	1.670E-01
INITIAL YIELD POINT	10.0

ELEMENT ARRAY

ELEMENT	I	J	K	L	M	N
1	1	7	9	4	8	5
2	1	9	3	5	6	2
3	7	13	15	10	14	11
4	7	15	9	11	12	8
5	13	19	21	16	20	17
6	13	21	15	17	18	14

BAND WIDTH = 18

LENGTH OF DYNAMIC STORAGE 2089 (DECIMAL) 004051 (OCTAL)

LOAD STEP 1 = 0. TOTAL LOAD = 2.00000E+01

FORCE NORM = 9.58679E-04

DF NORM	DF/F NORM RATIO
3.53526E-05	3.68764E-02
3.24195E-09	3.38168E-06

RADIUS OF CURVATURE = 1.949522E+01

NODAL POINT FORCES AND DISPLACEMENTS - STEP 1

POINT	X-DIS	Y-DIS	X-LOAD	Y-LOAD	X-RESIDUAL	Y-RESIDUAL
1	0.	7.658808E-07	3.833611E-04	0.	0.	1.786903E-12
2	0.	2.021878E-06	-9.560433E-07	0.	0.	-5.065420E-13
3	0.	7.576803E-07	-3.833394E-04	0.	0.	-1.783586E-12
4	-5.128054E-06	-1.796221E-06	0.	0.	-8.677939E-11	8.451294E-12
5	3.717133E-10	-5.410967E-07	0.	0.	-8.726615E-11	-7.669468E-14
6	5.119467E-06	-1.806739E-06	0.	0.	8.291152E-11	-8.811443E-12
7	-1.025087E-05	-9.462136E-06	0.	0.	2.489247E-10	3.182493E-12
8	1.313332E-09	-8.228513E-06	0.	0.	4.029522E-10	1.694121E-12
9	1.023231E-05	-9.501682E-06	0.	0.	-1.459898E-10	-6.731093E-12
10	-1.540827E-05	-2.225454E-05	0.	0.	-2.609954E-10	2.436666E-11
11	5.022494E-09	-2.103397E-05	0.	0.	-1.415999E-10	-1.020678E-11
12	1.531083E-05	-2.230468E-05	0.	0.	2.365009E-10	-3.241448E-11
13	-2.056081E-05	-4.008290E-05	0.	0.	-1.623822E-09	6.529475E-11
14	4.267225E-08	-3.895127E-05	0.	0.	-1.864845E-09	5.926284E-11
15	2.034053E-05	-4.026622E-05	0.	0.	1.087368E-09	-5.328735E-11
16	-2.558953E-05	-6.300192E-05	0.	0.	5.386724E-10	1.983962E-10
17	1.284580E-07	-6.179329E-05	0.	0.	1.476660E-09	-5.355800E-11
18	2.514371E-05	-6.298858E-05	0.	0.	-4.727146E-10	-2.295150E-10
19	-3.008995E-05	-8.995493E-05	-4.390942E-04	2.307715E-04	0.	0.
20	-8.999996E-08	-8.999993E-05	1.124359E-04	-4.522801E-04	0.	0.
21	2.990996E-05	-9.004493E-05	3.275920E-04	2.215086E-04	0.	0.

LAYER	MAX. EQUIV. PLASTIC STRAIN	ESTIMATED FATIGUE LIFE	AT NODE
1	[illegible].	1[illegible]	1

POINT STRAINS

ELASTIC STRAINS

POINT	EPS-X	EPS-Y	EPS-XY
1	-5.121721E-04	2.523380E-04	4.378696E-07
2	4.982439E-07	-3.782582E-07	2.281933E-07
3	5.124070E-04	-2.523979E-04	8.378513E-08
4	-5.121493E-04	2.511468E-04	2.106868E-06
5	1.634005E-07	1.028998E-07	-9.525310E-07
6	5.120079E-04	-2.522802E-04	-6.691594E-07
7	-5.102340E-04	2.507891E-04	5.230091E-06
8	2.237661E-06	-1.421387E-06	8.800281E-07
9	5.119805E-04	-2.521220E-04	-1.415030E-06
10	-5.140617E-04	2.457596E-04	2.439186E-05
11	2.172064E-06	3.233222E-07	-9.980864E-06
12	5.068518E-04	-2.510793E-04	-7.553479E-06
13	-4.996008E-04	2.412966E-04	2.861455E-05
14	9.451240E-06	-7.042507E-06	-1.044274E-05
15	5.041377E-04	-2.511903E-04	-3.048364E-05
16	-4.731579E-04	1.242810E-04	7.673845E-05
17	-1.725182E-05	6.517993E-05	-1.556522E-05
18	4.817799E-04	-2.386251E-04	-9.164963E-07
19	-4.193615E-04	1.586411E-08	1.017655E-04
20	9.229100E-06	4.932141E-09	-2.230864E-05
21	4.584959E-04	-1.127503E-04	-5.433398E-05

PLASTIC STRAINS

POINT	EPS-X	EPS-Y	EPS-XY
1	0.	0.	0.
2	0.	0.	0.
3	0.	0.	0.
4	0.	0.	0.
5	0.	0.	0.
6	0.	0.	0.
7	0.	0.	0.
8	0.	0.	0.
9	0.	0.	0.
10	0.	0.	0.
11	0.	0.	0.
12	0.	0.	0.
13	0.	0.	0.
14	0.	0.	0.
15	0.	0.	0.
16	0.	0.	0.
17	0.	0.	0.
18	0.	0.	0.
19	0.	0.	0.
20	0.	0.	0.
21	0.	0.	0.

TOTAL STRAINS

POINT	EPS-X	EPS-Y	EPS-XY
1	-5.121721E-04	2.523380E-04	4.378696E-07
2	4.982439E-07	-3.782582E-07	2.281933E-07
3	5.124070E-04	-2.523979E-04	8.378513E-08
4	-5.121493E-04	2.511468E-04	2.106868E-06
5	1.634005E-07	1.028998E-07	-9.525310E-07
6	5.120079E-04	-2.522802E-04	-6.691594E-07
7	-5.102340E-04	2.507891E-04	5.230091E-06
8	2.237661E-06	-1.421387E-06	8.800281E-07
9	5.119805E-04	-2.521220E-04	-1.415039E-06
10	-5.140617E-04	2.457596E-04	2.439186E-05
11	2.172064E-06	3.233222E-07	-9.980864E-06
12	5.068518E-04	-2.510793E-04	-7.553479E-06
13	-4.996008E-04	2.412966E-04	2.861455E-05
14	9.451240E-06	-7.042507E-06	-1.044274E-05
15	5.041377E-04	-2.511903E-04	-3.048364E-05
16	-4.731579E-04	1.242810E-04	7.673845E-05
17	-1.725182E-05	6.517993E-05	-1.556522E-05
18	4.817799E-04	-2.386251E-04	-9.164963E-07
19	-4.193615E-04	1.586411E-08	1.017655E-04
20	9.229100E-06	4.932141E-09	-2.230864E-05
21	4.584959E-04	-1.127503E-04	-5.433398E-05

POINT STRESSES

POINT	X-ORD	Y-ORD	SIG-X	SIG-Y	TAU-XY	SIG-Z	EQ. STRESS	PL. WORK	DWP
1	0.0000	.0000	-.1149	.0000	.0000	-.0379	.1014	-.00000	0.00000
2	0.0000	.0100	.0001	-.0000	.0000	.0000	.0001	-.00000	0.00000
3	0.0000	.0200	.1151	-.0000	.0000	.0379	.1016	-.00000	0.00000
4	.0100	-.0000	-.1150	-.0003	.0002	-.0381	.1012	-.00000	0.00000
5	.0100	.0100	.0001	.0001	-.0001	.0000	.0001	-.00000	0.00000
6	.0100	.0200	.1150	-.0000	-.0001	.0379	.1015	-.00000	0.00000
7	.0200	-.0000	-.1145	-.0002	.0005	-.0379	.1009	-.00000	0.00000
8	.0200	.0100	.0005	-.0001	.0001	.0001	.0005	-.00000	0.00000
9	.0200	.0200	.1150	.0000	-.0002	.0379	.1015	-.00000	0.00000
10	.0300	-.0000	-.1164	-.0022	.0020	-.0392	.1010	-.00000	0.00000
11	.0300	.0100	.0007	.0004	-.0008	.0004	.0014	-.00000	0.00000
12	.0300	.0200	.1136	-.0004	-.0007	.0373	.1007	-.00000	0.00000
13	.0400	-.0000	-.1127	-.0014	.0024	-.0377	.0985	-.00000	0.00000
14	.0400	.0100	.0018	-.0007	-.0008	.0004	.0028	-.00000	0.00000
15	.0400	.0200	.1128	-.0008	-.0025	.0369	.1004	-.00000	0.00000
16	.0500	-.0001	-.1220	-.0323	.0060	-.0509	.0826	-.00000	0.00000
17	.0500	.0099	.0044	.0168	-.0011	.0070	.0131	-.00000	0.00000
18	.0500	.0199	.1080	-.0004	-.0003	.0355	.0957	-.00000	0.00000
19	.0600	-.0001	-.1242	-.0613	.0078	-.0612	.0644	-.00000	0.00000
20	.0600	.0099	.0027	.0014	-.0017	.0013	.0032	-.00000	0.00000
21	.0600	.0199	.1195	.0335	-.0043	.0504	.0813	-.00000	0.00000

LOAD STEP 20 = 0. TOTAL LOAD = 5.00000E-02

FORCE NORM = 2.33775E-01

DF NORM	DF/F NORM RATIO
4.01617E-01	1.71796E+00
8.01697E-02	3.42935E-01
1.18167E-02	5.05472E-02
1.61309E-03	6.90016E-03
2.33109E-04	9.97151E-04

RADIUS OF CURVATURE = 3.184695E-02

NODAL POINT FORCES AND DISPLACEMENTS - STEP 20

POINT	X-DIS	Y-DIS	X-LOAD	Y-LOAD	X-RESIDUAL	Y-RESIDUAL
1	0.	1.021160E-03	6.856910E-02	0.	0.	-2.269801E-05
2	0.	1.372670E-03	-2.590045E-02	0.	0.	2.721828E-06
3	0.	7.670427E-04	-4.092467E-02	0.	0.	-1.241295E-06
4	-2.275216E-03	8.430014E-05	0.	0.	-6.628115E-05	4.269729E-05
5	-1.139247E-04	1.867411E-04	0.	0.	3.130186E-05	-3.360311E-05
6	1.823293E-03	-4.593395E-04	0.	0.	2.085349E-08	-3.663469E-06
7	-4.631886E-03	-2.329082E-03	0.	0.	3.978130E-05	-2.959242E-05
8	-4.591965E-04	-2.765087E-03	0.	0.	-3.040958E-05	6.073233E-06
9	3.299155E-03	-4.241141E-03	0.	0.	2.009979E-06	1.742350E-06
10	-7.901515E-03	-6.049334E-03	0.	0.	-8.242325E-05	6.081131E-05
11	-2.142203E-03	-8.131859E-03	0.	0.	-4.379866E-05	3.561549E-05
12	3.586590E-03	-1.002829E-02	0.	0.	3.173579E-06	1.655803E-06
13	-1.166700E-02	-1.139498E-02	0.	0.	-1.238756E-05	-2.087417E-05
14	-4.087967E-03	-1.445606E-02	0.	0.	8.491994E-05	-6.459577E-05
15	2.946051E-03	-1.815270E-02	0.	0.	3.153668E-05	1.087102E-06
16	-1.705829E-02	-1.784394E-02	0.	0.	-4.018735E-05	-4.212112E-05
17	-8.615449E-03	-2.288301E-02	0.	0.	-5.188088E-05	1.805868E-05
18	1.636377E-04	-2.738487E-02	0.	0.	6.495686E-05	6.901879E-05
19	-2.271844E-02	-2.550569E-02	1.330019E-02	9.126282E-02	0.	0.
20	-1.339805E-02	-3.188211E-02	-5.197096E-02	-8.192757E-02	0.	0.
21	-4.077655E-03	-3.825853E-02	3.685713E-02	-9.314154E-03	0.	0.

LAYER	MAX. EQUIV. PLASTIC STRAIN	ESTIMATED FATIGUE LIFE	AT NODE
1	1.60519E-01	29	12

POINT STRAINS

ELASTIC STRAINS

POINT	EPS-X	EPS-Y	EPS-XY
1	-1.511348E-01	6.369261E-02	3.116596E-02
2	8.991753E-03	-1.159244E-02	7.008192E-03
3	1.310921E-01	-6.218864E-02	-2.945451E-03
4	-1.647617E-01	6.070876E-02	6.620244E-02
5	2.778889E-03	1.123030E-04	-2.879681E-02
6	1.279247E-01	-5.830863E-02	-1.863958E-02
7	-1.433222E-01	4.487643E-02	7.540625E-02
8	1.200892E-02	-1.149049E-02	-2.330093E-03
9	1.406753E-01	-6.109633E-02	-5.826291E-02
10	-1.352081E-01	1.858073E-02	6.845244E-02
11	5.156610E-04	2.022815E-02	-2.202931E-02
12	1.227705E-01	-4.627971E-02	-4.979743E-02
13	-1.163476E-01	2.425237E-02	1.168757E-01
14	1.760657E-02	-5.666310E-03	-1.952837E-02
15	1.400775E-01	-2.745728E-02	-1.147507E-01
16	-9.125197E-02	-1.830308E-02	1.052340E-01
17	1.054080E-02	5.560251E-02	-6.334958E-03
18	1.264971E-01	-8.911238E-03	-1.343965E-01
19	-4.432264E-02	-8.983309E-02	1.376080E-01
20	6.592196E-02	3.439979E-03	-9.356658E-02
21	1.397606E-01	6.354181E-02	-2.102675E-01

PLASTIC STRAINS

POINT	EPS-X	EPS-Y	EPS-XY
1	-4.911409E-02	4.024583E-02	5.251549E-03
2	0.	0.	0.
3	5.881397E-02	-4.519035E-02	-6.918611E-04
4	-5.298482E-02	4.281836E-02	2.462640E-02
5	0.	0.	0.
6	5.838227E-02	-4.484832E-02	-1.702363E-02
7	-4.407788E-02	3.016300E-02	3.867976E-02
8	0.	0.	0.
9	5.681848E-02	-4.368118E-02	-4.196489E-02
10	-4.940571E-02	2.510067E-02	8.131137E-02
11	5.692954E-03	-1.360277E-02	-1.286526E-02
12	5.602542E-02	-4.358744E-02	-7.807180E-02
13	-2.411862E-02	9.782225E-03	4.687212E-02
14	0.	0.	0.
15	3.532783E-02	-2.644819E-02	-5.781237E-02
16	-3.946496E-03	-7.829902E-03	6.831633E-02
17	2.539619E-03	-1.771035E-02	-1.215639E-02
18	1.722012E-02	-1.217207E-02	-3.402357E-02
19	-3.741617E-03	1.685061E-03	1.474751E-02
20	-8.144097E-03	-9.093747E-03	6.486725E-02
21	1.271695E-02	-2.296931E-02	1.813482E-02

TOTAL STRAINS

POINT	EPS-X	EPS-Y	EPS-XY
1	-2.002489E-01	1.039384E-01	3.641751E-02
2	8.991753E-03	-1.159244E-02	7.008192E-03
3	1.899061E-01	-1.073790E-01	-3.637312E-03
4	-2.177465E-01	1.035271E-01	9.082884E-02
5	2.778889E-03	1.123030E-04	-2.879681E-02
6	1.863070E-01	-1.031569E-01	-3.566321E-02
7	-1.874001E-01	7.503943E-02	1.140860E-01
8	1.200892E-02	-1.149049E-02	-2.330093E-03
9	1.974938E-01	-1.047775E-01	-1.002278E-01
10	-1.846138E-01	4.368139E-02	1.497638E-01
11	6.208615E-03	6.625384E-03	-3.489458E-02
12	1.787959E-01	-8.986715E-02	-1.278692E-01
13	-1.404662E-01	3.403460E-02	1.637478E-01
14	1.760657E-02	-5.666310E-03	-1.952837E-02
15	1.754054E-01	-5.390547E-02	-1.725630E-01
16	-9.519846E-02	-2.613298E-02	1.735503E-01
17	1.308042E-02	3.789216E-02	-1.849134E-02
18	1.437172E-01	-2.108331E-02	-1.684201E-01
19	-4.806426E-02	-8.814803E-02	1.523555E-01
20	5.777786E-02	-5.653768E-03	-2.869933E-02
21	1.524776E-01	4.057250E-02	-1.921326E-01

POINT STRESSES

POINT	X-ORD	Y-ORD	SIG-X	SIG-Y	TAU-XY	SIG-Z	EQ. STRESS	PL. WORK	DWP
1	0.0000	.0010	-18.0866	-2.9341	1.6904	-8.4899	13.7225	1.11458	.00024
2	0.0000	.0114	.9935	-2.0935	.5336	-.3813	2.8337	-.00000	0.00000
3	0.0000	.0208	15.4787	.6009	-.2028	4.4050	13.3920	1.01801	0.00000
4	.0077	.0001	-18.6652	-6.4380	4.8511	-9.9967	13.7570	1.11301	0.00000
5	.0099	.0102	.5196	.9197	-2.1518	.4075	4.0153	-.00000	0.00000
6	.0118	.0195	14.0885	.4187	-2.6338	3.9119	13.1197	.94925	0.00000
7	.0154	-.0023	-16.8432	-9.0750	6.7863	-10.1971	13.8607	1.23691	0.00000
8	.0195	.0072	1.7806	-1.4790	-.9456	.0215	3.3421	-.00000	0.00000
9	.0233	.0158	10.5539	1.6476	-5.2888	3.0727	12.4512	1.00733	0.00000
10	.0221	-.0060	-19.4704	-16.4501	8.3451	-13.9770	15.2193	1.64142	0.00000
11	.0279	.0019	1.7014	5.9432	-.6934	2.1900	4.6314	-.00000	0.00000
12	.0336	.0100	6.4244	.6328	-5.6681	1.1941	11.2690	1.16104	0.00000
13	.0283	-.0114	-14.4040	-16.7843	7.0351	-12.1329	12.9221	.84544	-.00317
14	.0359	-.0045	3.4447	1.9431	-2.7373	.8610	5.2700	-.00000	0.00000
15	.0429	.0018	1.0629	-.1081	-6.4833	-.9882	11.6870	.77451	.00089
16	.0329	-.0178	-15.7560	-23.6009	5.3839	-14.8869	12.4929	.69870	-.00307
17	.0414	-.0129	2.5410	7.1986	2.4865	1.9840	7.1035	-.00000	0.00000
18	.0502	-.0074	-.8793	1.6851	-6.4366	-1.0116	11.4552	.38933	.00306
19	.0373	-.0255	-9.8960	-20.9868	1.2822	-11.5795	10.5878	.15094	-.00098
20	.0466	-.0219	.8453	3.7306	-4.0969	.9326	7.6442	-.00000	0.00000
21	.0559	-.0183	-.0710	5.5418	-5.0334	.5044	10.4377	.28874	.00165

```
03/17/70    SECURE    SCOPE 3.1.5         23 MAR 70
10.09.00.$INPUT=  00884
10.09.03.$FL=060000 CP=00000.000 PP=00008.227 XX=00000.000   FC=065600
10.09.03.$SEQUENCE,P908.
10.09.03.$NAME,532310,35125,TAYLOR,J.
10.09.03.$PROBLEM,400632.
10.09.03.$PROGRAM,AS189E.
10.09.03.$PRIORITY,C.
10.09.03.$MAXTIM,180.         3 MINUTES
10.09.03.$FLENGTH,60000.
10.09.03.SET(1)
10.09.05.INPUT(LC=100000)
10.09.11.TRUNCATED LABEL COMMON  CMESH
10.09.11.LAST PROGRAM PROCESSED  GINPUT
10.09.13.TRUNCATED LABEL COMMON  CMESH
10.09.13.LAST PROGRAM PROCESSED  MSHGEN
10.09.20.TRUNCATED LABEL COMMON  CMESH
10.09.20.LAST PROGRAM PROCESSED  STEP2
10.09.39.FILE OPENED ON DISK-----OUTPUT
10.09.44.$FL=053300 CP=00000.982 PP=00045.831 XX=00000.000   FC=072300
10.09.44.*-- FL USED DURING EXECUTION -(053300)
10.09.44.*-- FL REQUIRED FOR LOADING  -(063100)
10.09.45.$FL=050100 CP=00001.035 PP=00046.346 XX=00000.000   FC=075500
10.09.47.$FL=051400 CP=00001.172 PP=00046.844 XX=00000.000   FC=074200
10.11.05.EXIT
10.11.05. EOF ENCOUNTERED BY INPUTN
10.11.07.$CP  00068.710 SEC.
10.11.07.$PP  00052.110 SEC.
10.11.08.$XX  00000.000 SEC.
10.11.08.$DISK=  000000
10.11.08.EST. PRINTED OUTPUT  01968 LINES
10.11.08.EST. PUNCHED OUTPUT  00000 CARDS
10.11.08.                  0002.18 +  0003.77 =  0005.95
10.11.08.CONTROL POINT 2.
```

5.0 OPERATING PROCEDURES

5.1 Computing System Requirements

5.1.1 Hardware Requirements

1) High speed digital computer such as CDC 6500 or UNIVAC 1108.

2) At least 25K (decimal) memory - (larger memory permits solution of larger problems).

3) Two magnetic tape units if full restart capability is desired. One magnetic tape unit would be sufficient if input parameters are set properly (See section 6.6.4).

5.1.2 Software Requirements

1) The entire BLADER program is coded in Fortran IV. Although the BLADER program was compiled and checked out primarily on the CDC 6500 computing system at TRW, a great effort has been made to avoid using any programming practices which

1) (continued)

would cause significant problems in converting the program to other computing systems which have Fortran IV capability reasonably close to ASA standards.

2) The memory release feature of the CDC 6500 computing system at TRW is used, but is not really required and the usage thereof can be easily eliminated.

3) The Namelist feature of Fortran IV is used to read input data.

5.1.3 Input/Output Units

1) Fortran logical unit 5 is the standard input file (file INPUT on the CDC 6500 computing system).

2) Fortran logical unit 6 is the standard output file (file ØUTPUT on the CDC 6500 computing system).

3) One magnetic tape unit is required for the input restart if the input parameter RESTRT is nonzero. The Fortran logical unit number for this tape unit is determined by the parameter IRSIN. On the CDC 6500 computing system, the nominal value of IRSIN is 1 corresponding to file TAPE 1 (see section 6.6.4).

4) One magnetic tape unit is required for the output restart tape if the input parameter SAVETP is nonzero. The Fortran logical unit number for this tape unit is determined by the parameter IRSØUT. On the CDC 6500 computing system, the nominal value of IRSØUT is 3 corresponding to file TAPE 3 (see section 6.6.4).

5.2 Deck Setup

The deck setup for executing the BLADER program from a binary deck is very simple. The deck setup for executing from a binary program tape or from file service would be equally simple. In this report only the deck setup for executing from a binary deck on the CDC 6500 computing system at TRW is described. From this one basic deck setup, hopefully the deck setup for executing from a binary program tape or from file service or even for running on a different computing system will be suggested.

5.2.1 Administrative Control Cards

In this section is presented a list of the administrative control cards that are likely to be used when executing the BLADER program on the CDC 6500 computing system at TRW. Because the format of these cards may change from time to time, the reader is referred to section 4.3.6.1 of the TRW Systems Programming Handbook for details.

1) $ NAME

2) $ PRØBLEM

3) $ PRØGRAM, AS189.

4) $ PRIØRITY

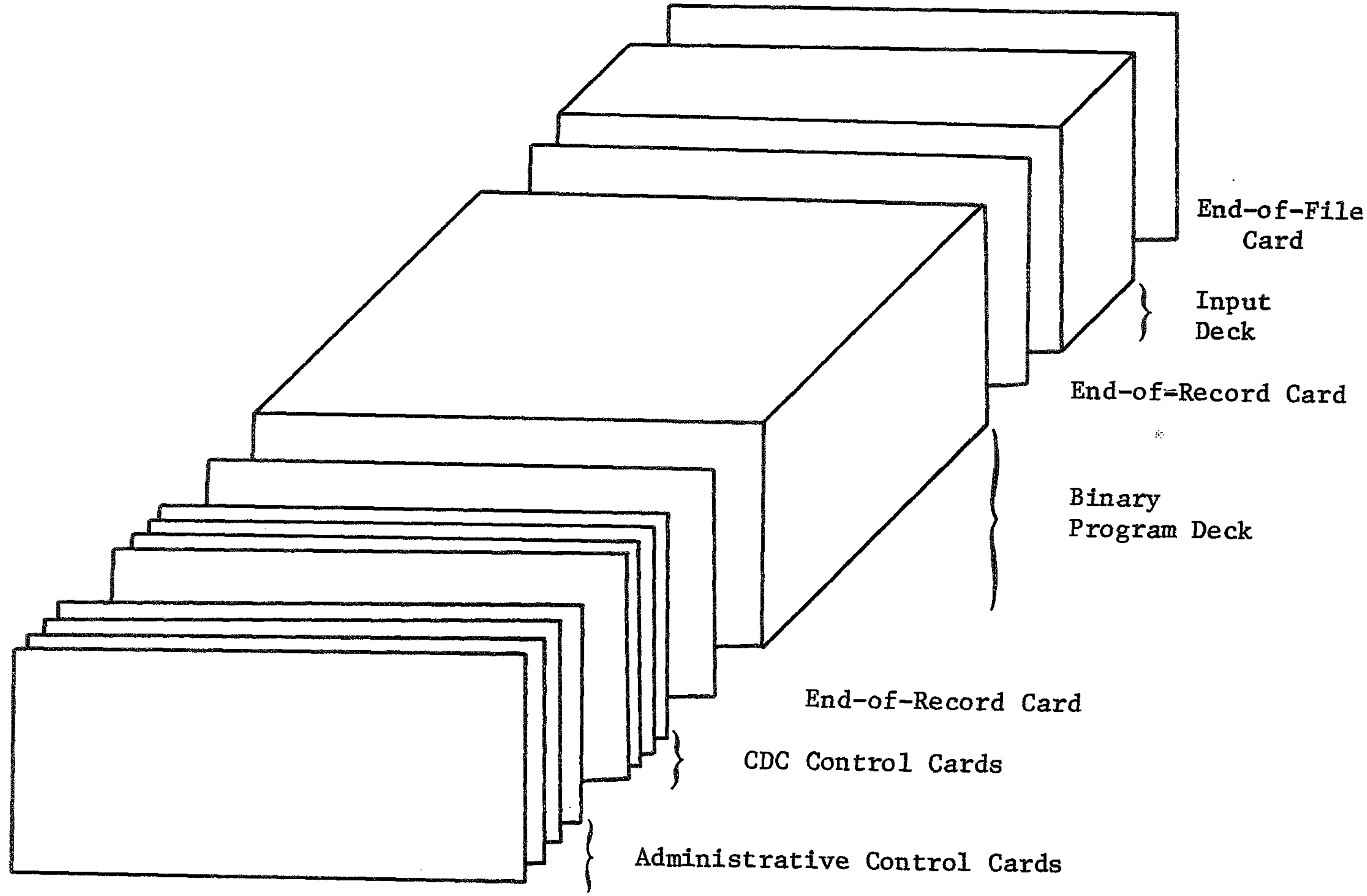
5) $ MAXTIM

6) $ FLENGTH (See section 5.6 Memory Requirements)

7) $ PRINT

8) $ TAPE

   For example, if we wish to run a case which uses the input restart tape 1234 mounted on unit 1 and which will write an output save tape mounted on unit 3, then we would include the following $ TAPE cards:
   $ TAPE, TAPE 1, 1234.
   $ TAPE, TAPE 3, S.



BASIC DECK SETUP

5.2.2 CDC Control Card's

In this section is presented a list of the CDC control cards that are required for executing the BLADER program from a binary deck. The general usage of the CDC control cards is described in the CDC SCØPE reference manual.

1) SET (1) - presets core to indefinites before program is loaded.

2) INPUT. - loads program and starts execution.
   When a large amount of printout is expected, the line count should be set to a large value, e.g. INPUT (LC = 1000000).

3) EXIT.

4) VIV.
   Actually, the VIV card is not a standard CDC control card. Rather, it is a special feature of the TRW 6500 computing system and is described in section 4.3.5.1 of the TRW Systems Programming Handbook.

5.2.3 End-of-record and End-of-file Cards

These are standard control cards for the CDC 6500 computing system. An end-of-record card has 7-8-9 punches in column 1, and an end-of-file card has 6-7-8-9 punches in column 1.

5.3 Library Subroutines

The following is a complete list of all library subroutines explicitly called by the BLADER program. This list does not include subroutines such as the I/Ø routines which may be called indirectly by READ, write, and other I/Ø statements.

1) MRELSE - Memory release routine for the CDC 6500 computing system at TRW. For executing the BLADER program on other computing systems, the two call statements calling MRELSE from the main program BLADER may be removed or a dummy routine for MRELSE may be provided.

2) ALØG - Natural logarithm subroutine: Obtains ln(x) for given x.

3) CØS - Cosine subroutine: Obtains COX(x) for given x (where x is in radians).

4) EXP - Exponential subroutine: Obtains $\exp(x) = e^x$ for given x.

5) PDUMP - Memory dump subroutine: PDUMP is a more or less standard memory dump routine available on many computing systems. The calling sequence is call PDUMP (L_1, U_1, f_1, L_2, U_2, f_2, ..., L_n, U_n, f_n) where L_i and U_i indicate the lower and upper limits of the memory region to be dumped and f_i is a flag indicating the format of the dump:

$$f_i = \begin{cases} 0 \text{ for octal dump,} \\ 1 \text{ for real cump,} \\ 2 \text{ for integer dump.} \end{cases}$$

If the Bladder Program is to be executed on a computing system which does not provide a PDUMP subroutine, then the CALL PDUMP statements in subroutine EERRXX may be deleted or replaced by another dumping procedure.

6) SIN - Sine subroutine: Obtains sin(x) for given x.

7) SQRT - Square root subroutine: Obtains $\sqrt{x}$ for given x.

5.4 Load Map

5.5 Subroutine Cross-References

The following is a load map and subroutine cross reference listing produced when executing the BLADER program on the CDC 6500 computing system.

```
22.38.01.  NORMAL              CONTROL                                           000100    053200    045310    005670
--TIME---LOAD MODE --L1--L2-----TYPE---------------USER---++---CALL---FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
         FWA LOADER  054066   FWA TABLES   050167
         -PROGRAM----ADDRESS-                    --LABELED---COMMON--
         BLADER      001140                                CDIMEN      000100
                                                           CMESH       000113
                                                           FFLAGS      000117
                                                           HEADER      000121
                                                           IDIMEN      000130
                                                           IFLAGS      000152
                                                           INDEX       000164
                                                           IOUNIT      000212
                                                           IPERM       000214
                                                           LBUCKT      000217
                                                           LSSAVE      000300
                                                           MESHIN      000324
                                                           PROPS       000335
                                                           RSDATA      000347
                                                           ST          000412
                                                           TEMP        000646
                                                           UV          001066
                                                           XYCOOR      001132
         BLOC        012212                                CDIMEN      000100
                                                           IDIMEN      000130
                                                           IFLAGS      000152
                                                           INDEX       000164
                                                           LBUCKT      000217
         DIMCHK      012435                                IDIMEN      000130
                                                           IFLAGS      000152
                                                           INDEX       000164
                                                           MESHIN      000324
                                                           PROPS       000335
         DISCHK      013101                                CDIMEN      000100
                                                           IDIMEN      000130
         FMCALC      012177                                IDIMEN      000130
                                                           IPERM       000214
                                                           PROPS       000335
                                                           XYCOOR      001132
         GEOM        013033                                IDIMEN      000130
                                                           IFLAGS      000152
                                                           INDEX       000164
```

		PROPS	000335
		ST	000412
		TEMP	000646
		UV	001066
		XYCOOR	001132
GINPUT	015737	CDIMEN	000100
		CMESH	000113
		FFLAGS	000117
		HEADER	000121
		IDIMEN	000130
		IFLAGS	000152
		INDEX	000164
		IOUNIT	000212
		MESHIN	000324
		PROPS	000335
		RSDATA	000347
MSHGEN	016667	CDIMEN	000100
		CMESH	000113
		FFLAGS	000117
		IDIMEN	000130
		IFLAGS	000152
		MESHIN	000324
		PROPS	000335
PRINT	017761	CDIMEN	000100
		IDIMEN	000132
		IFLAGS	000152
		INDEX	000164
		IOUNIT	000212
		LBUCKT	000217
		LSSAVE	0003[illegible]
		MESHIN	000324
		PROPS	000335
RESID	021624	CDIMEN	000100
		IDIMEN	000132
		IFLAGS	000152
		INDEX	000164
		IPERM	000214
		PROPS	000335
		ST	000412
		XYCOOR	001132
RRST23	022735	CDIMEN	000100

		CMESH	000113
		IDIMEN	000130
		IFLAGS	000152
		INDEX	000164
		IOUNIT	000212
		LSSAVE	000300
		MESHIN	000324
		PROPS	000335
		RSDATA	000347
SETIC	024233	CDIMEN	000100
		IDIMEN	000130
		IFLAGS	000152
		INDEX	000164
		IPERM	000214
		LBUCKT	000217
		PROPS	000335
		UV	001066
SETUP	026634	CDIMEN	000100
		FFLAGS	000117
		IDIMEN	000130
		IFLAGS	000152
		INDEX	000164
		IPERM	000214
		LBUCKT	000217
		LSSAVE	000300
		PROPS	000335
STEP2	031451	CDIMEN	000100
		CMESH	000113
		IDIMEN	000130
		IFLAGS	000152
		INDEX	000164
		LBUCKT	000217
		MESHIN	000324
		PROPS	000335
		ST	000412
STFNS	032645	CDIMEN	000100
		IDIMEN	000130
		IFLAGS	000152
		INDEX	000164
		IPERM	000214
		PROPS	000335

```
                                                     ST          000412
                                                     TEMP        000646
                                                     UV          001066
                                                     XYCOOR      001132
WRST2         035066                                 CDIMEN      000100
                                                     CMESH       000113
                                                     IDIMEN      000130
                                                     IFLAGS      000152
                                                     INDEX       000164
                                                     IOUNIT      000212
                                                     LSSAVE      000300
                                                     MESHIN      000324
EERRXX        035524                                 IDIMEN      000130
                                                     IFLAGS      000152
                                                     INDEX       000164
                                                     IOUNIT      000212
                                                     LBUCKT      000217
                                                     PROPS       000335
SYSTEM        035730                                 SCOPE2      035730
MRELSE        037135
OUTPTC        037163
SQRT          040540
ENDFIL        040603
REWINM        040676
INPUTN        040771
INPUTB        042533
OUTPTB        043050
SINCOS        043317
RBAREX        043416
ALNLOG         43475
EXP            43564
DUMP           43643
SIO$          044144
CPC           045027
GETBA         045271
--ENTRY-----ADDRESS-                        ---REFERENCES---
BLADER        011141
BLOC           12213          BLADER          1152

DIMCHK         12436          GINPUT          1036
```

DISCHK	013102	PRINT	020100		
		SETUP	030122		
FMCALC	0132[illegible]	PRINT	020031		
GEOM	013637	STEP2	031753	031760	
GINPUT	015740	BLADER	001151		
MSHGEN	016677	SETUP	027051	027064	
PRINT	017776	BLADER	001722	001744	
		EERRXX	035630	035652	
RESID	021632	BLADER	001573	001613	
RRST23	022772	SETUP	026752	027025	
SETIC	024263	BLADER	001511	001554	
SETUP	026663	BLADER	001240	001302	
STEP2	031471	BLADER	001401	001430	
STENS	032646	STEP2	031715		
WRST2	0351[illegible]3	SETUP	030707	030732	
EERRXX	035525	BLADER	001667		
QBNTRY	035731	BLADER	001142		
SYSTEM	036246	OUTPTC	037177	041344	
		SORT	041565		
		FEWINM	041756		
		INPUTN	041554	041603	042530
		INPUTB	042574		
		OUTPTB	043237		
		SINCOS	043362	043365	
		RBAPEX	043455		
		EXP	043617		

SYSTEMC	036212				
SYSTEMP	36241				
END	036136	BLADER	011757		
		BLOC	012421		
		DIMCHK	012643		
		DISCHK	013157		
		FMCALC	013464		
		GINPUT	016414		
		MSHGEN	01755[illegible]		
		PRINT	021317		
		RESID	022563		
		RRST23	024127		
		SETIC	0263[illegible]7		
		SETUP	03[illegible]741		
		STEP2	032545		
		WRST2	035507		
		EERRXX	035665		
STOP	036165	BLADER	011330		
		DIMCHK	012452		
		GINPUT	016363		
		RRST23	023042		
		SETUP	030630		
		EERRXX	035655	035663	
		DUMP	043745		
EXIT	036157	INPUTN	042523		
ABNORML	036175	OUTPTC	0372[illegible]	04[illegible]345	
		ENDFIL	041665		
		REWINM	04[illegible]757		
		INPUTN	0421[illegible]2	041636	042531
		INPUTB	042575		
		OUTPTB	04324[illegible]		
TERR	03664[illegible]				
NTERR	036643				
AERR	036646				
MRELSE	3713[illegible]	BLADER	0[illegible]116	1332	

JSPACE	037152
OUTPTC	037165

BLADER	001311	001313	001315	001316	001323	001325
	001326	001624	001626	001627	001650	001652
	001654	001655	001665	001666		
DIMCHK	012442	012443	012447	012450	012464	012465
	012500	012502	012504	012506	012510	012512
	012514	012516	012520	012522	012524	012526
	012527	012532	012534	012536	012540	012542
	012544	012545	012636	012637		
FMCALC	013435	013437	013441	013443	013444	013456
	013460	013461				
GINPUT	015743	015744	016017	016021	016022	016055
	016057	016062	016065	016066	016334	016336
	016337					
MSHGEN	017243	017245	017247	017251	017253	017255
	017257	017261	017263	017264	017267	017275
	017303	017311	017315	017362	017364	017372
	017374	017375	017407	017411	017412	017436
	017440	017442	017443	017522	017525	017533
	017541	017545				
PRINT	020052	020054	020055	020103	020105	020106
	020115	020116	020125	020126	020131	020137
	020146	020155	020164	020173	020202	020211
	020215	020237	020240	020414	020416	020420
	020422	020424	020425	020443	020444	020453
	020454	020457	020465	020503	020506	020511
	020512	020515	020516	020521	020527	020541
	020547	020552	020553	020556	020557	020562
	020570	020602	020610	020613	020614	020623
	020624	020627	020635	020650	020656	020661
	020662	020665	020666	020671	020677	020711
	020717	020722	020723	020726	020727	020732
	020746	020752	020760	020763	020764	020767
	020775	021003	021011	021024	021034	021042
	021046	021063	021065	021067	021071	021072
RRST23	023033	023035	023037	023040	023502	023504
	023505					
SETUP	027121	027123	027125	027126	027131	027135
	027141	027144	027150	027154	027164	027170
	027174	027212	027216	027222	027225	027231

			027235	027240	027244	027250	027274	027277
			027302	027306	027330	027332	027334	027335
			027350	027351	027374	027376	027400	027402
			027403	027422	027424	027426	027430	027431
			027450	027451	027460	027462	027473	027476
			027524	027526	027530	027532	027533	027565
			027567	027571	027573	027574	027622	027624
			027626	027630	027631	027670	027671	027714
			027716	027720	027721	027740	027742	027744
			027746	027747	030011	030012	030035	030037
			030041	030043	030044	030073	030075	030077
			030101	030102	030132	030133	030142	030143
			030152	030153	030170	030172	030174	030175
			030204	030207	030210	030227	030232	030235
			030236	030255	030257	030261	030262	030274
			030276	030300	030301	030330	030332	030334
			030335	030350	030351	030402	030404	030406
			030410	030411	030440	030442	030444	030446
			030447	030536	030540	030541	030601	030603
			030604	030616	030620	030621		
		STEP2	031525	031527	031532	031541	031550	031554
			031605	031607	031611	031613	031614	
		EERRXX	035530	035531	035553	035554		
		DUMP	044011	044017	044057	044074	044114	
KODER	37332							
SQRT	40541	BLADER	001621	001643				
		PRINT	020366					
		SETIC	025670	025722				
ENDFIL	040604	BLADER	001752					
		EERRXX	035657					
REWINM	040577	BLADER	001754					
		GINPUT	016024	016366				
		EERRXX	035661					
INPUTN	040772	GINPUT	016013					
		SETUP	027234					
NMLEOF	42513							

INPUTB	042535	GINPUT	016026	016032	016036			
		FRST23	023004	023006	023010	023012	023014	023016
			023020	023022	023023	023044	023061	023067
			023076	023105	023116	023122	023131	023141
			023145	023156	023165	023171	023173	023205
			023216	023230	023242	023254	023260	023271
			023301	023305	023315	023325	023331	023340
			023350	023362	023366	023375	023405	023411
			023420	023437	023442	023454	023460	023514
			023515	023525	023527	023531	023533	023535
			023537	023541	023543	023557	023605	023636
			023667	023720	023742	023756	023770	024002
			024014	024026	024040	024044	024053	024063
			024067	024076	024106	024112		
OUTPTB	043052	GINPUT	016373	016374	016402	016406	016410	016411
		PRINT	021110	021115	021117	021121	021123	021125
			021127	021131	021141	021154	021167	021202
			021206	021215	021225	021231	021240	021250
			021254					
		WRST2	035114	035116	035120	035122	035124	035126
			035130	035132	035133	035135	035147	035153
			035162	035171	035202	035206	035215	035221
			035225	035236	035245	035251	035253	035265
			035276	035310	035322	035334	035340	035351
			035355	035361	035371	035375	035401	035410
			035414	035422	035426	035435	035441	035445
			035454	035460	035466	035474	035500	
DUMPOUT	043316							
SIN	043320	MSHGEN	017065					
		STEP2	032131					
COS	043323	MSHGEN	017067					
		STEP2	032133					
RBAREX	043417	MSHGEN	017222					
		PRINT	020410					
ALOG	0435[illegible]1	MSHGEN	017445	017447				

		RBAREX	043435					
ALOG10	043476							
EXP	043565	MSHGEN	017500					
		RBAREX	043440					
PDUMP	043740	EERRXX	035534	035540	035552	035564		
DUMP	043743							
BKSPRU.	044440							
FIZBAK.	044450							
POSFIL.	044476	OUTPTC	037215					
		ENDFIL	040653					
RDPRU.	044556							
DAT.	044577	OUTPTC	037246	037233	037262	037524	040306	037222
CIO1.	044414	ENDFIL	040642	040655	040620			
		REWINM	040735	040724	040715			
		INPUTB	042623	042631	042636	042563	042734	042761
			043015					
		OUTPTB	043144	043071	043164	043133	043220	043136
			043257	043261	043323			
OPEN.	044146	SYSTEM	036551					
		OUTPTC	037205					
		ENDFIL	040614					
		INPUTN	042073	041415				
		INPUTB	042710	042657				
		OUTPTB	043145	043213				
SIO.	044272	OUTPTC	037261					
		INPUTB	042567					
		OUTPTB	043100					
ADVIN.	044505	SYSTEM	036453					
		ENDFIL	040633					
		REWINM	040732					
MVWDS.	044321	INPUTB	042511					

```
                                  OUTPTB     043113

         POSFI.     044515        ENDFIL     040627
                                  OUTPTB     043210

         FIZBA.     044524
         CPC        045075        MRELSE     037141     037154

         CPC02      045154
         CPC03      045227
         CPC04      045246
         CPC999     045261
         GETBA      045271        OUTPTC     037173
                                  ENDFIL     040610
                                  REWINM     040704
                                  INPUTN     041001     042034
                                  INPUTB     042654
                                  OUTPTB     043122

         ----UNSATISFIED EXTERNALS-----               ---REFERENCES---
```

5.6 Memory Requirements

The basic memory requirements for the BLADER program (excluding the variable memory requirements of the Dynamic Storage Array) are relatively modest. On the CDC 6500 computing system, the basic memory requirement is approximately 46000 (octal) $\approx$ 19,500 (decimal) cells. The memory required for the Dynamic Storage Array is extremely variable, depending on the size of the problem. On the CDC 6500 computing system, the program begins with 3000 (decimal) = 5670 (octal) cells in the D.S.A., but may increase or decrease the size of the D.S.A. - depending on the size of the problem. On a computing system which does not have a memory release feature such as that on the CDC 6500 computing system at TRW, it would be necessary to dimension the array BUCKET in the BLADER main program with the maximum dimensions desired for the D.S.A.

Tables 5.6.1 and 5.6.2 should be helpful in estimating the size of the Dynamic Storage Array for a given problem. It turns out that in most cases the only really significant parameters are the number of elements NUMEL and the number of nodal points NUMNP. All of the input parameters are described in section 2.3 of this report.

For a bladder analysis problem (BLANAL $\neq$ 0), many of the dimensional quantities (NUMEL, NUMNP, etc.) are calculated from the basic bladder dimensions as indicated in table 5.6.2. NLAYER, LWRAT, and NANGIN are the basic input dimensions.

Table 5.6.1

Dynamic Storage Requirements

		Rectangular Coordinates (RECCYL = 0)	Cylindrical Coordinates (RECCYL ≠ 0)	
NUMEL	X	102	120	Most significant dimensions
NUMNP	X	33	37	
NMAT	X	5	5	
NDP	X	2	2	
NUMBC	X	2	2	
NID	X	2	2	
NNDLIM	X	5	5	
Plus				
NLØAD	X	2	2	If NLØAD > 0
NLØAD	X	3	3	If NLØAD > 0 and neither NDP nor PRESS is entered.
NUMEL	X	-	3	If NLØAD < 0
NUMEL	X	1	1	If NMAT > 1
NLAYER	X	2	2	If BLANAL ≠ 0
NMAT	X	2	2	If FLIFE ≠ 0

Plus

2* NUMNP * BANDW for the band matrix; where BANDW is the band width of the system stiffness matrix.

Table 5.6.2

Bladder Analysis Dimensions

	Single Fold	Double Fold
NXINT	LWRAT * NLAYER	NANGIN
NPTSX	2 * NXINT + 1	2*NXINT + 1
NPTSY	2 * NLAYER + 1	2*NLAYER + 1
NUMEL	2 * NLAYER * NXINT	2*NLAYER*NXINT
NUMNP	NPTSX * NPTSY	
NMAT	Input	Input
NDP	0	0
NUMBC	NPTSY	2*NPTSY
NID	2 * NPTSY	0
NNDLIM	Input	Input
NLØAD	0	<0
BANDW	8 * NLAYER + 10	8*NLAYER + 10

Note: If FLIFE ≠ 0, then the array CYCTØF will require 40 * NMAT cells. However, since the CYCTØF array is overlaid by the band matrix, which is normally larger than the CYCTØF array, there is usually no need to allow extra storage for the CYCTØF array.

5.7 Print/Timing Estimates

Both the volume of printed output and the execution time are extremely variable being dependent upon such factors as the problem dimensions, the number of load steps, and the options selected. In this section an attempt is made to present information that will assist the user in estimating the volume of printed output and the execution time.

5.7.1 Print Volume

For the purposes of estimating the volume of printed output, the print may be divided into three categories: input printout, standard load step printout, and detailed load step printout. The following algorithms consider only the options which greatly affect the volume of printout and hence provide only an estimate of the printed output.

5.7.1 Print Volume (continued)

1) Input printout:

No. of lines $\approx$ C + E + P + Z where

C = no. of input cards,

E = no. of elements (multiplied by 2 if blank lines are included),

P = no. of points,

$$Z \approx \begin{cases} 100 \text{ if blank lines are included,} \\ 60 \text{ otherwise.} \end{cases}$$

2) Standard load step printout:

No. of lines $\approx$ MAXST * (I + P + L + Z) where

I = no. of iterations ($\leq$ ITER),

P = no. of points,

L = no. of sides loaded (see VPR in sections 6.2.3 and 6.2.4),

$$Z \approx \begin{cases} 20 \text{ if blank lines are included,} \\ 10 \text{ otherwise} \end{cases}$$

3) Detailed load step printout:

No. of lines $\approx \frac{\text{MAXST}}{\text{IPRIN}}$ (4 * P + Z)

where

P = no. of points,

$$Z \approx \begin{cases} 25 \text{ if blank lines are included,} \\ 12 \text{ otherwise.} \end{cases}$$

In some computing systems blank lines must be included in the estimate of the print volume while in other computing systems only real lines are counted.

