61-

NASA CR-66364

COMPUTER PROGRAMS FOR THE PLASTIC ANALYSIS OF STRUCTURES USING DISCRETE ELEMENT METHODS

by

H. Armen, Jr. and A. Pifko

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

N67-30768 (ACCESSION NUMBER)
191 (PAGES)
CR-66364 (NASA CR OR TMX OR AD NUMBER)
(THRU)
0 (CODE)
32 (CATEGORY)
FACILITY FORM 602

Prepared Under Contract NAS 1-5040 by
GRUMMAN AIRCRAFT ENGINEERING CORPORATION
Bethpage, New York

GPO PRICE $ _____

CFSTI PRICE(S) $ _____

Hard copy (HC) 3.00

Microfiche (MF) .65

ff 653 July 65

FOREWORD

This report was prepared by the Grumman Aircraft Engineering Corporation, Bethpage, New York, under Contract NAS 1-5040 entitled, "A Research Study for the Development of a Digital Method of Analysis of Supersonic Transport Aircraft Structures in the Plastic Range." The work was performed by the Research Department of Grumman Aircraft Engineering Corporation, with support from the Structural Mechanics Section of the Engineering Department.

The authors gratefully acknowledge the valuable contributions made by Mr. Paul Hornack and Miss Eloise Turner in the preparation of the computer programs. They would also like to thank Mr. Francis J. Nolan for much helpful advice on computing problems, for the preparation of a special eigenvalue subroutine, and the sections entitled, Solution of the Eigenvalue Problem Det $([A] - \lambda[B]) = 0$, and Discussion of Determinant Calculation.

COMPUTER PROGRAMS FOR THE PLASTIC ANALYSIS OF STRUCTURES USING DISCRETE ELEMENT METHODS

by

H. Armen, Jr. and A. Pifko

ABSTRACT

This report is a users document for two computer programs which were developed in a study reported in NASA CR-803 to extend finite element methods to calculate failure loads of aeronautical structures. The two programs deal with inelastic stress analysis of finite element structures in the presence of load cycling and the calculation of bifurcation type plastic buckling loads of rectangular plates. The stress analysis program takes into account the Bauschinger effect for biaxial stress states using a plasticity theory based on Ziegler's modification to Prager's kinematic hardening theory. The buckling program extends finite element methods to plastic buckling using Stowell's formulation for implementing a deformation plasticity theory into the buckling theory. The theoretical studies carried out in the development of these two programs as well as some research results obtained with the programs are given in NASA CR-803.

TABLE OF CONTENTS

Item	Page
Introduction	1
Cyclic Plastic Analysis Computer Programs	2
Program Description	2
Instructions for Preparing Data Tape	4
Preparation of Data Cards	6
Symbols and Format of the Data Cards	7
Restart Procedure	10
Special Instruction Cards	10
Output	11
Sample Output	13
Time Estimates	18
Buckling Computer Programs	19
General Program Discussion	19
Constant Stress Field Eigenvalue Program	20
General Eigenvalue Program	21
Program Input	21
Sequencing and Details of the Data Cards	22
Symbols and Format of the Data Cards	24
Program Output	28
Eigenvalue Programs	29
Determinant Program	30
Sample Output	30
List of Subroutines	37
Time Estimate	37
Solution of the Eigenvalue Problem Det ([A] - λ[B]) = 0	38
Discussion of Determinant Calculation	43

Item	Page

Appendix A - Flow Charts and Listings for the Plastic Analysis of Stable Structures 44

Stress Program 44

Strepp Program 65

Strain Program 80

Strapp Program 100

Appendix B - Flow Charts and Listings for Plastic Buckling of Rectangular Plates 114

Flow Charts 114

Listing for Eigenvalue Programs 125

Listing for Determinant Programs 165

INTRODUCTION

This report discusses and documents the computer programs written to implement the discrete element methods developed for the plastic analysis of complex structures in CR-803. The methods in CR-803 fall into two categories: those applicable to stable structures and useful in the prediction of failure associated with excessive plastic straining, and those applicable to the determination of buckling loads.

In the methods of the former category, particular attention is given to the case of intermittently applied loads causing successive excursions into the plastic range, including reversals of stress into the plastic range. In order to accommodate this case and to take the Bauschinger effect into account, the plasticity theory selected for use is the kinematic hardening theory of Prager, as modified by Ziegler.

The methods in this category are based on the application of two matrix relations coming out of a linear elastic analysis of the structure and involving the concept of initial strain. The effect of plasticity is introduced by interpreting plastic strains as initial strains.

The problem of plastic buckling is treated as a bifurcation phenomenon, with attention concentrated on flat rectangular plates. A previously developed discrete-element method for elastic buckling analysis is extended to take into account the effect of plastic deformation. It is based on the matrix displacement method of structural analysis used in conjunction with a deformation theory of plasticity.

The present report is separated into two parts. The first part is a discussion of the programs associated with the methods developed in CR-803 for the analysis of complex, stable structures subjected to loads causing biaxial membrane stress and stress reversal into the plastic range. The second part of the report discusses the programs written to implement the methods developed for the plastic buckling analysis of initially flat rectangular panels.

All the programs were written in FORTRAN IV to run on the IBM 7094-II. These are two-channel (A and B) machines with 8 tape drives on each channel; 32,768 words of core storage; on-line card reader; on-line printer and printer clock. The programs are set to run under FORTRAN SIBSYS control which uses an SJØB card for identification. Input is on logical tape 5 (A-2), print-out on logical tape 6 (B-1), and punched output on logical tape 7 (B-2), which is not used by these programs.

CYCLIC PLASTIC ANALYSIS COMPUTER PROGRAMS

Program Description

Incremental methods for the analysis of structures subjected to intermittent loads causing biaxial membrane stress and stress reversal into the plastic range are presented in Section 3 of CR-803. The methods employ the linear matrix equation associated with discrete-element analysis of elastic structures in conjunction with incremental constitutive relations from the plasticity theory. The governing incremental linear matrix equation, which includes the effects of initial strain, can be written for the increments of stress or increments of total strain in the elements. The use of the former type of equation leads to the so-called Stress Method, and the latter type of equation leads to the Strain Method. A

solution to the linear matrix equation, expressed in terms of either increments of stress or increments of total strain, may be obtained by utilizing one of two alternative procedures. The first procedure uses estimated values of plastic strain increment based on values computed in the preceding step; such a procedure may be termed a predictor procedure. A second procedure, termed the stepwise linearization procedure, incorporates the linear incremental plastic strain-stress relation directly into the linear matrix equation, thus avoiding the use of estimated values of plastic strain increment.