5.7.2 Execution Time

Although the problem options (i.e. coordinate system, bladder or standard strain analysis, etc.) have very little effect upon the running time, the problem dimensions, the number of load steps, and the convergence criterion have a great effect upon the running time. It is somewhat difficult to determine the relation between the problem dimensions and the running time. The band matrix size is approximately proportional to the square of the number of nodes, and the number of nodes is in turn proportional to the product of the number of intervals in each coordinate direction. The running time per iteration is roughly proportional to the size of the band matrix. Thus we can expect the running time per iteration to be roughly proportional to the square of the number of nodes. For a problem with a given number of nodes, the running time is almost directly proportional to the total number of iterations required, which in turn would be directly proportional to the number of load steps if all load steps required the same number of iterations. Thus a very crude estimate of the running time may be obtained from the formula

$$T = SIN^2$$

where

T = running time

I = total number of iterations

= MAXST * (average number of iterations per load step)

N = no. of nodes

S is a quantity which is roughly constant for a given computer and must be determined through experience in running the program.

Perhaps a few examples will give a rough idea of the running time at least for the CDC 6500 computing system. The running time for other computing systems will have to be determined by experience.

Run	No. of Nodes	No. of Load Steps	Total No. of Iterations	Running Time (sec.)	$S*10^3$	Comments
1	21	10	13	15.5	2.7	
2	21	12	15	20.3	3.07	(with restart)
3	21	10	12	19.8	3.74	cylindrical coords
4	45	10	36	136.0	1.87	
5	45	5	15	60.0	1.98	
6	75	4	4	32.05	1.42	
7	75	20	20	126.9	1.13	2706 lines of output
8	75	20	20	145.1	1.29	6396 lines of output

5.7.2 Execution Time (continued)

It is obvious from the table on the preceding page that the formula $T = SIN^2$ is only a rough formula since the value of $S = \frac{T}{IN^2}$ is not the same for all cases.

5.8 Error Messages

This section consists of a list of all error messages which may be printed out directly by the BLADER program. For convenience, the error messages are listed alphabetically, and for each message the name of the subroutine which prints the message is specified. Also, a brief explanation of the meaning of the message and suggested remedial procedures are presented for each message.

Lower case letters are used to indicate numerical or alphabetic variables in the message, and upper case letters are used to indicate the fixed portion of each message.

1. ***** A PLANE STRAIN CASE MAY NØT FØLLØW AN AXISYMMETRIC CASE.

   Subroutine: DIMCHK
   Explanation: Because the values of certain coefficients are built into the program by means of DATA statements for the rectangular coordinates option but are calculated for the cylindrical coordinates option, a plane strain problem may not be stacked behind an axisymmetric problem. (see section 2.1.6)
   Remedy: Reorder the cases placing all problems involving rectangular coordinates in front of all problems involving cylindrical coordinates.

2. ERRØR ... IDIMEN(i) = m IS LESS THAN ØLD VALUE = n

   Subroutine: GINPUT
   Explanation: Because of the Dynamic Storage feature, the problem dimensions may not be reduced on a Restart run. The values m and n are the new and old values of the quantity IDIMEN(i) where IDIMEN(2) = NUMEL, IDIMEN(3) ≡ NUMNP, etc... See the description of the labeled CØMMØN block IDIMEN in section 6.3 of this report

2. (continued)

for the significance of IDIMEN(i) for any specific value of i.

Remedy: Compare the new values and the old values of the problem dimensions and restructure the problem to avoid reducing the problem dimensions.

3. ERRØR IN PRØBLEM DIMENSIONS

Subroutine: DIMCHK

Explanation: An obvious error has been detected in the problem dimensions. The following is a list of possible errors which the program checks for:

a) $\alpha \leq 0$ where α may be any of the following: NUMEL, NUMNP, NUMCP, NUMBC, NMAT, MAXBW, MAXST.

b) NID < 0.

c) NDP < 0.

d) NUMCP $\geq$ NUMNP.

e) NLOAD $\neq$ 0 and NID $\neq$ 0.

f) NLOAD = 0 and NID = 0.

g) RECCYL = 0 and NLOAD < 0.

h) NNDLIM < 0 or NNDLIM > NUMNP.

i) BLANAL $\neq$ 0 and NLAYER $\leq$ 0.

j) BLANAL $\neq$ 0 and NXINT $\leq$ 0.

k) BLANAL $\neq$ 0 and NCYCF > 20.

l) BLANAL $\neq$ 0 and RFINAL $\leq$ 0.

Remedy: Check the input problem dimensions very carefully. Also, in the case of a bladder analysis (BLANAL $\neq$ 0) check the calculated problem dimensions and the input parameter RFINAL. In particular, check the printout of the general input data. More than one error may be present so check very carefully. Correct the offending quantities and resubmit the run.

5.8 Error Messages (continued)

4. ERRØR ... VPRFLG FRØM RESTART TAPE = valuea PRØGRAM ALREADY SET VPRFLG = valueb

Subroutine: RRST23

Explanation: There is an inconsistency between the old (valuea) and new (valueb) values of VPRFLG, the variable pressure flag, where

$$\text{VPRFLG} = \begin{cases} 1 & \text{if NLØAD} > 0 \text{ and if neither NDP nor PRESS} \\ & \text{is specified,} \\ 0 & \text{otherwise.} \end{cases}$$

Most likely one or more of the input quantities NLØAD, NDP, and PRESS was changed incorrectly on a restart run.

Remedy: Check the old and new values of the input quantities NLØAD, NDP, and PRESS. If NDP and PRESS were originally non-zero, then they may not be set to zero on a restart. Similarly, if NDP and PRESS were zero or unspecified on the original run, then they should be left unspecified on a restart run. Also if NLØAD was originally zero or negative, then it must be zero or negative on a restart run since the method of loading may not be changed on a restart run.

5. ERRØR ... m SIDES LØADED FOR ELEMENT n

Subroutine: SETUP

Explanation: The user has attempted to specify m (> 3) sides loaded for element n. Since the elements are triangular, no more than 3 sides of any element may be loaded. Also, no side of an element may be loaded with more than one load.

Remedy: Modify the pressure loading specifications (input arrays LØADE and LØADS)

5.8 Error Messages (continued)

6. ILLEGAL VALUE SPECIFIED FØR VARIABLE varnam WITH SUBSCRIPT(S) m, n

Subroutine: SETUP

Explanation: The program has detected an obvious error in the array indicated by the symbol varnam. The following table lists the possible errors.

varnam	Subscript(s)	Description of Possible Errors
DELTAP	1	DELTAP(1, 1) $\leq$ or DELTAP(2, 1) = 0.
MATEL	m	MATEL(m) $\leq$ 0 or MATEL(m) > NMAT.
CØRNER	1, n	CØRNER(1, n) $\leq$ 0 or CØRNER(1, n) > NUMNP.
NP	m, n	NP(m, n) $\leq$ 0 or NP(m, n) > NUMNP.
IBCØND	1, n	IBCØND(1, n) $\leq$ 0 or IBCØND(1, n) > NUMNP.
DLIM	1, n	DLIM(1, n) $\leq$ 0 or DLIM(1, n) > NUMNP or DLIM(2, n) $\leq$ 0 or DLIM(2, n) > 2.
LØADE	m	LØADE(m) $\leq$ 0 or LOADE(m) > NUMEL.
LØADS	m	$\lvert$LOADS(m)$\rvert$ > 3
DISPL	1, n	DISPL(1, n) $\leq$ 0 or DISPL(1, n) > NUMNP or DISPL(2, n) $\leq$ 0 or DISPL(2, n) > 2.

Remedy: Correct the offending values in the indicated array.

7. INPUT ERROR. CYCM(m) = value, BUT NØ VALUE WAS INPUT FOR CYCZ(m).

Subroutine: MSHGEN

Explanation: The arrays CYCM and CYCZ are in one-to-one correspondence. Thus, if a value is entered for CYCM(m), then a value must be entered for CYCZ(m).

Remedy: Delete the entry for CYCM(m) or enter a value for CYCZ(m).

8. LENGTH OF DYNAMIC STORAGE m (DECIMAL) nnnnnn (ØCTAL)
EXCEEDS MAXIMUM BUCKET SIZE.
MAXBKT = k.

Subroutine: BLADER

Explanation: The values m and nnnnnn specify the length of the Dynamic Storage array as decimal and octal integers respectively. The value m exceeds the maximum allowable value as indicated by the parameter MAXBKT.

Remedy: Increase the value of MAXBKT through the Namelist input or decrease the problem dimensions so that m $\leq$ MAXBKT. See section 5.6 for a discussion of the size of the Dynamic Storage array.

9. MAX BAND WIDTH ØF n EXCEEDED.

Subroutine: SETUP

Explanation: The integer n is the value of the parameter MAXBW and is less than the calculated band width of the band matrix A.

Remedy: Increase MAXBW so that it is at least as large as the calculated band width, or restructure the problem to reduce the band width of matrix A.

10. NEGATIVE ØR ZERØ AREA, ELEMENT = n

Subroutine: SETUP

Explanation: The area of element n calculated by the relation

$$A = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_3 - y_1)$$

is negative or zero, where (x_1, y_1), (x_2, y_2) and (x_3, y_3) are the coordinates of points 1, 2 and 3 (i.e. the corner points) of element n. If RECCYL = 0, then rectangular coordinates (x, y) are used, but if RECCYL $\neq$ 0, then cylindrical coordinates (r, z) are used.

Remedy: Renumber the nodes, change the point coordinates, or restructure the problem to avoid negative areas.

11. SØLUTIØN FAILED TØ CØNVERGE.

Subroutine: BLADER

Explanation: The solution of the problem (more specifically, the norm of the residual load vector DF) failed to converge within ITER iterations.

Remedy: Examine the sequence of values printed out for the norm of DF and the DF/F norm ratio. If the sequence appears to be converging satisfactorily, then perhaps merely increasing the value of the parameter ITER will allow the problem to converge within a few more iterations. Also, increasing the value of the convergence criterion CØNVER may permit the problem to converge and thus continue (but with reduced accuracy). If the DF norm and the DF/F norm ratio are not converging, then it will be necessary to change some of the problem parameters (such as YP, EM, etc. or the pressure loading parameters PRESS, MAXST, etc.) in order to get the problem to converge.

6.0 PROGRAM DESCRIPTION

6.1 Introduction

The contents of section 6 of this report describes various programming aspects of the BLADER program and is intended primarily for the benefit of any programmers who might have the responsibility of modifying the program in the future. This section is not as complete as we would like, but hopefully it describes the most significant programming aspects of the program and will prove of value to anyone who needs to understand the coding of the program.

6.2 Dynamic Storage Array

6.2.1 Introduction

The Dynamic Storage Array (D.S.A. for short) BUCKET, which is also referenced by the shorter name B for convenience, is a large block of memory in which all of the arrays which have variable dimensions are stored. Since all of the arrays in the D.S.A. may have variable dimensions, the origin of each array A_n relative to B is not fixed but rather is dependent upon the dimensions of the arrays A_1, A_2, ..., A_{n-1} which precede A_n in the D.S.A. The origins of the various individual arrays A_1, A_2, A_3, ..., A_N relative to B are stored in the labeled COMMON block LBUCKT. For example, the origin of the array NP is always B(1) and the origin of the array LOADE is $B(\ell_{NP} + 1)$, where ℓ_{NP} is the total dimension of the array NP (ℓ_{NP} = 6* NUMEL). The relative origin $\ell_{NP} + 1$ of the array LØADE is stored in LLØADE. Similarly, the relative origin of the array LØADS is stored in LLØADS and is equal to 1 + $\ell_{NP} + \ell_{LØADE}$, where $\ell_{LØADE}$ is the total dimension of the array LØADE. The origin of LOADS is equivalent to the location $B(LLØADE + \ell_{LØADE})$.

On the CDC 6500 version of the BLADER program, the Dynamic Storage Array B is assigned to blank COMMON for two important reasons:

1) By being assigned to blank COMMON, the array B may overlay the portion of memory which was occupied by the loader during the final operations of the loading process and which otherwise would not be used during execution.

2) The memory allocation/release feature (TRW subroutine MRELSE and CDC control card RFL) of the CDC software may be used to increase or decrease the size of the D.S.A. during execution. Thus, a user with a large problem can obtain the large amount of memory that is required to solve his problem (provided of course that the absolute upper limits of available memory, e.g. 50K or 131K, or whatever, are not exceeded), whereas the user with a small problem will request (and thus pay for) only a relatively small amount of memory and will receive better turn around time as an extra bonus.

6.2.2 Dynamic Storage Array Pointers

Unless explicitly stated otherwise, the name of the D.S.A. pointer for each of the arrays described in the following section is obtained by placing the character L in front of the array name. For example, the D.S.A. pointer for the array NP is LNP, and the D.S.A. pointer for the array BETA is LBETA. The main exceptions are the D.S.A. pointers for arrays whose names already contain six characters. In order to avoid having symbolic names with more than six characters, one or more characters have been deleted from the array name in order to make room for the L prefix. Thus, the pointer for the array MLAYER is LMLAYR, and the pointer for the array CYCTØF is LCYCF. All D.S.A. pointers are stored in the labeled COMMON block LBUCKT.

6.2.3 Summary List of Dynamic Storage Arrays

In this section is presented an alphabetical list of all arrays which are assigned to the Dynamic Storage Array. For each array, the corresponding D.S.A. pointer is given along with the position of that array in the following section, which gives a detailed description of each array.

Array Name	D.S.A. Pointer	Position in Section 6.2.4
A	LA	48
BETA	LBETA	7
CØRNER	LA	46
CØUNT	LCØUNT	18
CYCM	LCYCM	41
CYCTØF	LCYCF	45
CYCZ	LCYCZ	42
DELTAP	LDELTP	37
DEPTH	LDEPTH	44
DF	LDF	19
DIS	LDIS	21
DISPL	LDISPL	39
DLFLAG	LDLFLG	31
DLIM	LDLIM	30

6.2.3 Summary List of Dynamic Storage Arrays (continued)

Array Name	D.S.A. Pointer	Position in Section 6.2.4
DPR	LDPR	50
DR	LDR	20
DSD	LDSD	29
DTST	LDTST	27
DWP	LDWP	16
EEST	LEEST	11
EPST	LEPST	10
ESIG	LESIG	12
EST	LEST	26
ETST	LETST	9
EWP	LEWP	13
F	LF	22
IA	LDISPL	47
IBCØND	LBCØND	40
LØADE	LLØADE	2
LØADS	LLØADS	3
MATEL	LMATEL	6
MLAYER	LMLAYR	43
ND	LND	5
NEBC	LNEBC	4
NP	LNP	1
PST	LPST	25
RESI	LRESI	23
RØRD	LXØRD	14
SIG	LSIG	28
THETA	LTHETA	38
TPR	LTPR	49
TST	LTST	24
VEM	LVEM	32
VPR	LVPR	8
VTH	LVTH	34

6.2.3 Summary List of Dynamic Storage Arrays (continued)

Array Name	D.S.A. Pointer	Position in Section 6.2.4
VXI	LVXI	35
VXU	LVXU	33
VYP	LVYP	36
WP	LWP	17
XØRD	LXØRD	14
YØRD	LYØRD	15
ZØRD	LYØRD	15

6.2.4 Description of Arrays Stored in the Dynamic Storage Array

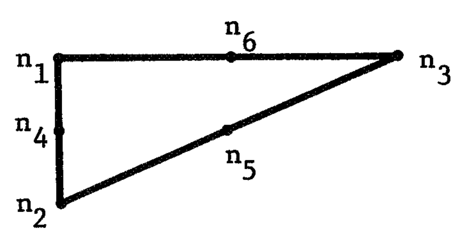
Name		Description
1) NP	-	NUMEL *6 array in which NP(i, j) specifies the nodal point index of the j-th point in element i.

However, in subroutine SETUP, the order of the subscripts is reversed so that in subroutine SETUP only NP is a 6* NUMEL array. Thus, in the Namelist input data NP(j, i) specifies the nodal point index of the j-th point in element i. This reversal of the order of the subscripts in subroutine SETUP was done in order to permit the implementation of the Namelist input feature and the Dynamic Storage feature without requiring an excessive amount of recoding. In the original version of the BLADER program, arrays such as NP were given fixed maximum dimensions such as 40*6. Unfortunately, the dimension that limited the size of the problem (e.g. the maximum number of elements) was chosen to be the first dimension. This choice of the order of the dimensions was perhaps slightly inconvenient, but not critical as far as the implementation of the Dynamic Storage feature was concerned. But when, for greater convenience in specifying the input data, the Namelist input feature was implemented, it was discovered that the Namelist

Name (continued) Description (continued)

feature would not permit variable dimensions. Thus, in order to "fool" the Fortran compiler, the order of the dimensions of the NP were reversed in subroutine SETUP. Because of the way a compiler works, the last dimension of an array passed through a calling sequence is a dummy dimension which serves only to identify the array as an n-dimensional array. Thus, in SETUP the array NP was dimensioned 6*1000. Just before exiting from SETUP, the program uses the temporary array IA to take the transpose of the array NP.

Consider the following diagram in which n_1, n_2 and n_3 represent the nodal point indices of the corner points of element i and n_4, n_5 and n_6 represent the nodal point indices of the midpoints of element i.



Then the values n_1, n_2, ..., n_6 are stored in columns 1-6 of the i-th row of the NP array (except in subroutine SETUP, where the rows and columns are interchanged).

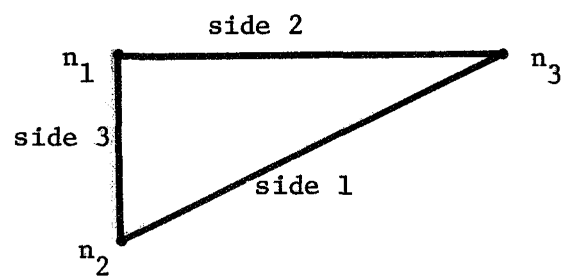
Actually any one of the three corner points may be selected as the starting point for numbering the nodes within the element so long as the three corner points are listed first in cyclic counter-clockwise order followed by the three midpoints again in cyclic counter-clockwise order. The midpoint between the first and second corner points must be the first midpoint listed.

If BLANAL = 0, then the NP array must be input. If BLANAL ≠ 0 (i.e., Bladder Analysis Option), then the NP array will be calculated in the subroutine SETUP.

2) LØADE - Vector with dimension NLØAD (provided that NLØAD > 0) in which LØADE(i) specifies the element index for the ith loading condition. If NLØAD ≤ 0, then the LØADE array is not used.

If BLANAL = 0 and if NLØAD > 0, then the array LØADE must be input. If BLANAL ≠ 0, then NLØAD is set ≤ 0 so that LØADE is not used.

3) LØADS - Vector with dimension NLØAD (provided that NLØAD > 0) in which LØADS(i) specifies which side of the element specified by LØADE(i) is loaded.



As shown in the above diagram, sides 1, 2, and 3 are opposite nodes n_1, n_2, and n_3 respectively where n_1, n_2, and n_3 are the nodal point indices stored in columns 1, 2, and 3 of the jth row of the NP array and where j is stored in LOADE(i).

The sign of LØADS(i) determines whether the loading is a compression (+ or no sign) or a tension (- sign). For example, if LØADE(5) = 12 and LØADE(5) = 2, then side 2 of element 12 is loaded with a compression. If LØADE(5) = -2 instead, then side 2 of element 12 is loaded with a tension. In either case, the magnitude of the loading is $\frac{\text{PRESS}}{\text{MAXST}}$ if

Name (continued) Description (continued)

NDP = 0 or is determined from the array DELTAP if NDP $\neq$ 0.

If NLØAD $\leq$ 0, then the array LØADS is not used.

If BLANAL = 0 and if NLØAD > 0, then the array LØADS must be input. If BLANAL $\neq$ 0, then NLØAD is set $\leq$ 0 so that LØADS is not used.

4) NEBC - Vector with dimension 2* NUMBC + NNDLIM that specifies which degrees of freedom are constrained by the boundary conditions. Part of the array NEBC is generated in the subroutine SETUP, and if NNDLIM > 0, part of the array NEBC may be generated in the subroutine DISCHK. If BLANAL = 0, then the input array IBCØND is used to set up that portion of the NEBC array which is generated by the subroutine SETUP. If BLANAL $\neq$ 0, then the nodes on the "left" and possibly the nodes on the "right" edge are used by subroutine SETUP to generate the NEBC array.

Suppose NEBC(i) = 2(n - 1) + k where k may have the value 1 or 2. Then for convenience we denote

$$\xi = \begin{cases} x & \text{if RECCYL} = 0 \text{ and } k = 1, \\ y & \text{if RECCYL} = 0 \text{ and } k = 2, \\ r & \text{if RECCYL} \neq 0 \text{ and } k = 1, \\ z & \text{if RECCYL} \neq 0 \text{ and } k = 2. \end{cases}$$

Then the value 2(n - 1) + k for NEBC(i) states that the i-th boundary condition requires node n to be constrained from motion in the direction specified by the coordinate ξ.

5) ND - Vector with dimension NID that specifies which degrees of freedom are constrained by imposed displacement conditions. If BLANAL = 0 and NID > 0, then the ND array is obtained from the input array DISPL. If NID > 0, then NLØAD must be zero. Similarly, if NID = 0, then NLØAD must be nonzero.

If BLANAL $\neq$ 0 and RECCYL = 0 (single fold bladder analysis option), then NID is set equal to 2* NPTSY and ND(1) ... ND(NID)

5) ND - (continued)

are set to 2N + 1, 2N + 2, ..., 2* NUMNP, where NPTSY = 2* LAYER + 1 and N = NUMNP - NPTSY. In other words, all of the nodes on the rightmost edge of the area to be analyzed will be subjected to imposed displacements in both the x- and y- directions.

If BLANAL $\neq$ 0 and RECCYL $\neq$ 0 (double fold bladder analysis), then NID will be set to zero.

Suppose ND(i) = 2(n - 1) + k where k may have the value 1 or 2. Then, using the symbol ξ as described under NEBC above, the value 2(n - 1) + k for ND(i) states that the n-th node is constrained by an imposed displacement in the ξ-direction. The magnitude of the imposed displacement is specified by DSD(i).

6) MATEL- Vector with dimension NUMEL (defined only if NMAT > 1) in which MATEL(i) specifies the index of the material of which element i is composed. If there is only one material, then the array MATEL is not necessary and is therefore not defined. If BLANAL $\neq$ 0 and NMAT > 1, then the MATEL array is generated by subroutine SETUP from the input array MLAYER, if BLANAL = 0 and NMAT > 1, then the array MATEL must be input.

7) BETA - NUMEL *6 array in which the variables β_1, β_2, ..., β_6 for the elasto-plastic constitutive law are stored. BETA(i, j) specifies β_j, the jth variable for element i.

The arrays BETA, ETST, EPST, EEST, and ESIG are cleared to zero by subroutine SETUP when the Dynamic Storage Array is cleared and then are updated by subroutine SETIC each load step.

8) VPR - Vector with dimension NLØAD in which VPR(i) specifies P_1, the total loading to be applied to side S_i = LØADS(i) of element LØADE(i). The VPR vector is defined and used only if NLØAD > 0 and if NDP and PRESS are not specified in the Namelist input.

9) ETST - NUMEL * 6 * K array (where K = 3 if RECCYL = 0 or K = 4 otherwise) in which the accumulated elemental total strains are stored. ETST(i, j, k) specifies the k-th component of strain for the j-th point of element i. The significance of k is described by

9) ETST - (continued)

the following table:

k	$\varepsilon_{T,k}$ for RECCYL = 0	$\varepsilon_{T,k}$ RECCYL $\neq$ 0
1	$\varepsilon_{T,x}$	$\varepsilon_{T,r}$
2	$\varepsilon_{T,y}$	$\varepsilon_{T,z}$
3	$\gamma_{T,xy}$	$\varepsilon_{T,\theta}$
4	-	$\gamma_{T,rz}$

See description of BETA above for addition details.

10) EPST - NUMEL * 6 * K array (where K = 3 if RECCYL = 0 or K = 4 otherwise) in which the accumulated elemental plastic strains are stored. EPST(i, j, k) specifies $\varepsilon_{P,k}$, the k-th component of strain for the j-th point of element i. The significance of k is indicated in the table under the description of ETST above (substitute $\varepsilon_{P,k}$ for $\varepsilon_{T,k}$ in each case). Also, see description of BETA above.

11) EEST - NUMEL * 6 * K array (where K = 3 if RECCYL = 0 or K = 4 otherwise) in which the accumulated elemental elastic strains are stored. EEST(i, j, k) specifies $\varepsilon_{E,k}$, the k-th component of strain for the j-th point of element i. The significance of k is indicated in the table under the description of ETST above (substitute $\varepsilon_{E,k}$ for $\varepsilon_{T,k}$ in each case). Also, see description of BETA above.

12) ESIG - NUMEL * 6 * 5 array in which the accumulated elemental stresses are stored. ESIG(i, j, k) contains the k-th component of stress σ_k for the j-th point of element i. The significance of k is described by the following chart:

12) ESIG - (continued)

k	σ_k for RECCYL = 0	σ_k for RECCYL $\neq$ 0
1	σ_x	σ_r
2	σ_y	σ_z
3	τ_{xy}	σ_θ
4	σ_z	τ_{rz}
5	$\bar{\sigma}$	$\bar{\sigma}$

See description of BETA above for additional details.

13) EWP - NUMEL *6 array in which EWP(i, j) specifies the accumulated plastic work for node j of element i.

The array EWP is initially set equal to 10^{-6} by subroutine SETUP and then is updated by subroutine SETIC each load step.

14) XØRD / RØRD - Vector with dimension NUMNP in which XØRD(i) specifies the ξ-component for nodal point i, where

$$\xi = \begin{cases} x & \text{if RECCYL} = 0, \\ r & \text{if RECCYL} \neq 0. \end{cases}$$

See the description of YØRD below.

15) YØRD / ZØRD - Vector with dimension NUMNP in which YØRD(i) specifies the ζ- component for nodal point i, where

$$\zeta = \begin{cases} y & \text{if RECCYL} = 0, \\ z & \text{if RECCYL} \neq 0. \end{cases}$$

The arrays XØRD and YØRD are initialized by subroutine SETUP and then are updated by subroutine SETIC each load step. If BLANAL = 0, then in subroutine SETUP, XØRD and YØRD are obtained from the input array CØRNER. If BLANAL $\neq$ 0, then XØRD and YØRD are calculated using the input array DEPTH and the value XLNGTH (the latter may be input directly or calculated from the array DEPTH).

In subroutine FMCALC the arrays XØRD and YØRD are referenced by the names RØRD and ZØRD respectively.

16) DWP - Vector with dimension NUMNP in which DWP(i) specifies the incremental average plastic work for nodal point i. The array DWP is calculated in subroutine SETIC and printed in subroutine PRINT.

17) WP - Vector with dimension NUMNP in which WP(i) specifies the accumulated average plastic work for nodal point i. The array WP is calculated in subroutine SETIC and printed in subroutine PRINT.

18) CØUNT- Vector with dimension NUMNP in which CØUNT(i) specifies the number of elements which contain nodal point i. The array CØUNT is calculated and used in subroutine SETIC.

19) DF - Vector with dimension 2* NUMNP in which DF(i) (where $i = 2(n-1) + k$) specifies the incremental load for the ξ-component of node n_1 where ξ and k are related as follows:

$$\xi = \begin{cases} x & \text{if RECCYL} = 0 \text{ and } k = 1, \\ y & \text{if RECCYL} = 0 \text{ and } k = 2, \\ r & \text{if RECCYL} \neq 0 \text{ and } k = 1, \\ z & \text{if RECCYL} \neq 0 \text{ and } k = 2. \end{cases}$$

The vector DF actually is used in two different ways in order to save memory space. Since this double usage has caused some confusion in the past, it is perhaps worthwhile to describe how DF is used.

First, in subroutine STEP2 the actual incremental loads are calculated and stored in the DF vector. Thus, on the first iteration of each load step, the vector DF will be initially set equal to the vector RESI (which will be zero if the input flag IRESI $\leq$ 0; if IRESI > 0, then RESI will contain residual loads from the previous load step). Next, still in STEP2, for each element the subroutines STFNS and GEØM are called to calculate the elemental stiffness matrix ST and the incremental load vector CF for the six points within that

19) DF - (continued)

element. Then the elemental incremental loads are added to the system incremental load vector DF. Next, the DF vector may be modified by the consideration of imposed displacement constraints. The vector DF and the system stiffness matrix A are used to calculate the incremental displacement vector DR using the matrix equation

$$[A]\ [DR] = [DF].$$

Finally (again only on the first iteration for each load step), the incremental load vector DF is added to the accumulated load vector F.

The second usage of the vector DF is by subroutine RESID, which calculates the residual nodal forces to find out how much the structure is out of equilibrium. Thus the DF vector is again cleared to zero. The residual forces for each element are calculated using the total elemental stresses ESIG and are stored in the vector CF. The residual load vector DF for the entire structure is formed by superimposing the vectors CF over all of the elements. Next, the DF vector is modified to account for the boundary conditions and perhaps for the imposed displacement constraints. Finally, unless NLØAD < 0, the accumulated load vector F is added to DF. If NLØAD < 0, then for each degree of freedom n specified in the boundary conditions, DF(n) is set to zero. The purpose of the iteration is to reduce the modulus of the DF vector below an acceptable limit.

If IRESI > 0, then the residual load vector DF calculated by subroutine RESID is stored into the vector RESI by subroutine PRINT.

20) DR - Vector with dimension 2* NUMNP in which DR(i) specifies the incremental displacement of the ξ-component of node n (where i, ξ, and n are defined as for DF described above). The vector

20) DR - (continued)

DR is calculated in subroutine STEP2 and then added to the vector DIS in subroutine SETIC.

21) DIS - Vector with dimension 2* NUMNP in which DIS(i) specifies the accumulated displacement of ξ-component of node n (where i, ξ, and n are defined as for DF described above).

The array DIS is initially cleared to zero by subroutine SETUP when the D.S.A. is cleared and then is updated by subroutine SETIC each load step by being increased by the vector DR. The array DIS may also be modified by the subroutine DISCHK if NNDLIM > 0.

22) F - Vector with dimension 2* NUMNP in which F(i) specifies the accumulated load for the ξ-component of node n (where i, ξ, and n are defined as for DF described above). See description of DF above.

23) RESI - Vector with dimension 2* NUMNP in which RESI(i) specifies the residual load for the ξ-component of node n (where i, ξ, and n are defined as for DF described above). See description of DF above.

24) TST - NUMNP * K array (where K = 3 if RECCYL = 0 or K = 4 otherwise) in which TST(i, k) specifies the accumulated total average nodal strain $\varepsilon_{T,k}$ for the i-th nodal point. See the description of ETST for the significance of k. The TST array is calculated in subroutine SETIC and is printed in subroutine PRINT.

25) PST - NUMNP * K array (where K = 3 if RECCYL = 0 or K = 4 otherwise) in which PST(i, k) specifies the accumulated plastic average nodal strain $\varepsilon_{P,k}$ for the i-th nodal point. See the description of ETST for the significance of k. The PST array is calculated in subroutine SETIC and is printed in subroutine PRINT.

Name (continued) Description (continued)

26) EST - NUMNP * K array (where K = 3 if RECCYL = 0 or K = 4 otherwise) in which EST(i, k) specifies the accumulated elastic average nodal strain $\varepsilon_{E,k}$ for the i-th nodal point. See the description of ETST for the significance of k. The EST array is calculated in subroutine SETIC and is printed in subroutine PRINT.

27) DTST - NUMNP * K array (where K = 4 if RECCYL = 0 or K = 5 otherwise) in which DTST(i, k) specifies the incremental total nodal strain $\varepsilon_{T,k}$ for the i-th nodal point. The significance of k is described by the following table:

k	$\varepsilon_{T,k}$ for RECCYL = 0	$\varepsilon_{T,k}$ for RECCYL $\neq$ 0
1	$\varepsilon_{T,x}$	$\varepsilon_{T,r}$
2	$\varepsilon_{T,y}$	$\varepsilon_{T,z}$
3	$\gamma_{T,xy}$	$\varepsilon_{T,\theta}$
4	rotation	$\gamma_{T,rz}$
5	--	rotation

The DTST array is calculated by subroutine SETIC using the locally defined arrays DETST and RØT.

28) SIG - NUMNP * 5 array in which SIG(i, k) specifies the accumulated average stress σ_k for the i-th nodal point. The significance of k is described under ESIG.

29) DSD - Vector of dimension NID whose elements are in a one-to-one correspondence with the elements of the vector ND. DSD(i) specifies the magnitude (including sign) of the ith imposed displacement. If BLANAL = 0 and NID > 0, then the DSD array is obtained from the input array DISPL. If BLANAL $\neq$ 0 and RECCYL = 0 (single fold option), then DSD(1),..., DSD(NID) are calculated in subroutine STEP2 using the quantities XLNGTH, RSFØLD, BLTHNS, etc.

Name (continued) Description (continued)

30) DLIM - 3* NNDLIM array in which DLIM(i, j) specifies the following information defining the j-th limiting nodal displacement condition:

i	Description of DLIM(i, j)
1	Nodal point index;
2	Component code: $= 1 \begin{cases} \text{for } x & \text{if RECCYL} = 0, \\ \text{for } r & \text{if RECCYL} \neq 0. \end{cases}$ $= 2 \begin{cases} \text{for } y & \text{if RECCYL} = 0, \\ \text{for } z & \text{if RECCYL} \neq 0; \end{cases}$
3	Magnitude (including sign) of the limiting displacement.

The array DLIM is used by subroutine DISCHK (if NNDLIM > 0) to check for limiting nodal displacements. The array DLIM is input if NNDLIM > 0.

31) DLFLAG - Vector of dimension NNDLIM that specifies which, if any, nodes have reached their limiting displacements. Suppose DLIM(1, i) contains the node index n. If DLFLAG(i) = 0, then node n has not yet reached its limiting displacement. If DLFLAG(i) ≠ 0, then node n has reached its limiting displacement. If NNDLIM > 0, then the vector DLFLAG is initially cleared to zero by subroutine SETUP when the D.S.A. is cleared and the appropriate elements of DLFLAG are set to 1.0 by subroutine DISCHK as the corresponding limiting displacement is reached.

The D.S.A. pointer of DLFLAG is LDLFLG.

32) VEM - Vector with dimension NMAT in which VEM(i) specifies E_i, the elastic modulus of the i-th material. VEM must be input.

33) VXU - Vector with dimension NMAT in which VXU(i) specifies ν_i, Poisson's ratio for the i-th material. VXU must be input.

Name (continued) Description (continued)

34) VTH - Vector with dimension NMAT in which VTH(i) specifies the initial thickness of the i-th material. VTH must be input. If BLANAL $\neq$ 0, the VTH array is set to indicate unit thickness.

35) VXI - Vector with dimension NMAT in which VXI(i) specifies $\chi_i = \frac{E_{Pi}}{E_i}$, the plastic/elastic modulus ratio for the ith material. VXI must be input.

36) VYP - Vector with dimension NMAT in which VYP specifies Y_i, the initial yield point for the i-th material. VYP must be input.

37) DELTAP - 2* NDP array in which DELTAP(1, i) specifies n_i, the load step index up to (and including) which the load increment ΔP_i is to be used. That is, if i = 1, then the load increment ΔP_1 is used from the initial conditions up to load step n_1. if i >1, then the load increment ΔP_i is to be used for each load step n where $n_{i-1} < n \leq n_i$. The load increment ΔP_i is specified in DELTAP(2, i). If NDP > 0, then the array DELTAP must be input. If NDP = 0 (or is not entered), then the array DELTAP must not be input.

The D.S.A. pointer for DELTAP is LDELTP.

38) THETA - NUMEL * 3 array in which THETA(n, i) specifies

$$\theta_i = \frac{\frac{x_j}{x_i} - 1}{\frac{x_j}{\Delta P} + 1},$$

where n is the element index; i is the point index within element n(i=1, 2, or 3); j = |NLØAD| (where NLØAD < 0); and P is the load step size. If NLOAD < 0, then the array is calculated in subroutine GEØM and is also used in subroutine SETIC.

39) DISPL - 3 * NID array which is used, if NID > 0 and if BLANAL = 0, to specify the imposed displacement constraints through the Namelist input data.

39) DISPL - (continued)

a) DISPL(1, i) specifies the node for which the i-th constraint is to be imposed;

b) DISPL(2, i) specifies the component code for the i-th constraint:

$$= 1 \begin{cases} \text{for } x & \text{if RECCYL} = 0 \\ \text{for } r & \text{if RECCYL} \neq 0 \end{cases}$$

$$= 2 \begin{cases} \text{for } y & \text{if RECCYL} = 0, \\ \text{for } z & \text{if RECCYL} \neq 0; \end{cases}$$

c) DISPL(3, i) specifies the magnitude (including sign) of the i-th imposed displacement.

The information specified in DISPL(1, i) and DISPL(2, i) for each i is used to generate an entry in the ND array. The value specified in DISPL(3,i) is divided by MAXST, the total number of load steps and the quotient is stored in the DSD array. After the Namelist input data is processed by subroutine SETUP, the memory space assigned to the array DISPL is no longer needed and hence is released by redefining the value in the D.S.A. pointer LA.

If NID = 0 or if BLANAL ≠ 0, then the array DISPL is not used.

40) IBCØND - 2* NUMBC array which is used, provided that BLANAL = 0, to specify the boundary conditions through the Namelist input data.

IBCØND(1, i) specifies n, the nodal index of the i-th node to be constrained, and IBCØND(2, i) specifies C, the corresponding constraint code, where

40) IBCØND - (continued)

$$C = \begin{cases} 1 & \text{if node n is to be constrained in the x-direction only,} \\ 2 & \text{if node n is to be constrained in the y-direction only,} \\ 0 & \text{if node n is to be constrained in both the x- and the y-directions.} \end{cases}$$

The information specified in the IBCØND array is used to generate the NEBC vector.

After the NEBC array is used, the IBCØND array is no longer needed, and thus the memory space assigned for the array IBCØND is released to be used for other purposes.

If BLANAL $\neq$ 0, then the NEBC array will be generated in a different manner. Therefore no memory space is reserved for the IBCØND array and the user must not attempt to specify any entries for IBCØND in the Namelist input.

The D.S.A. pointer for the array IBCØND is LBCØND.

41) CYCM - Vector with dimension NMAT in which CYCM(i) specifies M_i, a coefficient used to calculate the fatigue life of element layers which are composed of the i-th material. The arrays CYCM and CYCZ are calculated in subroutine SETUP and are used in subroutine PRINT only if BLANAL and FLIFE are both non-zero.

M_i and Z_i (see CYCZ below) are coefficients for a least squares fit to the relation

$$\varepsilon = MN^Z$$

for the i-th material. Thus, if $(\varepsilon_{max})_i$ is the maximum plastic strain for layer j and if layer j consists of material i, then the fatigue life N_j (i.e. number of cycles to failure) of layer j is calculated from the relation

41) CYCM - (continued)

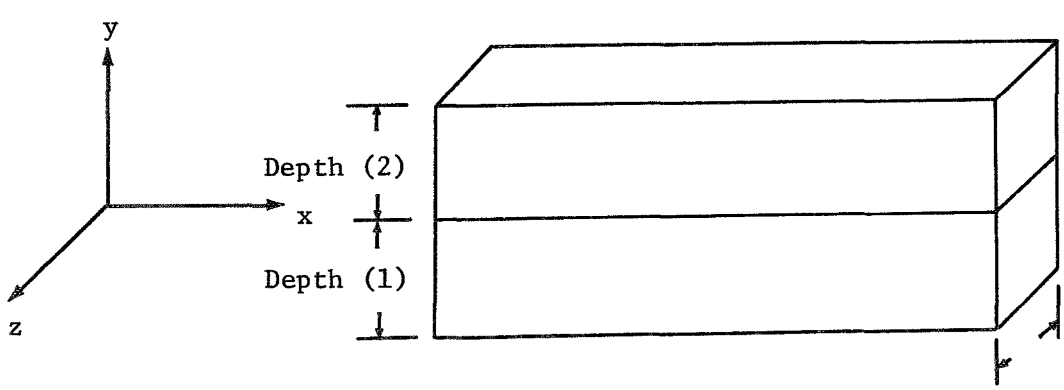
$$N_j = \left[\frac{(\varepsilon_{max})_j}{M_i}\right]^{1/Z_i}$$

42) CYCZ - Vector with dimension NMAT in which CYCZ(i) specifies Z_i, a coefficient used along with M_i stored in CYCM(i) to calculate the fatigue life of element layers which are composed of the i-th material. See the description of CYCM above for further details.

43) MLAYER - Vector with dimension NLAYER in which MLAYER(i) specifies, if BLANAL ≠ 0 and if NMAT > 1, the material index for the i-th element layer. If BLANAL = 0, the array MLAYER is not defined. If NMAT = 1, then it is not necessary to specify MLAYER since all elements are composed of the same material.

The D.S.A. pointer for MLAYER is LMLAYR.

44) DEPTH - Vector with dimension NLAYER in which DEPTH(i) specifies the initial "depth" of each element layer provided that BLANAL ≠ 0. In this case the array VTH is set to indicate unit "thickness" into the plane of the cross-section as indicated by the following example:



VTH(1) = VTH(2) = 1.0

44) DEPTH - (continued)

If BLANAL = 0, then the array DEPTH is not defined.

45) CYCTØF - 40* NMAT array in which a table of $N(\varepsilon)$ may be entered for each material provided that BLANAL $\neq$ 0 and FLIFE $\neq$ 0. $N(\varepsilon)$ is the number of cycles to failure as a function of the plastic strain ε.

In the CYCTØF array, CYCTØF(2i-1, j) and CYCTØF(2i, j) respectively specify $\varepsilon_{i,j}$ and $N_{i,j}$, the ith value of plastic strain and the corresponding number of cycles to failure for the j-th material. The index i may range from 1 to 20. Zero entries for either $\varepsilon_{i,j}$ or $N_{i,j}$ will cause the pair to be ignored.

If BLANAL $\neq$ 0 and FLIFE $\neq$ 0, then the array CYCTØF is used by subroutine SETUP to calculate the vectors CYCM and CYCZ. After CYCM and CYCZ are calculated, the array CYCTØF is no longer needed and therefore the memory space for CYCTØF is released for the storage of other arrays.

The D.S.A. pointer for CYCTOF is LCYCF.

46) CØRNER - 3* NUMCP array in which the coordinates of the corner points are specified if BLANAL = 0. If BLANAL $\neq$ 0, then the array CØRNER is not defined.

CØRNER(1, i) specifies the node index n. CØRNER(2, i) specifies x_n if RECCYL = 0 or r_n if RECCYL $\neq$ 0, and CØRNER(3, i) specifies y_n if RECCYL = 0 or z_n if RECCYL $\neq$ 0.

The array CØRNER is used by subroutine SETUP to generate the arrays XØRD and YØRD. The values x_n (or r_n) and y_n (or z_n) are stored in XØRD(n) and YØRD(n) respectively. After the arrays XØRD and YØRD are generated, the array CØRNER is no longer necessary and the memory space may be used for storing other quantities. Thus, the D.S.A. pointer for CØRNER is LA so that the array CØRNER is assigned to the same area of memory as the array A, which is not used until later in the program.

Name (continued) Description (continued)

47) IA - Vector of dimension 6* NUMEL which is used by subroutine SETUP to transpose the array NP. (See the description of NP for details.) The D.S.A. pointer for IA is LDISPL so that the IA array overlays the DISPL array and is subsequently overlayed by the A array.

48) A - (2* NUMNP) * IBAND array in which the system stiffness band matrix A is stored. Since the band width IBAND of matrix A is calculated by subroutine SETUP after all of the input data has been read and processed, the memory required for storing A is not requested until after SETUP returns control back to the main program. Then the D.S.A. pointer LA of the array A is adjusted so that the array A overlays arrays such as DISPL and IBCØND (if BLANAL = 0) or CYCTØF (if BLANAL ≠ 0), which are no longer needed. The pointer NBUCKT which indicates the total length of the D.S.A. is adjusted to allow for the A array, and the subroutine MRELSE is called to obtain the required amount of memory for the D.S.A.

49) TPR - Vector with dimension NLØAD in which TPR(i) specifies the accumulated load applied to side S_i = LØADS(i) of element LØADE(i).

$$TPR(i) = n\ DPR(i)$$

where n is the current load step index and where DPR(i) is as described below.

50) DPR - Vector with dimension NLØAD in which DPR(i) specifies the increment of load to be applied to side S_i = LOADS(i) of element LØADE(i) for each load step. The vectors TPR and DPR are used in connection with the vector VPR and are thus defined and used only if NLØAD > 0 and if NDP and PRESS are not specified in the Namelist input.

6.3 Labeled COMMON Blocks

This section describes the quantities which are defined in each labeled COMMON block.

All of the labeled COMMON blocks are defined in the main program. In general, a COMMON block is defined within a subroutine if one or more quantities in that COMMON block are used in the subroutine.

COMMON Block : CDIMEN

(Computed Dimensional Quantities)

1) NCDIM - Number of quantities in the CDIMEN COMMON block.

2) LREC1 - Length of first record of the restart tape. (33)

3) NEQ - Number of degrees of freedom. (2* NUMBC = dimension of several vectors: DF, DR, DIS, F, and RESI.)

4) NEQBC - Total number of boundary conditions: (NUMBC $\leq$ NEQBC $\leq$ 2* NUMBC + NNDLIM.)

5) IBAND - Band width of matrix A.

6) LENA - Number of cells of memory required for band matrix A. (NEQ * IBAND)

7) NEL6 - 6* NUMEL = dimension of several arrays: NP, BETA and EWP.

8) NEL64 - Dimension of several arrays: ETST, EPST and EEST;

$$\text{NEL64} = 6 * n * \text{NUMEL}$$

$$\text{WHERE} \quad n = \begin{cases} 3 & \text{if RECCYL} = 0, \\ 4 & \text{if RECCYL} \neq 0. \end{cases}$$

9) NPTSY - Number of points in "y-direction". (See description of Automatic Mesh Generation Procedure.)

(NPTSY is calculated and used only if BLANAL $\neq$ 0 and has the value 2* NLAYER + 1.)

10) NPTSX - Number of points in "x-direction." (Similar to NPTSY except NPTSX = 2* NXINT + 1)

11) NCYCF2- 2* NCYCF.

COMMON Block : CMESH

(Computed Quantities Used in Automatic Mesh Generation)

1) BLTHNS - Bladder thickness t where

$$t = \sum_{I = 1}^{NLAYER} DEPTH(I)$$

2) XINTN - Same as NXINT = No. of intervals in "x-direction" except XINTN is a "real" (i.e. floating point) quantity.

3) XYRAT - Ratio of total length in "x-direction" to total bladder thickness, i.e.

XLNGTH = XYRAT * BLTHNS

if XLNGTH is not explicitly entered in the Namelist input data. (XYRAT is the floating point equivalent of LWRAT.)

4) DPRAT -

$$C = \left(\frac{RFINAL}{RPF\emptyset LD}\right)^{\frac{1}{MAXST-1}}$$

Factor calculated in subroutine MSHGEN and used in subroutine STEP2 to modify the fold radius for a single fold analysis.

COMMON Block : FFLAGS

(Fixed Flags)

1) NØNTRY - = 100000 to indicate "no entry" for various Namelist input quantities.

2) BLANK - Word of blanks for clearing and checking the input header.

COMMON Block : HEADER

HEADER - Input header consisting of up to 64 characters (including blanks).

COMMON Block : IDIMEN

(Input Dimensional Quantities)

1) NDIMEN - Number of quantities in the IDIMEN COMMON block. (18)

2) NUMEL - Number of elements.

   a) Must be input if BLANAL = 0;

   b) Will be set to 2 * NLAYER * NXINT if BLANAL ≠ 0.

3) NUMNP - Number of nodal points:

   a) Must be input if BLANAL = 0;

   b) Will be set to NPTSX * NPTSY if BLANAL ≠ 0.

4) NUMCP - Number of corner points:

   a) Must be input if BLANAL = 0;

   b) Will be set to (NLAYER + 1) * (NXINT + 1) if BLANAL ≠ 0.

5) NUMBC - Number of boundary condition entries in IBCØND array and must be specified in Namelist input if BLANAL = 0. If BLANAL ≠ 0, then NUMBC will be set to n * NPTSY where

$$n = \begin{cases} 1 & \text{if RECCYL} = 0 \\ & \text{(Single fold option),} \\ 2 & \text{if RECCYL} \neq 0 \\ & \text{(Double fold option).} \end{cases}$$

6) NLØAD - Number of elements which are pressure loaded:

   a) A non-zero value must be specified for either NLØAD or NID (but not for both) if BLANAL = 0.

   b) If NLØAD > 0, then NLØAD specifies the number of entries in each of the arrays LØADE and LØADS (i.e. the number of elements which are pressure loaded).

   c) If RECCYL = 0, then NLØAD must be ≥ 0.

   d) If RECCYL ≠ 0, then a negative value may be entered for NLØAD. If NLOAD = -n (where n > 0), then n is the index of the node which is used as a reference for changing the radius of curvature of the secondary fold.

   e) If BLANAL ≠ 0, then

$$\text{NLØAD} = \begin{cases} 0 \text{ if RECCYL} = 0 \text{ (Single fold case)} \\ -(\text{NANGIN} * \text{NPTSY} * 1) \\ \quad \text{if RECCYL} \neq 0 \text{ (Double fold case).} \end{cases}$$

7) NID - Number of imposed displacement constraints:

a) See NLØAD above for compatibility requirements.

b) If BLANAL = 0 and
if NID > 0, then NID specifies the number of imposed displacement specifications in the array DISPL.

c) If BLANAL ≠ 0, then NID will be set

$$\text{equal to} \begin{cases} 2 & \text{if RECCYL} = 0 \text{ (Single fold case),} \\ 0 & \text{if RECCYL} \neq 0 \text{ (Double fold case).} \end{cases}$$

8) NMAT - Number of materials.
(nominal value is 1)

9) NDP - Number of pairs of entries in the DELTAP array.
(nominal value is 0)
If BLANAL ≠ 0, then NDP is set to 0.

10) MAXBW - Maximum band width allowed for the matrix A.
(nominal value is 50)

11) MAXST - Number of load steps.
(nominal value is 25)

12) NNDLIM - Number of limiting nodal displacement entries in the array DLIM.
(nominal value is 0)

13) MAXBKT - Maximum allowable size of the dynamic storage array BUCKET. If the program requests a larger amount of memory than permitted by MAXBKT, then an error message will be printed and the run will be terminated.
(nominal value is 25000 (decimal))

14) NLAYER - Number of element layers.
(Must be entered if BLANAL ≠ 0)

15) LWRAT - Ratio of the total length in the x-direction to the total bladder thickness and/or the ratio of the number of intervals in the x-direction to the number of element layers. LWRAT is significant only if BLANAL ≠ 0 and if RECCYL = 0.

COMMON Block : IDIMEN (continued)

15) LWRAT - (continued)
(See XLNGTH and NXINT for further details)
(nominal value is 2)

16) NANGIN - Number angular intervals for the double fold option
(i.e. BLANAL $\neq$ 0 and RECCYL $\neq$ 0).
(nominal value is 18)

17) NXINT - Number of intervals in the "x-direction":

a) Required only if BLANAL $\neq$ 0.
b) If RECCYL = 0, then the nominal value of NXINT is
LWRAT * NLAYER.
c) If RECCYL $\neq$ 0, then NXINT will be set equal to NANGIN.

18) NCYCF - Maximum number of pairs of entries (i.e. ε_i and N_i)
in the CYCTØF table for any material.

[nominal value is 0;
must not exceed 20.]

COMMON Block : IFLAGS
(Input Flags)

1) NFLGS - Number of quantities in the IFLAGS COMMON block. (10)

2) RESTRT - Restart dump number:
= n > 0 if the current run is to be a restart
from dump n of the input restart tape;

= 0 otherwise.
(nominal value is zero)

3) RECCYL - Coordinate system flag:
= 0 if rectangular coordinates are used (i.e. plane strain analysis). If, in addition, BLANAL $\neq$ 0, then a single fold bladder problem is being considered.

$\neq$ 0 if cylindrical coordinates are used (i.e. axisymmetric strain analysis). If, in addition, BLANAL $\neq$ 0, then a double fold bladder problem is being considered.
(nominal value = 0)

4) CØNVER - Convergence criterion ε:

$$\text{If} \quad \frac{\left| \sum_{i=1}^{N} (\Delta F_x)_i + \sum_{i=1}^{N} (\Delta F_y)_i \right|}{\left| \sum_{i=1}^{N} F_{x,i} + \sum_{i=1}^{N} F_{y,i} \right|} < \varepsilon$$

the solution for a given load step is considered to have converged. N is the number of nodes. $(\Delta F_x)_i$ and $(\Delta F_y)_i$ are the x- and y-components of the incremental load for the ith node. $F_{x,i}$ and $F_{y,i}$ are the x- and y-components of the accumulated load for the ith node. (Nominal value is 10^{-4}, except for the double fold option for which the nominal value is 0.1)

5) IPRIN - Print interval:

The strains (elastic, plastic, and total), the stresses, the plastic work (incremental and accumulated), and the coordinates for each nodal point will be printed out every IPRIN load steps.

(nominal value is 1)

6) ITER - Maximum number of iterations permitted for a single load step. If the solution does not converge within ITER iterations, then a warning message is printed out, the available data is printed, and the run is terminated.

(nominal value is 10)

7) IRESI - Residual loads option flag:

If IRESI > 0, then at the beginning of each load step the incremental load vector (DF) is set equal to the residual load vector (RESI) from the previous step.

If IRESI $\leq$ 0, then at the beginning of each load step the incremental load vector (DF) is cleared to zero.

(nominal value is 0)

COMMON Block : IDIMEN (continued)

8) SAVETP - Restart Flag:

$\neq$ 0 if it is desired to save data periodically on a "save tape" for a possible restart in a subsequent run;

= 0 otherwise.

(nominal value is 0)

9) BLANAL - Bladder Analysis option flag:

$\neq$ 0 if the Bladder Analysis option is desired,

= 0 otherwise.

(nominal value is 0)

10) FLIFE - Fatigue Life option flag:

$\neq$ 0 if the fatigue life is to be calculated and printed out;

= 0 otherwise.

(nominal value is 0)

COMMON Block : INDEX

(Indices and Computed Flags)

1) IFIX - First pass flag used by subroutines BLADER, STEP2, and GEØM.

= 0 on the first iteration of each load step;

$\neq$ 0 on subsequent iterations.

2) NV - Element index - In particular NV is set by subroutine STEP2 and passed to subroutines STFNS and GEØM.

3) ISTEP - Load step index

4) JSTEP - Index used with DELTAP array by subroutine STEP2.

5) NSTEP - Load step limit used with DELTAP array by subroutine STEP2.

6) IMAT - Material code for element NV; used by subroutines RESID, SETIC and STEP2.

7) LMAT - Material code for element NV-1; used by subroutines SETIC and STEP2.

8) ERRFLG - Error flag used by subroutines DIMCHK, EERRXX, GINPUT and PRINT.

= 1.0 until an error is detected.

= 0.0 if an error has been detected.

9) J34 - Index which is used in subroutines GEØM, PRINT, RESID, SETIC and STFNS to pick up or store τ_{xy} or τ_{rz} in the proper position in the arrays ESIG and DESIG. J34 is also used by subroutine SETIC to store γ_{xy} or γ_{rz} in the proper position in the arrays ETST, EPST, and EEST. Finally, J34 is also used in subroutine STFNS to determine the size of matrices which are used to calculate the stiffness matrix ST.

J34 is set in subroutine DIMCHK as follows:

$$J34 = \begin{matrix} 3 \text{ if RECCYL} = 0, \\ 4 \text{ if RECCYL} \neq 0. \end{matrix}$$

10) J43 - Index used by subroutines SETIC and STFNS to pick up and store σ_z or σ_θ in the proper position in the array ESIG. J43 is set in subroutine DIMCHK as follows:

$$J34 = \begin{matrix} 4 \text{ if RECCYL} = 0, \\ 3 \text{ if RECCYL} \neq 0. \end{matrix}$$

11) J45 - Index used by subroutine SETIC to store the rotation component of the incremental total nodal strain in the proper position in the array DTST. J45 is set in subroutine DIMCHK as follows:

$$J45 = \begin{matrix} 4 \text{ if RECCYL} = 0, \\ 5 \text{ if RECCYL} \neq 0. \end{matrix}$$

12) I63 - Index used by subroutine STFNS to determine the maximum value of I in the array C(I, J, K). I63 is set in subroutine DIMCHK as follows:

$$I63 = \begin{cases} 6 & \text{if RECCYL} = 0, \\ 3 & \text{if RECCYL} \neq 0. \end{cases}$$

13) I69 - Index used by subroutine STFNS to store the matrix UV in the proper position within the matrix RMM. I69 is set in subroutine DIMCHK as follows:

$$I69 = \begin{cases} 6 & \text{if RECCYL} = 0, \\ 9 & \text{if RECCYL} \neq 0. \end{cases}$$

14) I912 - Index used by subroutine STFNS to determine the number of rows and columns of the matrix AA and the number of rows in the matrix RMM. I912 is set by subroutine DIMCHK as follows:

$$I912 = \begin{cases} 9 & \text{if RECCYL} = 0, \\ 12 & \text{if RECCYL} \neq 0. \end{cases}$$

15) LSTEPO - Initial value of load step index ISTEP. If RESTRT = 0 (i.e. an initial run), then LSTEPO is set to 1 in subroutine GINPUT. If RESTRT ≠ 0 (i.e. a restart run), then LSTEPO is read in from the restart tape.

16) NPRIN - Index used by subroutine PRINT to determine when the next printout of the point strains, nodal coordinates, point stresses, and plastic work is to be made. When the load step index ISTEP reaches NPRIN, then the above quantities are printed out and NPRIN is increased by IPRIN.

NPRIN is initially set equal to IPRIN in subroutine GINPUT. If RESTRT ≠ 0 (i.e. on a restart run), NPRIN is set equal to LSTEPO + IPRIN, where LSTEPO is read from the restart tape.

17) NXTLS - Index used by subroutine PRINT to determine when the next dump of the restart data is to be made. When the load step index ISTEP reaches NXTLS, the load step dependent data is written onto the restart tape. NXTLS is initially set equal to LSTEPO - 1 in subroutine BLADER and then is incremented by an appropriate amount in subroutine PRINT on the first load step and each time a dump of the restart data is made.

18) INDMP - Index used by subroutine PRINT to determine the frequency of the dumps of the restart data. The load step dependent data is written onto the restart tape every INDMP load steps. INDMP is initially set equal to 5 in subroutine GINPUT. If there are nonzero entries in the input array LSSAVE, then INDMP will be set equal to LSSAVE(2, I) when the load step index ISTEP satisfies

$LSTEPO \leq ISTEP \leq LSSAVE(1, I)$ if $I = 1$,

or $LSSAVE(1, I-1) \leq ISTEP \leq LSSAVE(1, I)$ if $I > 1$.

19) LSLIM - Index used by subroutine PRINT to determine the frequency of the dumps of the restart data. LSLIM is the current value of LSSAVE(1, I) mentioned in the description of INDMP above.

20) IDUMP - Dump number - used by subroutine PRINT to identify the dumps of the restart data.

21) ILSS - Index used by subroutine PRINT to determine the frequency of the dumps of the restart data. ILSS specifies the current value of the index I which indexes the array LSSAVE mentioned in the description of INDMP above. ILSS is initially set equal to 1 in subroutine GINPUT and is incremented when appropriate in subroutine PRINT.

COMMON Block : INDEX (continued)

22) VPRFLG - Variable pressure loading flag:

VPRFLG ≠ 0 if NLØAD > 0, and if PRESS and NDP are not specified in the Namelist input;

VPRFLG = 0 otherwise.

COMMON Block : IØUNIT

(Logical Unit Names for Restart Tapes)

1) IRSIN - Logical unit for input restart tape (nominal value is 1).

2) IRSØUT - Logical unit for output restart tape (nominal value is 3).

COMMON Block : IPERM

(Cyclic Permutation of Integers 1,2,3)

IPERM(1) = 2

IPERM(2) = 3

IPERM(3) = 1

COMMON Block : LBUCKT

(Dynamic Storage Array Pointers)

	Pointer Name	Array Name(s)
1)	LNP	NP
2)	LLØADE	LØADE
3)	LLØADS	LØADS
4)	LNEBC	NEBC
5)	LND	ND
6)	LMATEL	MATEL
7)	LBETA	BETA
8)	LVPR	VPR
9)	LETST	ETST
10)	LEPST	EPST
11)	LEEST	EEST

COMMON Block : LBUCKT (continued)

	Pointer Name	Array Name(s)
12)	LESIG	ESIG
13)	LEWP	EWP
14)	LXØRD	XØRD
15)	LYØRD	YØRD
16)	LDWP	DWP
17)	LWP	WP
18)	LCØUNT	CØUNT
19)	LDF	DF
20)	LDR	DR
21)	LDIS	DIS
22)	LF	F
23)	LRESI	RESI
24)	LTST	TST
25)	LPST	PST
26)	LEST	EST
27)	LDTST	DTST
28)	LSIG	SIG
29)	LDSD	DSD
30)	LDLIM	DLIM
31)	LDLFLG	DLFLAG
32)	LVEM	VEM
33)	LVXU	VXU
34)	LVTH	VTH
35)	LVXI	VXI
36)	LVYP	VYP
37)	LDELTP	DELTAP
38)	LTHETA	THETA
39)	LDISPL	DISPL, IA
40)	LBCØND	IBCØND
41)	LCYCM	CYCM
42)	LCYCZ	CYCZ
43)	LMLAYR	MLAYER

COMMON Block : LBUCKT (continued)

Pointer Name	Array Name(s)
44) LDEPTH	DEPTH
45) LCYCF	CYCTØF
46) LA	A, CØRNER
47) LTPR	TPR
48) LDPR	DPR
49) NBUCKT	End of Dynamic Storage Array.

COMMON Block : LSSAVE

LSSAVE is a 2* 10 matrix which is used to control the frequency at which dumps of the restart data are made. A detailed description is given in Section 2.3.2 of this document.

COMMON Block : MESHIN
(Automatic Mesh Generation Input Data)

1) NMESH - Number of quantities in the MESHIN CØMMON block. (9)

2) XSTART - x_o = x-coordinate of left edge of the rectangular area (single fold option only).
(nominal value is 0)

3) YSTART - y_o = y-coordinate of bottom edge of the rectangular area (single fold option only).
(nominal value is 0)

4) XLNGTH - ℓ = total length of plane area in the x-direction (single fold option only).
(nominal value is LWRAT * BLTHNS. See footnote at end of section.)

5) RPFØLD - R_p = initial radius of primary fold: inner fold radius for double fold; radius to median line for single fold.
(nominal value is RPTRAT * BLTHNS. See footnote at end of section.)

COMMON Block : MESHIN (continued)

6) RSFØLD - R_s = radius of secondary fold (double fold option only). (nominal value is RSTRAT * BLTHNS. See footnote at end of section.)

Also used for current value of fold radius (to median line) for single fold.

7) RPTRAT - $\frac{R_p}{t}$ = ratio of primary fold radius R_p to total bladder thickness t. nominal value is 1000 for the single fold option and 10 for the double fold option.

8) RSTRAT - $\frac{R_s}{t}$ = ratio of secondary fold. R_s to total bladder thickness t (double fold option only). nominal value is 100. See footnote at end of section.

9) RFINAL - Final inner folding radius R_f (double fold option only). No nominal value. RFINAL must be specified if BLANAL $\neq$ 0 and RECCYL $\neq$ 0.

FOOTNOTE: The total bladder thickness t is stored in BLTHNS and is computed according to the relation

$$t = \sum_{i=1}^{NLAYER} DEPTH(i)$$

COMMON Block : PRØPS

(Material Properties and Miscellaneous Floating Point Quantities)

PRESS: Total pressure loading (input quantity). See description of PRESS under item (36) in section 2.3 of this report.

DPRESS: Incremental pressure loading: If BLANAL = 0, then DPRESS may be calculated from PRESS, from the DELTAP array, or from the VPR array. If BLANAL $\neq$ 0 and RECCYL $\neq$ 0 (double fold), then DPRESS is computed by

COMMON Block : PRØPS

DPRESS: (continued)

$$\text{DPRESS} = \frac{\text{(RFINAL - RSFØLD)}}{\text{MAXST}}$$

If BLANAL $\neq$ 0 and RECCYL = 0 (single fold), then DPRESS is set equal to zero.

EM - Elastic modulus for element NV.

XU - Poisson's ratio for element NV.

TH - Initial thickness of element NV.

XI - Plastic/elastic modulus ratio for element NV.

YP - Initial yield point for element NV.

AREA - Area of element NV.

TPRESS - Accumulated pressure loading

RMAXS - Maximum number of load steps (i.e. MAXST) in floating point format.

COMMON Block : RSDATA

(First Record of Restart Tape)

The COMMON block RSDATA consists of one array, RSDATA. At the time of this writing, the dimension of the array RSDATA is 35 (on the CDC 6500 computer). The significance of RSDATA(I) is specified below. In general, RSDATA(I) is a fixed point integer quantity and is sometimes referenced by the integer name IRSDAT(I). In subroutine RRST23, the quantity RSDATA(I) may be assigned another equivalent name for convenience.

If RESTRT $\neq$ 0 in the Namelist input, then the first record of the restart tape is read into the array RSDATA.

COMMON Block : RSDATA (continued)

I	RRST23 Equivalent Name	Description
1	INUMEL	NUMEL
2	INUMNP	NUMNP
3		NUMCP
4		NUMBC
5	INLØAD	NLØAD
6	INID	NID
7	INMAT	NMAT
8	INDP	NDP
9		MAXBW
10		MAXST
11	INDLIM	NNDLIM
12		MAXBKT
13		NLAYER
14		LWRAT
15		NANGIN
16		NXINT
17		NCYCF
18		RECCYL
19		CØNVER
20		IPRIN
21		ITER
22		IRESI
23		SAVETP
24		BLANAL
25		FLIFE
26		PRESS
27		HEADER(1)
28		HEADER(2)
29		HEADER(3)
30		HEADER(4)

COMMON Block : RSDATA (continued)

I	RRST23 Equivalent Name	Description
31		HEADER(5)
32		HEADER(6)
33		HEADER(7)
34	Not Used	
35	Not Used	

COMMON Block : ST

(Stiffness Matrix)

The COMMON block ST contains the 12x12 elemental stiffness matrix ST and the incremental element force vector CF (with dimension 12 corresponding to the 12 degrees of freedom for each element).

COMMON Block : TEMP

(Intermediate Stiffness Matrix)

The COMMON block TEMP contains a nx12 matrix which is a intermediate result in the calculation of the conventional elemental stiffness matrix (where n = 9 if RECCYL = 0 or n = 12 if RECCYL $\neq$ 0).

COMMON Block : UV

(U, V Matrices)

The COMMON block UV contains a 3x6x2 array UV in which UV(I, J, 1) specifies U_{ij} and UV(I, J, 2) specifies V_{ij}.

COMMON Block : XYCØØR

(x,y or r,z Coordinates)

	RECCYL = 0	RECCYL ≠ 0
X1	x_1	r_1
X2	x_2	r_2
X3	x_3	r_3
Y1	y_1	z_1
Y2	y_2	z_2
Y3	y_3	z_3

6.5 Summary List of Subroutines

This section presents a complete list of all subroutines which make up the BLADER program with the exception of standard Fortran IV library subroutines. A brief description of each subroutine is included.

1) BLADER - The main program - defines all labeled COMMON blocks and serves as the basic logic control routine.

2) BLØC - Initializes the pointers for the Dynamic Storage Array.

3) DIMCHK - Initializes several index quantities and checks the problem dimension input quantities for obvious errors.

4) DISCHK - Checks for nodes which have reached their limiting displacements.

5) EERRXX - Error procedure routine. Calls PDUMP to dump out various quantities and calls PRINT to print the terminal output.

6) FMCALC - Calculates the total force and the total moments for axisymmetric problems.

7) GEØM - Calculates the incremental elemental force vector for each element.

8) GINPUT - Read the general input data (i.e. problem dimensions, control flags, etc.) from the Namelist input cards. On option the first record of the restart tape will also be read and/or written.

9) MSHGEN - Mesh generation routine - generates the element mesh, node coordinates, boundary conditions, loading specifications, etc. for a bladder analysis problem.

10) PRINT - Main printout routine - prints out the displacements, accumulated loads, and residual loads. At specified load step intervals the nodal coordinates; nodal elastic, plastic and total strains; nodal stresses; and plastic work are printed out. On option the estimated fatigue life is calculated and printed out. Also on option, the load step dependent quantities are written on the restart tape.

6.5 Summary List of Subroutines (continued)

11) RESID - Determines the total error in nodal print equilibrium by calculating the residual nodal point forces.

12) RRST23 - Reads the second category of restart data (input arrays) and the third category (load step dependent quantities).

13) SETIC - Calculates the incremental and accumulates stresses and strains.

14) SETUP - Reads the input array data and initializes various quantities.

15) STEP2 - Assembles the incremental system stiffness matrix and solves for the incremental displacements.

16) STFNS - Calculates the instantaneous stiffness matrix ST for each element.

17) WRST2 - Writes the second category of restart data (input arrays) on the output restart tape.

6.6 Restart Feature

The usage of the restart procedure is described in section 2.5 of this report. In the current section, the programming aspects of the restart feature are described. In particular, the format of the restart tape is described.

The two input parameters SAVETP and RESTRT control the restart logic. An output restart tape will be generated only if SAVETP $\neq$ 0. An input restart tape will be read only if RESTRT $\neq$ 0. The value specified for RESTRT determines which restart dump (i.e. which load step) will be used to restart the problem.

Other input parameters such as NLØAD, NDP, BLANAL, etc. determine the format of the restart tape as described below. The data on the restart tape may be divided into three general categories:

1) Problem dimensions and other general input quantities.
2) Computed constants and input arrays.
3) Load step dependent quantities

The formats of the different categories are described below in sections 6.6.1, 6.6.2 and 6.6.3.

6.6.1 Problem Dimensions and Other General Input Quantities

The first category of restart data occupies the first record of the restart tape and is both written and read in subroutine GINPUT. The format is as follows:

Restart Tape (First Record)

1) NUMEL
2) NUMNP
3) NUMCP
4) NUMBC
5) NLØAD
6) NID
7) NMAT
8) NDP
9) MAXBW
10) MAXST
11) NNDLIM
12) MAXBKT
13) NLAYER
14) LWRAT
15) NANGIN
16) NXINT
17) NCYCF
18) RECCYL
19) CØNVER
20) IPRIN
21) ITER
22) IRESI
23) SAVETP
24) BLANAL
25) FLIFE

26) PRESS
27) HEADER(1)
28) HEADER(2)
29) HEADER(3)
30) HEADER(4)
31) HEADER(5)
32) HEADER(6)
33) HEADER(7)

6.6.2 Computed Constants and Input Arrays

The second category of restart data occupies several records. The number of records occupied depends on the values of the input parameters NLØAD, NNDLIM, NDP, NMAT, NID, BLANAL and FLIFE. In addition, the second parameter on the second record is VPRFLG which is nonzero if NLØAD > 0 and if neither NDP nor PRESS is specified. Otherwise, VPRFLG is zero. The value of VPRFLG determines whether or not the array VPR is written on the restart tape.

The second category of restart data is written in subroutine WRST2 and is read in subroutine RRST23.

The format of the second category of the restart data is as follows:

Restart Tape (Second Record)

1) NEQ (Input into INEQ)
2) VPRFLG
3) DPRAT
4) XLNGTH
5) BLTHNS
6) NPTSY - 2* NLAYER + 1
7) LSSAVE (1, 1)
8) LSSAVE (2, 1)
9) LSSAVE (1, 2)
. " .
. " .
. " .
26) LSSAVE (2, 10)

6.6.2 Computed Constants and Input Arrays (continued)

Restart Tape (Third and Subsequent Records)

1) [(NP(I, J); I = 1, 2, ..., NUMEL); J = 1, ..., 6]

2) [LØADE(I); I = 1, NLØAD] If NLØAD > 0
   [LØADS(I); I = 1, NLØAD]

3) [VPR(I); I = 1, NLØAD] If VPRFLG > 0

4) [DLIM(I, 1); 1 = 1, * NNDLIM] If NNDLIM > 0

5) [VEM(I); I = 1, NMAT]

6) [VXU(I); I = 1, NMAT]

7) [VTH(I); I = 1, NMAT]

8) [VXI(I); I = 1, NMAT]

9) [VYP(I); I = 1, NMAT]

10) [DELTAP(I, 1); I = 1, 2* NDP] If NDP > 0

11) [MATEL(I); I = 1, NUMEL] If NMAT > 1

12) [ND(I); I = 1, NID] If NID > 0
    [DSD(I); I = 1, NID]

13) [DEPTH(I); I = 1, NLAYER] If BLANAL ≠ 0

14) [MLAYER(I); I = 1, NLAYER]
    [CYCM(I); I = 1, NMAT] If FLIFE ≠ 0
    [CYCZ(I); I = 1, NMAT]

6.6.3 Load Step Dependent Quantities

The third category of restart data also occupies a variable number of records, depending on the value of input parameters and is written in subroutine PRINT and is read in subroutine RRST23. The format is as follows:

Restart Tape (First Record of Each Restart Dump)

1) IDUMP - Dump number
2) ISTEP - Load step number
3) TPRESS - Accumulated loading
4) NPRIN - Detailed print index
5) NEQBC - Number of boundary constraints
6) DPRESS - Incremental loading
7) RSFØLD - Current fold radius (single fold bladder analysis)

6.6.3 Load Step Dependent Quantities (continued)

Restart Tape (Second and Possibly Third and Fourth Records of Each Restart Dump)

1) BETA
2) ETST
3) EPST
4) EEST
5) ESIG
6) EWP
7) XØRD
8) YØRD
9) DIS
10) F
11) RESI
12) NEBC
13) DLFLAG (if NNDLIM > 0) - Third record
14) TPR (if VPRFLG ≠ 0) - Third or fourth record

6.6.4 Internal and External File Names

The internal symbolic file names for the input and output restart tapes are IRSIN and IRSØUT respectively. The corresponding external file names can be determined by means of arithmetic replacement statements or DATA statements in the main program BLADER. For example, on the CDC 6500 version of BLADER, the statements

```
IRSIN  = 1
IRSØUT = 3
```

specify that the external file names corresponding to IRSIN and IRSØUT are TAPE1 and TAPE3 respectively. On the CDC 6500 version, the desired external file names (e.g. TAPE1 and TAPE3) must also appear as parameters on the program statement in the main program. For example, the statement PROGRAM BLADER (INPUT, ØUTPUT, TAPE5 = INPUT, TAPE6 = ØUTPUT, TAPE1, TAPE3) permits Fortran logical units 1 and 3 to be used for the input and output restart tapes.

6.6.4 Internal and External File Names (continued)

When executing the BLADER program on other computing systems such as the UNIVAC 1108, the values assigned to IRSIN and IRSØUT may have to be modified to values which correspond to utility magnetic tapes. For example, in a certain UNIVAC 1108 computing system Fortran logical units 1 and 3 correspond to I/Ø devices other than magnetic tape units, but logical units 12 and 14 are two permissible magnetic tape units. Thus, we could set IRSIN = 12 and IRSØUT = 14 when executing BLADER on that computing system.

The parameters IRSIN and IRSØUT have been defined as permissible input quantities in the General Input Data allowing the file assignments to be changed during execution. Thus, for example, an output restart tape may be generated on unit 3 during the first of two stacked cases and then read in as an input restart tape during the second case provided that we set IRSIN = 3 in the General Input Data for the second case. It would be advisable to also set IRSØUT = 1 when we set IRSIN to avoid doing something foolish like writing on top of our input tape during the second case.

A user could get by with only one tape unit provided he does not wish to have distinct input and output restart tapes on the same run.