Computer programs using these two solution procedures have been developed and are identified by the following symbolic names:

STRESS - Stress method, stepwise linearization procedure

STREPP - Stress method, predictor procedure

STRAIN - Strain method, stepwise linearization procedure

STRAPP - Strain method, predictor procedure.

In all of the programs listed above:

Loading conditions are restricted to those which cause states of plane stress.

Cycling can only be specified by a load range, the amplitude of which can vary.

Any number of load cycles can be considered. A capability is provided to allow computations to be stopped at some intermediate (predetermined) point in a loading sequence and be subsequently continued.

Materials exhibiting elastic-plastic behavior involving ideal plasticity, linear or nonlinear strain hardening, or limited strain hardening can be treated.

Input and output formats are identical for all the programs. Input formats have been prepared so as to minimize the manipulation of data necessary to treat various structures.

The program will accept a binary input data tape on logical tape 11 (B-6), and will write a binary save tape to be used for a continuation run on logical tape 10 (B-5).

Instructions for Preparing Data Tape

The programs require two influence coefficient matrices, associated with the governing linear matrix equation, to be read from a binary input data tape. In the Stress Method, the first of these input matrices relates the element stresses in the redundant structure to initial strains; the second input matrix relates the element stresses to the applied loading. In the Strain Method, these matrices relate element strains to initial stresses and element strains to the applied loading, respectively.

Due to core storage limitations, it is necessary to delete rows and columns of the input matrices associated with regions of the given structure which are known, on the basis of an elastic analysis, to be subjected to low stress intensities and can be expected to remain elastic for loadings of practical interest. Because of the nature of the analytical methods used, explicit consideration need not be given to the stresses and strains in the elements of these regions. Their influence on the remainder of the structure is implicit in the influence coefficient data which are retained for use. Care must, of course, be taken, in the selection of these regions, that elements which are presumed to remain elastic do not, in fact, become plastic in the range of loadings applied.

The initial strain (or initial stress) influence coefficient matrix is a square array of order equal to three times the number of elements used to represent that portion of the idealized structure treated explicitly in the analysis. Because of computer core storage limitations, the entire square array is not initially read into the program. Instead, columns of this array, necessary to perform the required computations, are read individually from the binary data tape and stored in core. The choice and number of columns to be read and stored at any level of load depends upon the number of elements in the plastic range at that load. The number of columns needed is equal to three times the number of plastic elements. The resulting array is referred to as SIJ in the programs.

The influence coefficients relating stress (or strain) to the applied loading must be written as a column vector. The number of rows in this vector is equal to three times the number of elements used to represent that portion of the idealized structure considered explicitly in the analysis. If the structure is subjected to a system of multiple loads, the vector of influence coefficients associated with each load must be added to form one column vector. Consequently, only proportional loading can be treated in this case. The entire column vector is read into the program, and is referred to as SIM.

The input tape which contains these influence coefficient matrices must be generated in FORTRAN IV and all data must be contained in one file. The record size is equal to three times the number of elements considered in the idealized structure. The initial strain (or initial stress) array is written column-sort on the tape first, each column constituting a FORTRAN record. A dummy record in which the first word is 111111111111 is used to

separate the two input matrices. The SIM column vector follows the dummy record and an END OF FILE is used to signify the end of input data on the tape.

Preparation of Data Cards

The data cards are read in the following sequence and appear immediately after a SDATA card.

> The first data card is used as an indicator to determine whether or not the run is a continuation of a previous run; the card contains two parameters IRSTRT and RMXLD. When the loading sequence is not a continuation of a previous one, a blank card is used for the first data card.
>
> General input parameters which remain fixed during a loading sequence are contained in the second data card, and are: NØDE, NCØMP, MCØL, and INT.
>
> The third data card contains material parameters which remain constant during the loading sequence. These constants are represented by the following: SIGO, E, GNU, and ZFL.
>
> Data which must be read for each individual loading (or reloading) situation consist of the following: PMA, DELP, D, T, RMXLD, KLU, and NRSTRT. Additional data cards of this type must be added for each subsequent loading in a desired sequence.

The last card following the individual loading data cards must be blank. This card stops the program by reading a zero for PMA. In the event the job is to be continued at another time, the program will write a binary saved data tape in a prescribed format. The decision to write this tape must be made in advance and brought to the attention of the operator on a Special Instruction Card (see section concerning the Special Instructions). Built-in pauses in the program are as follows:

Pause 1: Mount input data tape on logical tape 9 drive A-5, and saved binary data (if a continuation run) on logical tape 11 drive B-6.

Pause 2: Demount input data tape and save tape (for restart) from drives A-5 and B-5, respectively, after the program has been completed.

Symbols and format of the data cards. -

Restart indicator card - Format 1

Columns	Field	Symbol	Comments
1-5	I5	IRSTRT	For IRSTRT > 0, the program to be run is a continuation of a previous run. Saved binary data are to be read from tape B-6, FORTRAN logical unit 11. For IRSTRT = 0, the program is a new run and no saved input tape are to be read.
6-20	E15.7	RMXLD	New maximum (or minimum) load level; to be defined when the run is a continuation run, and the new maximum load level is greater than the maximum load level of the previous run. For unloading and reloading, the RMXLD field should be left blank. The new maximum (or minimum) load level will be defined on the Format 4 data card.
21-80			Ignored.

General input parameters - Format 2

Columns	Field	Symbol	Comments
1-10	I10	NØDE	Number of elements of the idealized structure to be considered in the program. Because of core storage limitations, NØDE will not necessarily equal the total number of elements used in the idealization of the actual structure.
11-20	I10	NCØMP	Total number of stress (or strain) components (three times NODE).
21-30	I10	MCØL	Maximum number of columns of the SIJ array that can be read into core storage.
31-40	I10	INT	An index which multiplies the load increment to determine the interval of print out. Prints are generated when the load level, beyond the elastic range, has been incremented by an integral multiple of the interval of print out.
41-80			Ignored.

Material constants - Format 3

Columns	Field	Symbol	Comments
1-15	E15.7	SIGO	Yield stress in tension.
16-30	E15.7	E	Young's modulus.
31-45	E15.7	GNU	Poisson's ratio.
46-60	E15.7	ZFL	Used to detect the onset of perfect plasticity in cases of limited strain hardening (α_o^2). ZFL = 0 for elastic, perfectly plastic behavior; ZFL < 0 for elastic, unlimited strain hardening behavior.
61-80			Ignored.