7.0 SOURCE LISTINGS

This section contains listings of the main program and all subroutines of the BLADER program except for library subroutines.

```
                 PROGRAM BLADER(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE3 )BLAD0010
                                                                                  BLAD0020
          C      *************************************************************   BLAD0030
          C      LARGE STRAIN, ELASTOPLASTIC FINITE ELEMENT ANALYSIS PROGRAM      BLAD0040
                                                                                  BLAD0050
          C      SOLVES PLANE STRESS OR AXISYMMETRIC STRESS PROBLEMS              BLAD0060
          C      USING LST-P3 ELEMENT (SIX NODAL POINT TRIANGLE)                  BLAD0070
          C      SMALL OR FINITE DISPLACEMENT ANALYSIS                            BLAD0080
                                                                                  BLAD0090
          C                                                                       BLAD0100
          C      *************************************************************   BLAD0110
                                                                                  BLAD0120
                                                                                  BLAD0130
          C      ***** THE INDICATED DIMENSION OF THE ARRAY BUCKET IS A DUMMY.    BLAD0140
          C            THE ACTUAL DIMENSION NBUCKT IS CALCULATED BY SUBROUTINE BLOCBLAD0150
          C            AND THE REQUIRED STORAGE IS ALLOCATED BY SUBROUTINE MRELSE. BLAD0160
                                                                                  BLAD0170
000003           COMMON  BUCKET(3000)                                             BLAD0180
000003           DIMENSION  B(1)                                                  BLAD0190
000003           EQUIVALENCE  ( B,BUCKET )                                        BLAD0200
                                                                                  BLAD0210
000003           COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6 BLAD0220
                1   ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                  BLAD0230
000003           COMMON  /CMESH /  BLTHNS,XINTN ,XYRAT ,DPRAT                     BLAD0240
000003           COMMON  /FFLAGS/  NONTRY,BLANK                                   BLAD0250
000003           COMMON  /HEADER/  HEADER(10)                                     BLAD0260
000003           COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID  BLAD0270
                1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINBLAD0280
                2   ,NXINT ,NCYCF                                                 BLAD0290
000003           COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI BLAD0300
                1   ,SAVETP,BLANAL,FLIFE                                          BLAD0310
000003           COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,MSTEP ,IMAT  ,LMAT  BLAD0320
                1   ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN BLAD0330
                2   ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                    BLAD0340
000003           COMMON  /IOUNIT/  IRSIN,IRSOUT                                   BLAD0350
000003           COMMON  /IPERM/  IPERM(3)                                        BLAD0360
          C      ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE BLAD0370
```

```
              C           ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR   BLAD0380
              C           IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE    BLAD0390
              C           POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. *****  BLAD0400
 000003             COMMON  /LBUCKT/  LNP    ,LLOADE,LLOADS,LNEBC ,LND    ,LMATEL,LBETA BLAD0410
                   1   ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP  ,LXCRD ,LYORD ,LDWP   BLAD0420
                   2   ,LWP   ,LCOUNT,LDF   ,LDR   ,LDIS  ,LF    ,LRESI ,LTST  ,LPST   BLAD0430
                   3   ,LEST  ,LDTST ,LSIG  ,LDSD  ,LDLIM ,LDLFLG,LVEM  ,LVXU  ,LVTH   BLAD0440
                   4   ,LVXI  ,LVYP  ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYRBLAD0450
                   5   ,LDEPTH,LCYCF ,LA    ,LTPR  ,LDPR                                BLAD0460
                   6   ,NBUCKT                                                          BLAD0470
 000003             COMMON  /LSSAVE/  LSSAVE(2,10)                                      BLAD0480
 000003             COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATBLAD0490
                   1   ,RSTRAT,RFINAL                                                   BLAD0500
 000003             COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP     BLAD0510
                   1   ,AREA  ,TPRESS,RMAXS                                             BLAD0520
 000003             COMMON  /RSDATA/  RSDATA(36)                                        BLAD0530
 000003             COMMON  /ST    /  ST(12,12)     ,CF(12)                             BLAD0540
 000003             COMMON  /TEMP  /  TEMP(12,12)                                       BLAD0550
 000003             COMMON  /UV    /  UV(3,6,2)                                         BLAD0560
 000003             COMMON  /XYCOOR/  X1,X2,X3,Y1,Y2,Y3                                 BLAD0570
                                                                                        BLAD0580
 000003             DATA  NONTRY,BLANK / -100000,1H  /                                  BLAD0590
 000003             DATA  IRSIN,IRSOUT / 12,14 /                                        BLAD0600
 000003             DATA IPERM/2,3,1/                                                   BLAD0610
                                                                                        BLAD0620
                                                                                        BLAD0630
 000003             NCDIM= 11                                                           BLAD0640
 000004             NCIMEN= 18                                                          BLAD0650
 000005             NFLGS= 10                                                           BLAD0660
 000006             NMESH= 9                                                            BLAD0670
                                                                                        BLAD0680
              C     ***** READ GENERAL INPUT DATA *****                                 BLAD0690
 000007        10   CALL GINPUT                                                         BLAD0700
                                                                                        BLAD0710
              C     ***** CALCULATE BUCKET LOCATIONS OF ALL VARIABLY DIMENSIONED        BLAD0720
              C           ARRAYS *****                                                  BLAD0730
 000010             CALL BLOC                                                           BLAD0740
                                                                                        BLAD0750
```

```
000011           IF ( NBUCKT .GT. MAXBKT )  GO TO 270

          C      ***** REQUEST REQUIRED AMOUNT OF BUCKET STORAGE *****
000015           CALL MRELSE (BUCKET,NBUCKT)

          C      ***** READ THE REMAINDER OF THE INPUT DATA AND INITIALIZE VARIOUS
          C            QUANTITIES *****
000017           CALL SETUP ( B(LNP   ),B(LXORD ),B(LYORD ),B(LLOADE),B(LLOADS)
                1  ,B(LNEBC ),B(LND   ),B(LDSD  ),B(LVEM  ),B(LVXU  ),B(LVTH  )
                2  ,B(LVXI  ),B(LVYP  ),B(LDELTP),B(LMATEL),B(LDLIM ),B(LDISPL)
                3  ,B(LBCOND),B(LA    ),B(LDISPL),B(LMLAYR),B(LDEPTH),B(LCYCF )
                4  ,B(LCYCM ),B(LCYCZ ),B(LVPR  ),B(LTPR  ),B(LDPR  )  )

          C      ***** INCREASE LENGTH OF BUCKET STORAGE TO ALLOW FOR THE ARRAY A.
000141           LA     =LDISPL
000143           NBUCKT= LA + LENA

000145      270  WRITE (6,510)  NBUCKT,NBUCKT
000155      510  FORMAT (26HOLENGTH OF DYNAMIC STORAGE,I8,10H (DECIMAL),5X,O6
                1  ,8H (OCTAL) )

000155           IF ( NBUCKT .LE. MAXBKT )  GO TO 280
000160           WRITE (6,511)  MAXBKT
000165      511  FORMAT (39H EXCEEDS MAXIMUM BUCKET SIZE.  MAXBKT = I8 )
000165           STOP

000167      280  CALL MRELSE (BUCKET,NBUCKT)

000171           NXTLS= LSTEPO - 1


          C      ***** LOAD STEP LOOP *****

000173           DO 200 ISTEP=LSTEPO,MAXST


          C      ***** ITERATION TO INSURE EQUILIBRIUM *****
```

```
000174           IFIX=0                                                    BLAD1110
000175           DO 310 I=1,ITER                                           BLAD1150
                                                                           BLAD1160
           C     ***** APPLY AN INCREMENT OF LOAD TO THE STRUCTURE AND CALCULATE   BLAD1170
           C           THE RESULTING INCREMENTAL DISPLACEMENT *****        BLAD1180
000177      303  CALL STEP2 ( B(LA    ),B(LNP   ),B(LDELTP),B(LDF   ),B(LDR   )   BLAD1190
               1  ,B(LF    ),B(LRESI ),B(LNEBC ),B(LND   ),B(LDSD  )       BLAD1200
               2  ,B(LVEM  ),B(LVXU  ),B(LVTH  ),B(LMATEL),B(LDEPTH),B(LXORD )   BLAD1210
               3  ,B(LYORD ),B(LTPR  ),B(LDPR  ),NUMEL ,NEQ  )             BLAD1220
                                                                           BLAD1230
           C     ***** CALCULATE THE INCREMENTAL AND TOTAL STRESSES AND STRAINS ***BLAD1240
000267           CALL SETIC ( B(LNP   ),B(LBETA ),B(LESIG ),          B(LTST  )   BLAD1250
               1  ,B(LPST  ),B(LEST  ),B(LDTST ),B(LSIG  ),B(LETST ),B(LEPST )   BLAD1260
               2  ,B(LEEST ),B(LEWP  ),B(LXORD ),B(LYORD ),B(LDWP  ),B(LWP   )   BLAD1270
               3  ,B(LCOUNT),B(LDR   ),B(LDIS  ),B(LF    ),B(LVEM  )       BLAD1280
               4  ,B(LVXU  ),B(LVXI  ),B(LVYP  ),B(LMATEL),B(LTHETA),B(LNEBC )   BLAD1290
               5  ,NUMEL ,NUMNP )                                          BLAD1300
                                                                           BLAD1310
           C     ***** CALCULATE THE TOTAL ERROR IN NODAL POINT EQUILIBRIUM BY     BLAD1320
           C           CALCULATING THE RESIDUAL NODAL POINT FORCES *****  BLAD1330
000413           CALL RESID ( B(LNP   ),B(LESIG ),B(LXORD ),B(LYORD ),B(LDF   )   BLAD1340
               1  ,B(LF    ),B(LNEBC ),B(LND   ),B(LMATEL),B(LVTH  ),NUMEL  )   BLAD1350
                                                                           BLAD1360
000442           IF (IFIX .NE. 0)  GO TO 305                               BLAD1370
                                                                           BLAD1380
           C     ***** IFIX .EQ. 0. FIRST PASS FOR A GIVEN LOAD STEP. *****  BLAD1390
000443           IFIX=1                                                    BLAD1400
                                                                           BLAD1410
           C     ***** CALCULATE Y, THE NORM OF THE TOTAL FORCE VECTOR (F) *****   BLAD1420
000444           Y=0.                                                      BLAD1430
000445           DO 290 N=1,NEQ                                            BLAD1440
000447           F= B(LF+N-1)                                              BLAD1450
000452      290  Y= Y + F**2                                               BLAD1460
                                                                           BLAD1470
000456     295   Y=SQRT(Y)                                                 BLAD1480
000461           WRITE(6,504)  Y                                           BLAD1490
000466     504   FORMAT(1H0, 5X,13HFORCE NORM = ,E12.5//11X,7HDF NORM,9X   BLAD1500
               1  ,15HDF/F NORM RATIO)                                     BLAD1510
```

```
                                                                          BLAD1520
         C     ***** CALCULATE X, THE NORM OF THE ERROR VECTOR DF *****   BLAD1530
000466     305  X=0.                                                      BLAD1540
000467          DC 300 N=1,NEQ                                            BLAD1550
000471          DF= B(LDF+N-1)                                            BLAD1560
000474     300  X= X + DF**2                                              BLAD1570
                                                                          BLAD1580
000500    296   X=SQRT(X)                                                 BLAD1590
000503          CON=X/Y                                                   BLAD1600
000505          WRITE(6,505)  X,CON                                       BLAD1610
000514     505  FORMAT (2E20.5)                                           BLAD1620
                                                                          BLAD1630
         C      ***** TEST FOR CONVERGENCE                                BLAD1640
000514          IF (CON .LE. CONVER)  GO TO 30                            BLAD1650
                                                                          BLAD1660
000517     310  CONTINUE                                                  BLAD1670
                                                                          BLAD1680
000521          WRITE (6,520)                                             BLAD1690
000525     520  FORMAT (29H1SOLUTION FAILED TO CONVERGE. )                BLAD1700
000525          CALL EERRXX                                               BLAD1710
                                                                          BLAD1720
000526     30   CALL PRINT ( B(LXORD ),B(LYORD ),B(LDWP  ),B(LWP   ),B(LTST  )  BLAD1730
               1  ,B(LPST  ),B(LEST  ),B(LSIG  ),B(LDIS  ),B(LF    ),B(LDF   )  BLAD1740
               2  ,B(LRESI ),B(LNP   ),B(LMLAYR),B(LCYCM ),B(LCYCZ )       BLAD1750
               3  ,NUMNP ,NUMEL  )                                        BLAD1760
                                                                          BLAD1770
000603    200   CONTINUE                                                  BLAD1780
                                                                          BLAD1790
000606          IF ( SAVETP .EQ. 0.0 )  GO TO 10                          BLAD1800
                                                                          BLAD1810
000607          END FILE IRSOUT                                           BLAD1820
000611          REWIND IRSOUT                                             BLAD1830
000613          GO TO 10                                                  BLAD1840
000614          END                                                       BLAD1850
```

```
RUN 2.3I  05/28/70           BLADER



PROGRAM LENGTH INCLUDING I/O BUFFERS
011050

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10       -  000007    30       -  000526    270      -  000145    280      -  000167
295      -  000456    296      -  000500    303      -  000177    305      -  000466
504      -  000651    505      -  000663    510      -  000631    511      -  000642
520      -  000665

BLOCK NAMES AND LENGTHS
         -  005670    CDIMEN   -  000013    CMESH    -  000004    FFLAGS   -  000002
HEADER   -  000012    IDIMEN   -  000022    IFLAGS   -  000012    INDEX    -  000026
IOUNIT   -  000002    IPERM    -  000003    LBUCKT   -  000061    LSSAVE   -  000024
MESHIN   -  000011    PROPS    -  000012    RSDATA   -  000044    ST       -  000234
TEMP     -  000220    UV       -  000044    XYCOOR   -  000006

VARIABLE ASSIGNMENTS
B        -  000000C01 BLANK    -  000001C04 BUCKET   -  000000C01 CF       -  000220C20
CON      -  000737    CONVER   -  000003C07 DF       -  000736    F        -  000734
HEADER   -  000000C05 I        -  000731    IFIX     -  000000C10 IPERM    -  000000C12
IRSIN    -  000000C11 IRSOUT   -  000001C11 ISTEP    -  000002C10 ITER     -  000005C07
LA       -  000055C13 LBCOND   -  000047C13 LBETA    -  000006C13 LCOUNT   -  000021C13
LCYCF    -  000054C13 LCYCM    -  000050C13 LCYCZ    -  000051C13 LDELTP   -  000044C13
LDEPTH   -  000053C13 LDF      -  000022C13 LDIS     -  000024C13 LDISPL   -  000046C13
LDLIM    -  000035C13 LDPR     -  000057C13 LDR      -  000023C13 LDSD     -  000034C13
LDTST    -  000032C13 LDWP     -  000017C13 LEEST    -  000012C13 LENA     -  000005C02
LEPST    -  000011C13 LESIG    -  000013C13 LEST     -  000031C13 LETST    -  000010C13
LEWP     -  000014C13 LF       -  000025C13 LLOADE   -  000001C13 LLOADS   -  000002C13
LMATEL   -  000005C13 LMLAYR   -  000052C13 LND      -  000004C13 LNEBC    -  000003C13
LNP      -  000000C13 LPST     -  000030C13 LRESI    -  000026C13 LSIG     -  000033C13
LSSAVE   -  000000C14 LSTEPO   -  000016C10 LTHETA   -  000045C13 LTPR     -  000056C13
LTST     -  000027C13 LVEM     -  000037C13 LVPR     -  000007C13 LVTH     -  000041C13
LVXI     -  000042C13 LVXU     -  000040C13 LVYP     -  000043C13 LWP      -  000020C13
LXORD    -  000015C13 LYORD    -  000016C13 MAXBKT   -  000014C06 MAXST    -  000012C06
```

```
RUN 2.3I  05/28/70            BLADER

N       -  000733     NBUCKT -  000060C13 NCDIM   -  000000C02 NDIMEN -  000000C06
NEQ     -  000002C02 NFLGS   -  000000C07 NMESH   -  000000C15 NONTRY -  000000C04
NUMEL   -  000001C06 NUMNP   -  000002C06 NXTLS   -  000020C10 RSDATA -  000000C17
SAVETP  -  000007C07 ST      -  000000C20 TEMP    -  000000C21 UV     -  000000C22
X       -  000735     Y      -  000732

START OF CONSTANTS
000616

START OF TEMPORARIES
000672

START OF INDIRECTS
000676

EXTERNAL REFERENCES
         Q8NTRY    GINPUT    BLOC      MRELSE    SETUP     OUTPTC    STOP      STEP2
         SETIC     RESID     SQRT      EERRXX    PRINT     ENDFIL    REWINM    END

COMPILER SPACE
UNUSED -  010600    USED    -  037200
```

```
                 SUBROUTINE BLOC                                                 BLOC0010
                                                                                 BLOC0020
           C           CALCULATES BUCKET LOCATIONS OF ALL VARIABLY DIMENSIONED   BLOC0030
           C           ARRAYS                                                    BLOC0040
                                                                                 BLOC0050
000002           COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6  BLOC0060
                1   ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                 BLOC0070
000002           COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   BLOC0080
                1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINBLOC0090
                2   ,NXINT ,NCYCF                                                BLOC0100
000002           COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI BLOC0110
                1   ,SAVETP,BLANAL,FLIFE                                         BLOC0120
000002           COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT  BLOC0130
                1   ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN BLOC0140
                2   ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                   BLOC0150
           C     ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE BLOC0160
           C           ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR  BLOC0170
           C           IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE   BLOC0180
           C           POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. ***** BLOC0190
000002           COMMON  /LBUCKT/  LNP   ,LLOADE,LLOADS,LNEBC ,LND   ,LMATEL,LBETA BLOC0200
                1   ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP  ,LXORD ,LYORD ,LDWP  BLOC0210
                2   ,LWP   ,LCOUNT,LDF   ,LDR   ,LDIS  ,LF    ,LRESI ,LTST  ,LPST  BLOC0220
                3   ,LEST  ,LDTST ,LSIG  ,LDSD  ,LDLIM ,LDLFLG,LVEM  ,LVXU  ,LVTH  BLOC0230
                4   ,LVXI  ,LVYP  ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYRBLOC0240
                5   ,LDEPTH,LCYCF ,LA    ,LTPR  ,LDPR                            BLOC0250
                6   ,NBUCKT                                                      BLOC0260
                                                                                 BLOC0270
000002           NNLOAD= MAXO(NLOAD,0)                                           BLOC0280
000005           NEQ=2*NUMNP                                                     BLOC0290
000006           NNP4= 4*NUMNP                                                   BLOC0300
000010           NEL6= 6*NUMEL                                                   BLOC0310
                                                                                 BLOC0320
000012           IF ( RECCYL .NE. 0.0 )  GO TO 110                               BLOC0330
                                                                                 BLOC0340
           C     ***** RECTANGULAR COORDINATES *****                             BLOC0350
000013           NNP34= 3*NUMNP                                                  BLOC0360
000014           NNP45= NNP4                                                     BLOC0370
```

```
000016          NEL64= 3*NEL6                                          BLOC0380
000020          GO TO 120                                              BLOC0390
                                                                       BLOC0400
          C     ***** CYLINDRICAL COORDINATES *****                    BLOC0410
000020     110  NNP34= NNP4                                            BLOC0420
000021          NNP45= 5*NUMNP                                         BLOC0430
000023          NEL64= 4*NEL6                                          BLOC0440
                                                                       BLOC0450
000025     120  LNP   = 1                                              BLOC0460
000026          LLOADE= LNP    + NEL6                                  BLOC0470
000030          LLOADS= LLOADE + NNLOAD                                BLOC0480
000031          LNEBC = LLOADS + NNLOAD                                BLOC0490
000033          LND   = LNEBC  + 2*NUMBC + NNDLIM                      BLOC0500
000036          LMATEL= LND    + NID                                   BLOC0510
000037          LBETA = LMATEL                                         BLOC0520
000041          IF (NMAT .GT. 1)  LBETA = LMATEL + NUMEL               BLOC0530
000046          LVPR  = LBETA  + NEL6                                  BLOC0540
000050          LTPR  = LVPR   + NNLOAD                                BLOC0550
000051          LDPR  = LTPR   + NNLOAD                                BLOC0560
                                                                       BLOC0570
000053          LETST= LVPR                                            BLOC0580
000054          IF ( VPRFLG .NE. 0.0 )  LETST = LDPR + NNLOAD          BLOC0590
000060          LEPST = LETST  + NEL64                                 BLOC0600
000062          LEEST = LEPST  + NEL64                                 BLOC0610
000063          LESIG = LEEST  + NEL64                                 BLOC0620
000065          LEWP  = LESIG  + 5*NEL6                                BLOC0630
000067          LXORD = LEWP   + NEL6                                  BLOC0640
000071          LYORD = LXORD  + NUMNP                                 BLOC0650
000072          LDWP  = LYORD  + NUMNP                                 BLOC0660
000074          LWP   = LDWP   + NUMNP                                 BLOC0670
000075          LCOUNT= LWP    + NUMNP                                 BLOC0680
000077          LDF   = LCOUNT + NUMNP                                 BLOC0690
000100          LDR   = LDF    + NEQ                                   BLOC0700
000102          LDIS  = LDR    + NEQ                                   BLOC0710
000103          LF    = LDIS   + NEQ                                   BLOC0720
000105          LRESI = LF     + NEQ                                   BLOC0730
000106          LTST  = LRESI  + NEQ                                   BLOC0740
000110          LPST  = LTST   + NNP34                                 BLOC0750
```

```
000111          LEST  = LPST   + NNP34
000113          LDTST = LEST   + NNP34
000114          LSIG  = LDTST  + NNP45
000116          LDSD  = LSIG   + 5*NUMNP
000120          LDLIM = LDSD   + NID
000122          LDLFLG= LDLIM  + 3*NNDLIM
000124          LVEM  = LDLFLG + NNDLIM
000126          LVXU  = LVEM   + NMAT
000127          LVTH  = LVXU   + NMAT
000131          LVXI  = LVTH   + NMAT
000132          LVYP  = LVXI   + NMAT
000134          LDELTP= LVYP   + NMAT
000135          LTHETA= LDELTP + 2*NDP
000137          LDISPL= LTHETA
000141          IF (NLOAD .LT. 0)  LDISPL= LDISPL + 3*NUMEL

000145          IF ( BLANAL .NE. 0.0 )  GO TO 140

000146          LBCOND= LDISPL + 3*NID
000150          LA    = LBCOND + 2*NUMBC
000153          NN= MAXO(NEL6,3*NUMCP)
000157          GO TO 160

           C    ***** BLADDER ANALYSIS OPTION.
000160     140  IF ( FLIFE .EQ. 0 )  GO TO 150

000161          LCYCM= LDISPL
000162          LCYCZ= LCYCM + NMAT
000164          LDISPL= LCYCZ + NMAT

000166     150  LMLAYR= LDISPL
000167          LDEPTH= LMLAYR + NLAYER
000171          LA    = LDEPTH + NLAYER
000172          LCYCF= LA
000174          LDISPL= LA
000175          IF ( NCYCF .NE. 0 )  NN= MAXO(NEL6,40*NMAT)

000203     160  NBUCKT= LA     + NN
```

```
                                                                          BLOC1140
000205        RETURN                                                      BLOC1150
000206        END                                                         BLOC1160
```

```
RUN 2.3I  05/28/70          BLOC


SUBPROGRAM LENGTH
000223

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
110    -  000020    120    -  000025    140    -  000160    150    -  000166
160    -  000203

BLOCK NAMES AND LENGTHS
CDIMEN -  000013    IDIMEN -  000022    IFLAGS -  000012    INDEX  -  000026
LBUCKT -  000061

VARIABLE ASSIGNMENTS
BLANAL -  000010C03 FLIFE  -  000011C03 LA     -  000055C05 LBCOND -  000047C05
LBETA  -  000006C05 LCOUNT -  000021C05 LCYCF  -  000054C05 LCYCM  -  000050C05
LCYCZ  -  000051C05 LDELTP -  000044C05 LDEPTH -  000053C05 LDF    -  000022C05
LDIS   -  000024C05 LDISPL -  000046C05 LDLFLG -  000036C05 LDLIM  -  000035C05
LDPR   -  000057C05 LDR    -  000023C05 LDSD   -  000034C05 LDTST  -  000032C05
LDWP   -  000017C05 LEEST  -  000012C05 LEPST  -  000011C05 LESIG  -  000013C05
LEST   -  000031C05 LETST  -  000010C05 LEWP   -  000014C05 LF     -  000025C05
LLOADE -  000001C05 LLOADS -  000002C05 LMATEL -  000005C05 LMLAYR -  000052C05
LND    -  000004C05 LNEBC  -  000003C05 LNP    -  000000C05 LPST   -  000030C05
LRESI  -  000026C05 LSIG   -  000033C05 LTHETA -  000045C05 LTPR   -  000056C05
LTST   -  000027C05 LVEM   -  000037C05 LVPR   -  000007C05 LVTH   -  000041C05
LVXI   -  000042C05 LVXU   -  000040C05 LVYP   -  000043C05 LWP    -  000020C05
LXORD  -  000015C05 LYORD  -  000016C05 NBUCKT -  000060C05 NCYCF  -  000021C02
NDP    -  000010C02 NEL6   -  000006C01 NEL64  -  000007C01 NEQ    -  000002C01
NID    -  000006C02 NLAYER -  000015C02 NLOAD  -  000005C02 NMAT   -  000007C02
NN     -  000222    NNDLIM -  000013C02 NNLOAD -  000216    NNP34  -  000220
NNP4   -  000217    NNP45  -  000221    NUMBC  -  000004C02 NUMCP  -  000003C02
NUMEL  -  000001C02 NUMNP  -  000002C02 RECCYL -  000002C03 VPRFLG -  000025C04

START OF CONSTANTS
000210
```

```
RUN 2.3I  05/28/70          BLOC

START OF TEMPORARIES
000211

START OF INDIRECTS
000216

EXTERNAL REFERENCES
         END

COMPILER SPACE
UNUSED -  012200    USED   -  035600
```

```
                    SUBROUTINE DIMCHK                                                    DIMC0010
                                                                                         DIMC0020
              C           CHECKS PROBLEM DIMENSIONS SPECIFICATION FOR OBVIOUS ERRORS. DIMC0030
                                                                                         DIMC0040
000002              COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   DIMC0050
                   1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINDIMC0060
                   2    ,NXINT ,NCYCF                                                    DIMC0070
000002              COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER   ,IRESI DIMC0080
                   1    ,SAVETP,BLANAL,FLIFE                                             DIMC0090
000002              COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT   DIMC0100
                   1    ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN DIMC0110
                   2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                      DIMC0120
000002              COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATDIMC0130
                   1    ,RSTRAT,RFINAL                                                   DIMC0140
000002              COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP     DIMC0150
                   1    ,AREA  ,TPRESS,RMAXS                                             DIMC0160
                                                                                         DIMC0170
000002              DATA  PLANE / 0.0 /                                                  DIMC0180
                                                                                         DIMC0190
000002         500  FORMAT (29H1ERROR IN PROBLEM DIMENSIONS. )                           DIMC0200
000002         501  FORMAT (55H0***** A PLANE STRAIN CASE MAY NOT FOLLOW AN AXISYMMETRDIMC0210
                   1   8HIC CASE.  )                                                     DIMC0220
000002         516  FORMAT(16H NO. OF ELEMENTS                 ,I24/                     DIMC0230
                   2 35H NO. OF CORNER POINTS                  ,I5,/,                    DIMC0240
                   3 35H NO. OF NODAL POINTS                   ,I5,/,                    DIMC0250
                   4 35H NO. OF BOUNDARY CONDITIONS            ,I5,/,                    DIMC0260
                   5 35H NO. OF ELEMENTS LOADED                ,I5,/                     DIMC0270
                   6 35H NO. OF IMPOSED DISPLACEMENTS          ,I5,/                     DIMC0280
                   6 35H NUMBER OF MATERIALS                   ,I5,/,                    DIMC0290
                   7 35H NUMBER OF LOAD INCREMENT ENTRIES      ,I5,/,                    DIMC0300
                   7 35H MAXIMUM NUMBER OF STEPS               ,I5,/,                    DIMC0310
                   8 19H MAXIMUM BAND WIDTH                     I21 /                    DIMC0320
                   9 21H NO. OF DISPL. LIMITS                   I19 / )                  DIMC0330
000002         517  FORMAT (23H MAX. NO. OF ITERATIONS         ,I17 /                    DIMC0340
                   1 32H NO. OF STEPS PER DETAILED PRINT       ,I8  /                    DIMC0350
                   2 11H IRESI FLAG                            ,I29 /                    DIMC0360
                   3 22H CONVERGENCE CRITERION                 ,E23.5 /                  DIMC0370
```

```
                4 23H TOTAL PRESSURE LOADING            ,E22.5 / 1H1 )          DIMC0380
000002    518  FORMAT (23HO PLANE STRAIN ANALYSIS // )                          DIMC0390
000002    519  FORMAT (30HO AXISYMMETRIC STRAIN ANALYSIS // )                   DIMC0400
                                                                                DIMC0410
                                                                                DIMC0420
000002         IF ( RECCYL .NE. 0.0 )  GO TO 110                                DIMC0430
                                                                                DIMC0440
         C     ***** RECTANGULAR COORDINATES *****                              DIMC0450
000003         WRITE (6,518)                                                    DIMC0460
                                                                                DIMC0470
000007         IF (PLANE .EQ. 0.0)  GO TO 105                                   DIMC0480
                                                                                DIMC0490
         C     ***** ERROR. A PLANE STRAIN CASE MAY NOT FOLLOW AN AXISYMMETRIC  DIMC0500
         C           CASE BECAUSE DATA STATEMENTS IN SUBROUTINES  GEOM AND RESID DIMC0510
         C           SET VARIOUS COEFFICIENTS FOR THE PLANE STRAIN CASE *****   DIMC0520
000010         WRITE (6,501)                                                    DIMC0530
000014         STOP                                                             DIMC0540
                                                                                DIMC0550
000016    105  J34= 3                                                           DIMC0560
000017         J43= 4                                                           DIMC0570
000020         J45= 4                                                           DIMC0580
000021         I63= 6                                                           DIMC0590
000022         I69= 6                                                           DIMC0600
000023         I912= 9                                                          DIMC0610
000024         GO TO 140                                                        DIMC0620
                                                                                DIMC0630
         C     ***** CYLINDRICAL COORDINATES *****                              DIMC0640
000025    110  WRITE (6,519)                                                    DIMC0650
000031         PLANE= 1.0                                                       DIMC0660
000032         J34= 4                                                           DIMC0670
000033         J43= 3                                                           DIMC0680
000034         J45= 5                                                           DIMC0690
000035         I63= 3                                                           DIMC0700
000036         I69= 9                                                           DIMC0710
000037         I912= 12                                                         DIMC0720
                                                                                DIMC0730
                                                                                DIMC0740
000041    140  WRITE (6     ,516)  NUMEL,NUMCP,NUMNP,NUMBC,NLOAD,NID,NMAT,NDP   DIMC0750
```

```
               1   ,MAXST,MAXBW,NNDLIM
000073          WRITE (6,517)  ITER,IPRIN,IRESI,CONVER,PRESS

000111          IF (NUMEL .LE. 0)  GO TO 200
000113          IF (NUMNP .LE. 0)  GO TO 200
000114          IF ( (NUMCP .LE. 0) .OR. (NUMCP .GE. NUMNP) )  GO TO 200
000124          IF (NUMBC .LE. 0)  GO TO 200
000125          IF (NID   .LT. 0)  GO TO 200
000127          IF ( (NLOAD .NE. 0) .AND. (NID .NE. C) )  GO TO 200
000135          IF ( (NLOAD .EQ. 0) .AND. (NID .EQ. C) )  GO TO 200
000143          IF ( ( RECCYL .EQ. 0.0 ) .AND. ( NLOAD .LT. 0 ) )  GO TO 200
000151          IF (NMAT  .LE. 0)  GO TO 200
000152          IF (NDP   .LT. 0)  GO TO 200
000154          IF (MAXBW .LE. 0)  GO TO 200
000155          IF (MAXST .LE. 0)  GO TO 200
000157          IF ( (NNDLIM .LT. 0) .OR. (NNDLIM .GT. NUMNP) )  GO TO 200
000166          IF ( RESTRT .NE. 0.0 )  GO TO 900
000167          IF ( BLANAL .EQ. 0.0 )  GO TO 900

000170          IF ( NLAYER .LE. 0 )  GO TO 200
000171          IF ( NXINT .LE. 0 )  GO TO 200
000173          IF ( NCYCF .GT. 20 )  GO TO 200
000176          IF ( RFINAL .LE. 0.0 )  GO TO 200

         C      ***** NO APPARENT ERRORS *****
000177          GO TO 900

000200     200  WRITE (6,500)
000204          ERRFLG= 0.0

000205     900  RETURN
000206          END
```

```
RUN 2.3I  05/28/70          DIMCHK


SUBPROGRAM LENGTH
000445

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
105    -  000016    110    -  000025    140    -  000041    200    -  000200
500    -  000211    501    -  000216    516    -  000227    517    -  000337
518    -  000400    519    -  000405    900    -  000205

BLOCK NAMES AND LENGTHS
IDIMEN -  000022    IFLAGS -  000012    INDEX  -  000026    MESHIN -  000011
PROPS  -  000012

VARIABLE ASSIGNMENTS
BLANAL -  000010C02 CONVER -  000003C02 ERRFLG -  000007C03 IPRIN  -  000004C02
IRESI  -  000006C02 ITER   -  000005C02 I63    -  000013C03 I69    -  000014C03
I912   -  000015C03 J34    -  000010C03 J43    -  000011C03 J45    -  000012C03
MAXBW  -  000011C01 MAXST  -  000012C01 NCYCF  -  000021C01 NDP    -  000010C01
NID    -  000006C01 NLAYER -  000015C01 NLOAD  -  000005C01 NMAT   -  000007C01
NNDLIM -  000013C01 NUMBC  -  000004C01 NUMCP  -  000003C01 NUMEL  -  000001C01
NUMNP  -  000002C01 NXINT  -  000020C01 PLANE  -  000444    PRESS  -  000000C05
RECCYL -  000002C02 RESTRT -  000001C02 RFINAL -  000010C04

START OF CONSTANTS
000210

START OF TEMPORARIES
000416

START OF INDIRECTS
000444

EXTERNAL REFERENCES
         OUTPTC    STOP      END
```

```
RUN 2.3I  05/28/70          DIMCHK

COMPILER SPACE
UNUSED -  012100    USED   -  035700
```

```
                SUBROUTINE DISCHK (DIS,NEBC,DLIM,DLFLAG)                      DISC0010
        C             TEST FOR NODAL DISPLACEMENTS REACHING LIMITING DISPLACEMENT.DISC0020
                                                                              DISC0030
        C       ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.DISC0040
000007          DIMENSION  DIS(1),NEBC(1),DLIM(3,1),DLFLAG(1)                 DISC0050
                                                                              DISC0060
000007          COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ    ,NEQBC ,IBAND ,LENA   ,NEL6  DISC0070
               1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                              DISC0080
000007          COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   DISC0090
               1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINDISC0100
               2    ,NXINT ,NCYCF                                             DISC0110
                                                                              DISC0120
                                                                              DISC0130
000007          DO 200 I=1,NNDLIM                                             DISC0140
                                                                              DISC0150
                                                                              DISC0170
000010          IF ( DLFLAG(I) .NE. 0.0 )  GO TO 200                          DISC0160
000012          NODE= DLIM(1,I)                                               DISC0180
000015          IDIR= DLIM(2,I)                                               DISC0190
000021          DLIMIT= DLIM(3,I)                                             DISC0200
000023          II= 2*NODE + IDIR - 2                                         DISC0210
000026          DISPL= DIS(II)                                                DISC0220
000030          IF ( DLIMIT .GT. 0.0 )  GO TO 110                             DISC0230
                                                                              DISC0240
        C       ***** DLIMIT .LE. 0.0 *****                                   DISC0250
000033          IF ( DISPL .LE. DLIMIT )  GO TO 120                           DISC0260
000035          GO TO 200                                                     DISC0270
                                                                              DISC0280
        C       ***** DLIMIT .GT. 0.0 *****                                   DISC0290
000036     110  IF ( DISPL .LT. DLIMIT )  GO TO 20C                           DISC0300
                                                                              DISC0310
        C       ***** LIMITING DISPLACEMENT EXCEEDED *****                    DISC0320
000041     120  DLFLAG(I)= 1.0                                                DISC0330
000044          DIS(II)= DLIMIT                                               DISC0340
000046          NEQBC= NEQBC + 1                                              DISC0350
000050          NEBC(NEQBC)= II                                               DISC0360
000052     200  CONTINUE                                                      DISC0370
```

```
                                                                 DISC0380
                                                                 DISC0390
000055          RETURN                                           DISC0400
000055          END
```

```
RUN 2.3I  05/28/70          DISCHK



SUBPROGRAM LENGTH
000076

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
110     -  000036    120     -  000041    200     -  CC0052

BLOCK NAMES AND LENGTHS
CDIMEN -  000013    IDIMEN -  000022

VARIABLE ASSIGNMENTS
DISPL   -  000075    CLIMIT -  000073    I       -  CC0070    IDIR    -  000072
II      -  000074    NEQBC  -  000003C01 NNDLIM -  000013C02 NODE    -  000071

START OF CONSTANTS
000057

START OF TEMPORARIES
000061

START OF INDIRECTS
000063

EXTERNAL REFERENCES
         END

COMPILER SPACE
UNUSED -  013200    USED   -  034600
```

```
               SUBROUTINE EERRXX                                                 EERR0010
                                                                                 EERR0020
         C     ***** THE INDICATED DIMENSION OF THE ARRAY B IS A DUMMY.          EERR0030
         C           THE ACTUAL DIMENSION NBUCKT IS CALCULATED BY SUBROUTINE BLOCEERR0040
         C           AND THE REQUIRED STORAGE IS ALLOCATED BY SUBROUTINE MRELSE. EERR0050
                                                                                 EERR0060
000002         COMMON  B(1)                                                      EERR0070
                                                                                 EERR0080
000002         COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   EERR0090
              1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINEERR0100
              2   ,NXINT ,NCYCF                                                  EERR0110
000002         COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI EERR0120
              1   ,SAVETP,BLANAL,FLIFE                                           EERR0130
000002         COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT  EERR0140
              1   ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN EERR0150
              2   ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                     EERR0160
000002         COMMON  /IOUNIT/  IRSIN,IRSOUT                                    EERR0170
         C     ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE EERR0180
         C           ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR  EERR0190
         C           IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE   EERR0200
         C           POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. ***** EERR0210
000002         COMMON  /LBUCKT/  LNP   ,LLOADE,LLOADS,LNEBC ,LND   ,LMATEL,LBETA EERR0220
              1   ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP  ,LXCRD ,LYORD ,LDWP  EERR0230
              2   ,LWP   ,LCOUNT,LDF   ,LDR   ,LDIS  ,LF    ,LRESI ,LTST  ,LPST  EERR0240
              3   ,LEST  ,LDTST ,LSIG  ,LDSD  ,LDLIM ,LDLFLG,LVEM  ,LVXU  ,LVTH  EERR0250
              4   ,LVXI  ,LVYP  ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYREERR0260
              5   ,LDEPTH,LCYCF ,LA    ,LTPR  ,LDPR                              EERR0270
              6   ,NBUCKT                                                        EERR0280
000002         COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP    EERR0290
              1   ,AREA  ,TPRESS,RMAXS                                           EERR0300
                                                                                 EERR0310
                                                                                 EERR0320
000002         WRITE (6,10)                                                      EERR0330
000006    10   FORMAT (13H1LBUCKT ARRAY//)                                       EERR0340
000006         CALL PDUMP ( LNP,NBUCKT,0 )                                       EERR0350
000011         CALL PDUMP ( NDIMEN,NCYCF,2,  NFLGS,FLIFE,2,  IFIX,VPRFLG,2       EERR0360
              1  ,PRESS,RMAXS,1 )                                                EERR0370
```

```
RUN 2.3I  05/28/70           EERRXX

000025         WRITE (6,11)                                                   EERR0380
000031    11   FORMAT (//22HODYNAMIC STORAGE ARRAY// )                        EERR0390
000031         CALL PDUMP ( B(LNP),B(LBETA-1),2,  B(LBETA),B(LA),1 )          EERR0400
                                                                              EERR0410
000041         ERRFLG= 0.0                                                    EERR0420
000042         CALL PRINT ( B(LXORD ),B(LYORD ),B(LDWP  ),B(LWP   ),B(LTST  ) EERR0430
              1  ,B(LPST  ),B(LEST  ),B(LSIG  ),B(LDIS  ),B(LF    ),B(LDF   ) EERR0440
              2  ,B(LRESI ),B(LNP   ),B(LMLAYR),B(LCYCM ),B(LCYCZ )           EERR0450
              3  ,NUMNP ,NUMEL  )                                             EERR0460
                                                                              EERR0470
000127         IF ( SAVETP .EQ. 0.0 )  STOP                                   EERR0480
                                                                              EERR0490
000132         END FILE IRSOUT                                                EERR0500
000134         REWIND IRSOUT                                                  EERR0510
000136         STOP                                                           EERR0520
000140          END                                                           EERR0530
```

```
SUBPROGRAM LENGTH
000204

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10     -  000144    11     -  000153

BLOCK NAMES AND LENGTHS
       -  000001    IDIMEN -  000022    IFLAGS -  0C0012    INDEX  -  000026
IOUNIT -  000002    LBUCKT -  000061    PROPS  -  0C0012

VARIABLE ASSIGNMENTS
B      -  000000C01 ERRFLG -  000007C04 FLIFE  -  0C0011C03 IFIX   -  000000C04
IRSOUT -  000001C05 LA     -  000055C06 LBETA  -  0C0006C06 LCYCM  -  000050C06
LCYCZ  -  000051C06 LDF    -  000022C06 LDIS   -  0C0024C06 LDWP   -  000017C06
LEST   -  000031C06 LF     -  000025C06 LMLAYR -  0C0052C06 LNP    -  000000C06
LPST   -  000030C06 LRESI  -  000026C06 LSIG   -  0C0033C06 LTST   -  000027C06
LWP    -  000020C06 LXORD  -  000015C06 LYORD  -  0C0016C06 NBUCKT -  000060C06
NCYCF  -  000021C02 NDIMEN -  000000C02 NFLGS  -  CC0000C03 NUMEL  -  000001C02
NUMNP  -  000002C02 PRESS  -  000000C07 RMAXS  -  C00011C07 SAVETP -  000007C03
VPRFLG -  000025C04

START OF CONSTANTS
000142

START OF TEMPORARIES
000162

START OF INDIRECTS
000I64

EXTERNAL REFERENCES
         OUTPTC    PDUMP     PRINT     STOP      ENDFIL    REWINM    END

COMPILER SPACE
```

```
RUN 2.3I  05/28/70              EERRXX
UNUSED -  012600    USED    -   035200
```

```
                 . SUBROUTINE FMCALC (NP,ESIG,RORD,ZORD,DIS,NNUMEL )                       FMCA0010
                                                                                            FMCA0020
            C      ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.FMCA0030
000011             DIMENSION  NP(NNUMEL,6),ESIG(NNUMEL,6,5)                                 FMCA0040
000011             DIMENSION  RORD(1),ZORD(1),DIS(1)                                        FMCA0050
                                                                                            FMCA0060
000011             COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   FMCA0070
                  1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINFMCA0080
                  2   ,NXINT ,NCYCF                                                         FMCA0090
000011             COMMON  /IPERM/  IPERM(3)                                                FMCA0100
000011             COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP    FMCA0110
                  1   ,AREA  ,TPRESS,RMAXS                                                  FMCA0120
000011             COMMON  /XYCOOR/ R1,R2,R3,Z1,Z2,Z3                                       FMCA0130
                                                                                            FMCA0140
000011             DIMENSION  RR(3), ZZ(3)                                                  FMCA0150
000011             EQUIVALENCE  (RR,R1) , (ZZ,Z1)                                           FMCA0160
                                                                                            FMCA0170
000011             DIMENSION  BA(3,2), SIGTH(6)                                             FMCA0180
                                                                                            FMCA0190
000011             SUMFTH= 0.0                                                              FMCA0200
000012             SUMMR= 0.0                                                               FMCA0210
000013             SUMMZ= 0.0                                                               FMCA0220
                                                                                            FMCA0230
000014             IF (NLOAD .GE. 0)  GO TO 100                                             FMCA0240
                                                                                            FMCA0250
000015             DO  50 I=1,NUMNP                                                         FMCA0260
                                                                                            FMCA0280
000017       50    RORD(I)= RORD(I) + DPRESS                                                FMCA0270
000024      100    DO 200 NV=1,NUMEL                                                        FMCA0290
                                                                                            FMCA0300
000026             DO 110 I=1,3                                                             FMCA0310
000027             K1=NP(NV,I)                                                              FMCA0320
000034             RR(I)= RORD(K1)                                                          FMCA0330
000037             IF (NLOAD .LT. 0)  RR(I)= RR(I) - RORD(-NLOAD)                           FMCA0340
000045             ZZ(I)= ZORD(K1)                                                          FMCA0350
000051             J=IPERM(I)                                                               FMCA0360
000053             K2=NP(NV,J)                                                              FMCA0370
```

```
000060          M=IPERM(J)                                                    FMCA0380
000062          BA(M,2)=RORD(K2)-RORD(K1)                                     FMCA0390
000067   110    BA(M,1)=ZORD(K1)-ZORD(K2)                                     FMCA0400
000076          AREA= 0.5*( BA(3,2)*BA(2,1)-BA(2,2)*BA(3,1) )                 FMCA0410
                                                                              FMCA0420
000102          DC 120 I=1,6                                                  FMCA0430
000104    120   SIGTH(I)= ESIG(NV,I,3)                                        FMCA0440
                                                                              FMCA0450
000123          SUMFTH= SUMFTH + AREA/3.0*( SIGTH(4) + SIGTH(5) + SIGTH(6) )  FMCA0460
                                                                              FMCA0470
000131          SUMMR= SUMMR + AREA/60.0*(  (2.0*Z1-Z2-Z3)*SIGTH(1)           FMCA0480
               1  + (2.0*Z2-Z1-Z3)*SIGTH(2) + (2.C*Z3-Z1-Z2)*SIGTH(3)         FMCA0490
               2  + 8.0*( (Z1+Z2+0.5*Z3)*SIGTH(4) + (0.5*Z1+Z2+Z3)*SIGTH(5)   FMCA0500
               3  + (Z1+0.5*Z2+Z3)*SIGTH(6) )  )                              FMCA0510
                                                                              FMCA0520
000171          SUMMZ= SUMMZ + AREA/60.0*(  (2.0*R1-R2-R3)*SIGTH(1)           FMCA0530
               1  + (2.0*R2-R1-R3)*SIGTH(2) + (2.C*R3-R1-R2)*SIGTH(3)         FMCA0540
               2  + 8.0*( (R1+R2+0.5*R3)*SIGTH(4) + (0.5*R1+R2+R3)*SIGTH(5)   FMCA0550
               3  + (R1+0.5*R2+R3)*SIGTH(6) )  )                              FMCA0560
                                                                              FMCA0570
000232    200   CONTINUE                                                      FMCA0580
                                                                              FMCA0590
000235          WRITE (6,901)  SUMFTH,SUMMR,SUMMZ                             FMCA0600
000246    901   FORMAT (///24H  TOTAL FORCE  (F THETA),E16.5/                 FMCA0610
               1  20H  TOTAL MOMENT  (MR),E20.5,2CH  TOTAL MOMENT  (MZ),E20.5  ) FMCA0620
                                                                              FMCA0630
000246          IF (NLOAD .GE. 0.0)  GO TO 300                                FMCA0640
                                                                              FMCA0650
000253          R= RORD(-NLOAD)                                               FMCA0660
000256          WRITE (6,902)  R                                              FMCA0670
000263    902   FORMAT (4H0 R= E16.5  )                                       FMCA0680
                                                                              FMCA0690
000263    300   RETURN                                                        FMCA0700
000264           END                                                          FMCA0710
```

```
RUN 2.3I  05/28/70           FMCALC



SUBPROGRAM LENGTH
000434

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
100    -  000024    120    -  0C0104    300    -  0C0263    901    -  000300
902    -  000316

BLOCK NAMES AND LENGTHS
IDIMEN -  000022    IPERM  -  000003    PROPS  -  0C0012    XYCOOR -  000006

VARIABLE ASSIGNMENTS
AREA   -  000007C03 BA     -  000406    DPRESS -  CC0001C03 I      -  000425
IPERM  -  000000C02 J      -  000430    K1     -  0C0427    K2     -  000431
M      -  000432    NLOAD  -  000005C01 NUMEL  -  0C0001C01 NUMNP  -  000002C01
NV     -  000426    R      -  000433    RR     -  CC0000C04 R1     -  000000C04
R2     -  000001C04 R3     -  000002C04 SIGTH  -  CC0414    SUMFTH -  000422
SUMMR  -  000423    SUMMZ  -  000424    ZZ     -  0C0003C04 Z1     -  000003C04
Z2     -  000004C04 Z3     -  000005C04

START OF CONSTANTS
000266

START OF TEMPORARIES
000321

START OF INDIRECTS
000366

EXTERNAL REFERENCES
         OUTPTC    END

COMPILER SPACE
UNUSED -  012100    USED   -  035700
```

```
              SUBROUTINE GEOM (NP     ,XORD   ,YORD   ,DPR    ,LOADE ,LOADS ,THETA  GEOM0010
             1    ,ESIG   ,NNUMEL   )                                               GEOM0020
                                                                                    GEOM0030
        C             CALCULATES INCREMENTAL ELEMENTAL FORCE VECTOR.                GEOM0040
                                                                                    GEOM0050
        C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.GEOM0060
000014        DIMENSION    NP(NNUMEL,6),THETA(NNUMEL,3),ESIG(NNUMEL,6,5)            GEOM0070
000014        DIMENSION    XORD  (1),YORD  (1),DPR   (1),LOADE (1),LOADS (1)        GEOM0080
                                                                                    GEOM0090
000014        COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID       GEOM0100
             1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINGEOM0110
             2    ,NXINT ,NCYCF                                                     GEOM0120
000014        COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI GEOM0130
             1    ,SAVETP,BLANAL,FLIFE                                              GEOM0140
000014        COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT  GEOM0150
             1    ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN GEOM0160
             2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                        GEOM0170
000014        COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP    GEOM0180
             1    ,AREA  ,TPRESS,RMAXS                                              GEOM0190
000014        COMMON  /ST    /  ST(12,12)   ,CF(12)                                 GEOM0200
000014        COMMON  /TEMP  /  TEMP(12,12)                                         GEOM0210
000014        COMMON  /UV    /  UV(3,6,2)                                           GEOM0220
000014        COMMON  /XYCOOR/  X1,X2,X3,Y1,Y2,Y3                                   GEOM0230
                                                                                    GEOM0240
000014        DIMENSION HG(3,3,6),AX(3),AY(3),AT(3),R(3,6)                          GEOM0250
000014        DIMENSION  ISIDE(3), DELPR(3)                                         GEOM0260
                                                                                    GEOM0270
000014        REAL JXX(3,3),JYY(3,3),JXY(3,3),JTT(3,3)                              GEOM0280
                                                                                    GEOM0290
000014        EQUIVALENCE  (R1,X1) , (R2,X2) , (R3,X3)                              GEOM0300
                                                                                    GEOM0310
        C     ***** VALUES OF HG FOR RECTANGULAR COORDINATES ARE SET BY THE         GEOM0320
        C           FOLLOWING DATA STATEMENTS *****                                 GEOM0330
000014        DATA  HG/ 6., 0., 0., 0.,-2.,-1., 0.,-1.,-2.,                         GEOM0340
             1          -2., 0.,-1., 0., 6., 0.,-1., 0.,-2.,                        GEOM0350
             2          -2.,-1., 0.,-1.,-2., 0., 0., 0., 6.,                        GEOM0360
             3          12., 8., 4., 8.,12., 4., 4., 4., 4.,                        GEOM0370
```

```
              4        4.,  4.,  4.,  4.,12.,  8.,  4.,  8.,12.,          GEOM0380
              5       12.,  4.,  8.,  4.,  4.,  4.,  8.,  4.,12. /        GEOM0390
                                                                          GEOM0400
                                                                          GEOM0410
                                                                          GEOM0420
000014          IF ( RECCYL .EQ. 0.0 )  GO TO 300                         GEOM0430
                                                                          GEOM0440
         C      ***** CYLINDRICAL COORDINATES *****                       GEOM0450
000015          HG(1,1,1)=(36.*R1+3.*R2+3.*R3)/7.                         GEOM0460
000023          HG(1,2,1)=(3.*R1-2.*R2-R3)/7.                             GEOM0470
000031          HG(1,3,1)=(3.*R1-R2-2.*R3)/7.                             GEOM0480
000037          HG(2,2,1)=(-2.*R1-9.*R2-3.*R3)/7.                         GEOM0490
000046          HG(2,3,1)=(-R1-3.*R2-3.*R3)/7.                            GEOM0500
000053          HG(3,3,1)=(-2.*R1-3.*R2-9.*R3)/7.                         GEOM0510
000062          HG(1,1,2)=(-9.*R1-2.*R2-3.*R3)/7.                         GEOM0520
000071          HG(1,2,2)=(-2.*R1+3.*R2-R3)/7.                            GEOM0530
000077          HG(1,3,2)=(-3.*R1-R2-3.*R3)/7.                            GEOM0540
000105          HG(2,2,2)=(3.*R1+36.*R2+3.*R3)/7.                         GEOM0550
000113          HG(2,3,2)=(-R1+3.*R2-2.*R3)/7.                            GEOM0560
000121          HG(3,3,2)=(-3.*R1-2.*R2-9.*R3)/7.                         GEOM0570
000130          HG(1,1,3)=(-9.*R1-3.*R2-2.*R3)/7.                         GEOM0580
000137          HG(1,2,3)=(-3.*R1-3.*R2-R3)/7.                            GEOM0590
000145          HG(1,3,3)=(-2.*R1-R2+3.*R3)/7.                            GEOM0600
000153          HG(2,2,3)=(-3.*R1-9.*R2-2.*R3)/7.                         GEOM0610
000162          HG(2,3,3)=(-R1-2.*R2+3.*R3)/7.                            GEOM0620
000170          HG(3,3,3)=(3.*R1+3.*R2+36.*R3)/7.                         GEOM0630
000176          HG(1,1,4)=(48.*R1+24.*R2+12.*R3)/7.                       GEOM0640
000205          HG(1,2,4)=(24.*R1+24.*R2+8.*R3)/7.                        GEOM0650
000213          HG(1,3,4)=(12.*R1+8.*R2+8.*R3)/7.                         GEOM0660
000221          HG(2,2,4)=(24.*R1+48.*R2+12.*R3)/7.                       GEOM0670
000230          HG(2,3,4)=(8.*R1+12.*R2+8.*R3)/7.                         GEOM0680
000236          HG(3,3,4)=(8.*R1+8.*R2+12.*R3)/7.                         GEOM0690
000244          HG(1,1,5)=(12.*R1+8.*R2+8.*R3)/7.                         GEOM0700
000252          HG(1,2,5)=(8.*R1+12.*R2+8.*R3)/7.                         GEOM0710
000260          HG(1,3,5)=(8.*R1+8.*R2+12.*R3)/7.                         GEOM0720
000266          HG(2,2,5)=(12.*R1+48.*R2+24.*R3)/7.                       GEOM0730
000275          HG(2,3,5)=(8.*R1+24.*R2+24.*R3)/7.                        GEOM0740
000303          HG(3,3,5)=(12.*R1+24.*R2+48.*R3)/7.                       GEOM0750
```

```
000312          HG(1,1,6)=(48.*R1+12.*R2+24.*R3)/7.                      GEOM0760
000321          HG(1,2,6)=(12.*R1+8.*R2+8.*R3)/7.                        GEOM0770
000327          HG(1,3,6)=(24.*R1+8.*R2+24.*R3)/7.                       GEOM0780
000335          HG(2,2,6)=(8.*R1+12.*R2+8.*R3)/7.                        GEOM0790
000343          HG(2,3,6)=(8.*R1+8.*R2+12.*R3)/7.                        GEOM0800
000351          HG(3,3,6)=(24.*R1+12.*R2+48.*R3)/7.                      GEOM0810
                                                                         GEOM0820
000360          DO 340 K=1,6                                             GEOM0830
000361          DO 340 N=2,3                                             GEOM0840
000362          NM1= N - 1                                               GEOM0850
000364          DO 340 M=1,NM1                                           GEOM0860
000366    340   HG(N,M,K)=HG(M,N,K)                                      GEOM0870
                                                                         GEOM0880
          C     ***** CALCULATE GEOMETRIC STIFFNESS MATRIX *****         GEOM0890
                                                                         GEOM0900
000410    300   DO 350 I=1,3                                             GEOM0910
000412          DO 350 J=1,3                                             GEOM0920
000413          SJXX= 0.0                                                GEOM0930
000414          SJYY= 0.0                                                GEOM0940
000415          SJXY= 0.0                                                GEOM0950
000416          SJTT= 0.0                                                GEOM0960
000417          DO 345 K=1,6                                             GEOM0970
000420          HGIJK= HG(I,J,K)                                         GEOM0980
000426          SJXX= SJXX + HGIJK*ESIG(NV,K,1)                          GEOM0990
000442          SJYY= SJYY + HGIJK*ESIG(NV,K,2)                          GEOM1000
000454          SJXY= SJXY + HGIJK*ESIG(NV,K,J34)                        GEOM1010
000466          IF ( RECCYL .NE. 0.0 )  SJTT= SJTT + HGIJK*ESIG(NV,K,3)  GEOM1020
000501    345   CONTINUE                                                 GEOM1030
                                                                         GEOM1040
000503          JXX(I,J)= SJXX                                           GEOM1050
000507          JYY(I,J)= SJYY                                           GEOM1060
000513          JXY(I,J)= SJXY                                           GEOM1070
000517          JTT(I,J)= SJTT                                           GEOM1080
000523    350   CONTINUE                                                 GEOM1090
                                                                         GEOM1100
000527          IF ( RECCYL .EQ. 0.0 )  GO TO 355                        GEOM1110
                                                                         GEOM1120
          C     ***** CYLINDRICAL COORDINATES *****                      GEOM1130
```

```
000530         DO 353 I=1,18
000532    353  R(I,1)= 0.0
000537         R(1,1)=2.*AREA/R1
000541         R(2,2)=2.*AREA/R2
000544         R(3,3)=2.*AREA/R3

000546    355  DO 400 J=1,6

000550         DO 360 I=1,3
000551         AX(I)=0.
000553         AY(I)=0.
000555         AT(I)=0.
000557         DO 360 K=1,3
000561         AX(I)=AX(I)+JXX(I,K)*UV(K,J,1)+JXY(I,K)*UV(K,J,2)
000601         AY(I)=AY(I)+JXY(I,K)*UV(K,J,1)+JYY(I,K)*UV(K,J,2)
000620         IF ( RECCYL .EQ. 0.0 )  GO TO 360

          C    ***** CYLINDRICAL COORDINATES *****
000621         AT(I)=AT(I)+JTT(I,K)*R(K,J)
000632    360  CONTINUE

000636         DO 400 I=1,6
000640         X=0.
000641         DO 380 K=1,3
000643   380   X=X+UV(K,I,1)*AX(K)+UV(K,I,2)*AY(K)

000665         Y= X

000667         IF ( RECCYL .EQ. 0.0 )  GO TO 390

          C    ***** CYLINDRICAL COORDINATES *****
000670         DO 385 K=1,3
000671    385  X= X + R(K,I)*AT(K)

000702    390  X=X*TH/(720.*AREA)
000706         Y=Y*TH/(720.*AREA)
000711         ST(2*I-1,2*J-1)=ST(2*I-1,2*J-1)+Y
000720   400   ST(2*I,2*J)=ST(2*I,2*J)+X
```

```
                                                                         GEOM1520
                                                                         GEOM1530
          C     ***** IF IFIX = 1, THEN CF IS NOT CALCULATED SINCE THE LOAD VECTORGEOM1540
          C           CONSISTS OF THE RESIDUAL NODAL FORCES *****        GEOM1550
000731          IF (IFIX-1)  401,13,401                                  GEOM1560
                                                                         GEOM1570
          C     ***** IFIX .NE. 1. APPLY INCREMENT OF LOAD TO STRUCTURE *****    GEOM1580
000733     401  IF (NLOAD) 30,5,50                                       GEOM1590
                                                                         GEOM1600
000735     50   JLOAD=0                                                  GEOM1610
000736          DO 3 I=1,NLOAD                                           GEOM1620
000740          IF (LOADE(I)-NV) 3,4,3                                   GEOM1630
000743      4   JLOAD=JLOAD+1                                            GEOM1640
000745          ISIDE(JLOAD)=LOADS(I)                                    GEOM1650
000750          IF ( VPRFLG .NE. 0.0 )  DELPR(JLOAD)= DPR(I)             GEOM1660
000755      3   CONTINUE                                                 GEOM1670
                                                                         GEOM1680
000760          IF (JLOAD .EQ. 0)  GO TO 5                               GEOM1690
                                                                         GEOM1700
000761          P1=0.                                                    GEOM1710
000762          P2=0.                                                    GEOM1720
000763          P3=0.                                                    GEOM1730
000764      6   DO 7 I=1,JLOAD                                           GEOM1740
000766          IF ( VPRFLG .NE. 0.0 )  DPRESS= DELPR(I)                 GEOM1750
000771          IF (ISIDE(I) .EQ. 1) GO TO 8                             GEOM1760
000775          IF (ISIDE(I) .EQ. -1) GO TO 15                           GEOM1770
001000          IF (ISIDE(I) .EQ. 2) GO TO 9                             GEOM1780
001003          IF (ISIDE(I) .EQ. -2) GO TO 16                           GEOM1790
001006          IF (ISIDE(I) .EQ. 3) GO TO 10                            GEOM1800
001011          IF (ISIDE(I) .EQ. -3) GO TO 17                           GEOM1810
001014     8    P1= DPRESS                                               GEOM1820
001016          GO TO 7                                                  GEOM1830
001016     15   P1= -DPRESS                                              GEOM1840
001020          GO TO 7                                                  GEOM1850
001020     9    P2= DPRESS                                               GEOM1860
001022          GO TO 7                                                  GEOM1870
001022     16   P2= -DPRESS                                              GEOM1880
001024          GO TO 7                                                  GEOM1890
```

```
001024    10   P3= DPRESS
001026         GC TO 7
001026    17   P3= -DPRESS
001030     7   CONTINUE


001033         CF(1)=TH*(P2*(Y3-Y1)+P3*(Y1-Y2))/6.
001043         CF(2)=TH*(P2*(X1-X3)+P3*(X2-X1))/6.
001053         CF(3)=TH*(P1*(Y2-Y3)+P3*(Y1-Y2))/6.
001063         CF(4)=TH*(P1*(X3-X2)+P3*(X2-X1))/6.
001073         CF(5)=TH*(P1*(Y2-Y3)+P2*(Y3-Y1))/6.
001103         CF(6)=TH*(P1*(X3-X2)+P2*(X1-X3))/6.
001113         CF(7)=2.*TH*P3*(Y1-Y2)/3.
001120         CF(8)=2.*TH*P3*(X2-X1)/3.
001125         CF(9)=2.*TH*P1*(Y2-Y3)/3.
001132         CF(10)=2.*TH*P1*(X3-X2)/3.
001137         CF(11)=2.*TH*P2*(Y3-Y1)/3.
001144         CF(12)=2.*TH*P2*(X1-X3)/3.

001151         IF ( RECCYL .EQ. 0.0 )  GO TO 13

          C    ***** CYLINDRICAL COORDINATES *****
001152         CF(1)= CF(1)*X1
001154         CF(2)= CF(2)*X1
001155         CF(3)= CF(3)*X2
001157         CF(4)= CF(4)*X2
001160         CF(5)= CF(5)*X3
001162         CF(6)= CF(6)*X3
001163         CON1= 0.5*(X1+X2)
001166         CON2= 0.5*(X1+X3)
001171         CON3= 0.5*(X2+X3)
001174         CF(7)= CF(7)*CON1
001176         CF(8)= CF(8)*CON1
001177         CF(9)= CF(9)*CON2
001201         CF(10)= CF(10)*CON2
001202         CF(11)= CF(11)*CON3
001204         CF(12)= CF(12)*CON3
001205         GO TO 13
```

```
                                                                          GEOM2280
001206     5   DO 14 I=1,12                                               GEOM2290
001210    14   CF(I)=0.                                                   GEOM2300
001214         GO TO 13                                                   GEOM2310
                                                                          GEOM2320
001215    30   J= -NLOAD                                                  GEOM2330
001217         DIV= XORD(J)/DPRESS + 1.0                                  GEOM2340
001223         THETA(NV,1)= ( XORD(J)/R1 - 1.0 ) / DIV                    GEOM2350
001235         THETA1= THETA(NV,1)                                        GEOM2360
001243         THETA(NV,2)= ( XORD(J)/R2 - 1.0 ) / DIV                    GEOM2370
001256         THETA2= THETA(NV,2)                                        GEOM2380
001264         THETA(NV,3)= ( XORD(J)/R3 - 1.0 ) / DIV                    GEOM2390
001277         THETA3= THETA(NV,3)                                        GEOM2400
                                                                          GEOM2410
001305         DO 31 I=1,12                                               GEOM2420
001307    31   CF(I)=-TH*( TEMP(7,I)*THETA1 + TEMP(8,I)*THETA2 + TEMP(9,I)*THETA3GEOM2430
              1  )/24.0                                                   GEOM2440
                                                                          GEOM2450
001341         DO 33 I=1,6                                                GEOM2460
001343         DO 33 J=1,3                                                GEOM2470
001344         TTEMP= 0.0                                                 GEOM2480
001345         DO 32 K=1,3                                                GEOM2490
001347    32   TTEMP= TTEMP + R(K,I)*JTT(K,J)                             GEOM2500
001362    33   TEMP(I,J)= TTEMP                                           GEOM2510
001373         DO 34 I=1,6                                                GEOM2520
001374    34   CF(2*I-1)= CF(2*I-1) - TH*( TEMP(I,1)*THETA1 + TEMP(I,2)*THETA2   GEOM2530
              1   + TEMP(I,3)*THETA3 )/360.0                              GEOM2540
                                                                          GEOM2550
001422    13   RETURN                                                     GEOM2560
001423         END                                                        GEOM2570
```

```
RUN 2.3I  05/28/70          GEOM


SUBPROGRAM LENGTH
002I04

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
3       -  000755     4       -  000743     5       -  001206     6       -  000764
7       -  001030     8       -  001014     9       -  001020     10      -  001024
13      -  001422     15      -  001016     16      -  001022     17      -  001026
30      -  001215     31      -  001307     32      -  001347     34      -  001374
50      -  000735     300     -  000410     340     -  000366     345     -  000501
353     -  000532     355     -  000546     360     -  000632     380     -  000643
385     -  000671     390     -  000702     401     -  000733

BLOCK NAMES AND LENGTHS
IDIMEN  -  000022     IFLAGS  -  000012     INDEX   -  000026     PROPS   -  000012
ST      -  000234     TEMP    -  000220     UV      -  000044     XYCCOR  -  000006

VARIABLE ASSIGNMENTS
AREA    -  000007C04  AT      -  001754     AX      -  001746     AY      -  001751
CF      -  000220C05  CON1    -  002074     CON2    -  002075     CON3    -  002076
DELPR   -  002004     DIV     -  002077     DPRESS  -  000001C04  ESIG    -  000001
HG      -  001660     HGIJK   -  002065     I       -  002057     IFIX    -  000000C03
ISIDE   -  002001     J       -  002060     JLOAD   -  002070     JTT     -  002042
JXX     -  002007     JXY     -  002031     JYY     -  002020     J34     -  000010C03
K       -  002053     M       -  002056     N       -  002054     NLOAD   -  000005C01
NM1     -  002055     NNUMEL  -  000002     NV      -  000001C03  P1      -  002071
P2      -  002072     P3      -  002073     R       -  001757     RECCYL  -  000002C02
R1      -  000000C10  R2      -  000001C10  R3      -  000002C10  SJTT    -  002064
SJXX    -  002061     SJXY    -  002063     SJYY    -  002062     ST      -  000000C05
TEMP    -  000000C06  TH      -  000004C04  THETA   -  000000     THETA1  -  002100
THETA2  -  002101     THETA3  -  002102     TTEMP   -  002103     UV      -  000000C07
VPRFLG  -  000025C03  X       -  002066     X1      -  000000C10  X2      -  000001C10
X3      -  000002C10  Y       -  002067     Y1      -  000003C10  Y2      -  000004C10
Y3      -  000005C10
```

```
RUN 2.3I  05/28/70           GEOM

START OF CONSTANTS
001425

START OF TEMPORARIES
001531

START OF INDIRECTS
001643

EXTERNAL REFERENCES
         END

COMPILER SPACE
UNUSED -  006200    USED   -  041600
```

```
              SUBROUTINE GINPUT                                               GINP0010
                                                                              GINP0020
                                                                              GINP0030
000002        COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6  GINP0040
             1   ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                 GINP0050
000002        COMMON  /CMESH /  BLTHNS,XINTN ,XYRAT ,DPRAT                    GINP0060
000002        COMMON  /FFLAGS/  NONTRY,BLANK                                  GINP0070
000002        EQUIVALENCE  (FNONTR,NONTRY)                                    GINP0080
000002        COMMON  /HEADER/  HEADER(10)                                    GINP0090
000002        COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   GINP0100
             1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINGINP0110
             2   ,NXINT ,NCYCF                                                GINP0120
000002        DIMENSION  IDIMEN(13)                                           GINP0130
000002        EQUIVALENCE  (IDIMEN,NDIMEN)                                    GINP0140
000002        COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI GINP0150
             1   ,SAVETP,BLANAL,FLIFE                                         GINP0160
000002        DIMENSION  IFLAGS( 8)                                           GINP0170
000002        EQUIVALENCE  (IFLAGS,NFLGS)                                     GINP0180
000002        COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT  GINP0190
             1   ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN GINP0200
             2   ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                   GINP0210
000002        COMMON  /IOUNIT/  IRSIN,IRSOUT                                  GINP0220
000002        COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATGINP0230
             1   ,RSTRAT,RFINAL                                               GINP0240
000002        DIMENSION  MESHIN( 8)                                           GINP0250
000002        EQUIVALENCE  ( MESHIN,NMESH )                                   GINP0260
000002        COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP    GINP0270
             1   ,AREA  ,TPRESS,RMAXS                                         GINP0280
000002        COMMON  /RSDATA/  RSDATA(36)                                    GINP0290
000002        DIMENSION IRSDAT(36)                                            GINP0300
000002        EQUIVALENCE  (IRSDAT,RSDATA)                                    GINP0310
                                                                              GINP0320
000002        NAMELIST /AS189/  CONVER,HEADER,IPRIN ,IRESI ,ITER  ,MAXBW ,MAXST GINP0330
             1   ,NDP   ,NID   ,NLOAD ,NMAT  ,NNDLIM,NUMBC ,NUMCP ,NUMEL ,NUMNP GINP0340
             2   ,PRESS ,RECCYL,RESTRT,SAVETP,MAXBKT,BLANAL,NLAYER,LWRAT ,NANGINGINP0350
             3   ,NXINT ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRAT,RSTRAT,RFINALGINP0360
             4   ,NCYCF ,FLIFE ,IRSIN ,IRSOUT                                 GINP0370
```

```
                                                                                GINP0380
                                                                                GINP0390
000002          WRITE (6,917)                                                   GINP0400
000006     917  FORMAT ( 1H1,28X,42HLARGE STRAIN, ELASTOPLASTIC FINITE ELEMENT   GINP0410
               1   , 9H ANALYSIS//44X,22HPROGRAM AS189 (BLADER)/// )            GINP0420
                                                                                GINP0430
000006          ERRFLG= 1.0                                                     GINP0440
000010          LREC1= 36                                                       GINP0450
000011          IDUMP= 1                                                        GINP0460
000012          INDMP= 5                                                        GINP0470
000013          ILSS= 1                                                         GINP0480
000014          JSTEP= 1                                                        GINP0490
                                                                                GINP0500
          C     ***** INITIALIZE SPECIFIED INPUT QUANTITIES TO INDICATE NO VALUE GINP0510
          C           INPUT *****                                               GINP0520
000015          DO 110 I=2,NDIMEN                                               GINP0530
000016          IDIMEN(I)= NONTRY                                               GINP0540
000021     110  CONTINUE                                                        GINP0550
000023          DO 120 I= 3,NFLGS                                               GINP0560
000025          IFLAGS(I)= NONTRY                                               GINP0570
000030     120  CONTINUE                                                        GINP0580
000032          DO 130 I=2,NMESH                                                GINP0590
000034          MESHIN(I)= NONTRY                                               GINP0600
000037     130  CONTINUE                                                        GINP0610
000041          DO 140 I=1,10                                                   GINP0620
000043          HEADER(I)= BLANK                                                GINP0630
000046     140  CONTINUE                                                        GINP0640
                                                                                GINP0650
000050          RESTRT= 0.0                                                     GINP0660
000051          PRESS= FNONTR                                                   GINP0670
                                                                                GINP0680
          C     ***** READ PROBLEM DIMENSIONS AND OTHER GENERAL INPUT DATA ***** GINP0690
000052          READ (5,AS189)                                                  GINP0700
                                                                                GINP0710
                                                                                GINP0720
000055          IF (RESTRT .EQ. 0.0)  GO TO 300                                 GINP0730
                                                                                GINP0740
          C     ***** THIS RUN IS A RESTART FROM A PREVIOUSLY EXECUTED RUN ***** GINP0750
```

```
000056          WRITE (6,901)  RESTRT
000064     901  FORMAT (32HOTHIS RUN IS A RESTART FROM DUMP F5.0 )
000064          REWIND IRSIN

         C      ***** READ GENERAL INPUT QUANTITIES FROM RESTART TAPE INTO
         C            RSDATA ARRAY *****
000066          READ (IRSIN)  ( RSDATA(I),I=1,LREC1 )

000100          DO 220 I=2,NDIMEN

000102          IF ( IDIMEN(I) .NE. NONTRY )  GO TO 210

         C      ***** NO VALUE WAS SPECIFIED FOR IDIMEN(I). USE VALUE FROM
         C            RESTART TAPE *****
000105          IDIMEN(I)= IRSDAT(I-1)
000110          GO TO 220

000110     210  IF ( IDIMEN(I) .GE. IRSDAT(I-1) )  GO TO 220

         C      ***** ERROR. DIMENSIONS MAY NOT BE REDUCED ON A RESTART *****
000114          WRITE (6,902)  I,IDIMEN(I),IRSDAT(I-1)
000130     902  FORMAT (16HOERROR...IDIMEN( I4,2H)= I6,24H IS LESS THAN OLD VALUE=
               1  I6 )
000130          ERRFLG= 0.0

000131     220  CONTINUE


         C      ***** PICK UP OLD VALUES OF  RECCYL, CONVER, IPRIN, ITER, IRESI,
         C            AND SAVETP  FROM RESTART TAPE IF NEW VALUES ARE NOT ENTERED.
000134          II= NDIMEN
000135          DO 240 I=3,NFLGS
000137          IF ( IFLAGS(I) .NE. NONTRY )  GO TO 240

000142          IFLAGS(I)= IRSDAT(II)
000145     240  II= II + 1
```

```
000151          IF ( PRESS .EQ. FNONTR )  PRESS= RSDATA(II)
000156          II= II + 1


000160          DC 250 I=1,10
000161          IF ( HEADER(I) .NE. BLANK )  GO TO 4C0
000164     250  CONTINUE
           C
           C    ***** NO HEADER WAS INPUT. USE HEADER FROM RESTART TAPE. *****
000166          DO 260 I=1,10
000167          HEADER(I)= RSDATA(II)
000173     260  II= II + 1
000176          GC TO 400


           C    ***** THIS RUN IS NOT A RESTART. IF SPECIFIED QUANTITIES WERE NOT
           C          ENTERED IN NAMELIST INPUT  USE NOMINAL VALUES *****

000177     300  IF ( NMAT   .EQ. NONTRY )  NMAT   = 1
000202          IF ( MAXBW  .EQ. NONTRY )  MAXBW  = 5C
000205          IF ( MAXST  .EQ. NONTRY )  MAXST  = 25
000210          IF ( MAXBKT .EQ. NONTRY )  MAXBKT= 25000
000213          IF ( RECCYL .EQ. FNONTR )  RECCYL= 0.0
000216          IF ( IPRIN  .EQ. NONTRY )  IPRIN  = 1
000221          IF ( IRESI  .EQ. NONTRY )  IRESI  = 0
000224          IF ( SAVETP .EQ. FNONTR )  SAVETP= 0.0
000227          IF ( BLANAL .EQ. FNONTR )  BLANAL= C.0
000232          IF ( FLIFE .EQ. FNONTR )  FLIFE= C.C
000235          FLIFE= FLIFE*BLANAL


000237          IF ( BLANAL .EQ. 0.0 )  GC TO 350

           C    ***** BLADDER ANALYSIS OPTION *****

000240          IF ( NLAYER .LE. 0 )  GO TO 400