Individual loading data - Format 4

Columns	Field	Symbol	Comments
1-15	E15.7	PMA	Dummy applied load used to compute the load level necessary to cause yielding from an elastic state. PMA takes on the same sign as RMXLD.
16-30	E15.7	DELP	Load increment. DELP has the same sign as RMXLD.
31-45	E15.7	D	A parameter used in the Ramberg-Osgood Stress-Strain Relation ($\sigma_{0.7}$).
46-60	E15.7	T	The shape parameter used in the Ramberg-Osgood Stress-Strain Relation (n).
61-75	E15.7	RMXLD	Maximum load level (+), or minimum load level (-).
76-77	I2	KLU	Index for the type of material behavior being considered. KLU = 0 — Elastic, limited or unlimited strain hardening; KLU ≠ 0 — Elastic, perfectly plastic.
78-80	I3	NRSTRT	Index used to determine whether the program is to write a save tape after the load level has reached the value of RMXLD given on this card. NRSTRT = 0 — No save tape is to be written; NRSTRT ≠ 0 — Binary save tape is to be written.

Any number of Format 4 data cards will be accepted by the program. If it is desired to write a binary save tape for restarting the program at a later time, from the value of RMXLD indicated on the last Format 4 card, only that card will have a nonzero value for NRSTRT. A blank card must follow the last Format 4 card.

Restart procedure. -

The program has been written to enable the user to restart the program from a point of completion of a given loading sequence, once the necessary binary saved tape has been written. This restart capability enables the program to continue loading to a new maximum level above the previous level or to unload elastically from the previous level and reload in the reverse direction.

Only Format 1 and Format 4 data cards are used for a restart run. Information contained on the Format 2 and Format 3 data cards are provided on the saved binary tape. The first data card acts as a control to the program; any positive number for IRSTRT in columns 1-5 on the Format 1 card indicates a continuation of a previous run. Saved binary data are read from FORTRAN logical tape 11 (B-6). The final Format 4 data card also acts as a control to the program; any positive number for NRSTRT in columns 78-80 indicates that a binary save tape will be written on FORTRAN logical tape 10 (B-5).

Special instruction cards. -

The programs require special input tapes and may require the writing of a save tape to be used for a continuation run. For this reason Special Instruction Cards are employed to relate, to the computer operator, any information regarding the tapes to be mounted at the start of any run and the tapes to be saved at its conclusion. All programs require a Special Instruction Card which must contain the following information:

AT PAUSE 1 MOUNT AN INPUT REEL ON A-5; AT PAUSE 2 SAVE A-5.

In addition, if the run is an original loading sequence to be continued at a later time, or is a restart of a previous run to be completed, or is a restart to be continued at a later time, then

additional instructions are required. These instructions are given to the operator in the form of the following statements:

When original loading is to be continued:

AT PAUSE 1 MOUNT BLANK REEL ON B-5; AT PAUSE 2 SAVE B-5.

When a restart is to be completed:

AT PAUSE 1 MOUNT SAVED BINARY DATA ON B-6; AT PAUSE 2 SAVE B-6.

If restart is to be continued:

AT PAUSE 1 MOUNT BLANK REEL ON B-5 AND REEL CONTAINING SAVED BINARY DATA ON B-6; AT PAUSE 2 SAVE B-5 AND B-6.

Output

Printed output begins with a listing of the various input parameters used in the particular loading case. Calculations performed for each load increment need not be listed. The amount of generated output, during any loading sequence, is determined by the value of the interval of print out, INT. The following information is listed at each interval of printing:

Holorith Statement	Comments
LOAD LEVEL, P =	Current load level
ELEMENT STRESSES	
NØDE NØ	Node or element number
SIGMA - X SIGMA - Y TAU - XY	Components of the stress vector, listed row-sort for each element.

The following information is listed only for those elements that are plastic:

Holorith Statement	Comments
ELEMENT NØ	
C	The hardening coefficient. Values of C listed as ±1 indicate that the element is perfectly plastic.

DLAM	Scalar factor, $d\lambda$, used to evaluate the plastic strain increment.
DMU	Scalar factor, $d\mu$, used to evaluate the increment of translation of the center of the loading surface.
DELTA ALPHA - X DELTA ALPHA - Y DELTA ALPHA - XY	Components of the increment of translation of the center of the loading surface, $d\alpha_{ij}$.
DELTA EP - X DELTA EP - Y DELTA GAM - XY	Components of plastic strain increment, $d\epsilon_{ij}$.
TØT ALPHA - X TØT ALPHA - Y TØT ALPHA - XY	Components of total translation of the center of the yield surface α_{ij}.
PLASTIC EP - X PLASTIC EP - Y PLASTIC GAM - XY	Components of plastic strain, ϵ_{ij}.
STRAIN - X STRAIN - Y STRAIN - XY	Components of total strain, (elastic plus plastic), e_{ij}.
ELASTIC EP - X ELASTIC EP - Y ELASTIC GAM - XY	Components of elastic strain, e'_{ij}.

The generated output begins at a load level equal to the sum of the maximum elastic load plus the load increment. The output is listed for each interval of printout until the value of the load reaches the maximum (or minimum) load, specified by RMXLD on the Format 4 data card. The program determines whether the structure can be unloaded elastically to a zero-load state, i.e., whether the loads can be removed from the structure without additional plastic deformation occurring. If the states of stress and strain are such as to allow elastic unloading to a zero-load condition, then

values of residual stresses and strains are computed and listed. If however, states of stress and strain at the maximum (or minimum) load are such that plastic behavior commences prior to a complete unloading, then additional incremental computations are performed in accordance with the chosen method of analysis. These computations are continued until the load level is as close to zero as the load increments will allow. The values of stress and strain listed at this load level will be the residual values.

Sample output. -

A listing of output generated by the STRAIN Program is given on the next page. The structure for which the computations have been performed is the stiffened panel discussed in Section 4 of CR-803; the maximum load level is 15,055 lbs, the load increment used is 100 lbs; and elastic, limited strain-hardening behavior is assumed. Since the load level, P, is at the desired maximum load, a listing of residual stresses and strains follows the usual output.

LOAD LEVEL, P = 15055.

ELEMENT STRESSES

NODE NO	SIGMA-X	SIGMA-Y	TAU-X,Y
1	0.40993539E 03	0.52729180E 05	0.
2	0.19357253E 04	0.50237689E 05	0.
3	0.19997162E 04	0.50341998E 05	0.
4	0.36635876E 04	0.35282353E 05	0.
5	0.11948036E 04	0.20571577E 05	0.
6	0.56490750E 03	0.97879658E 04	0.
7	0.37233192E 03	0.26091374E 04	0.
8	0.45776367E-04	-0.37666453E 04	0.
9	-0.91172142E 04	0.39099899E 05	0.
10	-0.63770333E 04	0.42530307E 05	-0.56845198E 04
11	-0.10026782E 05	0.41597462E 05	-0.69411366E 04
12	-0.33854610E 04	0.33958463E 05	-0.21632437E 04
13	-0.39500787E 03	0.20920052E 05	-0.10534794E 03
14	0.41891903E 03	0.98229513E 04	0.76918271E 01
15	0.35054608E 03	0.25947570E 04	0.98382661E 02
16	0.30517578E-04	-0.37070724E 04	0.14457560E 03
17	-0.43234074E 04	0.25799431E 05	0.
18	-0.37015983E 04	0.30954811E 05	-0.81982489E 04
19	-0.10541645E 05	0.28033692E 05	-0.11814749E 05
20	-0.77460044E 04	0.28916422E 05	-0.65862449E 04
21	-0.28921286E 04	0.20434495E 05	-0.23352049E 04
22	0.23948272E 02	0.97167692E 04	-0.54568052E 03
23	0.30158047E 03	0.25352289E 04	0.14386502E 03
24	0.15258789E-04	-0.35353246E 04	0.30511376E 03
25	-0.11397438E 04	0.19102821E 05	0.
26	-0.12171480E 04	0.20355691E 05	-0.75509665E 04
27	-0.60857397E 04	0.19342152E 05	-0.12759217E 05
28	-0.57562693E 04	0.19865705E 05	-0.98318718E 04
29	-0.33063781E 04	0.16360700E 05	-0.53757850E 04
30	-0.43690738E 03	0.82385841E 04	-0.17211830E 04
31	0.38146973E-04	-0.29024561E 04	0.46636247E 03
32	-0.32264270E 03	0.16458616E 05	0.
33	-0.29955005E 03	0.16871911E 05	-0.71949854E 04
34	-0.22475244E 04	0.16019428E 05	-0.13141131E 05
35	-0.17163248E 04	0.14987119E 05	-0.11498318E 05
36	-0.45788177E 03	0.11315962E 05	-0.73569033E 04
37	0.39768164E 03	0.53168697E 04	-0.25619919E 04
38	0.	-0.20429645E 04	0.64227637E 03
39	0.30740338E 03	0.15272731E 05	0.
40	0.39134595E 03	0.15187136E 05	-0.64050560E 04
41	0.37905960E 04	0.14987978E 05	-0.13225795E 05
42	0.52100345E 04	0.11093373E 05	-0.12425824E 05
43	0.40561809E 04	0.52777498E 04	-0.71942065E 04
44	0.95482207E 03	0.17612293E 04	-0.20848166E 04
45	0.12996035E 04	0.15079522E 05	0.
46	0.13565886E 04	0.13801986E 05	-0.93656176E 04
47	0.15276900E 05	0.25390812E 05	-0.16069330E 05
48	0.12622224E 05	0.65254965E 04	-0.10729155E 05
49	0.49696732E 04	0.19053392E 04	-0.46236011E 04
50	0.48761243E 04	0.15054800E 05	0.
51	0.33697376E 04	0.15054800E 05	-0.13168468E 05
52	0.41094353E 05	-0.25437042E-03	-0.19216193E 05
53	0.15357369E 05	0.	-0.87900609E 04
54	0.12224934E 04	-0.76293945E-05	-0.28062864E 04
55	-0.32822423E 04	0.30517578E-04	-0.30439764E 03

ELEMENT NO	C	DLAM	DMU
1	-0.100000E 01	0.337566E-08	0.412231E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
-0.175987E 01	0.147527E 03	0.	-0.930633E-04	0.199048E-03	0.

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.132016E 04	0.158889E 05	0.	-0.295621E-02	0.567183E-02	0.

STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.446688E-02	0.108293E-01	0.	-0.151067E-02	0.515747E-02	0.

ELEMENT NO	C	DLAM	DMU
2	0.784198E-06	0.271067E-08	0.415881E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
0.917509E 01	0.155505E 03	0.	-0.446979E-04	0.983661E-04	0.

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.440133E 03	0.127690E 05	0.	-0.118694E-02	0.237321E-02	0.

STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.247474E-02	0.724154E-02	0.	-0.128780E-02	0.486833E-02	0.

ELEMENT NO	C	DLAM	DMU
3	0.798878E-06	0.273127E-08	0.410380E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
0.949776E 01	0.152330E 03	0.	-0.443702E-04	0.982223E-04	0.

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.489931E 03	0.131435E 05	0.	-0.120381E-02	0.240731E-02	0.

STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.248841E-02	0.728399E-02	0.	-0.128460E-02	0.487667E-02	0.

ELEMENT NO	C	DLAM	DMU
9	0.800871E-07	0.223348E-09	0.387430E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
-0.295198E 02	0.122376E 03	0.	-0.522916E-05	0.790567E-05	0.

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.140746E 04	0.757312E 04	0.	-0.176929E-03	0.279431E-03	0.

STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.222077E-02	0.438091E-02	0.	-0.204384E-02	0.410148E-02	0.

ELEMENT NO	C	DLAM	DMU
10	0.169373E-06	0.780469E-09	0.655237E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
-0.328929E 02	0.212498E 03	-0.287407E 02	-0.165735E-04	0.272701E-04	-0.102701E-04

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.130877E 04	0.301354E 05	-0.128115E 04	-0.341781E-03	0.573888E-03	-0.207284E-03
STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.221787E-02	0.493109E-02	-0.165628E-02	-0.187609E-02	0.435720E-02	-0.144900E-02

ELEMENT NO	C	DLAM	DMU
11	0.131991E-06	0.710556E-09	0.802706E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
-0.612334E 02	0.245566E 03	-0.414826E 02	-0.162891E-04	0.244477E-04	-0.110161E-04

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.233205E 04	0.110580E 05	-0.175422E 04	-0.302179E-03	0.464936E-03	-0.202491E-03
STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.250865E-02	0.483803E-02	-0.197180E-02	-0.220647E-02	0.437309E-02	-0.176931E-02

ELEMENT NO	C	DLAM	DMU
19	0.319190E-08	0.142234E-10	0.820449E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
-0.776339E 02	0.206961E 03	-0.870341E 02	-0.313982E-06	0.426083E-06	-0.452649E-06

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
-0.103042E 04	0.289054E 04	-0.116313E 04	-0.338829E-05	0.465506E-05	-0.478608E-05
STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
-0.186140E-02	0.306310E-02	-0.301639E-02	-0.185801E-02	0.305845E-02	-0.301160E-02

ELEMENT NO	C	DLAM	DMU
52	-0.100100E 01	0.646293E-09	0.987867E-02

DELTA ALPHA-X	DELTA ALPHA-Y	DELTA ALPHA-XY	DELTA EP-X	DELTA EP-Y	DELTA GAM-XY
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