```

```
000241           NPTSY= 2*NLAYER + 1                                     GINP1520
                                                                         GINP1530
000243           IF ( RECCYL .NE. 0.0 )  GO TO 330                       GINP1540
                                                                         GINP1550
          C      ***** SINGLE FOLD OPTION *****                          GINP1560
000244           IF ( LWRAT .EQ. NONTRY )  LWRAT= 2                      GINP1570
000247           IF ( NXINT .EQ. NONTRY )  NXINT= LWRAT*NLAYER           GINP1580
000254           NUMBC= NPTSY                                            GINP1590
000256           NLOAD= 0                                                GINP1600
000257           NID= 2*NPTSY                                            GINP1610
000260           XYRAT= LWRAT                                            GINP1620
000262           IF ( XSTART .EQ. FNONTR )  XSTART= 0.0                  GINP1630
000265           IF ( YSTART .EQ. FNONTR )  YSTART= 0.0                  GINP1640
000270           IF ( RPTRAT .EQ. FNONTR )  RPTRAT= 1000.0               GINP1650
000274           GO TO 340                                               GINP1660
                                                                         GINP1670
          C      ***** DOUBLE FOLD OPTION *****                          GINP1680
000275       330 IF ( NANGIN .EQ. NONTRY )  NANGIN= 18                   GINP1690
000300           NXINT= NANGIN                                           GINP1700
000302           NUMBC= 2*NPTSY                                          GINP1710
000303           NID= 0                                                  GINP1720
000304           NLOAD= -NANGIN*NPTSY - 1                                GINP1730
000307           IF ( CONVER .EQ. FNONTR )  CONVER= 0.1                  GINP1740
000313           IF ( ITER .EQ. NONTRY )  ITER= 1                        GINP1750
000316           IF ( RPTRAT .EQ. FNONTR )  RPTRAT= 10.0                 GINP1760
000322           IF ( RSTRAT .EQ. FNONTR )  RSTRAT= 100.0                GINP1770
                                                                         GINP1780
000326       340 NPTSX= 2*NXINT + 1                                      GINP1790
000330           NUMNP= NPTSY*NPTSX                                      GINP1800
000333           NUMCP= (NLAYER+1)*(NXINT+1)                             GINP1810
000337           NUMEL= 2*NLAYER*NXINT                                   GINP1820
000342           NDP= 0                                                  GINP1830
000343           XINTN= NXINT                                            GINP1840
000344           NCYCF2= 2*NCYCF                                         GINP1850
                                                                         GINP1860
                                                                         GINP1870
000346       350 IF ( CONVER .EQ. FNONTR )  CONVER= 1.E-4                GINP1880
000352           IF ( ITER   .EQ. NONTRY )  ITER  = 10                   GINP1890
```

```
000355          DO 360 I=2,NDIMEN                                                   GINP1900
000357          IF ( IDIMEN(I) .EQ. NONTRY )  IDIMEN(I)= 0                          GINP1910
000364     360  CONTINUE                                                            GINP1920
                                                                                    GINP1930
000367          LSTEPO= 1                                                           GINP1940
000370          NEQBC= 0                                                            GINP1950
000371          TPRESS= 0.0                                                         GINP1960
000372          RMAXS=MAXST                                                         GINP1970
                                                                                    GINP1980
                                                                                    GINP1990
000373     400  WRITE (6,907)  HEADER                                               GINP2000
000401     907  FORMAT (1H1,5X,10A6)                                                GINP2010
                                                                                    GINP2020
000401          NPRIN=IPRIN                                                         GINP2030
000403          VPRFLG= 0.0                                                         GINP2040
000404          IF ( (NLOAD .GT. 0) .AND. (PRESS .EQ. FNONTR) .AND. (NDP .EQ. 0) )GINP2050
               1  VPRFLG= 1.0                                                       GINP2060
                                                                                    GINP2070
              C     ***** CHECK PROBLEM DIMENSIONS SPECIFICATION FOR OBVIOUS ERRORS.  GINP2080
000421          CALL DIMCHK                                                         GINP2090
                                                                                    GINP2100
000422          IF ( ERRFLG .EQ. 0.0 )  STOP                                        GINP2110
                                                                                    GINP2120
                                                                                    GINP2130
000425          IF ( SAVETP .EQ. 0.0 )  GO TO 900                                   GINP2140
                                                                                    GINP2150
              C     ***** WRITE FIRST RECORD OF NEW RESTART TAPE *****              GINP2160
000426          REWIND IRSOUT                                                       GINP2170
000430          WRITE (IRSOUT)  ( IDIMEN(I),I=2,NDIMEN ), ( IFLAGS(I),I=3,NFLGS ) GINP2180
               1  ,PRESS,HEADER                                                     GINP2190
000453     900  RETURN                                                              GINP2200
000454           END                                                                GINP2210
```

```
RUN 2.3I  05/28/70          GINPUT


SUBPROGRAM LENGTH
000730

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
210     -  000110    220     -  000131    240     -  CC0145    300     -  000177
330     -  000275    340     -  000326    350     -  CC0346    360     -  000364
400     -  000373    900     -  000453    901     -  C00663    902     -  000671
907     -  000707    917     -  000644

BLOCK NAMES AND LENGTHS
CDIMEN  -  000013    CMESH   -  000004    FFLAGS  -  0C0002    HEADER  -  000012
IDIMEN  -  000022    IFLAGS  -  000012    INDEX   -  C00026    IOUNIT  -  000002
MESHIN  -  000011    PROPS   -  000012    RSDATA  -  000044

VARIABLE ASSIGNMENTS
BLANAL  -  000010C06 BLANK   -  000001C03 CONVER  -  0C0003C06 ERRFLG  -  000007C07
FLIFE   -  000011C06 FNONTR  -  000000C03 HEADER  -  C00000C04 I       -  000726
IDIMEN  -  000000C05 IDUMP   -  000023C07 IFLAGS  -  CC0000C06 II      -  000727
ILSS    -  000024C07 INDMP   -  000021C07 IPRIN   -  000004C06 IRESI   -  000006C06
IRSDAT  -  000000C13 IRSIN   -  000000C10 IRSOUT  -  CC0001C10 ITER    -  000005C06
JSTEP   -  000003C07 LREC1   -  000001C01 LSTEPO  -  C00016C07 LWRAT   -  000016C05
MAXBKT  -  000014C05 MAXBW   -  000011C05 MAXST   -  C00012C05 MESHIN  -  000000C11
NANGIN  -  000017C05 NCYCF   -  000021C05 NCYCF2  -  000012C01 NDIMEN  -  000000C05
NDP     -  000010C05 NEQBC   -  000003C01 NFLGS   -  000000C06 NID     -  000006C05
NLAYER  -  000015C05 NLOAD   -  000005C05 NMAT    -  0C0007C05 NMESH   -  000000C11
NNDLIM  -  000013C05 NONTRY  -  000000C03 NPRIN   -  0C0017C07 NPTSX   -  000011C01
NPTSY   -  000010C01 NUMBC   -  000004C05 NUMCP   -  000003C05 NUMEL   -  000001C05
NUMNP   -  000002C05 NXINT   -  000020C05 PRESS   -  000000C12 RECCYL  -  000002C06
RESTRT  -  000001C06 RFINAL  -  000010C11 RMAXS   -  000011C12 RPFOLD  -  000004C11
RPTRAT  -  000006C11 RSDATA  -  000000C13 RSFOLD  -  0C0005C11 RSTRAT  -  000007C11
SAVETP  -  000007C06 TPRESS  -  000010C12 VPRFLG  -  000025C07 XINTN   -  000001C02
XLNGTH  -  000003C11 XSTART  -  000001C11 XYRAT   -  000002C02 YSTART  -  000002C11

START OF CONSTANTS
```

```
RUN 2.3I  05/28/70            GINPUT

000642

START OF TEMPORARIES
000713

START OF INDIRECTS
000723

EXTERNAL REFERENCES
         OUTPTC    INPUTN    REWINM    INPUTB    DIMCHK    STOP     OUTPTB    END

COMPILER SPACE
UNUSED -  010500    USED   -  037300
```

```
                SUBROUTINE MSHGEN (NP     ,XORD   ,YORD   ,DEPTH ,CYCM   ,CYCZ   ,VTH   MSHG0010
               1     ,MLAYER,MATEL ,ND     ,NEBC   ,CYCTOF,IERROR )                     MSHG0020
                                                                                        MSHG0030
          C     ***** THE ARRAY NP IS STORED AS A 6*NUMEL ARRAY AS IN SETUP *****       MSHG0040
          C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.      MSHG0050
000020          DIMENSION  NP(6,1),CYCTOF(40,1)                                         MSHG0060
               1  ,XORD  (1),YORD  (1),DEPTH (1),CYCM  (1),CYCZ  (1),VTH   (1)          MSHG0070
               2  ,MLAYER(1),MATEL (1),ND    (1),NEBC  (1)                              MSHG0080
                                                                                        MSHG0090
000020          COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6        MSHG0100
               1   ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                         MSHG0110
000020          COMMON  /CMESH /  BLTHNS,XINTN ,XYRAT ,DPRAT                            MSHG0120
000020          COMMON  /FFLAGS/  NONTRY,BLANK                                          MSHG0130
000020          EQUIVALENCE  (FNONTR,NONTRY)                                            MSHG0140
000020          COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID         MSHG0150
               1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINMSHG0160
               2   ,NXINT ,NCYCF                                                        MSHG0170
000020          COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI       MSHG0180
               1   ,SAVETP,BLANAL,FLIFE                                                 MSHG0190
000020          COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATMSHG0200
               1   ,RSTRAT,RFINAL                                                       MSHG0210
000020          COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP          MSHG0220
               1   ,AREA  ,TPRESS,RMAXS                                                 MSHG0230
                                                                                        MSHG0240
000020     30   FORMAT (19H  NODAL COORDINATES/7H0 POINT,9X,5HX-ORD,10X,5HY-ORD//       MSHG0250
               1  (I7,2F15.5) )                                                         MSHG0260
000020     32   FORMAT (22H1NO. OF ELEMENT LAYERS,I18/  22H LENGTH TO WIDTH RATIO       MSHG0270
               1  ,I18/  25H NO. OF ANGULAR INTERVALS,I15/  19H NO. OF X-INTERVALSMSHG0280
               2  ,I21/  13H TOTAL LENGTH,E32.5/  20H PRIMARY FOLD RADIUS,E25.5/        MSHG0290
               3  22H SECONDARY FOLD RADIUS,E23.5/  21H FINAL FOLDING RADIUS,E24.5MSHG0300
               4  // )                                                                  MSHG0310
000020     36   FORMAT (20H0INPUT ERROR.  CYCM(I4,2H)=E15.7,19H, BUT NO VALUE WAS       MSHG0320
               1  15HINPUT FOR CYCZ(I4,2H).  )                                          MSHG0330
000020     37   FORMAT (1H1,19X,13HNO. OF CYCLES/16H  PLASTIC STRAIN,6X,                MSHG0340
               1  10HTO FAILURE,10X,12HFOR MATERIAL,I5 )                                MSHG0350
000020     38   FORMAT (E14.5,F14.0)                                                    MSHG0360
000020     39   FORMAT (29H1 PLASTIC STRAIN COEFFICIENTS/ 10H0 MATERIAL,8X,1HM,         MSHG0370
```

```
               1  15X,1HZ/ (I7,E17.5,E16.5) )                                  MSHG0380
                                                                               MSHG0390
                                                                               MSHG0400
         C      ***** GENERATE THE NP ARRAY. IF NMAT .GT. 0, ALSO GENERATE MATEL  MSHG0410
         C           ARRAY *****                                               MSHG0420
000020          I1= 1                                                          MSHG0430
000021          N= 1                                                           MSHG0440
000022          DO 1140 J=1,NXINT                                              MSHG0450
                                                                               MSHG0460
000023          DO 1120 I=1,NLAYER                                             MSHG0470
000024          I2= I1 + NPTSY                                                 MSHG0480
000026          I3= I2 + NPTSY                                                 MSHG0490
000030          NP(1,N)= I1                                                    MSHG0500
000034          NP(2,N)= I3                                                    MSHG0510
000037          NP(3,N)= I3 + 2                                                MSHG0520
000043          NP(4,N)= I2                                                    MSHG0530
000047          NP(5,N)= I3 + 1                                                MSHG0540
000053          NP(6,N)= I2 + 1                                                MSHG0550
                                                                               MSHG0560
000057          NP1= N + 1                                                     MSHG0570
000061          NP(1,NP1)= I1                                                  MSHG0580
000064          NP(2,NP1)= I3 + 2                                              MSHG0590
000070          NP(3,NP1)= I1 + 2                                              MSHG0600
000074          NP(4,NP1)= I2 + 1                                              MSHG0610
000100          NP(5,NP1)= I2 + 2                                              MSHG0620
000104          NP(6,NP1)= I1 + 1                                              MSHG0630
                                                                               MSHG0640
000110          IF ( NMAT .EQ. 1 )  GO TO 1110                                 MSHG0650
                                                                               MSHG0660
         C      ***** MORE THAN 1 MATERIAL. STORE MATERIAL I.D. FOR ELEMENTS N,N+1MSHG0670
000112          MLAYI= MLAYER(I)                                               MSHG0680
000115          MATEL(N)= MLAYI                                                MSHG0690
000120          MATEL(NP1)= MLAYI                                              MSHG0700
                                                                               MSHG0710
000123     1110 N= N + 2                                                       MSHG0720
000125          I1= I1 + 2                                                     MSHG0730
000126     1120 CONTINUE                                                       MSHG0740
                                                                               MSHG0750
```

```
000131              I1= I2 + 3                                                  MSHG0760
000133        1140  CONTINUE                                                    MSHG0770
                                                                                MSHG0780
            C       ***** SET VTH ARRAY TO INDICATE UNIT THICKNESS *****        MSHG0790
000135              DO 1150 I=1,NMAT                                            MSHG0800
000137              VTH(I)= 1.0                                                 MSHG0810
000142        1150  CONTINUE                                                    MSHG0820
                                                                                MSHG0830
            C       ***** CALCULATE COORDINATES OF NODAL POINTS *****           MSHG0840
                                                                                MSHG0850
            C       ***** FIRST CALCULATE TOTAL WIDTH (I.E. BLADDER THICKNESS ) ***** MSHG0860
000145              BLTHNS= 0.0                                                 MSHG0870
000146              DO 1210 I=1,NLAYER                                          MSHG0880
000147              BLTHNS= BLTHNS + DEPTH(I)                                   MSHG0890
000152        1210  CONTINUE                                                    MSHG0900
                                                                                MSHG0910
000154              IF ( RPFOLD .EQ. FNONTR )  RPFOLD= RPTRAT*BLTHNS            MSHG0920
                                                                                MSHG0930
000160              IF ( RECCYL .EQ. 0.0 )  GO TO 1300                          MSHG0940
                                                                                MSHG0950
            C       ***** DOUBLE FOLD OPTION *****                              MSHG0960
000161              DELANG= 1.5707963/XINTN                                     MSHG0970
000163              ANGLE= 0.0                                                  MSHG0980
000164              IF ( RSFOLD .EQ. FNONTR )  RSFOLD= RSTRAT*BLTHNS            MSHG0990
                                                                                MSHG1000
            C       ***** INITIALIZE LOADING PARAMETERS *****                   MSHG1010
000170              DPRESS= ( RFINAL - RSFOLD ) / RMAXS                         MSHG1020
                                                                                MSHG1030
000173              II= 1                                                       MSHG1040
000174              DO 1260 J=1,NPTSX                                           MSHG1050
000176              SINANG= SIN(ANGLE)                                          MSHG1060
000200              COSANG= COS(ANGLE)                                          MSHG1070
000202              R= RPFOLD                                                   MSHG1080
000203              IEVEN= 0                                                    MSHG1090
000204              ILAY= 0                                                     MSHG1100
                                                                                MSHG1110
000205              DO 1240 I=1,NPTSY                                           MSHG1120
000212              XORD(II)= RSFOLD - R*COSANG                                 MSHG1130
```

```
000217          YORD(II)= R*SINANG                                          MSHG1140
000222          II= II + 1                                                  MSHG1150
000223          IF ( I .EQ. NPTSY )  GO TO 1250                             MSHG1160
                                                                            MSHG1170
000225          IF ( IEVEN .NE. 0 )  GO TO 1220                             MSHG1180
                                                                            MSHG1190
           C    ***** I IS ODD.  PICK UP NEXT DELTA R *****                 MSHG1200
000226          IEVEN= 1                                                    MSHG1210
000227          ILAY= ILAY + 1                                              MSHG1220
000231          DELTAR= 0.5*DEPTH(ILAY)                                     MSHG1230
000234          GO TO 1230                                                  MSHG1240
                                                                            MSHG1250
           C    ***** I IS EVEN. USE SAME DELTA R *****                     MSHG1260
000234     1220 IEVEN= 0                                                    MSHG1270
                                                                            MSHG1280
000235     1230 R= R + DELTAR                                               MSHG1290
000237     1240 CONTINUE                                                    MSHG1300
                                                                            MSHG1310
000242     1250 ANGLE= ANGLE + DELANG                                       MSHG1320
000244     1260 CONTINUE                                                    MSHG1330
                                                                            MSHG1340
000247          GO TO 1400                                                  MSHG1350
                                                                            MSHG1360
           C    ***** SINGLE FOLD OPTION *****                              MSHG1370
           C    ***** CALCULATE LENGTH IF NOT ENTERED IN INPUT *****        MSHG1380
000247     1300 IF ( XLNGTH .EQ. FNONTR )  XLNGTH= XYRAT*BLTHNS             MSHG1390
000253          DELTAX= 0.5*XLNGTH/XINTN                                    MSHG1400
000256          X= XSTART                                                   MSHG1410
000257          II= 1                                                       MSHG1420
000260          DO 1340 J=1,NPTSX                                           MSHG1430
                                                                            MSHG1440
           C    ***** STORE X AND Y FOR POINT II, WHICH IS THE BOTTOM POINT OF  MSHG1450
           C          LAYER 1 AND IS IN COLUMN J. *****                     MSHG1460
000262          Y= YSTART                                                   MSHG1470
000264          XORD(II)= X                                                 MSHG1480
000266          YORD(II)= Y                                                 MSHG1490
000271          II= II + 1                                                  MSHG1500
                                                                            MSHG1510
```

```
000272          DO 1320 I=1,NLAYER                                             MSHG1520
        C       ***** POINTS II AND II+1 ARE THE MIDPOINT AND UPPER POINT      MSHG1530
        C             RESPECTIVELY OF LAYER I AND ARE IN COLUMN J *****        MSHG1540
000274          XORD(II)= X                                                    MSHG1550
000277          XORD(II+1)= X                                                  MSHG1560
000301          DY= DEPTH(I)                                                   MSHG1570
000303          YORD(II)= Y + 0.5*DY                                           MSHG1580
000310          Y= Y + DY                                                      MSHG1590
000312          YORD(II+1)= Y                                                  MSHG1600
000314          II= II + 2                                                     MSHG1610
000315     1320 CONTINUE                                                       MSHG1620
                                                                               MSHG1630
000320          X= X + DELTAX                                                  MSHG1640
000322     1340 CONTINUE                                                       MSHG1650
                                                                               MSHG1660
        C       ***** INITIALIZE PARAMETERS SPECIFYING IMPOSED DISPLACEMENTS *****MSHG1670
000324          DPRESS= 0.0                                                    MSHG1680
000325          RSFOLD= RPFOLD                                                 MSHG1690
000326          DPRAT= (RFINAL/RPFOLD)**(1.0/(RMAXS-1.0) )                     MSHG1700
000335          NN= NUMNP - NPTSY                                              MSHG1710
000337          NN= NN + NN + 1                                                MSHG1720
000341          DO 1360 I=1,NID                                                MSHG1730
000343          ND(I)= NN                                                      MSHG1740
000346          NN= NN + 1                                                     MSHG1750
000350     1360 CONTINUE                                                       MSHG1760
        C                                                                      MSHG1770
000352     1400 WRITE (6,32 )  NLAYER,LWRAT ,NANGIN,NXINT,XLNGTH,RPFOLD,RSFOLD MSHG1780
               1  ,RFINAL                                                      MSHG1790
                                                                               MSHG1800
000376          WRITE (6,30)  ( I,XORD(I),YORD(I),I=1,NUMNP )                  MSHG1810
                                                                               MSHG1820
        C       ***** SET BOUNDARY CONDITION FLAGS *****                       MSHG1830
                                                                               MSHG1840
        C       ***** FOR SINGLE FOLD OPTION CONSTRAIN ALL NODES ON LEFT EDGE  MSHG1850
        C             FROM MOVEMENT IN THE X-DIRECTION *****                   MSHG1860
000427          II= 1                                                          MSHG1870
        C       ***** FOR DOUBLE FOLD OPTION CONSTRAIN ALL NODES ON BOTH FREE  MSHG1880
        C             EDGES FROM MOVEMENT IN THE Z-DIRECTION *****             MSHG1890
```

```
000430          IF ( RECCYL .NE. 0.0 )  II= 2                                   MSHG1900
                                                                                MSHG1910
000436          DO 1420 I=1,NUMBC                                               MSHG1920
          C     ***** FOR THE DOUBLE FOLD OPTION THE LAST NPTSY POINTS ARE      MSHG1940
          C           CONSTRAINED AS WELL AS THE FIRST NPTSY POINTS *****       MSHG1950
000440          NEBC(I)= II                                                     MSHG1930
000443          IF ( I .EQ. NPTSY )  II= 2*(NUMNP-NPTSY)                        MSHG1960
000447          II= II + 2                                                      MSHG1970
000451     1420 CONTINUE                                                        MSHG1980
000453          NEQBC= NUMBC                                                    MSHG1990
                                                                                MSHG2000
000455     1500 IF ( FLIFE .EQ. 0.0 )  GO TO 2000                               MSHG2010
                                                                                MSHG2020
000456          IF ( NCYCF .EQ. 0 )  GO TO 1590                                 MSHG2030
                                                                                MSHG2040
          C     ***** CALCULATE M AND Z ARRAYS FOR FATIGUE LIFE CALCULATIONS *****MSHG2050
000457          FN= NCYCF                                                       MSHG2060
                                                                                MSHG2070
000461          DO 1580 M=1,NMAT                                                MSHG2080
          C     ***** IF CYCM(M) WAS INPUT, THEN  DO NOT CALCULATE CYCM(M) AND  MSHG2090
          C           CYCZ(M) FOR MATERIAL M *****                              MSHG2100
000462          IF ( CYCM(M) .EQ. FNONTR )  GO TO 1510                          MSHG2110
000465          IF ( CYCZ(M) .NE. FNONTR )  GO TO 1580                          MSHG2120
                                                                                MSHG2130
000470          IERROR= 0                                                       MSHG2140
000471          WRITE (6,36)  M,CYCM(M),M                                       MSHG2150
000507          GO TO 1580                                                      MSHG2160
                                                                                MSHG2170
          C     ***** CALCULATE CYCM(M) AND CYCZ(M) FOR MATERIAL M *****        MSHG2180
000513     1510 SUMLE = 0.0                                                     MSHG2190
000514          SUMLN = 0.0                                                     MSHG2200
000515          SUMLEN= 0.0                                                     MSHG2210
000516          SUMLN2= 0.0                                                     MSHG2220
000517          WRITE (6,37)  M                                                 MSHG2230
                                                                                MSHG2240
000524          DO 1550 I=2,NCYCF2,2                                            MSHG2250
000531          PLSTI= CYCTOF(I-1,M)                                            MSHG2260
000536          FNCTOF= CYCTOF(I,M)                                             MSHG2270
```

```
000543          IF ( PLSTI*FNCTOF .EQ. C.0 )  GO TO 1550                 MSHG2280
000545          WRITE (6,38)  PLSTI,FNCTOF                               MSHG2290
000555          ALNEI= ALOG(PLSTI)                                       MSHG2300
000560          ALNNI= ALOG(FNCTOF)                                      MSHG2310
000562          SUMLE= SUMLE + ALNEI                                     MSHG2320
000564          SUMLN= SUMLN + ALNNI                                     MSHG2330
000566          SUMLEN= SUMLEN + ALNEI*ALNNI                             MSHG2340
000571          SUMLN2= SUMLN2 + ALNNI*ALNNI                             MSHG2350
000573     1550 CONTINUE                                                 MSHG2360
                                                                         MSHG2370
000601          DENOM= FN*SUMLN2 - SUMLN*SUMLN                           MSHG2380
000604          ALNM= ( SUMLN2*SUMLE - SUMLN*SUMLEN ) / DENOM            MSHG2390
000611          CYCM(M)= EXP(ALNM)                                       MSHG2400
000620          CYCZ(M)= ( FN*SUMLEN - SUMLN*SUMLE ) / DENOM             MSHG2410
000626     1580 CONTINUE                                                 MSHG2420
                                                                         MSHG2430
000631     1590 WRITE (6,39)  ( M,CYCM(M),CYCZ(M),M=1,NMAT )             MSHG2440
                                                                         MSHG2450
000657     2000 RETURN                                                   MSHG2460
000660          END                                                      MSHG2470
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
NONTRY
```

```
SUBPROGRAM LENGTH
001072

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
30     -  000663     32     -  000674     36     -  000730     37     -  000743
38     -  000757     39     -  000762     1110   -  000123     1220   -  000234
1230   -  000235     1250   -  000242     1300   -  000247     1400   -  000352
1500   -  000455     1510   -  000513     1550   -  000573     1580   -  000626
1590   -  000631     2000   -  000657

BLOCK NAMES AND LENGTHS
CDIMEN -  000013     CMESH  -  000004     FFLAGS -  000002     IDIMEN -  000022
IFLAGS -  000012     MESHIN -  000011     PROPS  -  000012

VARIABLE ASSIGNMENTS
ALNEI  -  001066     ALNM   -  001071     ALNNI  -  001067     ANGLE  -  001041
BLTHNS -  000000C02 COSANG -  001044     CYCTOF -  000005     DELANG -  001040
DELTAR -  001050     DELTAX -  001051     DENOM  -  001070     DPRAT  -  000003C02
DPRESS -  000001C07 DY     -  001054     FLIFE  -  000011C05 FN     -  001056
FNCTOF -  001065     FNONTR -  000000C03 I      -  001033     IERROR -  000006
IEVEN  -  001046     II     -  001042     ILAY   -  001047     I1     -  001030
I2     -  001034     I3     -  001035     J      -  001032     LWRAT  -  000016C04
M      -  001057     MATEL  -  000002     MLAYER -  000001     MLAYI  -  001037
N      -  001031     NANGIN -  000017C04 NCYCF  -  000021C04 NCYCF2 -  000012C01
ND     -  000003     NEBC   -  000004     NEQBC  -  000003C01 NID    -  000006C04
NLAYER -  000015C04 NMAT   -  000007C04 NN     -  001055     NPTSX  -  000011C01
NPTSY  -  000010C01 NP1    -  001036     NUMBC  -  000004C04 NUMNP  -  000002C04
NXINT  -  000020C04 PLSTI  -  001064     R      -  001045     RECCYL -  000002C05
RFINAL -  000010C06 RMAXS  -  000011C07 RPFOLD -  000004C06 RPTRAT -  000006C06
RSFOLD -  000005C06 RSTRAT -  000007C06 SINANG -  001043     SUMLE  -  001060
SUMLEN -  001062     SUMLN  -  001061     SUMLN2 -  001063     VTH    -  000000
X      -  001052     XINTN  -  000001C02 XLNGTH -  000003C06 XSTART -  000001C06
XYRAT  -  000002C02 Y      -  001053     YSTART -  000002C06
```

```
RUN 2.3I  05/28/70          MSHGEN

START OF CONSTANTS
000662

START OF TEMPORARIES
001000

START OF INDIRECTS
001011

EXTERNAL REFERENCES
         SIN       COS       RBAREX    OUTPTC    ALOG      EXP       END

COMPILER SPACE
UNUSED -  010300    USED   -  037500
```

```
                SUBROUTINE PRINT (XORD   ,YORD   ,DWP    ,WP     ,TST    ,PST    ,EST    PRIN0010
               1    ,SIG    ,DIS    ,F      ,DF     ,RESI   ,NP     ,MLAYER,CYCM   ,CYCZ   PRIN0020
               2    ,NNUMNP,NNUMEL )                                                      PRIN0030
                                                                                          PRIN0040
          C           PRINTS OUT X AND Y COMPONENTS OF TOTAL DISPLACEMENTS, TOTAL PRIN0050
          C           FORCES, AND RESIDUAL FORCES                                 PRIN0060
          C           PRINTS OUT STRAINS, STRESSES, AND PLASTIC WORK FOR EACH     PRIN0070
          C           NODAL POINT *****                                           PRIN0080
                                                                                  PRIN0090
          C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.PRIN0100
000025          DIMENSION    XORD  (1),YORD  (1),DWP   (1),WP    (1)              PRIN0110
000025          DIMENSION    DIS   (1),F     (1),DF    (1),RESI  (1)              PRIN0120
               1  ,MLAYER(1),CYCM  (1),CYCZ  (1)                                  PRIN0130
000025          DIMENSION    TST(NNUMNP,4) , PST(NNUMNP,4) , EST(NNUMNP,4)        PRIN0140
               1             ,SIG(NNUMNP,5) , NP (NNUMEL,6)                       PRIN0150
                                                                                  PRIN0160
          C     ***** THE INDICATED DIMENSION OF THE ARRAY B IS A DUMMY.          PRIN0170
          C           THE ACTUAL DIMENSION NBUCKT IS CALCULATED BY SUBROUTINE BLOCPRIN0180
          C           AND THE REQUIRED STORAGE IS ALLOCATED BY SUBROUTINE MRELSE. PRIN0190
                                                                                  PRIN0200
000025          COMMON  B(1)                                                      PRIN0210
                                                                                  PRIN0220
000025          COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6  PRIN0230
               1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                  PRIN0240
000025          COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   PRIN0250
               1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINPRIN0260
               2    ,NXINT ,NCYCF                                                 PRIN0270
000025          COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI PRIN0280
               1    ,SAVETP,BLANAL,FLIFE                                          PRIN0290
000025          COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT  PRIN0300
               1    ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN PRIN0310
               2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                    PRIN0320
000025          COMMON  /IOUNIT/  IRSIN,IRSOUT                                    PRIN0330
          C     ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE PRIN0340
          C           ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR  PRIN0350
          C           IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE   PRIN0360
          C           POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. ***** PRIN0370
```

```
000025          COMMON  /LBUCKT/  LNP    ,LLOADE,LLOADS,LNEBC ,LND    ,LMATEL,LBETA PRINO380
               1    ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP   ,LXORD ,LYORD ,LDWP  PRINO390
               2    ,LWP   ,LCOUNT,LDF   ,LDR   ,LDIS  ,LF     ,LRESI ,LTST  ,LPST  PRINO400
               3    ,LEST  ,LDTST ,LSIG  ,LDSD  ,LDLIM ,LDLFLG,LVEM  ,LVXU  ,LVTH  PRINO410
               4    ,LVXI  ,LVYP  ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYRPRINO420
               5    ,LDEPTH,LCYCF ,LA    ,LTPR  ,LDPR                              PRINO430
               6    ,NBUCKT                                                        PRINO440
000025          COMMON  /LSSAVE/  LSSAVE(2,10)                                     PRINO450
000025          COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATPRINO460
               1    ,RSTRAT,RFINAL                                                 PRINO470
000025          COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP    PRINO480
               1    ,AREA  ,TPRESS,RMAXS                                           PRINO490
                                                                                   PRINO500
000025     161  FORMAT ( 5HODUMP I5,38H OF RESTART DATA OCCURRED ON LOAD STEP I5   PRINO510
               1  ,5X,16HTOTAL PRESSURE = E15.6 )                                  PRINO520
000025     162  FORMAT (44H1NODAL POINT FORCES AND DISPLACEMENTS - STEP,I5// )     PRINO530
000025     163  FORMAT ( I5,2E15.6,2(E20.6,E15.6) )                                PRINO540
000025     164  FORMAT (6H POINT,5X,5HX-DIS,10X,5HY-DIS,15X,6HX-LOAD,9X,6HY-LOAD   PRINO550
               2  12X,10HX-RESIDUAL,5X,10HY-RESIDUAL / )                           PRINO560
000025     165  FORMAT (6H POINT,5X,5HR-DIS,10X,5HZ-DIS,15X,6HR-LOAD,9X,6HZ-LOAD   PRINO570
               1  12X,10HR-RESIDUAL,5X,10HZ-RESIDUAL / )                           PRINO580
000025     166  FORMAT (22HORADIUS OF CURVATURE = E15.6 )                          PRINO590
000025     530  FORMAT (1H1,31X,13HPOINT STRAINS///16X,15HELASTIC STRAINS/ )       PRINO600
000025     531  FORMAT ( 6HOPOINT,10X,5HEPS-X,15X,5HEPS-Y,15X,6HEPS-XY/ )          PRINO610
000025     532  FORMAT (1H1,15X,15HPLASTIC STRAINS/ )                              PRINO620
000025     533  FORMAT (1H1,15X,13HTOTAL STRAINS/ )                                PRINO630
000025     534  FORMAT (16H1 POINT STRESSES//7H  POINT,4X,5HR-ORD,5X,5HZ-ORD,      PRINO640
               1  13X,5HSIG-R,5X,5HSIG-Z,5X,5HSIG-O,4X,6HTAU-RZ,3X,10HEQ. STRESS,  PRINO650
               2    3X,8HPL. WORK,5X,3HDWP,/ )                                     PRINO660
000025     535  FORMAT (16H1 POINT STRESSES//7H  POINT,4X,5HX-ORD,5X,5HY-ORD,      PRINO670
               1  13X,5HSIG-X,5X,5HSIG-Y,4X,6HTAU-XY,5X,5HSIG-Z,3X,10HEQ. STRESS,  PRINO680
               2    3X,8HPL. WORK,5X,3HDWP,/,1X)                                   PRINO690
000025     538  FORMAT (I6,1X,2F10.4,7X,4F10.4,F11.4,5X,F8.5,3X,F8.5 )             PRINO700
000025     540  FORMAT (I5,3E20.6)                                                 PRINO710
000025     541  FORMAT (I5,4E20.6)                                                 PRINO720
000025     550  FORMAT (1H1,13X,11HMAX. EQUIV.,9X,9HESTIMATED/7H  LAYER,5X         PRINO730
               1  14HPLASTIC STRAIN,7X,12HFATIGUE LIFE,5X,7HAT NODE )              PRINO740
000025     551  FORMAT (I5,E20.5,F15.0,I15)                                        PRINO750
```

```
000025     631  FORMAT ( 6HOPOINT,10X,5HEPS-R,15X,5HEPS-Z,15X,5HEPS-O,15X,6HEPS-RZPRIN0760
               1  / )                                                          PRIN0770
                                                                               PRIN0780
                                                                               PRIN0790
000025          IF (RECCYL .EQ. 0.0)  GO TO 105                                PRIN0800
                                                                               PRIN0810
          C     ***** CYLINDRICAL COORDINATES *****                            PRIN0820
          C     ***** CALCULATE TOTAL FORCE AND TOTAL MOMENTS *****            PRIN0830
000026          CALL FMCALC (B(LNP   ),B(LESIG ),B(LXORD ),B(LYORD ),B(LDIS  )   PRIN0840
               1   ,NUMEL  )                                                   PRIN0850
000051          GO TO 110                                                      PRIN0860
                                                                               PRIN0870
000055     105  IF ( (BLANAL .EQ. 0.0) .OR. (RECCYL .NE. 0.0) )  GO TO 110      PRIN0880
                                                                               PRIN0890
          C     ***** SINGLE FOLD OPTION *****                                 PRIN0900
000063          RADIUS= 0.5*XORD(NPTSY+1)**2 / ( YORD(1) - YORD(NPTSY+1) )     PRIN0910
000070          WRITE (6,166)  RADIUS                                          PRIN0920
                                                                               PRIN0930
000075     110  IF (NNDLIM .EQ. 0)  GO TO 120                                  PRIN0940
                                                                               PRIN0950
          C     ***** CHECK FOR NODAL DISPLACEMENTS EXCEEDING LIMITING         PRIN0960
          C           DISPLACEMENT *****                                       PRIN0970
000102          CALL DISCHK ( B(LDIS),B(LNEBC),B(LDLIM),B(LDLFLG) )            PRIN0980
                                                                               PRIN0990
          C     ***** PRINTOUT X AND Y COMPONENTS OF TOTAL DISPLACEMENTS, TOTAL  PRIN1000
          C           FORCES, AND RESIDUAL FORCES *****                        PRIN1010
000120     120  WRITE(6,162)  ISTEP                                            PRIN1020
                                                                               PRIN1030
000126          IF ( RECCYL .NE. 0.0 )  GO TO 140                              PRIN1040
                                                                               PRIN1050
          C     ***** RECTANGULAR COORDINATES *****                            PRIN1060
000133          WRITE (6,164)                                                  PRIN1070
000136          GO TO 150                                                      PRIN1080
                                                                               PRIN1090
          C     ***** CYLINDRICAL COORDINATES *****                            PRIN1100
000142     140  WRITE (6,165)                                                  PRIN1110
                                                                               PRIN1120
000146     150  WRITE(6,163)  (N,DIS(2*N-1),DIS(2*N),F(2*N-1),F(2*N),DF(2*N-1),  PRIN1130
```

```
                  1   DF(2*N),N=1,NUMNP)

000235              IF (IRESI .LE. 0)  GO TO 200

000242              DO 160 N=1,NEQ
000244              RESI(N)=DF(N)
000250         160  CONTINUE


000253         200  IF ( FLIFE .EQ. 0.0 ) GO TO 300

            C       ***** CALCULATE AND PRINTOUT FATIGUE LIFE OF EACH LAYER *****
000254              WRITE (6,550)
000260              NVADD= 2*NLAYER - 1
000262              PST43= 0.0
000263              LMAT= 1

000264              DO 280 L=1,NLAYER
000271              NV= 2*L - 1
000273              EPMAX= -1.0
000275              DO 270 J=1,NXINT

000276              DO 260 I=1,9
000277              II= I
000301              IF ( I - 7 )  250,220,230
            C       ***** I = 7.  CALCULATE POINT 3 OF ELEMENT NV + 1 *****
000304         220  NV= NV + 1
000306              II= 3
000307              GO TO 250
000307         230  IF ( I .NE. 8 )  GO TO 240
            C       ***** I = 8.  CALCULATE POINT 5 OF ELEMENT NV + 1 *****
000311              II= 5
000312              GO TO 250
            C       ***** I = 9.  CALCULATE POINT 6 OF ELEMENT NV + 1 *****
000313         240  II= 6

000314         250  NODE= NP(NV,II)
000322              PST1= PST(NODE,1)
```

```
000330              PST2= PST(NODE,2)                                                  PRIN1520
000335              IF ( RECCYL .NE. 0.0 )  PST43= PST(NODE,3)                         PRIN1530
000344              EP2= 0.5*( (PST1-PST2)**2 + (PST1-PST43)**2 + (PST2-PST43)**2 )    PRIN1540
                  1  + 3.0*PST(NODE,J34)**2                                            PRIN1550
000365              IF ( EP2 .LE. EPMAX )  GO TO 260                                   PRIN1560
000367              EPMAX= EP2                                                         PRIN1570
000371              NODEMX= NODE                                                       PRIN1580
000372      260     CONTINUE                                                           PRIN1590
                                                                                       PRIN1600
000374              NV= NV + NVADD                                                     PRIN1610
000376      270     CONTINUE                                                           PRIN1620
                                                                                       PRIN1630
000400              IF ( EPMAX .GT. 0.0 )  GO TO 275                                   PRIN1640
000403              FLIFEL= 1.E10                                                      PRIN1650
000404              GO TO 276                                                          PRIN1660
                                                                                       PRIN1670
000404      275     EPMAX= SQRT(EPMAX)                                                 PRIN1680
000407              IF ( NMAT .GT. 1 )  LMAT= MLAYER(L)                                PRIN1690
000420              FLIFEL= ( EPMAX/CYCM(LMAT) )**(1.C/CYCZ(LMAT) )                    PRIN1700
000431      276     WRITE (6,551)  L,EPMAX,FLIFEL,NODEMX                               PRIN1710
000445      280     CONTINUE                                                           PRIN1720
                                                                                       PRIN1730
000453      300     IF ( ERRFLG .EQ. 0.0 )  GO TO 310                                  PRIN1740
                                                                                       PRIN1750
000454              IF (ISTEP .NE. NPRIN)  GO TO 400                                   PRIN1760
                                                                                       PRIN1770
000456              NPRIN=NPRIN+IPRIN                                                  PRIN1780
                                                                                       PRIN1790
          C         ***** PRINTOUT STRAINS, STRESSES, AND PLASTIC WORK FOR EACH POINT.PRIN1800
                                                                                       PRIN1810
000460      310     WRITE(6,530)                                                       PRIN1820
                                                                                       PRIN1830
000464              IF ( RECCYL .NE. 0.0 )  GO TO 320                                  PRIN1840
                                                                                       PRIN1850
          C         ***** RECTANGULAR COORDINATES *****                                PRIN1860
000471              WRITE (6,531)                                                      PRIN1870
000474              WRITE (6,540)  ( N,(EST(N,I),I=1,3),N=1,NUMNP )                    PRIN1880
000526              WRITE (6,532)                                                      PRIN1890
```

```
000532          WRITE (6,531)
000536          WRITE (6,540)  ( N,(PST(N,I),I=1,3),N=1,NUMNP )
000567          WRITE (6,533)
000573          WRITE (6,531)
000577          WRITE (6,540)  ( N,(TST(N,I),I=1,3),N=1,NUMNP )
000630          WRITE(6,535)
000634          GO TO 350

          C     ***** CYLINDRICAL COORDINATES *****
000640     320  WRITE (6,631)
000644          WRITE (6,541)  ( N,(EST(N,I),I=1,4),N=1,NUMNP )
000676          WRITE (6,532)
000702          WRITE (6,631)
000706          WRITE (6,541)  ( N,(PST(N,I),I=1,4),N=1,NUMNP )
000737          WRITE (6,533)
000743          WRITE (6,631)
000747          WRITE (6,541)  ( N,(TST(N,I),I=1,4),N=1,NUMNP )
001000          WRITE (6,534)

001004     350  WRITE(6,538) (N,XORD(N),YORD(N),(SIG(N,I),I=1,5),WP(N),DWP(N),
               1   N=1,NUMNP)


          C     ***** IF ERRFLG .EQ. 0.0, DO NOT BOTHER TO SAVE RESTART DATA *****
001066     400  IF (ERRFLG .EQ. 0.0)  GO TO 900
001073          IF ( SAVETP .EQ. 0.0 )  GO TO 900

001074          IF ( ISTEP .EQ. LSTEPO )  GO TO 420

001076          IF ( ISTEP .LT. NXTLS )  GO TO 900

          C     ***** ISTEP = NXTLS. TIME TO SAVE RESTART DATA *****
          C     ***** WRITE  THE FOLLOWING ARRAYS ON THE RESTART TAPE ...
          C           BETA, ETST, EPST, EEST, ESIG, EWP, XORD, YORD, DIS, F, RESI,
          C           NEBC, DLFLAG, AND TPR *****
001100          WRITE (6,161)  IDUMP,ISTEP,TPRESS
001112          LO= LVPR - 1
001114          L1= LDWP - 1
```

```
001116          L2= LTST - 1                                                     PRIN2280
001120          L3= LNEBC + NEQBC                                                PRIN2290
001122          L4= LVEM - 1                                                     PRIN2300
001124          L5= LDPR - 1                                                     PRIN2310
001126          WRITE (IRSOUT)  IDUMP,ISTEP,TPRESS,NPRIN,NEQBC,DPRESS,RSFOLD     PRIN2320
               1  ,( B(I),I=LBETA,L0)                                            PRIN2330
               1  ,( B(I),I=LETST,L1 ), ( B(I),I=LDIS,L2 ), ( B(I),I=LNEBC,L3 )  PRIN2340
001226          IF ( NNDLIM .GT. 0 )  WRITE (IRSOUT)        ( B(I),I=LDLFLG,L4 ) PRIN2350
001255          IF ( VPRFLG .NE. 0.0 )  WRITE (IRSOUT)  ( B(I),I=LTPR,L5 )       PRIN2360
                                                                                 PRIN2370
001300          IDUMP= IDUMP + 1                                                 PRIN2380
                                                                                 PRIN2390
001302          IF ( ISTEP .LT. LSLIM )  GO TO 440                               PRIN2400
                                                                                 PRIN2410
           C    ***** ISTEP .GE. LSLIM. TIME TO CHANGE INTERVALS BETWEEN DUMPS OF PRIN2420
           C          RESTART DATA *****                                         PRIN2430
001304     410  ILSS= ILSS + 1                                                   PRIN2440
001306          IF ( ILSS .LE. 10 )  GO TO 420                                   PRIN2450
                                                                                 PRIN2460
           C    ***** ILSS .GT. 10. USE CURRENT VALUE OF INDMP FOR REST OF RUN.  PRIN2470
001310          LSLIM= MAXST                                                     PRIN2480
001312          GO TO 440                                                        PRIN2490
                                                                                 PRIN2500
           C    ***** ILSS .LE. 10. PICK UP NEXT VALUE OF INDMP IF IT IS NON-ZERO.PRIN2510
001312     420  LSLIM= LSSAVE(1,ILSS)                                            PRIN2520
001315          IF ( LSLIM .LE. ISTEP )  GO TO 410                               PRIN2530
001320          ITEMP= LSSAVE(2,ILSS)                                            PRIN2540
001323          IF ( ITEMP .GT. 0 )  INDMP= ITEMP                                PRIN2550
                                                                                 PRIN2560
001326     440  NXTLS= MINO( NXTLS+INDMP, LSLIM, MAXST )                         PRIN2570
                                                                                 PRIN2580
001334     900  RETURN                                                           PRIN2590
001335          END                                                              PRIN2600
```

```
RUN 2.3I  05/28/70          PRINT


SUBPROGRAM LENGTH
001643

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
105    -  000055     110    -  000075     120    -  000120     140    -  000142
150    -  000146     161    -  001343     162    -  001356     163    -  001365
164    -  001371     165    -  001405     166    -  001421     200    -  000253
220    -  000304     230    -  000307     240    -  000313     250    -  000314
260    -  000372     275    -  000404     276    -  000431     300    -  000453
310    -  000460     320    -  000640     350    -  001004     400    -  001066
410    -  001304     420    -  001312     440    -  001326     530    -  001426
531    -  001435     532    -  001444     533    -  001451     534    -  001455
535    -  001476     538    -  001517     540    -  001525     541    -  001530
550    -  001533     551    -  001550     631    -  001554     900    -  001334

BLOCK NAMES AND LENGTHS
       -  000001     CDIMEN -  000013     IDIMEN -  000022     IFLAGS -  000012
INDEX  -  000026     IOUNIT -  000002     LBUCKT -  000061     LSSAVE -  000024
MESHIN -  000011     PROPS  -  000012

VARIABLE ASSIGNMENTS
B      -  000000C01  BLANAL -  000010C04  CYCM   -  000010     CYCZ   -  000011
DF     -  000004     DIS    -  000002     DPRESS -  000001C12  EPMAX  -  001622
EP2    -  001631     ERRFLG -  000007C05  EST    -  000000     F      -  000003
FLIFE  -  000011C04  FLIFEL -  001633     I      -  001624     IDUMP  -  000023C05
II     -  001625     ILSS   -  000024C05  INDMP  -  000021C05  IPRIN  -  000004C04
IRESI  -  000006C04  IRSOUT -  000001C06  ISTEP  -  000002C05  ITEMP  -  001642
J      -  001623     J34    -  000010C05  L      -  001621     LBETA  -  000006C07
LDIS   -  000024C07  LDLFLG -  000036C07  LDLIM  -  000035C07  LDPR   -  000057C07
LDWP   -  000017C07  LESIG  -  000013C07  LETST  -  000010C07  LMAT   -  000006C05
LNEBC  -  000003C07  LNP    -  000000C07  LSLIM  -  000022C05  LSSAVE -  000000C10
LSTEPO -  000016C05  LTPR   -  000056C07  LTST   -  000027C07  LVEM   -  000037C07
LVPR   -  000007C07  LXORD  -  000015C07  LYORD  -  000016C07  L0     -  001634
L1     -  001635     L2     -  001636     L3     -  001637     L4     -  001640
```

```
RUN 2.3I  05/28/70          PRINT

L5     -  001641    MAXST  -  000012C03 MLAYER -  000007    N      -  001616
NEQ    -  000002C02 NEQBC  -  000003C02 NLAYER -  000015C03 NMAT   -  000007C03
NNDLIM -  000013C03 NNUMEL -  000013    NNUMNP -  000012    NODE   -  001626
NODEMX -  001632    NP     -  000006    NPRIN  -  000017C05 NPTSY  -  000010C02
NUMEL  -  000001C03 NUMNP  -  000002C03 NV     -  000001C05 NVADD  -  001617
NXINT  -  000020C03 NXTLS  -  000020C05 PST1   -  001627    PST2   -  001630
PST43  -  001620    RADIUS -  001615    RECCYL -  000002C04 RESI   -  000005
RSFOLD -  000005C11 SAVETP -  000007C04 SIG    -  000001    TPRESS -  000010C12
VPRFLG -  000025C05

START OF CONSTANTS
001337

START OF TEMPORARIES
001572

START OF INDIRECTS
001610

EXTERNAL REFERENCES
         FMCALC    OUTPTC    DISCHK    SQRT      RBAREX    OUTPTB    END

COMPILER SPACE
UNUSED -  006200    USED   -  041600
```

```
              SUBROUTINE RESID (NP     ,ESIG  ,XORD  ,YORD  ,DF    ,F     ,NEBC
             1   ,ND    ,MATEL ,VTH    ,NNUMEL  )

        C          CALCULATES THE TOTAL ERROR IN NODAL POINT EQUILIBRIUM BY
        C          CALCULATING THE RESIDUAL NODAL POINT FORCES *****

        C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.
000016        DIMENSION    NP     (NNUMEL,6)    ,ESIG  (NNUMEL,6,5)
000016        DIMENSION    XORD  (1),YORD  (1),DF     (1),F      (1),NEBC  (1)
             1  ,ND     (1),MATEL (1),VTH    (1)

000016        COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ    ,NEQBC ,IBAND ,LENA  ,NEL6
             1   ,NEL64 ,NPTSY ,NPTSX ,NCYCF2
000016        COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID
             1   ,NMAT  ,NDP    ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGIN
             2   ,NXINT ,NCYCF
000016        COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI
             1   ,SAVETP,BLANAL,FLIFE
000016        COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT
             1   ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN
             2   ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG
000016        COMMON  /IPERM/  IPERM(3)
000016        COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP
             1   ,AREA  ,TPRESS,RMAXS
000016        COMMON  /ST    /  ST(12,12)    ,CF(12)
000016        COMMON  /XYCOOR/ R1,R2,R3,Z1,Z2,Z3

000016        DIMENSION          BA(3,2),W1(3,6),W2(3,6)

000016        DIMENSION  RR(3), ZZ(3)
000016        EQUIVALENCE  (RR,R1) , (ZZ,Z1)

000016        DATA W1/ 2.,-1.,-1.,-1., 2.,-1.,-1.,-1., 2.,
             1         3., 3.,-1.,-1., 3., 3., 3.,-1., 3./
000016        DATA W2/ 2.,-1.,-1.,-1., 2.,-1.,-1.,-1., 2.,
             1         8., 8., 4., 4., 8., 8., 8., 4., 8./
```

```
                                                                              RESI0380
         C      ***** CLEAR RESIDUAL FORCES (DF) *****                        RESI0390
000016          DO 150 I=1,NEQ                                                RESI0400
                                                                              RESI0420
         C      ***** ELEMENT LOOP *****                                      RESI0430
                                                                              RESI0440
000017   150    DF(I)=0.                                                      RESI0410
000023          DO 100 NV=1,NUMEL                                             RESI0450
                                                                              RESI0460
                                                                              RESI0480
         C      ***** IF THERE IS MORE THAN ONE MATERIAL, PICK UP THE THICKNESS  RESI0490
         C            OF ELEMENT NV *****                                     RESI0500
000025          IF (NMAT .EQ. 1)  GO TO 90                                    RESI0470
000027          IMAT= MATEL(NV)                                               RESI0510
000032          TH= VTH(IMAT)                                                 RESI0520
                                                                              RESI0530
000035    90    DO 110 I=1,3                                                  RESI0540
000037          K1=NP(NV,I)                                                   RESI0550
000045          RR(I)= XORD(K1)                                               RESI0560
000050          J=IPERM(I)                                                    RESI0570
000053          K2=NP(NV,J)                                                   RESI0580
000060          M=IPERM(J)                                                    RESI0590
000063          BA(M,2)=XORD(K2)-XORD(K1)                                     RESI0600
000070   110    BA(M,1)=YORD(K1)-YORD(K2)                                     RESI0610
                                                                              RESI0620
000077          AREA=BA(3,2)*BA(2,1)-BA(2,2)*BA(3,1)                          RESI0630
                                                                              RESI0640
000103          DO 120 I=1,12                                                 RESI0650
000105   120    CF(I)=0.                                                      RESI0660
                                                                              RESI0670
000111          IF ( RECCYL .EQ. 0.0 )  GO TO 125                             RESI0680
                                                                              RESI0690
         C      ***** CYLINDRICAL COORDINATES *****                           RESI0700
000112          W1(1,1)=18.*R1+3.*R2+3.*R3                                    RESI0710
000120          W1(1,2)=-5.*R1-6.*R2-R3                                       RESI0720
000125          W1(1,3)=-5.*R1-R2-6.*R3                                       RESI0730
000132          W1(1,4)=24.*R1+8.*R2+4.*R3                                    RESI0740
000140          W1(1,5)=4.*R1-8.*R2-8.*R3                                     RESI0750
```

```
000145          W1(1,6)=24.*R1+4.*R2+8.*R3                                   RESI0760
000153          W1(2,1)=-6.*R1-5.*R2-R3                                      RESI0770
000160          W1(2,2)=3.*R1+18.*R2+3.*R3                                   RESI0780
000165          W1(2,3)=-R1-5.*R2-6.*R3                                      RESI0790
000172          W1(2,4)=8.*R1+24.*R2+4.*R3                                   RESI0800
000200          W1(2,5)=4.*R1+24.*R2+8.*R3                                   RESI0810
000206          W1(2,6)=-8.*R1+4.*R2-8.*R3                                   RESI0820
000214          W1(3,1)=-6.*R1-R2-5.*R3                                      RESI0830
000221          W1(3,2)=-R1-6.*R2-5.*R3                                      RESI0840
000226          W1(3,3)=3.*R1+3.*R2+18.*R3                                   RESI0850
000233          W1(3,4)=-8.*R1-8.*R2+4.*R3                                   RESI0860
000241          W1(3,5)=4.*R1+8.*R2+24.*R3                                   RESI0870
000247          W1(3,6)=8.*R1+4.*R2+24.*R3                                   RESI0880
000255          W2(1,1)=24.*R1                                               RESI0890
000257          W2(1,2)=-8.*R1-4.*R3                                         RESI0900
000262          W2(1,3)=-8.*R1-4.*R2                                         RESI0910
000266          W2(1,4)=48.*R1+32.*R2+16.*R3                                 RESI0920
000274          W2(1,5)=16.*(R1+R2+R3)                                       RESI0930
000300          W2(1,6)=48.*R1+16.*R2+32.*R3                                 RESI0940
000306          W2(2,1)=-8.*R2-4.*R3                                         RESI0950
000311          W2(2,2)=24.*R2                                               RESI0960
000313          W2(2,3)=-4.*R1-8.*R2                                         RESI0970
000317          W2(2,4)=32.*R1+48.*R2+16.*R3                                 RESI0980
000325          W2(2,5)=16.*R1+48.*R2+32.*R3                                 RESI0990
000333          W2(2,6)=16.*(R1+R2+R3)                                       RESI1000
000337          W2(3,1)=-4.*R2-8.*R3                                         RESI1010
000342          W2(3,2)=-4.*R1-8.*R3                                         RESI1020
000346          W2(3,3)=24.*R3                                               RESI1030
000350          W2(3,4)=16.*(R1+R2+R3)                                       RESI1040
000354          W2(3,5)=16.*R1+32.*R2+48.*R3                                 RESI1050
000362          W2(3,6)=32.*R1+16.*R2+48.*R3                                 RESI1060
                                                                             RESI1070
           C    ***** LOOP FOR POINTS WITHIN ELEMENT NV *****                RESI1080
                                                                             RESI1090
           C    ***** CALCULATE RESIDUAL FORCES FOR ELEMENT NV *****         RESI1100
000370      125 DO 130 I=1,6                                                 RESI1110
000372          ESIG1= ESIG(NV,I,1)                                          RESI1120
000404          ESIG2= ESIG(NV,I,2)                                          RESI1130
```

```
000415          ESIG34= ESIG(NV,I,J34)
000425          W11I= W1(1,I)
000430          W12I= W1(2,I)
000434          W13I= W1(3,I)
000437          W21I= W2(1,I)
000443          W22I= W2(2,I)
000446          W23I= W2(3,I)
000452          TERM1= BA(1,1)*ESIG1  + BA(1,2)*ESIG34
000455          TERM2= BA(1,1)*ESIG34 + BA(1,2)*ESIG2
000461          TERM3= BA(2,1)*ESIG1  + BA(2,2)*ESIG34
000465          TERM4= BA(2,1)*ESIG34 + BA(2,2)*ESIG2
000471          TERM5= BA(3,1)*ESIG1  + BA(3,2)*ESIG34
000475          TERM6= BA(3,1)*ESIG34 + BA(3,2)*ESIG2
000501          CF(1)= CF(1) + W11I*TERM1
000504          CF(2)= CF(2) + W11I*TERM2
000507          CF(3)= CF(3) + W12I*TERM3
000512          CF(4)= CF(4) + W12I*TERM4
000515          CF(5)= CF(5) + W13I*TERM5
000520          CF(6)= CF(6) + W13I*TERM6
000523          CF(7)= CF(7) + W21I*TERM3 + W22I*TERM1
000530          CF(8)= CF(8) + W21I*TERM4 + W22I*TERM2
000535          CF(9)= CF(9) + W22I*TERM5 + W23I*TERM3
000542          CF(10)= CF(10) + W22I*TERM6 + W23I*TERM4
000547          CF(11)= CF(11) + W23I*TERM1 + W21I*TERM5
000554     130  CF(12)= CF(12) + W23I*TERM2 + W21I*TERM6

000563          CON1= 30.0

000565          IF ( RECCYL .EQ. 0.0 )  GO TO 135

          C     ***** CYLINDRICAL COORDINATES *****
000566          CON1= 360.0

000567          DO 200 I=1,3
000571          K=NP(NV,I)
000577          DO 200 J=1,6
000600    200   CF(2*I-1)=CF(2*I-1)+AREA*W2(I,J)*ESIG(NV,J,3)/(4.*XORD(K))
```

```
000635    135  DO 140 I=1,12
000637   140   CF(I)=-TH*CF(I)/CON1

         C     ***** ACCUMULATE THE RESIDUAL FORCES FOR EACH NODAL POINT *****
000645         DO 160 I=1,6
000646         L=NP(NV,I)
000654         DF(2*L-1)=DF(2*L-1)+CF(2*I-1)
000661   160   DF(2*L)=DF(2*L)+CF(2*I)

000667   100   CONTINUE


000672         DO 560 M=1,NEQBC
000673         NR=NEBC(M)
000676         F(NR)=-DF(NR)
000701         IF (NLOAD .LT. 0)  DF(NR)= 0.
000705    560  CONTINUE
000710         IF (NLOAD .LT. 0)  GO TO 900

000711         IF (NID .LE. 0) GO TO 700

000713         DO 660 M=1,NID
000714         NR=ND(M)
000717   660   F(NR)=-DF(NR)

         C     ***** ADD THE ACCUMLATED NODAL FORCES (F) TO THE RESIDUAL FORCES.
000725    700  DO 170 I=1,NEQ
000727   170   DF(I)=DF(I)+F(I)
000735    900  RETURN
000736         END
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
Z1
```

```
RUN 2.3I  05/28/70          RESID


SUBPROGRAM LENGTH
001111

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
90     -  000035    125    -  000370    135    -  000635    200    -  000600
560    -  000705    700    -  000725    900    -  000735

BLOCK NAMES AND LENGTHS
CDIMEN -  000013    IDIMEN -  000022    IFLAGS -  000012    INDEX  -  000026
IPERM  -  000003    PROPS  -  000012    ST     -  000234    XYCOOR -  000006

VARIABLE ASSIGNMENTS
AREA   -  000007C06 BA     -  001007    CF     -  000220C07 CON1   -  001105
ESIG1  -  001066    ESIG2  -  001067    ESIG34 -  001070    I      -  001061
IMAT   -  000005C04 IPERM  -  000000C05 J      -  001063    J34    -  000010C04
K      -  001106    K1     -  001062    K2     -  001064    L      -  001107
M      -  001065    MATEL  -  000002    ND     -  000001    NEBC   -  000000
NEQ    -  000002C01 NEQBC  -  000003C01 NID    -  000006C02 NLOAD  -  000005C02
NMAT   -  000007C02 NNUMEL -  000004    NR     -  001110    NUMEL  -  000001C02
NV     -  000001C04 RECCYL -  000002C03 RR     -  000000C10 R1     -  000000C10
R2     -  000001C10 R3     -  000002C10 ST     -  000000C07 TERM1  -  001077
TERM2  -  001100    TERM3  -  001101    TERM4  -  001102    TERM5  -  001103
TERM6  -  001104    TH     -  000004C06 VTH    -  000003    W1     -  001015
W11I   -  001071    W12I   -  001072    W13I   -  001073    W2     -  001037
W21I   -  001074    W22I   -  001075    W23I   -  001076    ZZ     -  000003C10

START OF CONSTANTS
000740

START OF TEMPORARIES
000763

START OF INDIRECTS
000773
```

```
RUN 2.3I  05/28/70            RESID

EXTERNAL REFERENCES
         END

COMPILER SPACE
UNUSED -  010100    USED   -  037700
```

```
                    SUBROUTINE RRST23 (NP     ,LOADE ,LOADS ,DLIM   ,VEM    ,VXU    ,VTH   RRST0010
                   1     ,VXI   ,VYP    ,DELTAP,MATEL ,DISPL ,XORD   ,YORD   ,DIS    ,F     RRST0020
                   2     ,RESI  ,NEBC   ,DLFLAG,BETA  ,ETST  ,EPST   ,EEST   ,ESIG   ,EWP   RRST0030
                   3     ,MLAYER,CYCM   ,CYCZ  ,DEPTH ,VPR   ,TPR    ,ND     ,DSD           RRST0040
                   4     ,NNUMEL )                                                         RRST0050
                                                                                           RRST0060
              C            READS THE SECOND CATEGORY OF RESTART DATA ( INPUT ARRAYS )      RRST0070
              C            AND THE THIRD CATEGORY ( LOAD STEP DEDENDENT QUANTITIES )       RRST0080
                                                                                           RRST0090
              C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.RRST0100
000045              DIMENSION    NP  (6,1),LOADE (1),LOADS (1),DLIM  (1),VEM   (1)     RRST0110
                   1  ,VXU   (1),VTH   (1),VXI   (1),VYP   (1),DELTAP(1),MATEL (1)     RRST0120
                   2  ,DISPL (1),XORD  (1),YORD  (1),DIS   (1),F     (1),RESI  (1)     RRST0130
                   3  ,NEBC  (1),DLFLAG(1),MLAYER(1),CYCM  (1),CYCZ  (1),DEPTH (1)     RRST0140
                   4  ,VPR   (1),TPR   (1),ND    (1),DSD   (1)                         RRST0150
000045              DIMENSION    BETA(NNUMEL,6)  ,ETST(NNUMEL,6,4),EPST(NNUMEL,6,4)    RRST0160
                   1             ,EEST(NNUMEL,6,4),ESIG(NNUMEL,6,5),EWP (NNUMEL,6)     RRST0170
                                                                                       RRST0180
000045              COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6   RRST0190
                   1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                  RRST0200
000045              COMMON  /CMESH /  BLTHNS,XINTN ,XYRAT ,DPRAT                       RRST0210
000045              COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID    RRST0220
                   1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINRRST0230
                   2    ,NXINT ,NCYCF                                                 RRST0240
000045              COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI  RRST0250
                   1    ,SAVETP,BLANAL,FLIFE                                          RRST0260
000045              COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT   RRST0270
                   1    ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN RRST0280
                   2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                    RRST0290
000045              COMMON  /IOUNIT/  IRSIN,IRSOUT                                    RRST0300
000045              COMMON  /LSSAVE/  LSSAVE(2,10)                                    RRST0310
000045              COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATRRST0320
                   1    ,RSTRAT,RFINAL                                                RRST0330
000045              COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP     RRST0340
                   1    ,AREA  ,TPRESS,RMAXS                                          RRST0350
000045              COMMON  /RSDATA/  RSDATA(36)                                      RRST0360
000045              DIMENSION  IRS(36),RS(36)                                         RRST0370
```

```
000045          EQUIVALENCE    ( IRS,RSDATA( 1) )                                      RRST0380
000045          EQUIVALENCE    ( RS,INUMEL,RSDATA( 1) ), ( INUMNP,RS( 2) )             RRST0390
               1  ,( INLOAD,RS( 5) ), ( INID  ,RS( 6) ), ( INMAT ,RS( 7) )             RRST0400
               2  ,( INDP  ,RS( 8) ), ( INDLIM,RS(11) )                                RRST0410
                                                                                       RRST0420
                                                                                       RRST0430
000045          NLAYER= IRS(13)                                                        RRST0440
                                                                                       RRST0450
000046          READ (IRSIN) INEQ,VPRFL ,DPRAT,XLNGTH,BLTHNS,NPTSY,LSSAVE              RRST0460
                                                                                       RRST0470
000067          IF ( VPRFL .EQ. VPRFLG )  GO TO 200                                    RRST0480
000075          WRITE (6,11)  VPRFL , VPRFLG                                           RRST0490
000104     11   FORMAT (36H0ERROR... VPRFLG FROM RESTART TAPE =F5.1,5X,                RRST0500
               1  28HPROGRAM ALREADY SET VPRFLG =F5.1 )                                RRST0510
000104          STOP                                                                   RRST0520
                                                                                       RRST0530
          C     ***** READ NP ARRAY IN TRANSPOSE ORDER FOR USE BY SUBROUTINE SETUPRRST0540
000106    200   READ (IRSIN)  (  ( NP(I,J),J=1,INUMEL ), I=1,6  )                      RRST0550
                                                                                       RRST0560
000133    300   IF ( INLOAD .LE. 0 )  GO TO 400                                        RRST0570
000140          READ (IRSIN)  ( LOADE(I),I=1,INLOAD ), ( LOADS(I),I=1,INLOAD )         RRST0580
000166          IF ( VPRFLG .NE. 0.0 )  READ (IRSIN)  ( VPR(I),I=1,INLOAD )            RRST0590
                                                                                       RRST0600
000215    400   IF ( INDLIM .LE. 0 )  GO TO 500                                        RRST0610
000217          NNDL3= 3*INDLIM                                                        RRST0620
000220          READ (IRSIN)  ( DLIM(I  ),I=1,NNDL3 )                                  RRST0630
                                                                                       RRST0640
000235    500   READ (IRSIN)  ( VEM(I),I=1,INMAT ), ( VXU(I),I=1,INMAT )               RRST0650
               1  , ( VTH(I),I=1,INMAT ),( VXI(I),I=1,INMAT ),( VYP(I),I=1,INMAT )RRST0660
                                                                                       RRST0670
000324    600   IF ( INDP .LE. 0 )  GO TO 700                                          RRST0680
000331          INDP2= 2*INDP                                                          RRST0690
000333          READ (IRSIN)  ( DELTAP(I  ),I=1,INDP2 )                                RRST0700
                                                                                       RRST0710
000351    700   IF ( INMAT .LE. 1 )  GO TO 800                                         RRST0720
000357          READ (IRSIN)  ( MATEL(I),I=1,INUMEL )                                  RRST0730
                                                                                       RRST0740
000375    800   IF ( INID .LE. 0 )  GO TO 900                                          RRST0750
```

```
RUN 2.3I  05/28/70          RRST23

000402          READ (IRSIN)  ( ND(I),I=1,INID ), ( DSD(I),I=1,INID )             RRST0760
                                                                                  RRST0770
000432     900  IF ( BLANAL .EQ. 0.0 )  GO TO 200C                                RRST0780
                READ (IRSIN)  ( DEPTH(I),I=1,NLAYER )                             RRST0790
000437          IF ( FLIFE .NE. 0.0 )   READ (IRSIN)  ( MLAYER(I),I=1,NLAYER )    RRST0800
000455         1  ,( CYCM(I),I=1,INMAT ), ( CYCZ(I),I=1,INMAT )                   RRST0810
                                                                                  RRST0820
                                                                                  RRST0830
          C     ***** READ LOAD STEP DEPENDENT QUANTITIES FROM RESTART TAPE ***** RRST0840
                                                                                  RRST0850
          C     ***** SKIP OVER TO THE PROPER DUMP ON THE RESTART TAPE *****      RRST0860
000530    2000  NSKIP= RESTRT - 1.0                                               RRST0870
000533          IF ( NSKIP .EQ. 0 )  GO TO 2200                                   RRST0880
000534          MSKIP= NSKIP                                                      RRST0890
000536          IF ( INDLIM .GT. 0 )  NSKIP= NSKIP + NSKIP                        RRST0900
000540          IF ( VPRFLG .NE. 0.0 )   NSKIP= NSKIP + MSKIP                     RRST0910
000543          WRITE (6,10)  NSKIP                                               RRST0920
000551      10  FORMAT (48HONO. OF RECORDS OF INPUT RESTART TAPE SKIPPED  =I5 )   RRST0930
000551          DO 2100 I=1,NSKIP                                                 RRST0940
000556          READ (IRSIN)                                                      RRST0950
000561    2100  CONTINUE                                                          RRST0960
                                                                                  RRST0970
000567    2200  READ (IRSIN)  JDUMP,LSTEPO,TPRESS,NPRN ,NEQBC,DPRESS,RSFOLD        RRST0980
               1  , (  ( BETA(I,J),I=1,INUMEL ), J=1,6  )                         RRST0990
               2  , (  ( ( ETST(I,J,K),I=1,INUMEL ) ,J=1,6 ) , K=1,J34  )         RRST1000
               3  , (  ( ( EPST(I,J,K),I=1,INUMEL ) ,J=1,6 ) , K=1,J34  )         RRST1010
               4  , (  ( ( EEST(I,J,K),I=1,INUMEL ) ,J=1,6 ) , K=1,J34  )         RRST1020
               5  , (  ( ( ESIG(I,J,K),I=1,INUMEL ) ,J=1,6 ) , K=1,5  )           RRST1030
               6  , (  ( EWP(I,J),I=1,INUMEL ) ,J=1,6  )                          RRST1040
               7  , ( XORD(I),I=1,INUMNP )                                        RRST1050
               8  , ( YORD(I),I=1,INUMNP )                                        RRST1060
               9  , ( DIS(I),I=1,INEQ )                                           RRST1070
               1  , ( F(I),I=1,INEQ )                                             RRST1080
               2  , ( RESI(I),I=1,INEQ )                                          RRST1090
               3  , ( NEBC(I),I=1,NEQBC )                                         RRST1100
                                                                                  RRST1110
001110          IF ( INDLIM .GT. 0 )  READ (IRSIN)  ( DLFLAG(I),I=1,INDLIM )      RRST1120
                                                                                  RRST1130
```

```
001137         IF ( VPRFLG .NE. 0.0 )  READ (IRSIN)  ( TPR(I),I=1,INLOAD )        RRST1140
                                                                                  RRST1150
001162         RMAXS=MAXST - LSTEPO                                               RRST1160
001165         NPRIN= LSTEPO + IPRIN                                              RRST1170
001167         LSTEPO= LSTEPO + 1                                                 RRST1180
001170         RETURN                                                             RRST1190
001171          END                                                               RRST1200
```

```
SUBPROGRAM LENGTH
001276

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10       -  001216     11       -  001201     200      -  000106     300      -  000133
400      -  000215     500      -  000235     600      -  000324     700      -  000351
800      -  000375     900      -  000432     2000     -  000530     2200     -  000567

BLOCK NAMES AND LENGTHS
CDIMEN   -  000013     CMESH    -  000004     IDIMEN   -  000022     IFLAGS   -  000012
INDEX    -  000026     IOUNIT   -  000002     LSSAVE   -  000024     MESHIN   -  000011
PROPS    -  000012     RSDATA   -  000044

VARIABLE ASSIGNMENTS
BETA     -  000015     BLANAL   -  000010C04  BLTHNS   -  000000C02  CYCM     -  000024
CYCZ     -  000025     DELTAP   -  000003     DEPTH    -  000026     DIS      -  000010
DISPL    -  000005     DLFLAG   -  000014     DPRAT    -  000003C02  DPRESS   -  000001C11
DSD      -  000032     EEST     -  000020     EPST     -  000017     ESIG     -  000021
ETST     -  000016     EWP      -  000022     F        -  000011     FLIFE    -  000011C04
I        -  001265     INDLIM   -  000012C12  INDP     -  000007C12  INDP2    -  001270
INEQ     -  001263     INID     -  000005C12  INLOAD   -  000004C12  INMAT    -  000006C12
INUMEL   -  000000C12  INUMNP   -  000001C12  IPRIN    -  000004C04  IRS      -  000000C12
IRSIN    -  000000C06  J        -  001266     JDUMP    -  001273     J34      -  000010C05
K        -  001275     LSSAVE   -  000000C07  LSTEPO   -  000016C05  MATEL    -  000004
MAXST    -  000012C03  MLAYER   -  000023     MSKIP    -  001272     ND       -  000031
NEBC     -  000013     NEQBC    -  000003C01  NLAYER   -  000015C03  NNDL3    -  001267
NNUMEL   -  000033     NPRIN    -  000017C05  NPRN     -  001274     NPTSY    -  000010C01
NSKIP    -  001271     RESI     -  000012     RESTRT   -  000001C04  RMAXS    -  000011C11
RS       -  000000C12  RSDATA   -  000000C12  RSFOLD   -  000005C10  TPR      -  000030
TPRESS   -  000010C11  VPR      -  000027     VPRFL    -  001264     VPRFLG   -  000025C05
VTH      -  000000     VXI      -  000001     VYP      -  000002     XLNGTH   -  000003C10
XORD     -  000006     YORD     -  000007

START OF CONSTANTS
```

```
RUN 2.3I  05/28/70          RRST23

001173

START OF TEMPORARIES
001225

START OF INDIRECTS
001263

EXTERNAL REFERENCES
         INPUTB    OUTPTC    STOP      END

COMPILER SPACE
UNUSED -  007400    USED   -  040400
```

```
                 SUBROUTINE SETIC (NP     ,BETA   ,ESIG   ,       TST    ,PST    ,EST    SETI0010
                1    ,DTST   ,SIG    ,ETST   ,EPST   ,EEST   ,EWP    ,XORD   ,YORD   ,DWP    SETI0020
                2    ,WP     ,COUNT  ,DR     ,DIS    ,F      ,VEM    ,VXU    ,VXI            SETI0030
                3    ,VYP    ,MATEL  ,THETA  ,NEBC   ,NNUMEL,NNUMNP )                        SETI0040
                                                                                             SETI0050
          C           CALCULATES THE INCREMENTAL AND TOTAL STRESSES AND STRAINS              SETI0060
                                                                                             SETI0070
          C      ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.SETI0080
000040           DIMENSION    NP  (NNUMEL,6)  ,BETA(NNUMEL,6)  ,ESIG(NNUMEL,6,5)     SETI0090
                1                             ,TST (NNUMNP,4)  ,PST (NNUMNP,4)       SETI0100
                2               ,EST (NNUMNP,4)  ,DTST(NNUMNP,5)  ,SIG (NNUMNP,5)    SETI0110
                3               ,ETST(NNUMEL,6,4),EPST(NNUMEL,6,4),EEST(NNUMEL,6,4)  SETI0120
                4               ,EWP (NNUMEL,6)  ,THETA(NNUMEL,3)                    SETI0130
000040           DIMENSION    XORD  (1),YORD  (1),DWP   (1),WP    (1),COUNT (1)      SETI0140
                1  ,DR    (1),DIS   (1),F     (1),          VEM   (1),VXU   (1)      SETI0150
                2  ,VXI   (1),VYP   (1),MATEL (1),NEBC  (1)                          SETI0160
                                                                                     SETI0170
000040           DIMENSION  DETST(6,4),ROT(6),BA(3,2),DEPST(4)                       SETI0180
000040           DIMENSION DEEST(4), S(4), DESIG(4), SIGP(6,4)                       SETI0190
                                                                                     SETI0200
          C      ***** THE INDICATED DIMENSION OF THE ARRAY B IS A DUMMY.            SETI0210
          C            THE ACTUAL DIMENSION NBUCKT IS CALCULATED BY SUBROUTINE BLOCSETI0220
          C            AND THE REQUIRED STORAGE IS ALLOCATED BY SUBROUTINE MRELSE. SETI0230
                                                                                     SETI0240
000040           COMMON  B(1)                                                        SETI0250
                                                                                     SETI0260
000040           COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6    SETI0270
                1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                    SETI0280
000040           COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID     SETI0290
                1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINSETI0300
                2    ,NXINT ,NCYCF                                                   SETI0310
000040           COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI   SETI0320
                1    ,SAVETP,BLANAL,FLIFE                                            SETI0330
000040           COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT    SETI0340
                1    ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN SETI0350
                2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                      SETI0360
000040           COMMON  /IPERM/   IPERM(3)                                          SETI0370
```

```
        C     ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE SETI0380
        C           ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR  SETI0390
        C           IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE   SETI0400
        C           POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. ***** SETI0410
000040        COMMON  /LBUCKT/ LNP    ,LLOADE,LLOADS,LNEBC ,LND    ,LMATEL,LBETA SETI0420
             1   ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP   ,LXORD ,LYORD ,LDWP  SETI0430
             2   ,LWP   ,LCOUNT,LDF   ,LDR   ,LDIS  ,LF     ,LRESI ,LTST  ,LPST  SETI0440
             3   ,LEST  ,LDTST ,LSIG  ,LDSD  ,LDLIM ,LDLFLG,LVEM  ,LVXU  ,LVTH  SETI0450
             4   ,LVXI  ,LVYP  ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYRSETI0460
             5   ,LDEPTH,LCYCF ,LA    ,LTPR  ,LDPR                              SETI0470
             6   ,NBUCKT                                                        SETI0480
000040        COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP     SETI0490
             1   ,AREA  ,TPRESS,RMAXS                                           SETI0500
000040        COMMON  /UV    /  UV(3,6,2)                                       SETI0510
                                                                                SETI0520
000040        ER=EM/(1.-XU-2.*XU*XU)                                            SETI0530
                                                                                SETI0540
        C     ***** CLEAR DWP, WP, AND COUNT ARRAYS *****                       SETI0550
000044        LL= LDF-1                                                         SETI0560
000046        DO 110 I=LDWP,LL                                                  SETI0570
                                                                                SETI0590
        C     ***** CLEAR TST, PST, AND EST ARRAYS *****                        SETI0600
000050   110  B(I)= 0.0                                                         SETI0580
000055        LL= LDTST-1                                                       SETI0610
000057        DO 115 I=LTST,LL                                                  SETI0620
                                                                                SETI0640
        C     ***** CLEAR SIG ARRAY *****                                       SETI0650
000061   115  B(I)= 0.0                                                         SETI0630
000066        LL= LDSD - 1                                                      SETI0660
000070        DO 120 I=LSIG,LL                                                  SETI0670
                                                                                SETI0690
000072   120  B(I)= 0.0                                                         SETI0680
000077        DO 130 N=1,NEQ                                                    SETI0700
                                                                                SETI0720
                                                                                SETI0730
000100   130  DIS(N)=DIS(N)+DR(N)                                               SETI0710
000107        LMAT= 0                                                           SETI0740
                                                                                SETI0750
```

```
          C      ***** ELEMENT LOOP *****                                SETI0760
                                                                         SETI0770
000110           DO 300 NV=1,NUMEL                                       SETI0780
                                                                         SETI0790
                                                                         SETI0810
000111           IF (NMAT .EQ. 1)  GO TO 140                             SETI0800
000113           IMAT= MATEL(NV)                                         SETI0820
000116           IF (IMAT .EQ. LMAT)  GO TO 140                          SETI0830
                                                                         SETI0840
          C      ***** STORE PROPERTIES FOR THE CURRENT ELEMENT *****    SETI0850
000120           EM= VEM(IMAT)                                           SETI0860
000123           XU= VXU(IMAT)                                           SETI0870
000126           XI= VXI(IMAT)                                           SETI0880
000131           YP= VYP(IMAT)                                           SETI0890
000134           ER=EM/(1.-XU-2.*XU*XU)                                  SETI0900
000140           LMAT= IMAT                                              SETI0910
                                                                         SETI0920
000142     140   DO 150 I=1,3                                            SETI0930
000144           DO 151 K=1,J34                                          SETI0940
000145    151    DETST(I,K)=0.                                           SETI0950
000155           ROT(I)=0.                                               SETI0960
000157           K1=NP(NV,I)                                             SETI0970
000164           J=IPERM(I)                                              SETI0980
000167           K2=NP(NV,J)                                             SETI0990
000174           M=IPERM(J)                                              SETI1000
000177           BA(M,2)=XORD(K2)-XORD(K1)                               SETI1010
000204    150    BA(M,1)=YORD(K1)-YORD(K2)                               SETI1020
                                                                         SETI1030
000214           AREA=BA(3,2)*BA(2,1)-BA(2,2)*BA(3,1)                    SETI1040
                                                                         SETI1050
          C      ***** CALCULATE UV MATRIX *****                         SETI1060
000220           DO 160 L=1,3                                            SETI1070
000222           L1=IPERM(L)                                             SETI1080
000225           L2=IPERM(L1)                                            SETI1090
000227           L3=L+3                                                  SETI1100
                                                                         SETI1110
000231           DO 160 N=1,2                                            SETI1120
000233           DO=BA(L,N)/AREA                                         SETI1130
```

```
000240          D1=BA(L1,N)/AREA                                                   SETI1140
000245          UV(L,L,N)=3.*D0                                                    SETI1150
000252          UV(L1,L,N)=-D0                                                     SETI1160
000260          UV(L2,L,N)=-D0                                                     SETI1170
000266          UV(L,L3,N)=4.*D1                                                   SETI1180
000274          UV(L1,L3,N)=4.*D0                                                  SETI1190
000302     160  UV(L2,L3,N)=0.                                                     SETI1200
                                                                                   SETI1210
          C     ***** CALCULATE STRESS DEVIATORS AND STORE TEMPORARILY IN SIGP ***SETI1220
000313          DO 180 J=1,6                                                       SETI1230
000315          X=ESIG(NV,J,1)                                                     SETI1240
000327          Y=ESIG(NV,J,2)                                                     SETI1250
000340          Z=ESIG(NV,J,J43)                                                   SETI1260
000350          COMM=(X+Y+Z)/3.                                                    SETI1270
000354          SIGP(J,1)=X-COMM                                                   SETI1280
000357          SIGP(J,2)=Y-COMM                                                   SETI1290
000362          SIGP(J,J34)= 2.0*ESIG(NV,J,J34)                                    SETI1300
000376          SIGP(J,J43)= Z - COMM                                              SETI1310
                                                                                   SETI1320
000403          K=2*NP(NV,J)                                                       SETI1330
                                                                                   SETI1340
          C     ***** CALCULATE INCREMENTAL TOTAL STRAINS AND ROTATION FOR EACH    SETI1350
          C           CORNER OF ELEMENT NV *****                                   SETI1360
000411          DO 180 I=1,3                                                       SETI1370
000412          UVIJ1= UV(I,J,1)                                                   SETI1380
000416          UVIJ2= UV(I,J,2)                                                   SETI1390
000422          DETST(I,1)= DETST(I,1) + UVIJ1*DR(K-1)                             SETI1400
000427          DETST(I,2)= DETST(I,2) + UVIJ2*DR(K)                               SETI1410
000434          TERM1= UVIJ2*DR(K-1)                                               SETI1420
000440          TERM2= UVIJ1*DR(K)                                                 SETI1430
000443          DETST(I,J34)= DETST(I,J34) + TERM1 + TERM2                         SETI1440
000452     180  ROT(I)= ROT(I) + TERM1 - TERM2                                     SETI1450
                                                                                   SETI1460
000462          IF ( RECCYL .EQ. 0.0 )  GO TO 182                                  SETI1470
                                                                                   SETI1480
          C     ***** CYLINDRICAL COORDINATES *****                                SETI1490
000463          DO 183 I=1,3                                                       SETI1500
000464          J= NP(NV,I)                                                        SETI1510
```

```
000472         TEMP= DR(2*J-1)/XORD(J)
000476         IF (NLOAD .LT. 0)  TEMP= TEMP + THETA(NV,I)
000506    183  DETST(I,3)= TEMP + 0.5*TEMP**2

        C      ***** CALCULATE SECOND ORDER TERMS OF INCREMENT TOTAL STRAINS ****
000515    182  DO 185 I=1,3
000517         UX= DETST(I,1)
000522         VY= DETST(I,2)
000524         UY= 0.5*(DETST(I,J34)+ROT(I) )
000533         VX= 0.5*(DETST(I,J34)-ROT(I) )
000542         DETST(I,1)= UX + 0.5*(UX**2+VX**2)
000550         DETST(I,2)= VY + 0.5*(VY**2+UY**2)
000555    185  DETST(I,J34)= DETST(I,J34) + UX*UY + VX*VY

        C      ***** OBTAIN INCREMENTAL TOTAL STRAINS AND ROTATIONS AT THE
        C            MIDPOINTS BY AVERAGING THE VALUE AT THH PROPER CORNER POINTS
000567    187  DO 190 I=1,3
000571         I1=IPERM(I)
000574         L=I+3
000576         ROT(L)=(ROT(I)+ROT(I1))/2.

000605         DO 190 J=1,J34
000607   190   DETST(L,J)=(DETST(I,J)+DETST(I1,J))/2.


        C      ***** LOOP FOR POINTS WITHIN ELEMENT NV *****

000635         DO 300 I=1,6
000637         N=NP(NV,I)

000645         DO 220 J=1,J34
000646         DEPST(J)=0.
000650   220   DEEST(J)=DETST(I,J)

000660         IF (BETA(NV,I))  250,250,251
000665   251   S(1)=SIGP(I,1)
000670         S(2)=SIGP(I,2)
000672         S(3)=SIGP(I,3)
```

```
000674         S(J34)= 0.5*SIGP(I,J34)                                   SETI1900
                                                                         SETI1910
         C     ***** CALCULATE PLASTIC AND ELASTIC INCREMENTAL STRAINS   SETI1920
         C           (DEPST AND DEEST) *****                             SETI1930
000702         DO 240 J=1,J34                                            SETI1940
000703         DO 230 K=1,J34                                            SETI1950
000704   230   DEPST(J)=DEPST(J)+DETST(I,K)*S(K)*BETA(NV,I)*SIGP(I,J)    SETI1960
000736   240   DEEST(J)=DEEST(J)-DEPST(J)                                SETI1970
                                                                         SETI1980
         C     ***** CALCULATE INCREMENTAL STRESSES (DESIG) *****        SETI1990
000744   250   DESIG(1)=ER*((1.-XU)*DEEST(1)+XU*DEEST(2))                SETI2000
000752         DESIG(2)=ER*(XU*DEEST(1)+(1.-XU)*DEEST(2))                SETI2010
000757         DESIG(J34)= ER*(0.5-XU)*DEEST(J34)                        SETI2020
000764         DESIG(J43)= ER*XU*(DEEST(1)+DEEST(2))                     SETI2030
                                                                         SETI2040
000771         IF ( RECCYL .EQ. 0.0 )  GO TO 260                         SETI2050
                                                                         SETI2060
         C     ***** CYLINDRICAL COORDINATES *****                       SETI2070
000772         TTEMP= ER*XU*DEEST(3)                                     SETI2080
000775         DESIG(1)= DESIG(1) + TTEMP                                SETI2090
000777         DESIG(2)= DESIG(2) + TTEMP                                SETI2100
001001         DESIG(3)= DESIG(3) + ER*DEEST(3) - TTEMP                  SETI2110
                                                                         SETI2120
         C     ***** CALCULATE THE INCREMENT OF PLASTIC WORK (DEWP) ***** SETI2130
001005    260  DEWP=0.                                                   SETI2140
                                                                         SETI2150
001006         DO 273 J=1,J34                                            SETI2160
                                                                         SETI2180
         C     ***** ADD INCREMENTAL ELEMENTAL STRESSES (DESIG) TO TOTAL SETI2190
         C           ELEMENTAL STRESSES (ESIG) *****                     SETI2200
001010   273   DEWP=DEWP+(ESIG(NV,I,J)+.5*DESIG(J))*DEPST(J)             SETI2170
001040         DO 270 J=1,4                                              SETI2210
001042   270   ESIG(NV,I,J)=ESIG(NV,I,J)+DESIG(J)                        SETI2220
                                                                         SETI2230
         C     ***** ADD INCREMENT OF PLASTIC WORK (DEWP) TO EWP(NV,I), THE TOTALSETI2240
         C           PLASTIC WORK FOR NODE I OF ELEMENT NV IF DEWP IS POSITIVE.  SETI2250
001061         IF (DEWP)  271,271,272                                    SETI2260
001063   272   EWP(NV,I)=EWP(NV,I)+DEWP                                  SETI2270
```

```
                                                                              SETI2280
          C     ***** ACCUMULATE EWP(NV,I) IN WP(N) (WHERE N IS THE EXTERNAL NODE SETI2290
          C           INDEX OF THE I-TH NODE IN ELEMENT NV) *****             SETI2300
001072     271  WP(N)=WP(N)+EWP(NV,I)                                         SETI2310
                                                                              SETI2320
          C     ***** ADD DEWP TO DWP(N), WHICH WILL BE USED TO CALCULATE THE SETI2330
          C           AVERAGE INCREMENTAL PLASTIC WORK FOR NODE N *****       SETI2340
001102          DWP(N)=DWP(N)+DEWP                                            SETI2350
                                                                              SETI2360
          C     ***** ADD THE APPROPRIATE INCREMENTAL STRAINS TO THE ELEMENTAL SETI2370
          C           TOTAL STRAINS (ETST), PLASTIC STRAINS (EPST), AND ELASTIC SETI2380
          C           STRAINS (EEST) *****                                    SETI2390
001106     275  DO 280 J=1,J34                                                SETI2400
001110          ETST(NV,I,J)=ETST(NV,I,J)+DETST(I,J)                          SETI2410
001125          EPST(NV,I,J)=EPST(NV,I,J)+DEPST(J)                            SETI2420
001137          EEST(NV,I,J)=EEST(NV,I,J)+DEEST(J)                            SETI2430
                                                                              SETI2440
          C     ***** CALCULATE THE NODAL POINT INCREMENTAL STRAINS (DTST), TOTAL SETI2450
          C           STRAINS (TST), PLASTIC STRAINS (PST), AND ELASTIC STRAINS SETI2460
          C           (EST) *****                                             SETI2470
001151          DTST(N,J)=DETST(I,J)                                          SETI2480
001162          TST(N,J)=TST(N,J)+ETST(NV,I,J)                                SETI2490
001177          PST(N,J)=PST(N,J)+EPST(NV,I,J)                                SETI2500
001215    280   EST(N,J)=EST(N,J)+EEST(NV,I,J)                                SETI2510
                                                                              SETI2520
001234          DTST(N,J45)= ROT(I)                                           SETI2530
                                                                              SETI2540
                                                                              SETI2550
          C     ***** CORRECT THE TOTAL ELEMENTAL STRESSES (ESIG) *****       SETI2560
001244          E11=DETST(I,1)                                                SETI2570
001246          E22=DETST(I,2)                                                SETI2580
001250          E33= 0.0                                                      SETI2590
001251          IF ( RECCYL .NE. 0.0 )  E33= DETST(I,3)                       SETI2600
001254          GAM12=DETST(I,J34)                                            SETI2610
001261          OMEGA=ROT(I)                                                  SETI2620
001263          C1=GAM12+OMEGA                                                SETI2630
001265          C2=GAM12-OMEGA                                                SETI2640
001267          S11=ESIG(NV,I,1)                                              SETI2650
```

```
001300          S22=ESIG(NV,I,2)
001310          S12=ESIG(NV,I,J34)
001320          S33=ESIG(NV,I,J43)
001330          ESIG1= ( 1.0+E11-E22-E33 )*S11 + C1*S12
001336          ESIG(NV,I,1)= ESIG1
001347          ESIG2= ( 1.0-E11+E22-E33 )*S22 + C2*S12
001355          ESIG(NV,I,2)= ESIG2
001366          ESIG34= 0.5*( C2*S11+C1*S22 ) + ( 1.0-E33 )*S12
001376          ESIG(NV,I,J34)= ESIG34
001406          ESIG43= ( 1.0-E11-E22+E33 )*S33
001413          ESIG(NV,I,J43)= ESIG43
001423          ESIG5= SQRT(  0.5*( (ESIG1-ESIG2)**2 + (ESIG1-ESIG43)**2
               1  + (ESIG2-ESIG43)**2 + 6.0*ESIG34**2 )  )
001441          ESIG(NV,I,5)= ESIG5


         C      ***** CONSTRAIN EACH PLASTIC NODE POINT TO LIE ON THE YIELD
         C            SURFACE *****
001455          YEQSG=SQRT(YP**2+2.*XI*EM*EWP(NV,I)/(1.-XI))
001473          COMM= ESIG5/YEQSG
001475          BETA(NV,I)=0.

001505          IF (DEWP)  380,340,350

001507   340    IF (COMM-0.999)  380,380,350

001512    350   DO 360 J=1,5
001514   360    ESIG(NV,I,J)=ESIG(NV,I,J)/COMM

001531          CONS=2.*XI/(3.*(1.-XI))
001536          DENM=2.*ESIG(NV,I,5)**2*(CONS*(1.+XU)+1.)
001553          BETA(NV,I)=3./DENM

         C      ***** COUNT THE NUMBER OF ELEMENTS COMMON TO NODE N *****
001561    380   COUNT(N)=COUNT(N)+1.

         C      ***** ADD INCREMENTAL NODAL POINT STRESSES TO THE TOTAL NODAL
         C            POINT STRESSES *****
```

```
001565          DO 300 J=1,5
001566          SIG(N,J)=SIG(N,J)+ESIG(NV,I,J)
001610    300   CONTINUE



          C     ***** COMPUTE AVERAGE NODAL QUANTITIES *****
001617          DO 400 N=1,NUMNP
001621          CNT=COUNT(N)
001624          WP(N)=WP(N)/CNT
001627          DWP(N)=DWP(N)/CNT

001632     305  DO 310 J=1,J34
001634          DTST(N,J)=DTST(N,J)/CNT
001642          TST(N,J)=TST(N,J)/CNT
001650          PST(N,J)=PST(N,J)/CNT
001655    310   EST(N,J)=EST(N,J)/CNT

001665          DTST(N,J45)= DTST(N,J45)/CNT

001673          DO 320 J=1,5
001675    320   SIG(N,J)=SIG(N,J)/CNT

001706    400   CONTINUE


          C     ***** TRANSFORM FORCE VECTOR (F) *****
001711          DO 500 N=1,NUMNP
001712          E11=DTST(N,1)
001720          E22=DTST(N,2)
001726          E33= 0.0
001727          IF ( RECCYL .NE. 0.0 )  E33= DTST(N,3)
001736          E12=DTST(N,J34)
001744          OMEGA=DTST(N,J45)
001752          C1=E12+OMEGA
001754          C2=E12-OMEGA
          C     ***** SET F1 = 0 IF 2*N-1 = NEBC(I) FOR ANY I.
          C           SET F2 = 0 IF 2*N = NEBC(I) FOR ANY I.
```

```
001756          F1FLG= 1.0                                           SETI3420
001760          F2FLG= 1.0                                           SETI3430
001761          NN= N+N                                              SETI3440
001763          DO 430 I=1,NEQBC                                     SETI3450
001764          IF ( NN-1 .NE. NEBC(I) )  GO TO 410                  SETI3460
001770          F1FLG= 0.0                                           SETI3470
001771          GO TO 420                                            SETI3480
001771     410  IF ( NN .NE. NEBC(I) )  GO TO 430                    SETI3490
001775          F2FLG= 0.0                                           SETI3500
001776     420  IF ( (F1FLG+F2FLG) .EQ. 0.0 )  GO TO 440             SETI3510
002000     430  CONTINUE                                             SETI3520
002003     440  F1= F1FLG*F(NN-1)                                    SETI3530
002007          F2= F2FLG*F(NN)                                      SETI3540
002012          F(2*N-1)=(1.-E22-E33)*F1+0.5*C1*F2                   SETI3550
002023    500   F(2*N)=0.5*C2*F1+(1.-E11-E33)*F2                     SETI3560
                                                                     SETI3570
         C      ***** UPDATE NODAL POINT COORDINATES *****           SETI3580
002036          DO 510 N=1,NUMNP                                     SETI3590
002040          XORD(N)=XORD(N)+DR(2*N-1)                            SETI3600
002045     510  YORD(N)=YORD(N)+DR(2*N)                              SETI3610
002054          RETURN                                               SETI3620
002055          END                                                  SETI3630
```

```
RUN 2.3I  05/28/70          SETIC


SUBPROGRAM LENGTH
002403

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
140     -  000142    151     -  000145    182     -  000515    183     -  000506
187     -  000567    190     -  000607    230     -  000704    250     -  000744
251     -  000665    260     -  001005    270     -  001042    271     -  001072
272     -  001063    273     -  001010    275     -  001106    305     -  001632
320     -  001675    340     -  001507    350     -  001512    360     -  001514
380     -  001561    410     -  001771    420     -  001776    430     -  002000
440     -  002003

BLOCK NAMES AND LENGTHS
        -  000001    CDIMEN  -  000013    IDIMEN  -  000022    IFLAGS  -  000012
INDEX   -  000026    IPERM   -  000003    LBUCKT  -  000061    PROPS   -  000012
UV      -  000044

VARIABLE ASSIGNMENTS
AREA    -  000007C10 B       -  000000C01 BA      -  002234    CNT     -  002374
COMM    -  002334    CONS    -  002372    COUNT   -  000012    C1      -  002356
C2      -  002357    DEEST   -  002246    DENM    -  002373    DEPST   -  002242
DESIG   -  002256    DETST   -  002176    DEWP    -  002350    DIS     -  000014
DR      -  000013    DTST    -  000000    DWP     -  000010    D0      -  002327
D1      -  002330    EEST    -  000004    EM      -  000002C10 EPST    -  000003
ER      -  002312    ESIG1   -  002364    ESIG2   -  002365    ESIG34  -  002366
ESIG43  -  002367    ESIG5   -  002370    ETST    -  000002    EWP     -  000005
E11     -  002351    E12     -  002375    E22     -  002352    E33     -  002353
F       -  000015    F1      -  002401    F1FLG   -  002376    F2      -  002402
F2FLG   -  002377    GAM12   -  002354    I       -  002314    IMAT    -  000005C05
IPERM   -  000000C06 I1      -  002346    J       -  002320    J34     -  000010C05
J43     -  000011C05 J45     -  000012C05 K       -  002316    K1      -  002317
K2      -  002321    L       -  002323    LDF     -  000022C07 LDSD    -  000034C07
LDTST   -  000032C07 LDWP    -  000017C07 LL      -  002313    LMAT    -  000006C05
LSIG    -  000033C07 LTST    -  000027C07 L1      -  002324    L2      -  002325
```

```
RUN 2.3I  05/28/70          SETIC

L3      -  002326    M      -  002322    MATEL  -  000022    N      -  002315
NEBC    -  000024    NEQ    -  000002C02 NEQBC  -  CC0003C02 NLOAD  -  000005C03
NMAT    -  000007C03 NN     -  002400    NNUMEL -  0C0025    NNUMNP -  000026
NUMEL   -  000001C03 NUMNP  -  000002C03 NV     -  000001C05 OMEGA  -  002355
RECCYL  -  000002C04 ROT    -  002226    S      -  002252    SIG    -  000001
SIGP    -  002262    S11    -  002360    S12    -  002362    S22    -  002361
S33     -  002363    TEMP   -  002341    TERM1  -  002337    TERM2  -  002340
THETA   -  000023    TTEMP  -  002347    UV     -  C00000C11 UVIJ1  -  002335
UVIJ2   -  002336    UX     -  002342    UY     -  002344    VEM    -  000016
VX      -  002345    VXI    -  000020    VXU    -  CC0017    VY     -  002343
VYP     -  000021    WP     -  000011    X      -  002331    XI     -  000005C10
XORD    -  000006    XU     -  000003C10 Y      -  002332    YEQSG  -  002371
YORD    -  000007    YP     -  000006C10 Z      -  002333

START OF CONSTANTS
002057

START OF TEMPORARIES
002074

START OF INDIRECTS
002136

EXTERNAL REFERENCES
         SQRT      END

COMPILER SPACE
UNUSED -  004600    USED   -  043200
```

```
              SUBROUTINE SETUP (NP     ,XORD   ,YORD   ,LOADE ,LOADS ,NEBC   ,ND     SETU0010
             1    ,DSD    ,VEM    ,VXU    ,VTH    ,VXI    ,VYP   ,DELTAP,MATEL ,DLIM   SETU0020
             2    ,DISPL  ,IBCOND,CORNER,IA     ,MLAYER,DEPTH ,CYCTOF,CYCM   ,CYCZ   SETU0030
             3    ,VPR    ,TPR    ,DPR    )                                          SETU0040
                                                                                     SETU0050
        C          READS MOST OF INPUT DATA AND INITIALIZES PERTINENT                SETU0060
        C          QUANTITIES                                                        SETU0070
                                                                                     SETU0080
        C     ***** NOTE IF BLANAL .NE. 0.0, THEN THE ARRAY                          SETU0090
        C          CYCTOF WILL OCCURY THE SAME MEMORY AS MATRIX A.                   SETU0100
        C          THE ARRAYS IBCOND AND CORNER WILL NOT BE USED.                    SETU0110
                                                                                     SETU0120
        C     ***** THE ARRAY IA WILL WIPE OUT IBCOND AND CORNER (OR CYCTOF          SETU0130
        C          IF BLANAL .NE. 0.0)                                               SETU0140
                                                                                     SETU0150
        C     ***** FOR THE CONVENIENCE OF THE USER, THE ARRAY NP IS READ IN         SETU0160
        C          IN ROW-WISE FASHION, BUT FOR COMPATIBILITY WITH THE REST          SETU0170
        C          OF THE PROGRAM THE ARRAY NP IS TRANSPOSED IN STORAGE BEFORE SETU0180
        C          EXITING. *****                                                    SETU0190
                                                                                     SETU0200
        C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.SETU0210
000037        DIMENSION  NP(6,1000),DELTAP(2,1000),DLIM(3,1000),IBCOND(2,1000)  SETU0220
             1  ,DISPL(3,1000),CORNER(3,1000)                                        SETU0230
             2  ,CYCTOF(40,1000)                                                     SETU0240
000037        DIMENSION      XORD  (999),YORD  (999),LOADE (999),LOADS (999)       SETU0250
             1  ,NEBC  (999),ND    (999),DSD   (999),VEM   (999),VXU   (999)       SETU0260
             2  ,VTH   (999),VXI   (999),VYP   (999),MATEL (999),IA    (999)       SETU0270
             3  ,MLAYER(999),DEPTH (999),CYCM  (999),CYCZ  (999),VPR   (999)       SETU0280
             4  ,TPR   (999),DPR   (999)                                             SETU0290
                                                                                     SETU0300
        C     ***** THE INDICATED DIMENSION OF THE ARRAY B IS A DUMMY.               SETU0310
        C          THE ACTUAL DIMENSION NBUCKT IS CALCULATED BY SUBROUTINE BLOCSETU0320
        C          AND THE REQUIRED STORAGE IS ALLOCATED BY SUBROUTINE MRELSE. SETU0330
                                                                                     SETU0340
000037        COMMON  B(1)                                                           SETU0350
                                                                                     SETU0360
000037        COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ    ,NEQBC ,IBAND ,LENA   ,NEL6   SETU0370
```

```
               1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                  SETU0380
000037          COMMON  /FFLAGS/  NONTRY,BLANK                                   SETU0390
000037          EQUIVALENCE  (FNONTR,NONTRY)                                     SETU0400
000037          COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID   SETU0410
               1    ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINSETU0420
               2    ,NXINT ,NCYCF                                                SETU0430
000037          COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI SETU0440
               1    ,SAVETP,BLANAL,FLIFE                                         SETU0450
000037          COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT  SETU0460
               1    ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN SETU0470
               2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS  ,VPRFLG                   SETU0480
000037          COMMON  /IPERM/   IPERM(3)                                       SETU0490
           C    ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE SETU0500
           C          ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR  SETU0510
           C          IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE   SETU0520
           C          POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. ***** SETU0530
000037          COMMON  /LBUCKT/  LNP   ,LLOADE,LLOADS,LNEBC ,LND   ,LMATEL,LBETA SETU0540
               1    ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP  ,LXORD ,LYORD ,LDWP  SETU0550
               2    ,LWP   ,LCOUNT,LDF   ,LDR   ,LDIS  ,LF    ,LRESI ,LTST  ,LPST  SETU0560
               3    ,LEST  ,LDTST ,LSIG  ,LDSD  ,LDLIM ,LDLFLG,LVEM  ,LVXU  ,LVTH  SETU0570
               4    ,LVXI  ,LVYP  ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYRSETU0580
               5    ,LDEPTH,LCYCF ,LA    ,LTPR  ,LDPR                            SETU0590
               6    ,NBUCKT                                                      SETU0600
000037          COMMON  /LSSAVE/  LSSAVE(2,10)                                   SETU0610
000037          COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP    SETU0620
               1    ,AREA  ,TPRESS,RMAXS                                         SETU0630
                                                                                 SETU0640
000037          EQUIVALENCE  (IBANDW,IBAND )                                     SETU0650
                                                                                 SETU0660
000037          DATA  HPRESS,HDELTP,HMATEL,HCORNR,HNP   ,HBCOND,HDLIM ,HLOADE    SETU0670
               1      ,HLOADS,HDISPL                                             SETU0680
               2     / 5HPRESS ,6HDELTAP,5HMATEL ,6HCORNER,2HNP    ,6HIBCOND     SETU0690
               3      ,4HDLIM  ,5HLOADE ,5HLOADS ,5HDISPL  /                     SETU0700
                                                                                 SETU0710
000037          NAMELIST /DATA /  DELTAP,DISPL ,DLIM  ,IBCOND,LOADE ,LOADS ,MATEL SETU0720
               1    ,NP    ,VEM   ,VTH   ,VXI   ,VXU   ,VYP   ,CORNER,XORD  ,YORD  SETU0730
               2    ,LSSAVE,MLAYER,DEPTH ,CYCTOF,CYCM  ,CYCZ  ,VPR               SETU0740
                                                                                 SETU0750
```

```
000037     1     FORMAT (37H0ILLEGAL VALUE SPECIFIED FOR VARIABLE,5X,A6,5X         SETU0760
               1   ,17HWITH SUBSCRIPT(S),2I6 )                                       SETU0770
000037     19    FORMAT (  31H1 MATERIAL AND GEOM. PROPERTIES,///                    SETU0780
               1 16H ELASTIC MODULUS,17X,5F12.2/(33X,5F12.2) )                       SETU0790
000037       20 FORMAT(1H1,15HELEMENT ARRAY  ,//,8H ELEMENT,5X,1HI,5X,1HJ,5X,1HK,   SETU0800
               1 5X,1HL,5X,1HM,5X,1HN,//)                                            SETU0810
000037     21    FORMAT (15H0POISSONS RATIO,16X,5F12.3/(31X,5F12.3) )                SETU0820
000037     22    FORMAT (18H0INITIAL THICKNESS,13X,5F12.3/(31X,5F12.3) )             SETU0830
000037     23    FORMAT (30H0PLASTIC/ELASTIC MODULUS RATIO,3X,5E12.3/(33X,5E12.3) )SETU0840
000037     24    FORMAT (20H0INITIAL YIELD POINT,11X,5F12.1/(31X,5F12.1) )           SETU0850
000037     25    FORMAT (25H0ARC SEGMENT (IN DEGREES),6X,5F12.3/(31X,5F12.3) )       SETU0860
000037       26 FORMAT(1H0,2X,7I6)                                                   SETU0870
000037     27    FORMAT (32H1VARIABLE MATERIAL SPECIFICATION//                       SETU0880
               1    2(20H  ELEMENT  MATERIAL ),6H  ETC./(I7,I9,4(I11,I9)) )          SETU0890
000037       29 FORMAT(1H1,19H CORNER POINT ARRAY,//,6H POINT,10X,5HX-ORD,10X,      SETU0900
               1 5HY-ORD,/,1X)                                                       SETU0910
000037       31 FORMAT(1H ,I6,2F15.5)                                                SETU0920
000037       33 FORMAT(1H0,I5,I16)                                                   SETU0930
000037     34    FORMAT (1H+,E41.5)                                                  SETU0940
000037       35 FORMAT(1H1,14H PRESSURE LOAD,//,8H ELEMENT,8X,11HSIDE LOADED)        SETU0950
000037     36    FORMAT (1H+,E61.5,E20.5)                                            SETU0960
000037       40 FORMAT(1H1,20H BOUNDARY CONDITIONS,//,6H POINT,2X,4HNFIX,/,1X)       SETU0970
000037     41    FORMAT(1H1,22H IMPOSED DISPLACEMENTS,//,8H NODE    ,8X,             SETU0980
               1    11HDIRECTION  ,8X,9HMAGNITUDE,/)                                 SETU0990
000037     43    FORMAT(1H ,I6,9X,I6,9X,E12.5)                                       SETU1000
000037       44 FORMAT(1H0,2I6)                                                      SETU1010
000037     45    FORMAT (27H1 NODAL DISPLACEMENT LIMITS//                            SETU1020
               1  30H  NODE  DIRECT.   DISPL. LIMIT/ )                               SETU1030
000037     46    FORMAT (I5,I7,E17.5)                                                SETU1040
000037     48    FORMAT (1H+,30X,10HTOTAL LOAD)                                      SETU1050
000037     49    FORMAT (1H+,50X,10HINCR. LOAD,10X,11HACCUM. LOAD )                  SETU1060
000037     61    FORMAT ( 9H0ERROR... I5,25H SIDES LOADED FOR ELEMENT,I8 )           SETU1070
000037       62 FORMAT (33H0NEGATIVE OR ZERO AREA, ELEMENT = I5)                     SETU1080
000037       65 FORMAT (13H0BAND WIDTH = I5)                                         SETU1090
000037       70 FORMAT (18H0MAX BAND WIDTH OF I3, 9H EXCEEDED)                       SETU1100
                                                                                     SETU1110
                                                                                     SETU1120
000037           II= 1                                                               SETU1130
```

```
000040          IERROR = 1                                                                SETU1140
                                                                                          SETU1150
          C     ***** CLEAR BUCKET STORAGE REGION *****                                   SETU1160
000041          NCLEAR= NBUCKT - 1                                                        SETU1170
000043          DO 100 I=1,NCLEAR                                                         SETU1180
000044     100  B(I)= 0.0                                                                 SETU1190
000051          IF ( FLIFE .EQ. 0.0 )  GO TO 103                                          SETU1200
000052          DO 102 I=1,NMAT                                                           SETU1210
000053          CYCM(I)= FNONTR                                                           SETU1220
000056     102  CONTINUE                                                                  SETU1230
                                                                                          SETU1240
          C     ***** READ INPUT ARRAYS AND LOAD STEP DEPENDENT QUANTITIES FROM           SETU1250
          C           RESTART TAPE *****                                                  SETU1260
000061     103  IF ( RESTRT .NE. 0.0 )                                                    SETU1270
               1      CALL RRST23 (NP    ,LOADE ,LOADS ,DLIM  ,VEM   ,VXU   ,VTH          SETU1280
               2    ,VXI   ,VYP   ,DELTAP,MATEL ,DISPL ,XORD  ,YORD                       SETU1290
               3  ,B(LDIS  ),B(LF    ),B(LRESI ),NEBC     ,B(LDLFLG),B(LBETA )            SETU1300
               4  ,B(LETST ),B(LEPST ),B(LEEST ),B(LESIG ),B(LEWP  ),MLAYER               SETU1310
               5  ,CYCM  ,CYCZ  ,DEPTH ,VPR   ,TPR   ,ND    ,DSD   ,NUMEL  )              SETU1320
                                                                                          SETU1330
          C     ***** READ REMAINDER OF NAMELIST INPUT DATA FOR THIS CASE *****           SETU1340
000176          READ (5,DATA)                                                             SETU1350
                                                                                          SETU1360
                                                                                          SETU1370
000201          IF ( (BLANAL .EQ. 0.0) .OR. (RESTRT .NE. 0.0) )  GO TO 2000               SETU1380
                                                                                          SETU1390
          C     ***** BLADDER ANALYSIS OPTION.                                            SETU1400
          C           THIS IS AN INITIAL RUN, NOT A RESTART *****                         SETU1410
                                                                                          SETU1420
000213          CALL       MSHGEN (NP    ,XORD  ,YORD  ,DEPTH ,CYCM  ,CYCZ  ,VTH          SETU1430
               1    ,MLAYER,MATEL ,ND    ,NEBC  ,CYCTOF,IERROR )                          SETU1440
                                                                                          SETU1450
000231    2000  IF ( (BLANAL .NE. 0.0) .OR. (VPRFLG .NE. 0.0) )  GO TO 110                SETU1460
000243          IF (NDP .NE. 0)  GO TO 105                                                SETU1470
                                                                                          SETU1480
000244          DPRESS= (PRESS - TPRESS)/ RMAXS                                           SETU1490
000247          GO TO 110                                                                 SETU1500
                                                                                          SETU1510
```

```
        C      ***** VARIABLE LOAD STEP SIZES *****                              SETU1520
000247   105   NSTEP= DELTAP(1,1)                                                SETU1530
000252         DPRESS= DELTAP(2,1)                                               SETU1540
000254         IF ( (NSTEP .GT. 0) .AND. (DPRESS .NE. 0.0) )  GO TO 110          SETU1550
000263         IERROR= 0                                                         SETU1560
000264         WRITE (6,1)  HDELTP,II                                            SETU1570
                                                                                 SETU1580
        C      ***** MATERIAL PROPERTIES *****                                   SETU1590
000273   110   WRITE (6     ,19)  (VEM(I),I=1,NMAT)                              SETU1600
000306         WRITE (6     ,21)  (VXU(I),I=1,NMAT)                              SETU1610
                                                                                 SETU1620
000321         IF ( RECCYL .EQ. 0.0 )  GO TO 115                                 SETU1630
000326         IF ( BLANAL .NE. 0.0 )  GO TO 115                                 SETU1640
                                                                                 SETU1650
        C      ***** CYLINDRICAL COORDINATES. CONVERT THICKNESS FROM DEGREES     SETU1660
        C            TO RADIANS ******                                           SETU1670
000327         WRITE (6     ,25)  (VTH(I),I=1,NMAT)                              SETU1680
000341         DO 112 I=1,NMAT                                                   SETU1690
000346   112   VTH(I)= 0.0174532925*VTH(I)                                       SETU1700
000354         GO TO 116                                                         SETU1710
                                                                                 SETU1720
000354   115   WRITE (6     ,22)  (VTH(I),I=1,NMAT)                              SETU1730
                                                                                 SETU1740
000367   116   WRITE (6     ,23)  (VXI(I),I=1,NMAT)                              SETU1750
000402         WRITE (6     ,24)  (VYP(I),I=1,NMAT)                              SETU1760
                                                                                 SETU1770
000415         IF (NMAT .GT. 1)  GO TO 120                                       SETU1780
                                                                                 SETU1790
        C      ***** ONLY ONE MATERIAL. PROPERTIES ARE CONSTANT *****            SETU1800
000424         EM= VEM(1)                                                        SETU1810
000425         XU= VXU(1)                                                        SETU1820
000427         TH= VTH(1)                                                        SETU1830
000431         XI= VXI(1)                                                        SETU1840
000433         YP= VYP(1)                                                        SETU1850
000435         GO TO 130                                                         SETU1860
                                                                                 SETU1870
        C      ***** MATERIAL SPECIFICATION FOR EACH ELEMENT (IF NMAT .GT. 1) ***SETU1880
000436   120   WRITE (6     ,27)  ( I,MATEL(I),I=1,NUMEL )                       SETU1890
```

```
000453           DO 125 I=1,NUMEL                                                    SETU1900
000460           IF ( (MATEL(I) .GT. 0) .AND. (MATEL(I) .LE. NMAT) )  GO TO 125      SETU1910
000472           IERROR= 0                                                           SETU1920
000473           WRITE (6,1)  HMATEL,I                                               SETU1930
000502      125  CONTINUE                                                            SETU1940
                                                                                     SETU1950
           C     ***** COORDINATES OF CORNER POINTS *****                            SETU1960
000510      130  IF ( RESTRT .NE. 0.0 )  GO TO 135                                   SETU1970
000511           IF ( BLANAL .NE. 0.0 )  GO TO 135                                   SETU1980
                                                                                     SETU1990
000512           WRITE (6,29)                                                        SETU2000
000516           DO 140 I=1,NUMCP                                                    SETU2010
000523           N= CORNER(1,I)                                                      SETU2020
000527           X= CORNER(2,I)                                                      SETU2030
000533           Y= CORNER(3,I)                                                      SETU2040
000536           WRITE (6,31)  N,X,Y                                                 SETU2050
000550           IF ( (N .GT. 0) .AND. (N .LE. NUMNP) )  GO TO 141                   SETU2060
000564           IERROR= 0                                                           SETU2070
000565           WRITE (6,1)  HCORNR,II,I                                            SETU2080
000576           GO TO 140                                                           SETU2090
                                                                                     SETU2100
000602      141  XORD(N)= X                                                          SETU2110
000605           YORD(N)= Y                                                          SETU2120
000607      140  CONTINUE                                                            SETU2130
                                                                                     SETU2140
                                                                                     SETU2150
000612      135  WRITE (6,20)                                                        SETU2160
                                                                                     SETU2170
000616           DO 146 N=1,NUMEL                                                    SETU2180
                                                                                     SETU2190
           C     ***** NODE SPECIFICATION FOR EACH ELEMENT *****                     SETU2200
                                                                                     SETU2220
000623           WRITE (6,26)  N,(NP(I,N),I=1,6)                                     SETU2210
000643           IF ( BLANAL .NE. 0.0 )  GO TO 146                                   SETU2230
                                                                                     SETU2240
000650           DO 145 I=1,6                                                        SETU2250
000651           K=NP(I,N)                                                           SETU2260
000655           IF ( (K .GT. 0) .AND. (K .LE. NUMNP) )  GO TO 142                   SETU2270
```

```
000666           IERROR= 0                                                   SETU2280
000667           WRITE (6,1)  HNP,I,N                                        SETU2290
                                                                             SETU2300
000700     142   IF (I .GT. 3)  GO TO 145                                    SETU2310
                                                                             SETU2320
000707           IF ( RESTRT .NE. 0.0 )  GO TO 145                           SETU2330
                                                                             SETU2340
          C      ***** CALCULATE COORDINATES OF ALL MIDPOINTS *****          SETU2350
000710           J=IPERM(I)                                                  SETU2360
000712           L=NP(J,N)                                                   SETU2370
000716           IF ( (L .GT. 0) .AND. (L .LE. NUMNP) )  GO TO 143           SETU2380
000727           IERROR = 0                                                  SETU2390
000730           WRITE (6,1)  HNP,J,N                                        SETU2400
                                                                             SETU2410
000741     143   IP3= I + 3                                                  SETU2420
000743           M=NP(IP3,N)                                                 SETU2430
000753           IF ( (M .GT. 0) .AND. (M .LE. NUMNP) )  GO TO 144           SETU2440
000764           IERROR = 0                                                  SETU2450
000765           WRITE (6,1)  HNP,IP3,N                                      SETU2460
                                                                             SETU2470
000776     144   IF (IERROR .EQ. 0)  GO TO 145                               SETU2480
001003           XORD(M)=0.5*(XORD(K)+XORD(L))                               SETU2490
001012           YORD(M)=0.5*(YORD(K)+YORD(L))                               SETU2500
001022     145   CONTINUE                                                    SETU2510
                                                                             SETU2520
001024     146   CONTINUE                                                    SETU2530
                                                                             SETU2540
                                                                             SETU2550
001027           IF ( RESTRT .NE. 0.0 )  GO TO 185                           SETU2560
001030           IF ( BLANAL .NE. 0.0 )  GO TO 185                           SETU2570
                                                                             SETU2580
          C      ***** BOUNDARY CONDITIONS *****                             SETU2590
001031           J= NEQBC                                                    SETU2600
001032           WRITE (6,40)                                                SETU2610
001036           DO 200 I=1,NUMBC                                            SETU2620
001043           N= IBCOND(1,I)                                              SETU2630
001046           IF ( (RESTRT .NE. 0.0) .AND. (N .EQ. 0) )  GO TO 200        SETU2640
                                                                             SETU2650
```

```
001054            NFIX= IBCOND(2,I)
001057            WRITE (6,44)  N,NFIX

001066            IF ( (N .GT. 0) .AND. (N .LE. NUMNP) )  GO TO 165
001102            IERROR= 0
001103            WRITE (6,1)  HBCOND,II,I
001114            GO TO 200

001120      165   JY=2*N
001122            JX=JY-1
001124            IF (NFIX-1) 170,175,180

          C       ***** NODE CONSTRAINED IN BOTH X AND Y DIRECTIONS *****
001126       170  J=J+2
001130            NEBC(J-1)=JX
001132            NEBC(J)=JY
001135            GO TO 200

          C       ***** NODE CONSTRAINED IN ONLY THE X DIRECTION *****
001135       175  J=J+1
001137            NEBC(J)=JX
001141            GO TO 200

          C       ***** NODE CONSTRAINED IN ONLY THE Y DIRECTION *****
001142       180  J=J+1
001144            NEBC(J)=JY

001146       200  CONTINUE

001151            NEQBC=J


001152      185   IF (NNDLIM .EQ. 0)  GO TO 195

          C       ***** LIMITING DISPLACEMENTS *****
001153            WRITE (6,45)
001157            DO 190 I=1,NNDLIM
001164            NODE= DLIM(1,I)
```

```
001170          IDIR= DLIM(2,I)
001174          DLIMIT= DLIM(3,I)
001200          WRITE (6,46)  NODE,IDIR,DLIMIT
001211          IF ( (NODE .GT. 0) .AND. (NODE .LE. NUMNP) .AND. (IDIR .GT. 0)
               1  .AND. (IDIR .LT. 3) )  GO TO 190
001235          IERROR= 0
001236          WRITE (6,1)  HDLIM,II,I
001247     190  CONTINUE


001255          IF ( RESTRT .NE. 0.0 )  CALL DISCHK ( B(LDIS),NEBC,DLIM,B(LDLFLG))

001273     195  IF (NLOAD .LE. 0)  GO TO 160

           C    ***** PRESSURE LOADING SPECIFICATIONS *****
001275          WRITE (6,35)
001300          IF ( VPRFLG .EQ. 0.0 )  GO TO 196
001305          WRITE (6,48)
001310          IF ( RESTRT .EQ. 0.0 )  GO TO 196
001315          WRITE (6,49)
001320     196  DO 150 N=1,NLOAD
001325          LELEM= LOADE(N)
001330          LSIDE= LOADS(N)
001332          WRITE (6,33) LELEM,LSIDE
001342          IF ( VPRFLG .EQ. 0.0 )  GO TO 153
           C    ***** VARIABLE PRESSURE LOADING DESIRED.  COMPUTE PRESSURE
           C          INCREMENT FOR SIDE LOADS(N) OF ELEMENT LOADE(N) *****
001347          WRITE (6,34)  VPR(N)
001355          DPR(N)= ( VPR(N) - TPR(N) ) / RMAXS
001365          IF ( RESTRT .EQ. 0.0 )  GO TO 153
001372          WRITE (6,36)  DPR(N),TPR(N)

001403     153  IF ( (LELEM .GT. 0) .AND. (LELEM .LE. NUMEL) )  GO TO 151
001417          IERROR= 0
001420          WRITE (6,1)  HLOADE,N
001427     151  IF ( IABS(LSIDE) .LT. 4 )  GO TO 150
001436          IERROR = 0
001437          WRITE (6,1)  HLOADS,N