TOT ALPHA-X	TOT ALPHA-Y	TOT ALPHA-XY	PLASTIC EP-X	PLASTIC EP-Y	PLASTIC GAM-XY
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

STRAIN-X	STRAIN-Y	STRAIN-XY	ELASTIC EP-X	ELASTIC EP-Y	ELASTIC GAM-XY
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.488825E-02

NODE NO	RESIDUAL STRESS SIGMA - X	SIGMA - Y	TAU - XY	RESIDUAL STRAIN STRAIN - X	STRAIN - Y	STRAIN - XY
1	0.69090036E 04	-0.11919927E 05	-0.	-0.19282739E-02	0.43000075E-02	-0.
2	0.68869494E 04	-0.10408120E 04	-0.	-0.48113836E-03	0.20686132E-02	-0.
3	0.71146174E 04	-0.16270742E 04	-0.	-0.45844778E-03	0.20385412E-02	-0.
4	0.51376756E 04	0.39714151E 04	-0.	0.38688736E-03	0.23824632E-03	-0.
5	0.17184607E 04	0.10867551E 04	-0.	0.13651315E-03	0.56001657E-04	-0.
6	0.41664470E 03	0.38117433E 03	-0.	0.29636510E-04	0.25115776E-04	-0.
7	0.12763276E 03	-0.36090948E 02	-0.	0.13574515E-04	-0.72922330E-05	-0.
8	0.50971681E-05	-0.56527019E 03	-0.	0.16625594E-04	-0.55418647E-04	-0.
9	-0.27256634E 04	-0.85987455E 04	-0.	-0.19124646E-03	-0.48341644E-03	-0.
10	-0.16005698E 04	-0.23487207E 04	-0.38732283E 03	-0.42962004E-03	0.39069729E-03	-0.30601371E-03
11	-0.25166190E 04	-0.13599274E 04	-0.31636000E 03	-0.50890890E-03	0.40562820E-03	-0.28313155E-03
12	0.70216895E 02	0.38504010E 04	0.29378726E 03	-0.10636307E-03	0.37542509E-03	0.74886949E-04
13	0.81186545E 03	0.15264426E 04	0.62191930E 03	0.34699281E-04	0.12577284E-03	0.15852845E-03
14	0.35709903E 03	0.46219864E 03	0.22885140E 03	0.21415631E-04	0.34810680E-04	0.58334672E-04
15	0.11417951E 03	-0.23180607E 02	0.77374373E 02	0.11875852E-04	-0.56308295E-05	0.19722879E-04
16	-0.	-0.54484896E 03	0.48362456E 02	0.16024969E-04	-0.53416564E-04	0.12327684E-04
17	-0.54460578E 03	-0.33709817E 04	-0.	0.45753797E-04	-0.31447059E-03	-0.
18	-0.64460258E 03	-0.12888739E 04	-0.11298158E 04	-0.25288277E-04	-0.10740128E-03	-0.28799228E-03
19	-0.18357399E 04	-0.17183705E 04	-0.16622270E 04	-0.13282248E-03	-0.10982028E-03	-0.42849100E-03
20	-0.17101344E 04	0.20554635E 04	-0.85144694E 03	-0.22811504E-03	0.25181410E-03	-0.21703549E-03
21	-0.40060420E 03	0.16279608E 04	0.13355397E 03	-0.87156122E-04	0.17138648E-03	0.34043168E-04
22	0.15310562E 03	0.57711852E 03	0.22617415E 03	-0.19637191E-05	0.52077140E-04	0.57652235E-04
23	0.82609639E 02	0.10328128E 02	0.13603864E 03	0.77952157E-05	-0.14171336E-05	0.34676517E-04
24	0.50971681E-05	-0.48789026E 03	0.95968864E 02	0.14349713E-04	-0.47832378E-04	0.24462651E-04
25	0.38731861E 02	-0.62126108E 03	-0.	0.22069626E-04	-0.62047121E-04	-0.
26	-0.47870440E 02	-0.52162942E 03	-0.68148957E 03	0.10648861E-04	-0.49732185E-04	-0.17371303E-03
27	-0.23935220E 03	-0.49692444E 03	-0.12527356E 04	-0.88504779E-05	-0.41678312E-04	-0.31932478E-03
28	-0.67157625E 03	0.25452215E 03	-0.10407256E 04	-0.73326754E-04	0.44705394E-04	-0.26528302E-03
29	-0.58015505E 03	0.85418108E 03	-0.38903681E 03	-0.82000919E-04	0.10080662E-03	-0.99166248E-04
30	-0.17073923E 03	0.50748917E 03	0.42805691E 02	-0.31665292E-04	0.54775582E-04	0.10911255E-04
31	0.12742920E-05	-0.29470728E 03	0.11052256E 03	0.86678611E-05	-0.28892870E-04	0.28172418E-04
32	0.21765444E 02	-0.16657473E 03	-0.	0.70331240E-05	-0.16971016E-04	-0.
33	0.97255750E 01	-0.16162042E 03	-0.41900022E 03	0.57070293E-05	-0.16131185E-04	-0.10680398E-03
34	0.72970988E 02	-0.51196089E 02	-0.80440418E 03	0.86597857E-05	-0.71654299E-05	-0.20504420E-03
35	-0.27579933E 02	0.31729579E 02	-0.75459826E 03	-0.36371379E-05	0.39219175E-05	-0.19234857E-03
36	-0.17822103E 03	0.28603904E 03	-0.45843980E 03	-0.25885563E-04	0.33284838E-04	-0.11685720E-03
37	-0.14192218E 03	0.26875989E 03	-0.93538946E 02	-0.21818641E-04	0.30523190E-04	-0.23843260E-04
38	-0.	-0.13421806E 03	0.79861403E 02	0.39475901E-05	-0.13158634E-04	0.20356828E-04
39	0.93352858E 02	-0.11408802E 02	-0.	0.94877941E-05	-0.38641823E-05	-0.
40	0.49766801E 02	0.67682905E 02	-0.52174706E 02	0.28884245E-05	0.51718495E-05	-0.13299435E-04
41	0.48204354E 03	-0.18084464E 03	-0.43534569E 03	0.52578130E-04	-0.31907617E-04	-0.11097047E-03
42	0.53916271E 03	-0.19066770E 03	-0.58367340E 03	0.58466963E-04	-0.34550638E-04	-0.14877949E-03
43	0.17225127E 03	-0.52209120E 02	-0.29084332E 03	0.18422942E-04	-0.10184754E-04	-0.74136534E-04
44	-0.47595441E 02	0.64918604E 02	-0.86926310E 02	-0.65755904E-05	0.77644350E-05	-0.22157814E-04
45	0.15800124E 03	-0.72242765E 02	-0.	0.17615105E-04	-0.11729719E-04	-0.
46	0.22517850E 03	0.25095494E 02	0.70009741E 02	0.21338220E-04	-0.41625544E-05	0.17845620E-04
47	0.25357939E 04	0.25300211E 04	0.11807964E 03	0.17419486E-03	0.17345911E-03	0.30098733E-04
48	0.72765879E 03	-0.66323702E 03	0.35890416E 02	0.90846068E-04	-0.86424966E-04	0.91485376E-05
49	0.14509291E 03	-0.10389021E 03	-0.49797805E 02	0.17280390E-04	-0.14452753E-04	-0.12693558E-04
50	-0.13211690E 03	-0.63714601E-06	-0.	-0.12952637E-04	0.38857911E-05	-0.
51	-0.19882756E 03	0.25485840E-05	-0.14985142E 03	-0.19492897E-04	0.58478695E-05	-0.38197422E-04
52	-0.24247269E 04	-0.	0.36703556E 03	0.86412207E-03	-0.47950471E-03	-0.14214781E-02
53	-0.42564300E 03	-0.12742920E-05	0.58501854E 03	-0.41729707E-04	0.12518912E-04	0.14912237E-03
54	0.85843100E 02	-0.66900332E-05	0.66605195E 02	0.84159904E-05	-0.25247977E-05	0.16977794E-04
55	-0.10058197E 03	0.63714601E-06	-0.15763800E 02	-0.98609775E-05	0.29582932E-05	-0.40182236E-05