```

```
001446    150  CONTINUE                                                SETU3420
                                                                       SETU3430
001454         DO 155 NV=1,NUMEL                                       SETU3440
001456         ICNT= 0                                                 SETU3450
001457         DO 152 N= 1,NLOAD                                       SETU3460
001461         IF ( NV .EQ. LOADE(N) )  ICNT= ICNT + 1                 SETU3470
001465    152  CONTINUE                                                SETU3480
001470         IF ( ICNT .LT. 4 )  GO TO 155                           SETU3490
001472         IERROR= 0                                               SETU3500
001473         WRITE (6,61)  ICNT,NV                                   SETU3510
001502    155  CONTINUE                                                SETU3520
                                                                       SETU3530
                                                                       SETU3540
001510   160   IF (NID .LE. 0)  GO TO 220                              SETU3550
001512         IF ( BLANAL .NE. 0.0 )  GO TO 220                       SETU3560
                                                                       SETU3570
         C     ***** IMPOSED DISPLACEMENTS *****                       SETU3580
001513         WRITE (6,41)                                            SETU3590
001516         DO 210 I=1,NID                                          SETU3600
001523         N= DISPL(1,I)                                           SETU3610
001527         IF ( (RESTRT .NE. 0.0) .AND. (N .EQ. 0) )  GO TO 210    SETU3620
001535         J= DISPL(2,I)                                           SETU3630
001541         DISP= DISPL(3,I)                                        SETU3640
001544         WRITE (6,43)  N,J,DISP                                  SETU3650
001556         IF ( (N .GT. 0) .AND. (N .LE. NUMNP) .AND.              SETU3660
              1     (J .GT. 0) .AND. (J .LT. 3) )  GO TO 205           SETU3670
001602         IERROR= 0                                               SETU3680
001603         WRITE (6,1)  HDISPL,II,I                                SETU3690
001614    205  L=2*(N-1)+J                                             SETU3700
001617         ND(I)=L                                                 SETU3710
001622         DSD(I)=DISP /RMAXS                                      SETU3720
001626    210  CONTINUE                                                SETU3730
                                                                       SETU3740
                                                                       SETU3750
         C     CHECK FOR ZERO OR NEGATIVE ELEMENT AREAS                SETU3760
         C     AND COMPUTE BAND WIDTH                                  SETU3770
         C                                                             SETU3780
001634     220 NPD = 0                                                 SETU3790
```

```
001635        DO 250  N = 1,NUMEL                                        SETU3800
001637        I = NP(1,N)                                                SETU3810
001643        J = NP(2,N)                                                SETU3820
001646        K = NP(3,N)                                                SETU3830
001652        X = XORD(I)                                                SETU3840
001654        Y = YORD(I)                                                SETU3850
001657        AREA = (XORD(J)-X)*(YORD(K)-Y) - (XORD(K)-X)*(YORD(J)-Y)   SETU3860
001675        IF (AREA.GT.0.)  GO TO 240                                 SETU3870
001700        IERROR= 0                                                  SETU3880
001700        WRITE(6,62)N                                               SETU3890
001706    240 DO 250  I = 1,5                                            SETU3900
001713        K = I + 1                                                  SETU3910
001715        DO 250  J = K,6                                            SETU3920
001716        M = NP(I,N) - NP(J,N)                                      SETU3930
001725        L = IABS(M)                                                SETU3940
001727        IF (L.GT.NPD)  NPD = L                                     SETU3950
001733    250 CONTINUE                                                   SETU3960
001742        IBANDW = 2*NPD + 2                                         SETU3970
001744        WRITE(6,65)IBANDW                                          SETU3980
001751        IF (IBANDW.LE.MAXBW)  GO TO 280                            SETU3990
001757        IERROR= 0                                                  SETU4000
001760        WRITE (6,70)MAXBW                                          SETU4010
                                                                         SETU4020
001766   280  IF (IERROR .EQ. 0)  STOP                                   SETU4030
                                                                         SETU4040
001775        LENA= NEQ*IBANDW                                           SETU4050
                                                                         SETU4060
002000        IF ( RESTRT .NE. 0.0 )  GO TO 400                          SETU4070
                                                                         SETU4080
                                                                         SETU4090
        C     ***** INITIALIZE EWP ARRAY *****                           SETU4100
002001        LL= LXORD-1                                                SETU4110
002003        DO 330 I=LEWP,LL                                           SETU4120
                                                                         SETU4140
        C     ***** TRANSPOSE THE NP ARRAY USING THE IA ARRAY AS SCRATCH CORE.  SETU4150
002005   330  B(I)= -1.E-6                                               SETU4130
002012   400  L= 1                                                       SETU4160
002013        DO 410 I=1,6                                               SETU4170
```

```
002015          DC 410 N=1,NUMEL                                                          SETU4180
002016          IA(L)= NP(I,N)                                                            SETU4190
002024     410  L= L+1                                                                    SETU4200
002032          NEL6= 6*NUMEL                                                             SETU4210
002034          DO 420 I=1,NEL6                                                           SETU4220
                                                                                          SETU4240
002036     420  NP(I,1)= IA(I)                                                            SETU4230
002047          IF ( SAVETP .NE. 0.0 )                                                    SETU4250
               1      CALL WRST2 (NP    ,LOADE ,LOADS ,DLIM  ,VEM   ,VXU   ,VTH    SETU4260
               1   ,VXI   ,VYP   ,DELTAP,MATEL ,MLAYER,CYCM  ,CYCZ  ,ND    ,DSD    SETU4270
               2   ,VPR   ,DEPTH )                                                 SETU4280
002103          RETURN                                                                    SETU4290
002104          END                                                                       SETU4300
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
NONTRY
IBAND
```

```
RUN 2.3I  05/28/70          SETUP


SUBPROGRAM LENGTH
002604

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1        - 002216    19       - 002230    20       - 002244    21       - 002256
22       - 002264    23       - 002272    24       - 002302    25       - 002311
26       - 002320    27       - 002323    29       - 002340    31       - 002351
33       - 002354    34       - 002357    35       - 002362    36       - 002371
40       - 002374    41       - 002403    43       - 002416    44       - 002422
45       - 002425    46       - 002440    48       - 002443    49       - 002447
61       - 002455    62       - 002464    65       - 002472    70       - 002476
103      - 000061    105      - 000247    110      - 000273    115      - 000354
116      - 000367    120      - 000436    125      - 000502    130      - 000510
135      - 000612    140      - 000607    141      - 000602    142      - 000700
143      - 000741    144      - 000776    145      - 001022    146      - 001024
150      - 001446    151      - 001427    152      - 001465    153      - 001403
155      - 001502    160      - 001510    165      - 001120    170      - 001126
175      - 001135    180      - 001142    185      - 001152    190      - 001247
195      - 001273    196      - 001320    200      - 001146    205      - 001614
210      - 001626    220      - 001634    240      - 001706    250      - 001733
280      - 001766    400      - 002012    420      - 002036    2000     - 000231

BLOCK NAMES AND LENGTHS
         - 000001    CDIMEN   - 000013    FFLAGS   - 000002    IDIMEN   - 000022
IFLAGS   - 000012    INDEX    - 000026    IPERM    - 000003    LBUCKT   - 000061
LSSAVE   - 000024    PROPS    - 000012

VARIABLE ASSIGNMENTS
AREA     - 000007C12 B        - 000000C01 BLANAL   - 000010C05 CORNER   - 000014
CYCM     - 000021    CYCTOF   - 000020    CYCZ     - 000022    DELTAP   - 000007
DEPTH    - 000017    DISP     - 002601    DISPL    - 000012    DLIM     - 000011
DLIMIT   - 002575    DPR      - 000025    DPRESS   - 000001C12 DSD      - 000001
EM       - 000002C12 FLIFE    - 000011C05 FNONTR   - 000000C03 HBCOND   - 002547
HCORNR   - 002545    HDELTP   - 002543    HDISPL   - 002553    HDLIM    - 002550
```

```
RUN 2.3I  05/28/70          SETUP

HLOADE -  002551     HLOADS -  002552     HMATEL -  002544     HNP    -  002546
HPRESS -  002542     I      -  002557     IA     -  000015     IBANDW -  000004C02
IBCOND -  000013     ICNT   -  002600     IDIR   -  002574     IERROR -  002555
II     -  002554     IPERM  -  000000C07  IP3    -  002566     J      -  002564
JX     -  002572     JY     -  002571     K      -  002563     L      -  002565
LBETA  -  000006C10  LDIS   -  000024C10  LDLFLG -  000036C10  LEEST  -  000012C10
LELEM  -  002576     LENA   -  000005C02  LEPST  -  000011C10  LESIG  -  000013C10
LETST  -  000010C10  LEWP   -  000014C10  LF     -  000025C10  LL     -  002603
LRESI  -  000026C10  LSIDE  -  002577     LSSAVE -  000000C11  LXORD  -  000015C10
M      -  002567     MATEL  -  000010     MAXBW  -  000011C04  MLAYER -  000016
N      -  002560     NBUCKT -  000060C10  NCLEAR -  002556     ND     -  000000
NDP    -  000010C04  NEL6   -  000006C02  NEQ    -  000002C02  NEQBC  -  000003C02
NFIX   -  002570     NID    -  000006C04  NLOAD  -  000005C04  NMAT   -  000007C04
NNDLIM -  000013C04  NODE   -  002573     NPD    -  002602     NSTEP  -  000004C06
NUMBC  -  000004C04  NUMCP  -  000003C04  NUMEL  -  000001C04  NUMNP  -  000002C04
NV     -  000001C06  PRESS  -  000000C12  RECCYL -  000002C05  RESTRT -  000001C05
RMAXS  -  000011C12  SAVETP -  000007C05  TH     -  000004C12  TPR    -  000024
TPRESS -  000010C12  VEM    -  000002     VPR    -  000023     VPRFLG -  000025C06
VTH    -  000004     VXI    -  000005     VXU    -  000003     VYP    -  000006
X      -  002561     XI     -  000005C12  XU     -  000003C12  Y      -  002562
YP     -  000006C12

START OF CONSTANTS
002215

START OF TEMPORARIES
002512

START OF INDIRECTS
002530

EXTERNAL REFERENCES
        RRST23    INPUTN    MSHGEN    OUTPTC    DISCHK    STOP      WRST2     END

COMPILER SPACE
UNUSED -  003600     USED   -  044200
```

```
               SUBROUTINE STEP2 (A      ,NP     ,DELTAP,DF     ,DR     ,F      ,RESI  STEP0010
              1    ,NEBC  ,ND     ,DSD    ,VEM    ,VXU    ,VTH    ,MATEL ,DEPTH ,XORD  STEP0020
              2    ,YORD  ,TPR    ,DPR    ,NNUMEL,NNEQ  )                              STEP0030
                                                                                      STEP0040
         C          ASSEMBLES THE INCREMENTAL SYSTEM STIFFNESS MATRIX AND             STEP0050
         C          SOLVES FOR INCREMENTAL DISPLACEMENTS.                             STEP0060
                                                                                      STEP0070
         C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.STEP0080
000030         DIMENSION    A(NNEQ,1),NP(NNUMEL,6),DELTAP(2,1)                        STEP0090
000030         DIMENSION     DF     (1),DR     (1),F      (1),RESI   (1)              STEP0100
              1  ,NEBC  (1),ND     (1),DSD    (1),VEM    (1),VXU    (1),VTH    (1)    STEP0110
              2  ,MATEL (1),DEPTH (1),XORD   (1),YORD   (1),TPR    (1),DPR    (1)     STEP0120
                                                                                      STEP0130
         C     ***** THE INDICATED DIMENSION OF THE ARRAY B IS A DUMMY.               STEP0140
         C          THE ACTUAL DIMENSION NBUCKT IS CALCULATED BY SUBROUTINE BLOCSTEP0150
         C          AND THE REQUIRED STORAGE IS ALLOCATED BY SUBROUTINE MRELSE. STEP0160
                                                                                      STEP0170
000030         COMMON  B(1)                                                           STEP0180
                                                                                      STEP0190
000030         COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ    ,NEQBC ,IBAND ,LENA   ,NEL6   STEP0200
              1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                       STEP0210
000030         COMMON  /CMESH /  BLTHNS,XINTN ,XYRAT ,DPRAT                           STEP0220
000030         COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID        STEP0230
              1    ,NMAT  ,NDP    ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINSTEP0240
              2    ,NXINT ,NCYCF                                                      STEP0250
000030         COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER   ,IRESI  STEP0260
              1    ,SAVETP,BLANAL,FLIFE                                               STEP0270
000030         COMMON  /INDEX /  IFIX   ,NV     ,ISTEP ,JSTEP ,NSTEP ,IMAT   ,LMAT   STEP0280
              1    ,ERRFLG,J34   ,J43    ,J45    ,I63    ,I69    ,I912   ,LSTEPO,NPRIN STEP0290
              2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS   ,VPRFLG                        STEP0300
         C     ***** THE ARRAYS TPR AND DPR ARE STORED IMMEDIATELY FOLLOWING THE STEP0310
         C          ARRAY UPR. THUS THE POSITION OF THE POINTERS LTPR AND LDPR  STEP0320
         C          IN THE COMMON BLOCK LBUCKT DOES NOT INDICATE THE RELATIVE   STEP0330
         C          POSITION OF TPR AND DPR IN THE DYNAMIC STORAGE ARRAY. ***** STEP0340
000030         COMMON  /LBUCKT/  LNP    ,LLOADE,LLOADS,LNEBC ,LND    ,LMATEL,LBETA STEP0350
              1    ,LVPR  ,LETST ,LEPST ,LEEST ,LESIG ,LEWP   ,LXCRD ,LYORD ,LDWP   STEP0360
              2    ,LWP    ,LCOUNT,LDF    ,LDR    ,LDIS   ,LF     ,LRESI ,LTST  ,LPST   STEP0370
```

```
                 3    ,LEST   ,LDTST ,LSIG   ,LDSD   ,LDLIM ,LDLFLG,LVEM   ,LVXU   ,LVTH  STEP0380
                 4    ,LVXI   ,LVYP   ,LDELTP,LTHETA,LDISPL,LBCOND,LCYCM ,LCYCZ ,LMLAYRSTEP0390
                 5    ,LDEPTH,LCYCF ,LA     ,LTPR   ,LDPR                                 STEP0400
                 6    ,NBUCKT                                                              STEP0410
000030            COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRATSTEP0420
                 1    ,RSTRAT,RFINAL                                                       STEP0430
000030            COMMON  /PROPS /  PRESS ,DPRESS,EM     ,XU     ,TH     ,XI     ,YP    STEP0440
                 1    ,AREA  ,TPRESS,RMAXS                                                 STEP0450
000030            COMMON  /ST    /  ST(12,12)     ,CF(12)                                  STEP0460
                                                                                           STEP0470
000030       902  FORMAT(1H1,5X,11HLOAD STEP   ,I4,3H   =E15.5,5X,12HTOTAL LOAD =E15.5STEP0480
                 1  // )                                                                   STEP0490
000030       903  FORMAT (1H1,5X,9HLOAD STEP,I6///9X,11HACCUM. LOAD//(I5,E15.5) )         STEP0500
                                                                                           STEP0510
                                                                                           STEP0520
000030            IF (IFIX .NE. 0)  GO TO 70                                               STEP0530
                                                                                           STEP0540
000031            IF ( BLANAL .EQ. 0.0 )  GO TO 40                                         STEP0550
                                                                                           STEP0560
         C        ***** BLADDER ANALYSIS OPTION.                                           STEP0570
000032            IF ( RECCYL .NE. 0.0 )  GO TO 60                                         STEP0580
                                                                                           STEP0590
         C        ***** SINGLE FOLD OPTION *****                                           STEP0600
000033        30  TPRESS= RSFOLD                                                           STEP0610
000035            GO TO 60                                                                 STEP0620
                                                                                           STEP0630
000035        40  IF (NDP .GT. 1)  GO TO 50                                                STEP0640
                                                                                           STEP0650
000041            IF ( VPRFLG .EQ. 0.0 )  GO TO 60                                         STEP0660
                                                                                           STEP0670
         C        ***** VARIABLE PRESSURE LOADING DESIRED.  PRINT PRESSURE                 STEP0680
         C              INCREMENT AND TOTAL PRESSURE FOR EACH SIDE LOADED *****            STEP0690
000042            DO 45 I=1,NLOAD                                                          STEP0700
000043            TPR(I)= TPR(I) + DPR(I)                                                  STEP0710
000050        45  CONTINUE                                                                 STEP0720
000052            WRITE (6,903)  ISTEP,(I,TPR(I),I=1,NLOAD)                                STEP0730
000075            GO TO 111                                                                STEP0740
                                                                                           STEP0750
```

```
         C      ***** MORE THAN 1 PRESSURE INCREMENT SPECIFIED *****                STEP0760
000101    50    IF (ISTEP .LE. NSTEP)  GO TO 60                                      STEP0770
                                                                                     STEP0780
000104          JSTEP= JSTEP + 1                                                     STEP0790
000105          IF ( JSTEP .LE. NDP )  GO TO 55                                      STEP0800
                                                                                     STEP0810
         C      ***** NO MORE ENTRIES IN DELTAP ARRAY. USE CURRENT VALUE OF          STEP0820
         C            DPRESS FOR THE REST OF THE RUN *****                           STEP0830
000110          NSTEP= MAXST                                                         STEP0840
000111          GO TO 60                                                             STEP0850
                                                                                     STEP0860
000112    55    NSTEP= DELTAP(1,JSTEP)                                               STEP0870
000116          DPRESS= DELTAP(2,JSTEP)                                              STEP0880
000120          GO TO 50                                                             STEP0890
                                                                                     STEP0900
000121    60    TPRESS= TPRESS + DPRESS                                              STEP0910
000123          WRITE(6,902)  ISTEP,DPRESS,TPRESS                                    STEP0920
                                                                                     STEP0930
         C      ***** IFIX .EQ. 0. AN INCREMENT OF LOAD IS BEING APPLIED. STORE      STEP0940
         C            RESIDUAL LOAD VECTOR INTO INCREMENTAL LOAD VECTOR (DF) *****STEP0950
000135   111    DO 113 I=1,NEQ                                                       STEP0960
                                                                                     STEP0980
                                                                                     STEP0990
         C      ***** CLEAR SYSTEM STIFFNESS MATRIX (A) AND INCREMENTAL              STEP1000
         C            DISPLACEMENT VECTOR (DR) *****                                 STEP1010
000142   113    DF(I)=RESI(I)                                                        STEP0970
000150    70    DO 100 I=1,LENA                                                      STEP1020
000152   100    A(I,1)= 0.0                                                          STEP1030
000163          DO 110 I=1,NEQ                                                       STEP1040
                                                                                     STEP1060
                                                                                     STEP1070
000164   110    DR(I)=0.                                                             STEP1050
000170   112    LMAT= 0                                                              STEP1080
                                                                                     STEP1090
         C      ***** ELEMENT LOOP *****                                             STEP1100
                                                                                     STEP1110
000171          DO 1 NV=1,NUMEL                                                      STEP1120
                                                                                     STEP1130
```

```
                                                                   STEP1150
000173          IF (NMAT .EQ. 1)  GO TO 120                        STEP1140
000175          IMAT= MATEL(NV)                                    STEP1160
000200          IF (IMAT .EQ. LMAT)  GO TO 120                     STEP1170
                                                                   STEP1180
         C      ***** PICK UP EM, XU, AND TH FOR ELEMENT NV (WHICH CONSISTS OF   STEP1190
         C            MATERIAL IMAT) *****                         STEP1200
000202          EM= VEM(IMAT)                                      STEP1210
000205          XU= VXU(IMAT)                                      STEP1220
000210          TH= VTH(IMAT)                                      STEP1230
000213          LMAT= IMAT                                         STEP1240
                                                                   STEP1250
         C      ***** CALCULATE THE INSTANTANEOUS STIFFNESS MATRIX AND LOAD      STEP1260
         C            VECTOR FOR ELEMENT NV.                       STEP1270
000214    120   CALL STFNS ( B(LNP   ),B(LBETA ),B(LESIG ),B(LXORD ),B(LYORD)   STEP1280
               1  ,NUMEL )                                         STEP1290
                                                                   STEP1300
000237          CALL GEOM ( B(LNP   ),B(LXORD ),B(LYORD ),B(LDPR  ),B(LLOADE)   STEP1310
               1 ,B(LLOADS),B(LTHETA),B(LESIG ),NUMEL )            STEP1320
                                                                   STEP1330
         C      ***** ADD THE STIFFNESS MATRIX (ST) FOR ELEMENT NV TO THE SYSTEM STEP1340
         C            STIFFNESS MATRIX (A) *****                   STEP1350
000302          DO 150 I=1,6                                       STEP1360
000307          IK=2*I-1                                           STEP1370
000311          L=NP(NV,I)                                         STEP1380
000317          NR=2*L-1                                           STEP1390
000321          DO 150 J=1,6                                       STEP1400
000322          JK=2*J-1                                           STEP1410
000324          M=NP(NV,J)                                         STEP1420
000332          IF (L-M) 149,149,150                               STEP1430
000334   149    NC=2*(M-L)+1                                       STEP1440
000337          A(NR+1,NC)=A(NR+1,NC)+ST(IK+1,JK+1)                STEP1450
000350          A(NR,NC+1)=A(NR,NC+1)+ST(IK,JK+1)                  STEP1460
000362          A(NR,NC)=A(NR,NC)+ST(IK,JK)                        STEP1470
000373          IF (NC-1) 150,150,148                              STEP1480
000376   148    A(NR+1,NC-1)=A(NR+1,NC-1)+ST(IK+1,JK)              STEP1490
000410   150    CONTINUE                                           STEP1500
                                                                   STEP1510
```

```
                                                                               STEP1520
000414          IF (IFIX .EQ. 1)  GO TO 1                                      STEP1530
                                                                               STEP1540
        C       ***** ADD THE FORCE VECTOR (CF) FOR ELEMENT NV TO THE INCREMENTAL STEP1550
        C            LOAD VECTOR (DF) *****                                    STEP1560
000416  114     DO 160 I=1,6                                                   STEP1570
000420          L=NP(NV,I)                                                     STEP1580
000426          DF(2*L-1)=DF(2*L-1)+CF(2*I-1)                                  STEP1590
000433  160     DF(2*L)=DF(2*L)+CF(2*I)                                        STEP1600
                                                                               STEP1610
000441    1     CONTINUE                                                       STEP1620
                                                                               STEP1630
                                                                               STEP1640
000444          IF (NID .LE. 0)  GO TO 200                                     STEP1650
                                                                               STEP1660
000445          IF ( BLANAL .EQ. 0.0 )  GO TO 165                              STEP1670
000446          IF ( IFIX .NE. 0 )  GO TO 165                                  STEP1680
                                                                               STEP1690
        C       ****** NID .GT. 0 AND BLANAL .NE. 0.  SINGLE FOLD OPTION *****  STEP1700
000447          ALPHA= XLNGTH/RSFOLD                                           STEP1710
000451          SINA= SIN(ALPHA)                                               STEP1720
000454          COSA= COS(ALPHA)                                               STEP1730
000456          IEVEN= 0                                                       STEP1740
000457          ILAY= 0                                                        STEP1750
000460          XYFLG= 0.0                                                     STEP1760
        C       ***** CALCULATE  RO = CURRENT INNER FOLD RADIUS *****          STEP1770
000461          RO= RSFOLD - 0.5*BLTHNS                                        STEP1780
000464          R= RO                                                          STEP1790
000465          RSFOLD= DPRAT*RSFOLD                                           STEP1800
000467          NN= NUMNP - NPTSY                                              STEP1810
                                                                               STEP1820
        C       ***** CONSIDER IMPOSED DISPLACEMENTS *****                     STEP1830
000471  165     DO 190 I=1,NID                                                 STEP1840
000476          NR=ND(I)                                                       STEP1850
000501          DISPL= 0.0                                                     STEP1860
                                                                               STEP1870
000502          IF ( BLANAL .EQ. 0.0 )  GO TO 169                              STEP1880
                                                                               STEP1890
```

```
        C      ***** SINGLE FOLD OPTION *****                                 STEP1900
000503         IF ( IFIX .NE. 0 )  GO TO 175                                  STEP1910
                                                                              STEP1920
000504         IF ( XYFLG .NE. 0.0 )  GO TO 166                               STEP1930
                                                                              STEP1940
        C      ***** CALCULATE X-COMPONENT OF IMPOSED DISPLACEMENT *****      STEP1950
000505         DISPL= R*SINA - XORD(NN+1)                                     STEP1960
000511         XYFLG= 1.0                                                     STEP1970
000512         GO TO 175                                                      STEP1980
                                                                              STEP1990
        C      ***** CALCULATE Y-COMPONENT OF IMPOSED DISPLACEMENT *****      STEP2000
000513   166   DISPL= R*COSA - RO - YORD(NN+1)                                STEP2010
000521         XYFLG= 0.0                                                     STEP2020
000522         IF ( IEVEN .NE. 0 )  GO TO 167                                 STEP2030
                                                                              STEP2040
        C      ***** I IS ODD.  PICK UP NEXT DELTA R *****                    STEP2050
000523         IF ( I .EQ. NID )  GO TO 175                                   STEP2060
000525         IEVEN= 1                                                       STEP2070
000526         ILAY= ILAY + 1                                                 STEP2080
000527         DELTAR= 0.5*DEPTH(ILAY)                                        STEP2090
000533         GO TO 168                                                      STEP2100
                                                                              STEP2110
        C      ***** I IS EVEN. USE SAME DELTA R *****                        STEP2120
000533   167   IEVEN= 0                                                       STEP2130
                                                                              STEP2140
000534   168   R= R + DELTAR                                                  STEP2150
000536         NN= NN + 1                                                     STEP2160
000540         GO TO 175                                                      STEP2170
                                                                              STEP2180
000540   169   IF ( IFIX .EQ. 0 )  DISPL= DSD(I)                              STEP2190
                                                                              STEP2200
000544   175   A(NR,1)=1.                                                     STEP2210
                                                                              STEP2220
000552         DO 180 J=2,IBAND                                               STEP2230
000553         L=NR+J-1                                                       STEP2240
000556         IF (NEQ .LT. L)  GO TO 170                                     STEP2250
000560         DF(L)=DF(L)-A(NR,J)*DISPL                                      STEP2260
000570  170    A(NR,J)=0.                                                     STEP2270
```

```
000575          K=NR-J+1                                                            STEP2280
000600          IF (K .LE. 0)  GO TO 180                                            STEP2290
                                                                                    STEP2300
000601          DF(K)=DF(K)-A(K,J)*DISPL                                            STEP2310
000610          A(K,J)=0.                                                           STEP2320
000615    180   CONTINUE                                                            STEP2330
                                                                                    STEP2340
000620    190   DF(NR)=DISPL                                                        STEP2350
                                                                                    STEP2360
                                                                                    STEP2370
000625     200  DO 161 I=1,NEQ                                                      STEP2380
                                                                                    STEP2400
000627    161   DR(I)=DF(I)                                                         STEP2390
000635          DO 240 M=1,NEQBC                                                    STEP2410
000636          NR=NEBC(M)                                                          STEP2420
000641          A(NR,1)=1.                                                          STEP2430
000647          DR(NR)=0.                                                           STEP2440
                                                                                    STEP2450
000651          DO 230 J=2,IBAND                                                    STEP2460
000652          A(NR,J)=0.                                                          STEP2470
000657          L=NR-J+1                                                            STEP2480
000662          IF (L .LE. 0)  GO TO 230                                            STEP2490
000663          A(L,J)=0.                                                           STEP2500
000670    230   CONTINUE                                                            STEP2510
                                                                                    STEP2520
000673    240   CONTINUE                                                            STEP2530
                                                                                    STEP2540
                                                                                    STEP2550
          C     ***** TRANSFORM BAND MATRIX INTO AN UPPER TRIANGULAR MATRIX         STEP2560
          C           ON THE FORWARD PASS *****                                     STEP2570
000675          NR=NEQ-1                                                            STEP2580
000677          DO 350 N=1,NR                                                       STEP2590
000701          M=N-1                                                               STEP2600
000703          PIVOT=A(N,1)                                                        STEP2610
000711          MR=IBAND                                                            STEP2620
000712          LESS=NEQ-M                                                          STEP2630
000714          IF (LESS-IBAND) 321,322,322                                         STEP2640
000717    321   MR=LESS                                                             STEP2650
```

```
                                                                       STEP2660
000721    322  DO 320 L=2,MR                                           STEP2670
               C=A(N,L)/PIVOT                                          STEP2680
000723
000731         I=M+L                                                   STEP2690
000733         J=0                                                     STEP2700
                                                                       STEP2710
000734         DO 300 K=L,MR                                           STEP2720
000736         J=J+1                                                   STEP2730
000740   300   A(I,J)=A(I,J)-C*A(N,K)                                  STEP2740
                                                                       STEP2750
000755         DR(I)=DR(I)-C*DR(N)                                     STEP2760
000762   320   A(N,L)=C                                                STEP2770
                                                                       STEP2780
000772   350   DR(N)=DR(N)/PIVOT                                       STEP2790
                                                                       STEP2800
000777         DR(NEQ)=DR(NEQ)/A(NEQ,1)                                STEP2810
                                                                       STEP2820
                                                                       STEP2830
         C     ***** CALCULATE DISPLACEMENTS (DR) ON THE BACKWARD PASS *****   STEP2840
001005         DO 400 I=2,NEQ                                          STEP2850
001007         M=NEQ-I                                                 STEP2860
001011         N=M+1                                                   STEP2870
001013         MR=IBAND                                                STEP2880
001015         IF (I-IBAND) 401,402,402                                STEP2890
001017   401   MR=I                                                    STEP2900
                                                                       STEP2910
001021    402  DO 400 K=2,MR                                           STEP2920
001023         L=M+K                                                   STEP2930
001025   400   DR(N)=DR(N)-A(N,K)*DR(L)                                STEP2940
                                                                       STEP2950
                                                                       STEP2960
001042         IF (NID .LE. 0) GO TO 720                               STEP2970
001044         DO 710 I=1,NEQ                                          STEP2980
                                                                       STEP3000
001045   710   DF(I)=0.                                                STEP2990
001051    720  IF (IFIX-1)  121,132,132                                STEP3010
                                                                       STEP3020
         C     ***** IFIX .NE. 1. ADD INCREMENTAL FORCE VECTOR (DF) TO TOTAL   STEP3030
```

```
        C             FORCE VECTOR (F) *****                                   STEP3040
001054   121    DO 123 N=1,NEQ                                                 STEP3050
                                                                               STEP3070
001056   123    F(N)=F(N)+DF(N)                                                STEP3060
001064    132   RETURN                                                         STEP3080
001065          END                                                            STEP3090
```

```
SUBPROGRAM LENGTH
001163

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1      - 000441    30     - 000033    40     - 000035    50     - 000101
55     - 000112    60     - 000121    70     - 000150    100    - 000152
111    - 000135    112    - 000170    114    - 000416    120    - 000214
121    - 001054    132    - 001064    148    - 000376    149    - 000334
150    - 000410    165    - 000471    166    - 000513    167    - 000533
168    - 000534    169    - 000540    170    - 000570    175    - 000544
180    - 000615    200    - 000625    230    - 000670    321    - 000717
322    - 000721    401    - 001017    402    - 001021    720    - 001051
902    - 001072    903    - 001102

BLOCK NAMES AND LENGTHS
       - 000001    CDIMEN - 000013    CMESH  - 000004    IDIMEN - 000022
IFLAGS - 000012    INDEX  - 000026    LBUCKT - 000061    MESHIN - 000011
PROPS  - 000012    ST     - 000234

VARIABLE ASSIGNMENTS
ALPHA  - 001142    B      - 000000C01 BLANAL - 000010C05 BLTHNS - 000000C03
C      - 001162    CF     - 000220C12 COSA   - 001144    DELTAR - 001154
DEPTH  - 000010    DISPL  - 001153    DPR    - 000014    DPRAT  - 000003C03
DPRESS - 000001C11 DSD    - 000003    EM     - 000002C11 I      - 001132
IBAND  - 000004C02 IEVEN  - 001145    IFIX   - 000000C06 IK     - 001133
ILAY   - 001146    IMAT   - 000005C06 ISTEP  - 000002C06 J      - 001136
JK     - 001137    JSTEP  - 000003C06 K      - 001155    L      - 001134
LBETA  - 000006C07 LDPR   - 000057C07 LENA   - 000005C02 LESIG  - 000013C07
LESS   - 001161    LLOADE - 000001C07 LLOADS - 000002C07 LMAT   - 000006C06
LNP    - 000000C07 LTHETA - 000045C07 LXORD  - 000015C07 LYORD  - 000016C07
M      - 001140    MATEL  - 000007    MAXST  - 000012C04 MR     - 001160
N      - 001156    NC     - 001141    ND     - 000002    NDP    - 000010C04
NEBC   - 000001    NEQ    - 000002C02 NEQBC  - 000003C02 NID    - 000006C04
NLOAD  - 000005C04 NMAT   - 000007C04 NN     - 001152    NNEQ   - 000016
```

```
RUN 2.3I  05/28/70          STEP2

NNUMEL -  000015    NPTSY  -  000010C02 NR     -  001135    NSTEP  -  000004C06
NUMEL  -  000001C04 NUMNP  -  000002C04 NV     -  000001C06 PIVOT  -  001157
R      -  001151    RECCYL -  000002C05 RESI   -  000000    RSFOLD -  000005C10
RO     -  001150    SINA   -  001143    ST     -  000000C12 TH     -  000004C11
TPR    -  000013    TPRESS -  000010C11 VEM    -  000004    VPRFLG -  000025C06
VTH    -  000006    VXU    -  000005    XLNGTH -  000003C10 XORD   -  000011
XU     -  000003C11 XYFLG  -  001147    YORD   -  000012

START OF CONSTANTS
001067

START OF TEMPORARIES
001114

START OF INDIRECTS
001122

EXTERNAL REFERENCES
         OUTPTC    STFNS     GEOM      SIN       COS       END

COMPILER SPACE
UNUSED -  007600    USED   -  040200
```

```
                 SUBROUTINE STFNS (NP    ,BETA  ,ESIG  ,XORD  ,YORD  ,NNUMEL )      STFN0010
                                                                                    STFN0020
          C            CALCULATES THE INSTANTANEOUS STIFFNESS MATRIX FOR EACH       STFN0030
          C            ELEMENT.                                                     STFN0040
                                                                                    STFN0050
          C      ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.STFN0060
000011           DIMENSION    NP(NNUMEL,6) , BETA(NNUMEL,6) , ESIG(NNUMEL,6,5)      STFN0070
000011           DIMENSION    XORD  (1),YORD  (1)                                   STFN0080
                                                                                    STFN0090
000011           COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ   ,NEQBC ,IBAND ,LENA  ,NEL6   STFN0100
                1   ,NEL64 ,NPTSY ,NPTSX ,NCYCF2                                    STFN0110
000011           COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID    STFN0120
                1   ,NMAT  ,NDP   ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGINSTFN0130
                2   ,NXINT ,NCYCF                                                   STFN0140
000011           COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER  ,IRESI  STFN0150
                1   ,SAVETP,BLANAL,FLIFE                                            STFN0160
000011           COMMON  /INDEX /  IFIX  ,NV    ,ISTEP ,JSTEP ,NSTEP ,IMAT  ,LMAT   STFN0170
                1   ,ERRFLG,J34   ,J43   ,J45   ,I63   ,I69   ,I912  ,LSTEPO,NPRIN  STFN0180
                2   ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS                               STFN0190
000011           COMMON  /IPERM/  IPERM(3)                                          STFN0200
000011           COMMON  /PROPS /  PRESS ,DPRESS,EM    ,XU    ,TH    ,XI    ,YP     STFN0210
                1   ,AREA  ,TPRESS,RMAXS                                            STFN0220
000011           COMMON  /ST    /  ST(12,12)    ,CF(12)                             STFN0230
000011           COMMON  /TEMP  /  TEMP(12,12)                                      STFN0240
000011           COMMON  /UV    /  UV(3,6,2)                                        STFN0250
000011           COMMON  /XYCOOR/  X1,X2,X3,Y1,Y2,Y3                                STFN0260
                                                                                    STFN0270
000011           DIMENSION C(6,4,4),          BA(3,2),SGP(6,4),RMM(12,12)           STFN0280
000011           DIMENSION  AA(12,12),FN(3,3)                                       STFN0290
000011           DIMENSION  CR1(6), CR2(6), CR3(6), CR4(6), CR5(6), CR6(6)          STFN0300
000011           DIMENSION  C33C1( 6),C33C2( 6),C33C3( 6)                           STFN0310
                                                                                    STFN0320
000011           EQUIVALENCE  (R1,X1) , (R2,X2) , (R3,X3)                           STFN0330
                                                                                    STFN0340
000011           DATA  CR1 / 46.0, 7.0, 7.0, 2.0, 1.0, 2.0 /                        STFN0350
000011           DATA  CR2 /  2.0, 7.0, 1.0,46.0, 7.0, 2.0 /                        STFN0360
000011           DATA  CR3 /  2.0, 1.0, 7.0, 2.0, 7.0,46.0 /                        STFN0370
```

```
000011          DATA  CR4 / 50.0,35.0,15.0,50.0,15.0,10.0 /          STFN0380
000011          DATA  CR5 / 10.0,15.0,15.0,50.0,35.0,50.0 /          STFN0390
000011          DATA  CR6 / 50.0,15.0,35.0,10.0,15.0,50.0 /          STFN0400
                                                                     STFN0410
                                                                     STFN0420
000011          I1=NP(NV,1)                                          STFN0430
000015          I2=NP(NV,2)                                          STFN0440
000022          I3=NP(NV,3)                                          STFN0450
000026          X1=XORD(I1)                                          STFN0460
000030          X2=XORD(I2)                                          STFN0470
000032          X3=XORD(I3)                                          STFN0480
000034          Y1=YORD(I1)                                          STFN0490
000036          Y2=YORD(I2)                                          STFN0500
000040          Y3=YORD(I3)                                          STFN0510
                                                                     STFN0520
          C     ***** CALCULATE GEOMETRIC MATRIX  BA *****           STFN0530
000042          DO 120 I=1,3                                         STFN0540
000044          J=IPERM(I)                                           STFN0550
000047          K=NP(NV,I)                                           STFN0560
000053          K1=NP(NV,J)                                          STFN0570
000060          M=IPERM(J)                                           STFN0580
000062          BA(M,2)=XORD(K1)-XORD(K)                             STFN0590
000067    120   BA(M,1)=YORD(K)-YORD(K1)                             STFN0600
                                                                     STFN0610
000076          AREA=0.5*(BA(3,2)*BA(2,1)-BA(2,2)*BA(3,1))           STFN0620
                                                                     STFN0630
000102          ER=EM/(1.-XU-2.*XU*XU)                               STFN0640
000107          CON1= EM/(1.0+XU)                                    STFN0650
000112          CON2= ER*(1.0-XU)                                    STFN0660
000115          CON3= ER*XU                                          STFN0670
000117          CON5= ER*(0.5-XU)                                    STFN0680
                                                                     STFN0690
          C     ***** CALCULATE STRESS DEVIATORS *****               STFN0700
000122          DO 130 I=1,6                                         STFN0710
000123          S11=ESIG(NV,I,1)                                     STFN0720
000134          S22=ESIG(NV,I,2)                                     STFN0730
000144          S33=ESIG(NV,I,J43)                                   STFN0740
000153          HYDR=(S11+S22+S33)/3.                                STFN0750
```

```
000157         SGP(I,1)=S11-HYDR                                              STFN0760
000163         SGP(I,2)=S22-HYDR                                              STFN0770
000166         SGP(I,J34)= ESIG(NV,I,J34)                                     STFN0780
000177   130   SGP(I,J43)= S33 - HYDR                                         STFN0790
                                                                              STFN0800
       C       ***** CALCULATE INSTANTANEOUS STRESS-STRAIN MATRIX  C *****    STFN0810
000206         DO 190 I=1,I63                                                 STFN0820
                                                                              STFN0830
                                                                              STFN0850
000210         CON4= BETA(NV,I)*CON1                                          STFN0840
000216         DO 140 J=1,J34                                                 STFN0860
000217         DO 140 K=J,J34                                                 STFN0870
000220   140   C(I,J,K)= - CON4*SGP(I,J)*SGP(I,K)                             STFN0880
                                                                              STFN0890
000247         C(I,1,1)= C(I,1,1) + CON2                                      STFN0900
000252         C(I,2,2)= C(I,2,2) + CON2                                      STFN0910
000255         C(I,3,3)= C(I,3,3) + CON5                                      STFN0920
000260         C(I,1,2)= C(I,1,2) + CON3                                      STFN0930
                                                                              STFN0940
000263         IF ( RECCYL .EQ. 0.0 )  GO TO 190                              STFN0950
                                                                              STFN0960
       C       ***** CYLINDRICAL COORDINATES *****                            STFN0970
000264         C(I,1,3)= C(I,1,3) + CON3                                      STFN0980
000267         C(I,2,3)= C(I,2,3) + CON3                                      STFN0990
000272         C(I,3,3)= C(I,3,3) + 0.5*ER                                    STFN1000
000276         C(I,4,4)= C(I,4,4) + CON5                                      STFN1010
                                                                              STFN1020
000301   190   CONTINUE                                                       STFN1030
                                                                              STFN1040
                                                                              STFN1050
000304         IF ( RECCYL .EQ. 0.0 )  GO TO 300                              STFN1060
                                                                              STFN1070
       C       ***** CYLINDRICAL COORDINATES *****                            STFN1080
                                                                              STFN1090
000305   290   C2R1 =  2.0*R1                                                 STFN1100
000307         C3R1 =  3.0*R1                                                 STFN1110
000311         C12R1= 12.0*R1                                                 STFN1120
000313         C2R2 =  2.0*R2                                                 STFN1130
```

```
000315          C3R2 =  3.0*R2
000317          C12R2= 12.0*R2
000321          C2R3 =  2.0*R3
000323          C3R3 =  3.0*R3
000325          C12R3= 12.0*R3

           C    ***** CALCULATE COEFFICIENTS USED IN COMPUTING 3*3 MATRICES N(I,J)
000327          C33C1( 1)=  C12R1   + C3R2    + C3R3
000332          C33C1( 2)=  C3R1    + C2R2    + R3
000335          C33C1( 3)=  C3R1    + R2      + C2R3
000340          C33C1( 4)=  C2R1    + C3R2    + R3
000343          C33C1( 5)=  R1      + R2      + R3
000346          C33C1( 6)=  C2R1    + R2      + C3R3
000351          C33C2( 1)=  C33C1( 2)
000353          C33C2( 2)=  C33C1( 4)
000354          C33C2( 3)=  C33C1( 5)
000356          C33C2( 4)=  C3R1    + C12R2   + C3R3
000361          C33C2( 5)=  R1      + C3R2    + C2R3
000364          C33C2( 6)=  R1      + C2R2    + C3R3
000367          C33C3( 1)=  C33C1( 3)
000370          C33C3( 2)=  C33C1( 5)
000372          C33C3( 3)=  C33C1( 6)
000373          C33C3( 4)=  C33C2( 5)
000375          C33C3( 5)=  C33C2( 6)
000376          C33C3( 6)=  C3R1    + C3R2    + C12R3


           C    ***** CALCULATE ELEMENTAL STIFFNESS MATRIX  AA ******

           C    ***** I AND J ARE THE ROW AND COLUMN INDICES OF THE SUBMATRICES
           C          N(I,J) WHICH MAKE UP THE LARGE MATRIX AA *****

           C    ***** K AND L ARE THE ROW AND COLUMN INDICES OF THE INDIVIDUAL
           C          ELEMENTS WITHIN THE LARGE MATRIX AA *****

           C    ***** M AND N ARE THE ROW AND COLUMN INDICES OF THE INDIVIDUAL
           C          ELEMENTS WITHIN EACH MATRIX N(I,J) *****
```

```
000401     300  KK= 1
000402          DO 490  I=1,J34
000404          LL= KK
000406          DO 480  J=I,J34

000410          C1= C(1,I,J)
000416          C2= C(2,I,J)
000424          C3= C(3,I,J)
000432          C4= C(4,I,J)
000440          C5= C(5,I,J)
000446          C6= C(6,I,J)


          C     ***** CALCULATE MATRIX N(I,J) *****

000454     400  II= 1

000455          DO 450 M=1,3

000457          NMIN= M

000461          DO 440 N=NMIN,3

000462          IF ( RECCYL .NE. 0.0 )  GO TO 412

          C     ***** RECTANGULAR COORDINATES *****
000463     411  AAA= ( C1*CR1(II) + C2*CR2(II) + C3*CR3(II) + C4*CR4(II)
               1     + C5*CR5(II) + C6*CR6(II) ) / 80.0
000510          GO TO 420

          C     ***** CYLINDRICAL COORDINATES *****
000510     412  AAA= ( C1*C33C1(II) + C2*C33C2(II) + C3*C33C3(II) ) / 15.0

          C     ***** STORE ELEMENT (M,N) IN MATRIX N(I,J) *****
000522     420  FN(M,N)= AAA

000526          IF (M .NE. N)  FN(N,M)= AAA
```

```
000534    430  II= II + 1                                               STFN1900
000536    440  CONTINUE                                                 STFN1910
                                                                        STFN1920
000540    450  CONTINUE                                                 STFN1930
                                                                        STFN1940
                                                                        STFN1950
          C    ***** STORE SUBMATRIX N(I,J) IN LARGE MATRIX AA *****    STFN1960
                                                                        STFN1970
000542    455  K= KK                                                    STFN1980
000544         DO 470 M=1,3                                             STFN1990
000545         L= LL                                                    STFN2000
000547         DO 460 N=1,3                                             STFN2010
                                                                        STFN2020
000550         AAA= FN(M,N)                                             STFN2030
000554         AA(K,L)= AAA                                             STFN2040
                                                                        STFN2050
000560         IF (I .NE. J)  AA(L,K)= AAA                              STFN2060
                                                                        STFN2070
000567         L= L + 1                                                 STFN2080
000571    460  CONTINUE                                                 STFN2090
                                                                        STFN2100
000573         K= K +1                                                  STFN2110
000574    470  CONTINUE                                                 STFN2120
                                                                        STFN2130
                                                                        STFN2140
000576         LL= LL + 3                                               STFN2150
000600    480  CONTINUE                                                 STFN2160
                                                                        STFN2170
000602         KK= KK + 3                                               STFN2180
000604    490  CONTINUE                                                 STFN2190
                                                                        STFN2200
                                                                        STFN2210
          C    ***** COORDINATE TRANSFORMATION ******                   STFN2220
                                                                        STFN2230
000606         DO 210 L=1,3                                             STFN2240
000610         L1=IPERM(L)                                              STFN2250
000613         L2=IPERM(L1)                                             STFN2260
000615         L3=L+3                                                   STFN2270
```

```
000617          DO 210 N=1,2
000621          DO=BA(L,N)
000625          D1=BA(L1,N)
000631          UV(L,L,N)=3.*DO
000636          UV(L1,L,N)=-DO
000644          UV(L2,L,N)=-DO
000652          UV(L,L3,N)=4.*D1
000660          UV(L1,L3,N)=4.*DO
000666    210   UV(L2,L3,N)=0.

000677          DO 211 I=1,144
000701    211   RMM(I,1)=0.

000706          DO 212 I=1,3
000707          DO 212 J=1,6
000710          RMM(I,J)=UV(I,J,1)
000716          RMM(I+3,J+6)=UV(I,J,2)
000723          II= I + I69
000725          RMM(II,J)= UV(I,J,2)
000733    212   RMM(II,J+6)= UV(I,J,1)

000745          IF ( RECCYL .EQ. 0.0 )  GO TO 214

          C     ***** CYLINDRICAL COORDINATES *****
000746          I4= NP(NV,4)
000753          I5= NP(NV,5)
000757          I6= NP(NV,6)
000764          AREA2= 2.0*AREA
000766          RMM( 7,1)= AREA2/R1
000770          RMM( 8,2)= AREA2/R2
000772          RMM( 9,3)= AREA2/R3

000774    214   DO 213 I=1,I912
000776          DO 213 J=1,6
000777          TEMP(I,2*J-1)=RMM(I,J)
001007    213   TEMP(I,2*J)=RMM(I,J+6)

001022          DO 215 I=1,I912
```

```
001023          DO 215 J=1,12                                          STFN2660
001024   215    RMM(I,J)=TEMP(I,J)                                     STFN2670
                                                                       STFN2680
         C      ***** CALCULATE CONVENTIONAL STIFFNESS MATRIX *****    STFN2690
001042          DO 134 I=1,I912                                        STFN2700
001043          DO 134 J=1,12                                          STFN2710
001044          TTEMP= 0.0                                             STFN2720
001045          DO 1340 K=1,I912                                       STFN2730
                                                                       STFN2750
001047    1340 TTEMP= TTEMP + AA(I,K)*RMM(K,J)                         STFN2740
001064    134   TEMP(I,J)= TTEMP                                       STFN2760
                                                                       STFN2770
001075          DO 135 I=1,12                                          STFN2780
001077          DO 135 J=1,12                                          STFN2790
001100          SST= 0.0                                               STFN2800
001101          DO 1350 K=1,I912                                       STFN2810
                                                                       STFN2830
001103    1350 SST= SST + RMM(K,I)*TEMP(K,J)                           STFN2820
001120    135   ST(I,J)= SST*TH/(48.0*AREA)                            STFN2840
                                                                       STFN2850
                                                                       STFN2860
001133          RETURN                                                 STFN2870
001133           END                                                   STFN2880
```

```
RUN 2.3I  05/28/70          STFNS


SUBPROGRAM LENGTH
002221

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
140     -  000220     190     -  000301     211     -  000701     214     -  000774
215     -  001024     290     -  000305     300     -  000401     400     -  000454
411     -  000463     412     -  000510     420     -  000522     430     -  000534
455     -  000542     1340    -  001047     1350    -  001103

BLOCK NAMES AND LENGTHS
CDIMEN  -  000013     IDIMEN  -  000022     IFLAGS  -  000012     INDEX   -  000025
IPERM   -  000003     PROPS   -  000012     ST      -  000234     TEMP    -  000220
UV      -  000044     XYCOOR  -  000006

VARIABLE ASSIGNMENTS
AA      -  001617     AAA     -  002204     AREA    -  000007C06 AREA2   -  002216
BA      -  001341     C       -  001201     CF      -  000220C07 CON1    -  002147
CON2    -  002150     CON3    -  002151     CON4    -  002157     CON5    -  002152
CR1     -  002050     CR2     -  002056     CR3     -  002064     CR4     -  002072
CR5     -  002100     CR6     -  002106     C1      -  002173     C12R1   -  002162
C12R2   -  002165     C12R3   -  002170     C2      -  002174     C2R1    -  002160
C2R2    -  002163     C2R3    -  002166     C3      -  002175     C3R1    -  002161
C3R2    -  002164     C3R3    -  002167     C33C1   -  002114     C33C2   -  002122
C33C3   -  002130     C4      -  002176     C5      -  002177     C6      -  002200
DO      -  002211     D1      -  002212     EM      -  000002C06 ER      -  002146
FN      -  002037     HYDR    -  002156     I       -  002141     II      -  002201
IPERM   -  000000C05 I1      -  002136     I2      -  002137     I3      -  002140
I4      -  002213     I5      -  002214     I6      -  002215     I63     -  000013C04
I69     -  000014C04 I912    -  000015C04 J       -  002142     J34     -  000010C04
J43     -  000011C04 K       -  002143     KK      -  002171     K1      -  002144
L       -  002205     LL      -  002172     L1      -  002206     L2      -  002207
L3      -  002210     M       -  002145     N       -  002203     NMIN    -  002202
NV      -  000001C04 RECCYL  -  000002C03 RMM     -  001377     R1      -  000000C12
R2      -  000001C12 R3      -  000002C12 SGP     -  001347     SST     -  002220
```

```
ST      -  000000C07 S11     -  002153    S22     -  0C2154    S33     -  002155
TEMP    -  000000C10 TH      -  000004C06 TTEMP   -  0C2217    UV      -  000000C11
XU      -  000003C06 X1      -  000000C12 X2      -  0C0001C12 X3      -  000002C12
Y1      -  000003C12 Y2      -  000004C12 Y3      -  0C0005C12

START OF CONSTANTS
001135

START OF TEMPORARIES
001152

START OF INDIRECTS
001166

EXTERNAL REFERENCES
         END

COMPILER SPACE
UNUSED -  007100    USED    -  040700
```

```
                SUBROUTINE WRST2 (NP     ,LOADE ,LOADS ,DLIM   ,VEM    ,VXU    ,VTH
               1    ,VXI    ,VYP    ,DELTAP,MATEL ,MLAYER,CYCM   ,CYCZ   ,ND     ,DSD
               2    ,VPR    ,DEPTH )

          C           WRITES THE SECOND CATEGORY OF RESTART DATA (INPUT ARRAYS)
          C           ONTO THE RESTART TAPE

          C     ***** THE FOLLOWING DIMENSION SPECIFICATIONS ARE DUMMY DIMENSIONS.
000025          DIMENSION    NP     (1),LOADE (1),LOADS (1),DLIM   (1),VEM    (1)
               1   ,VXU    (1),VTH    (1),VXI    (1),VYP    (1),DELTAP(1),MATEL (1)
               2   ,MLAYER(1),CYCM   (1),CYCZ   (1),ND     (1),DSD    (1),VPR    (1)
               3   ,DEPTH (1)

000025          COMMON  /CDIMEN/  NCDIM ,LREC1 ,NEQ    ,NEQBC ,IBAND ,LENA   ,NEL6
               1    ,NEL64 ,NPTSY ,NPTSX ,NCYCF2
000025          COMMON  /CMESH /  BLTHNS,XINTN ,XYRAT ,DPRAT
000025          COMMON  /IDIMEN/  NDIMEN,NUMEL ,NUMNP ,NUMCP ,NUMBC ,NLOAD ,NID
               1    ,NMAT  ,NDP    ,MAXBW ,MAXST ,NNDLIM,MAXBKT,NLAYER,LWRAT ,NANGIN
               2    ,NXINT ,NCYCF
000025          COMMON  /IFLAGS/  NFLGS ,RESTRT,RECCYL,CONVER,IPRIN ,ITER   ,IRESI
               1    ,SAVETP,BLANAL,FLIFE
000025          COMMON  /INDEX /  IFIX   ,NV     ,ISTEP ,JSTEP ,NSTEP ,IMAT   ,LMAT
               1    ,ERRFLG,J34    ,J43    ,J45    ,I63    ,I69    ,I912  ,LSTEPO,NPRIN
               2    ,NXTLS ,INDMP ,LSLIM ,IDUMP ,ILSS   ,VPRFLG
000025          COMMON  /IOUNIT/  IRSIN,IRSOUT
000025          COMMON  /LSSAVE/  LSSAVE(2,10)
000025          COMMON  /MESHIN/  NMESH ,XSTART,YSTART,XLNGTH,RPFOLD,RSFOLD,RPTRAT
               1    ,RSTRAT,RFINAL


000025          WRITE (IRSOUT) NEQ,VPRFLG,DPRAT,XLNGTH,BLTHNS,NPTSY,LSSAVE

000046          WRITE (IRSOUT)  ( NP(I   ),I=1,NEL6 )

000066     300  IF ( NLOAD .LE. 0 )  GO TO 400
000073          WRITE (IRSOUT)  ( LOADE(I),I=1,NLOAD ),  ( LOADS(I),I=1,NLOAD )
000121          IF ( VPRFLG .NE. 0.0 )  WRITE (IRSOUT)  ( VPR(I),I=1,NLOAD )
```

```
                                                                                WRST0380
000144     400  IF ( NNDLIM .LE. 0 )  GO TO 500                                 WRST0390
                NNDL3= 3*NNDLIM                                                 WRST0400
000146
000147          WRITE (IRSOUT)  ( DLIM(I  ),I=1,NNDL3 )                         WRST0410
                                                                                WRST0420
000164     500  WRITE (IRSOUT)  ( VEM(I),I=1,NMAT ), ( VXU(I),I=1,NMAT )        WRST0430
               1  , ( VTH(I),I=1,NMAT ), ( VXI(I),I=1,NMAT ),( VYP(I),I=1,NMAT ) WRST0440
                                                                                WRST0450
000253     600  IF ( NDP .LE. 0 )  GO TO 700                                    WRST0460
000260          NDP2= 2*NDP                                                     WRST0470
000262          WRITE (IRSOUT)  ( DELTAP(I  ),I=1,NDP2 )                        WRST0480
                                                                                WRST0490
000274     700  IF ( NMAT .LE. 1 )  GO TO 800                                   WRST0500
000302          WRITE (IRSOUT)  ( MATEL(I),I=1,NUMEL )                          WRST0510
                                                                                WRST0520
000314     800  IF ( NID .LE. 0 )  GO TO 900                                    WRST0530
000321          WRITE (IRSOUT)  ( ND(I),I=1,NID ), ( DSD(I),I=1,NID )           WRST0540
                                                                                WRST0550
000341     900  IF ( BLANAL .EQ. 0.0 )  GO TO 2000                              WRST0560
000346          WRITE (IRSOUT) ( DEPTH(I),I=1,NLAYER )                          WRST0570
000360          IF ( FLIFE .NE. 0.0 )  WRITE (IRSOUT)  ( MLAYER(I),I=1,NLAYER ) WRST0580
               1  ,( CYCM(I),I=1,NMAT ), ( CYCZ(I),I=1,NMAT )                   WRST0590
                                                                                WRST0600
000417    2000  RETURN                                                          WRST0610
000420           END                                                            WRST0620
```

```
SUBPROGRAM LENGTH
000436

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
300     -  000066    400     -  000144    500     -  000164    600     -  000253
700     -  000274    800     -  000314    900     -  000341    2000    -  000417

BLOCK NAMES AND LENGTHS
CDIMEN  -  000013    CMESH   -  000004    IDIMEN  -  000022    IFLAGS  -  000012
INDEX   -  000026    IDUNIT  -  000002    LSSAVE  -  000024    MESHIN  -  000011

VARIABLE ASSIGNMENTS
BLANAL  -  000010C04 BLTHNS  -  000000C02 CYCM    -  000006    CYCZ    -  000007
DELTAP  -  000003    DEPTH   -  000013    DPRAT   -  000003C02 DSD     -  000011
FLIFE   -  000011C04 I       -  000433    IRSOUT  -  000001C06 LSSAVE  -  000000C07
MATEL   -  000004    MLAYER  -  000005    ND      -  000010    NDP     -  000010C03
NDP2    -  000435    NEL6    -  000006C01 NEQ     -  000002C01 NID     -  000006C03
NLAYER  -  000015C03 NLOAD   -  000005C03 NMAT    -  000007C03 NNDLIM  -  000013C03
NNDL3   -  000434    NPTSY   -  000010C01 NUMEL   -  000001C03 VPR     -  000012
VPRFLG  -  000025C05 VTH     -  000000    VXI     -  000001    VYP     -  000002
XLNGTH  -  000003C10

START OF CONSTANTS
000422

START OF TEMPORARIES
000424

START OF INDIRECTS
000433

EXTERNAL REFERENCES
          OUTPTB    END
```

```
RUN 2.3I  05/28/70          WRST2

COMPILER SPACE
UNUSED -  011600    USED   -  036200
```