Time Estimates

There is some difficulty associated with establishing an accurate guide for time estimates. The time required to complete any loading program will depend upon a number of variables. These variables include: the number of elements considered in the analysis; the number of elements that become plastic during the course of loading; the load increment; and the interval of printout.

A table of actual time used by the individual programs for various load increments is given below. The table should be used only as a basis of comparison of time used by the various programs, and not for establishing time estimates for any other structure or loading condition.

The values of the required time for the various programs were determined from computations performed on the stiffened panel. The incremental procedures associated with each of the programs began at the maximum elastic load, 8000 lbs, and continued up to a load level of 20,000 lbs.

Table I - Computing Time, Minutes

Load Increment	Program Name			
	STRESS	STEPP	STRAIN	STRAPP
10		28.00		27.82
50	14.50		12.95	8.02
100	8.11		8.63	5.25
250	4.11		5.53	2.62
500	3.20		4.81	2.25
750	3.60		4.18	1.23
1000	3.05		4.10	1.16

BUCKLING COMPUTER PROGRAMS

General Program Discussion

The formulation of the plastic buckling problem for initially flat panels, within the framework of the stiffness method of discrete-element analysis, is found in Section 6 of CR-803.

Based on this formulation, the computer programs must perform the following two basic operations:

1) Form the stiffness matrix of the over-all structure.

2) Solve for the state of stress necessary to cause the vanishing of the determinant of the stiffness matrix by either an eigenvalue calculation or by directly evaluating the determinant of the stiffness matrix.

The first stage of the calculations is now standard within discrete-element analysis. That is, the stiffness matrix is evaluated for each finite element in the structural idealization and then, after taking account of boundary conditions, these individual stiffness matrices are assembled into the over-all stiffness matrix of the structure. Consistent with the formulation in CR-803, the stiffness matrix for each element is formed by the program in two parts: the element bending stiffness matrix, from subroutine BSTIF; and the element initial stress stiffness matrix, from subroutine MSTIF. The element stiffness matrices are assembled into the over-all stiffness matrices by subroutine STACK.

The second stage of the computations, in the case of a plastic buckling problem, involves a sequence of eigenvalue calculations or the evaluation of the determinant of the stiffness matrix for various stress levels. An elastic buckling problem requires only one eigenvalue calculation.

Two of the three programs developed involve eigenvalue calculations. The remaining program evaluates the determinant of the stiffness matrix directly for successive load levels, and detects the vanishing of the determinant in order to predict buckling. A succession of determinant calculations is required both for elastic and for plastic buckling problems. The subroutine used for the evaluation of the determinant is discussed in the section labeled, Discussion of Determinant Calculations.

The eigenvalue formulation of the buckling problem involves the evaluation of the eigenvalues of one of two alternative equations of the general form, $[A]\{X\} = \lambda[B]\{X\}$. The eigenvalues of this equation can be found by recasting it in the form, $[B^{-1}][A]\{X\} = \lambda\{X\}$. The matrix $[B^{-1}][A]$ is not, in general, symmetric if $[A]$ and $[B]$ are symmetric. If $[B]$ is positive definite, however, a symmetric matrix, $[C]$, can be found which is similar to $[B^{-1}][A]$. The eigenvalue of interest is then found from $[C]\{X\} = \lambda\{X\}$. The development of this symmetric matrix has a twofold advantage: since only the lower triangle of the $[C]$ matrix need be stored in computer core, there is significant storage saving, and subroutines exist which can efficiently evaluate the eigenvalues of real symmetric matrices. A detailed discussion of the algorithms used in these operations can be found in the section labeled, Solution of the Eigenvalue Problem Det $([A] - \lambda[B]) = 0$.

The two eigenvalue programs differ in their degree of generality and in the form of the eigenvalue equation solved.

Constant stress field eigenvalue program. -

This program is written specifically for a constant stress field situation with the shearing stress equal to zero. In it, the positive definite nature of the initial stress stiffness matrix, $[K_m]$, permits the development of the symmetric matrix used to evaluate the eigenvalues of:

$$[K_B]\{\delta\} = -\lambda[K_m]\{\delta\} \tag{1}$$

where $[K_B]$ and λ are to be interpreted here as being appropriate to either an elastic or a plastic buckling calculation. The eigenvalue of interest is the smallest one.

A significant computational advantage is gained in the solution of the plastic buckling problem as a consequence of the simple nature of the constant stress field. This fact is used to reduce significantly the amount of computing time necessary to develop the symmetric matrix, $[C]$, for successive eigenvalue calculations. Consequently, this program should be used for plastic buckling calculations when the stress field is constant and the shearing stress is zero.

General eigenvalue program. -

This program is capable of handling variable stress fields, including shearing stress. In it, the positive definite nature of the bending stiffness matrix, $[K_B]$, permits the development of the symmetric matrix used to evaluate the eigenvalues of:

$$-[K_m]\{\delta\} = \frac{1}{\lambda}[K_B]\{\delta\} , \tag{2}$$

where $[K_B]$, $[K_m]$, λ are to be interpreted as being appropriate to either an elastic or plastic buckling calculation. In this case it is the largest eigenvalue which is of interest.