****

TRW/S+ISD SCOPE SYSTEM
SYSTEM STATUS REPORT NO. 80
MAY 25, 1970

****

1. THE -RETURN- CONTROL CARD WILL NOW AUTOMATICALLY REWIND OR UNLOAD.

2. THE -DMP- CONTROL CARD WILL NOT BE HONORED FOR CLASSIFIED JOBS THAT DO NOT HAVE FILE -OUTPUT- ON TAPE.

3. A NEW VERSION OF SUBROUTINE QRAL IS AVAILABLE FROM THE LIBRARIAN WHICH CORRECTS AN ERROR ASSOCIATED WITH DIVISION BY ZERO.

****

```
  05/28/70   CDC 6500 SECURE SCOPE 3.1.5  25 MAY 1970
16.33.58.$INPUT=  03311
16.33.59.$SEQUENCE,E609.
16.33.59.$NAME,532310,35125,TAYLOR,J.
16.33.59.$PROBLEM,334005.
16.33.59.$PROGRAM,AS189E.
16.33.59.$MAXTIM,60.         1 MINUTES
16.33.59.$FLENGTH,50000.     20480 CELLS
16.33.59.$TAPE,OUTPUT,S,PR,X,C1.
16.35.08. (54 ASSIGNED)      *****REEL=C667
16.35.10.$FL=050000 CP=00000.000 PP=00003.917 XX=00000.000  FC=143200
16.35.10.RUN(S)              COMPILE WITH LIST,NO EXECUTE
16.35.12.FILE OPENED ON DISK-----LGO
16.35.18.MT 54  WRITE PARITY ERROR   RECOVERED   PRU = 000347
16.35.54.$CP  00023.647 SEC.
16.35.54.$PP  00050.146 SEC.
16.35.54.$XX  00000.000 SEC.
16.35.54.$DISK=  000005
16.35.54.EST. PRINTED OUTPUT  00000 LINES
16.35.54.EST. PUNCHED OUTPUT  00000 CARDS
16.35.54.ESTIMATED COST = $0004.46 +$0005.00 =$0009.46
16.35.54.CONTROL POINT 6. FC=142600
```