Program Input

All programs are provided with an identical input format. The sequencing and details of the data cards follow. Symbols used for input data are presented in the section entitled, Symbols and Format of the Data Cards.

Sequencing and details of the data cards. -

Card 1 contains general program control variables: NJTS, KLU, KLU2, KLU3, KLU4, KLU5, NEIG, LØT. The variable KLU4 is always zero for the constant stress field eigenvalue program.

Card 2 contains plate properties: ANU, E, H, S7, NCØ. In the case of elastic buckling, the problem is formulated so that only ANU is required.

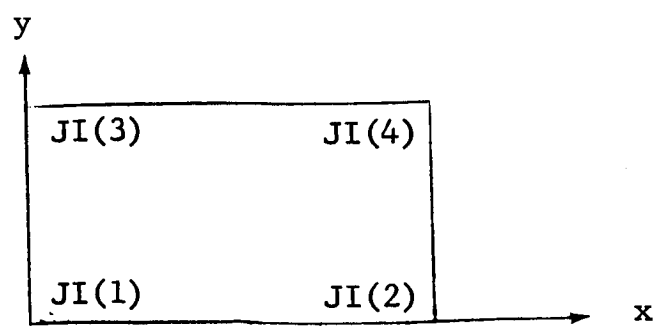
Card 3 is prepared using one of two formats:

a) The short format contains the following data necessary to compute plate idealization for equal size elements, XSZ, YSZ, XN1, YN1, NMEM. In the event that the dimensions of the elements vary along the coordinate axes, the short format given above must be replaced by the long format, requiring a considerable number of additional data cards.

b) The long format:

Group I-Data Cards have one card for each node point in the idealization. Each card contains: JT, X, Y.

Group II-Data Cards have one card for each element. Each card contains: MEM, JI(1), JI(2), JI(3), JI(4). The order used in the specification of the JI array associated with a discrete element is shown below.



Card 4 is prepared using one of two formats:

a) For a constant stress field:

One card contains APH, BTA, GAM, EN, DEN. The quantities EN and DEN are always positive. The quantities APH, BTA and GAM take on a positive or negative sign according to the sense of the corresponding stress.

b) For a variable stress field:

Group III-Data Cards consist of one card for each element, each card containing: PX, DSX, PY, DSY, PXY, DSXY.

Plastic buckling calculations using the eigenvalue programs require as input an initial estimate of the buckling stress resultants and the increments of these stress resultants. For a constant stress field situation this is specified by EN and DEN, along with APH, BTA, GAM. In the case of a variable stress state, this is specified by PX, PY, PXY, DSX, DSY, DSXY. The initial estimates should be less than, or equal to, the corresponding elastic buckling values. They can be closely determined by first performing calculations based on a coarse idealization with large stress resultant increments, and then using the resulting solution as input for subsequent calculations with a finer idealization. The quantities, EN and DEN, are only required for a plastic buckling calculation. In the case of elastic buckling with a variable stress field, DSX, DSY, and DSXY are not required, and PX, PY, PXY represent some nominal state of stress with the desired distribution. The determinant program requires as input initial values for the stress resultants for both elastic and plastic buckling calculations. Care must be taken to ensure that these stresses are below the actual buckling stresses.

Card 5 - Group IV-Data Cards

This grouping is needed at the beginning of each problem to specify the node boundary conditions. It can be omitted when the calculation follows one with the same boundary conditions. The grouping has one card for each node, and specifies the node boundary conditions.[1] Each card contains: JØINT, JD(1), JD(2), JD(3), JD(4).

Card 6, the last card of each run, contains a job termination variable, STØPIT.

Symbols and format of the data cards. -

Card 1:

Columns	Field	Symbol	Comments
1-3	I3	NJTS	Number of node points in plate idealization.
4-6	I3	KLU	KLU = 0, Elastic buckling run. KLU ≠ 0, Plastic buckling run.
7-9	I3	KLU2	KLU2 = 0, No intermediate output. KLU2 ≠ 0, Intermediate output will be printed.
10-12	I3	KLU3	KLU3 = 1, Idealization is computed (Short Format). KLU3 ≠ 1, Plate idealization is read as input from Group I and Group II-Data Cards (Long Format).
13-15	I3	KLU4	KLU4 = 0, Constant stress field. KLU4 ≠ 0, Variable stress field, read from Group III-Data Cards.

[1]The rectangular element due to Bogner, Fox and Schmidt used in the present work requires the specification of four conditions at each node.

Columns	Field	Symbol	Comments
16-18	I3	KLU5	KLU5 = 0, Read node boundary conditions. KLU5 ≠ 0, Omit Data Cards-Group IV.
19-21	I3	NEIG	Number of eigenvalues desired.
22-24	I3	LØT	Quantity used in specifying the desired accuracy of a plastic buckling run.
25-80			Ignored.

Card 2:

Columns	Field	Symbol	Comments
1-14	E14.8	ANU	Poisson's ratio.
14-28	E14.8	E	Young's modulus.
29-42	E14.8	H	Plate thickness.
43-49	I7	NCØ	Shape parameter, n, used in Ramberg-Osgood stress-strain representation.
50-60	F10.2	S7	$\sigma_{0.7}$, used in Ramberg-Osgood stress-strain representation.
61-80			Ignored.

Card 3 (Short Format):

Columns	Field	Symbol	Comments
1-6	F6.0	XSZ	Length of plate along x axis, or the length of that portion of the plate considered when symmetry is taken into account.
7-12	F6.0	YSZ	Length of plate along y axis, or the length of that portion of the plate considered when symmetry is taken into account.

Columns	Field	Symbol	Comments
13-18	F6.0	XN1	Number of plate elements along the x axis.
19-24	F6.0	YN1	Number of plate elements along the y axis.
25-26	I2	NMEM	Number of plate elements in the idealization.
27-80			Ignored.

or

Card 3 Group I-Data Cards (Long Format)

Columns	Field	Symbol	Comments
1-4	I4	JT	Node number.
5-16	E12.1	X	x coordinate of node.
17-28	E12.1	Y	y coordinate of node.
29-80			Ignored.

Group II-Data Cards

Columns	Field	Symbol	Comments
1-3	I3	MEM	Element number.
6-8	I3	JI(1)	Node points associated with the "MEMth" discrete element.
9-11	I3	JI(2)	
12-14	I3	JI(3)	
15-17	I3	JI(4)	
18-80			Ignored

4a) Card 4 (Constant Stress Field)

Columns	Field	Symbol	Comments
1-10	F10.2	APH	Ratio of direct stress, σ_x, to stress intensity, σ_*.[2]
11-20	F10.2	BTA	Ratio of direct stress, σ_y, to stress intensity, σ_*.[2]
21-30	F10.2	GAM	Ratio of shearing stress, τ_{xy}, to stress intensity, σ_*.[2]
31-40	F10.2	EN	Stress resultant intensity, $h\sigma_*$.[2]
41-50	F10.2	DEN	Increment of stress resultant intensity, ΔN_*.[2]
51-80			Ignored.

or

4b) Group III-Data Cards (Variable Stress Field)

Columns	Field	Symbol	Comments
1-10	F10.2	PX	Element stress resultant, N_x.[2]
11-20	F10.2	DSX	Increment of element stress resultant, ΔN_x.[2]
21-30	F10.2	PY	Element stress resultant, N_y.[2]
31-40	F10.2	DSY	Increment of element stress resultant, ΔN_y.[2]
41-50	F10.2	PXY	Element stress resultant, N_{xy}.[2]
51-60	F10.2	DSXY	Increment of element stress resultant, ΔN_{xy}.[2]
61-80			Ignored.

[2]These quantities, as defined above, apply in the case of plastic buckling when the eigenvalue programs are used. In the case of elastic buckling calculations and of plastic buckling calculations when the determinant program is used, these quantities are divided by the flexural rigidity, D.

5. Group IV - Data Cards

Columns	Field	Symbol	Comments
1-3	I3	JØINT	Node number.
4-6	I3	JD(1)	JD(1) = 0, w_j is fully restrained. JD(1) ≠ 0, w_j is free at node j.
7-9	I3	JD(2)	JD(2) = 0, w_{xj} is fully restrained. JD(2) ≠ 0, w_{xj} is free at node j.
10-12	I3	JD(3)	JD(3) = 0, w_{yj} is fully restrained. JD(3) ≠ 0, w_{yj} is free at node j.
13-15	I3	JD(4)	JD(4) = 0, w_{xyj} is fully restrained. JD(4) ≠ 0, w_{xyj} is free at node j.
16-80			Ignored.

6. Card 6 (Last Card)

Columns	Field	Symbol	Comments
1-5	F5.0	STØPIT	STØPIT = 0, program will accept additional input data and continue to run. STØPIT ≠ 0, program stops, all jobs completed.
6-80			Ignored.

Program Output

The printout begins with a statement specifying whether the run is an elastic or plastic buckling calculation. The input variables defined in the section labeled, Program Input, along with the node coordinates and the JII array, follow. The stress field, specified by APH, BTA, GAM, EN, DEN for a constant stress field,

or PX, DSX, PY, DSY, PXY, DSXY for a variable stress field, is printed next. Nodal boundary conditions, specified as either zero or one for each degree of freedom, follow.

Eigenvalue programs. -

The calculated output depends on the program used. The eigenvalue programs print a holorith statement, FUTILE WAS TROUBLE FREE, if the computation involving the formation of the lower triangular matrices is successfully completed. In the event that this calculation cannot be completed, the program prints, C IS NØT PØSITIVE DEFINITE or ØVERFLØW, and then returns to the beginning of the program. A listing of the eigenvalues follows. In the case of the constant stress field eigenvalue program, all the eigenvalues are printed. The general eigenvalue program allows the option of printing any number of consecutive eigenvalues starting with the largest one (specified by NEIG). Consistent with the plastic buckling formulation, a sequence of eigenvalues associated with different loadings will be printed, until the desired solution is obtained. This solution satisfies the plastic buckling criterion, which requires that the largest eigenvalue be equal to one in the general eigenvalue program, or the smallest eigenvalue be equal to one in the constant stress field program. When this solution is reached, ERRØR is printed out. This variable is equal to $(\bar{\lambda} - 1)$, where $\bar{\lambda}$ is the smallest eigenvalue for the constant stress field program, and is the inverse of the largest eigenvalue for the general eigenvalue program. It is a measure of the closeness of the desired eigenvalue to one. The allowable error is specified by the input variable, LØT (absolute value of ERRØR $\leq 10^{-\text{LØT}}$). The final printed output is the buckling stress resultants. In the case of an elastic buckling calculation, the stress resultants are

divided by the plate flexural rigidity, D, and listed as NX/D, NY/D, NXY/D. In the case of a plastic buckling calculation, the stress resultants, NX, NY, NXY, are printed.

Determinant program. -

This program prints the determinant of the stiffness matrix for a sequence of load levels. The determinant information is printed under the following headings:

CYCLE	number of determinants printed
DETERMINANT	the value of the determinant if it can be stored in a floating point number
KEY	KEY = 0 if determinant can be printed, KEY = 1 if determinant cannot be printed due to overflow or underflow
KAY, VALUE	if KEY = 1, KAY and VALUE are printed such that the determinant equals VALUE * 2^{Kay} (see section entitled Discussion of Determinant Calculation)

When the value of the determinant changes sign, the program prints, THE SIGN ØF DET C HAS CHANGED. This will be printed any number of times until the error criterion specified by LØT is satisfied. When the determinant is close enough to zero, the stress resultants are printed in the same manner as in the eigenvalue programs.

Sample output. -

Sample output is shown for both the elastic and plastic buckling of a simply-supported square plate with a uniform compressive load on two opposite edges. Symmetry conditions were taken into account, so that only one-quarter of the plate was considered. In order to limit the amount of output, a coarse network consisting of only four elements was used for the plate idealization. Calculations

based on this idealization are shown for: 1) elastic buckling, using the general eigenvalue program; 2) plastic buckling, using both the constant stress field program and the determinant program. The input parameters common to all three cases are only shown once.

THIS IS AN ELASTIC BUCKLING RUN

KLU= 0 KLU2= 0 KLU3= 1 KLU4= 0 KLU5= 0 NEIG= 3 LOT= -0

ANU= 0.50000000E 00 E= 0.09999999E 08 H= 0.11201899E 01 NCD= 10 S7= 0.99999999E 05

XSZ= 10.0 YSZ= 10.0 XN1= 2.0 YN1= 2.0 NJTS= 9 NMEM= 4

JOINT	X	Y
1	0.	0.
2	5.00	0.
3	10.00	0.
4	0.	5.00
5	5.00	5.00
6	10.00	5.00
7	0.	10.00
8	5.00	10.00
9	10.00	10.00

JII ARRAY

1	2	4	5
2	3	5	6
4	5	7	8
5	6	8	9

APH= -1.00 BTA= 0. GAM= 0. EN= -0. DEN= -0.

NODE	W	WX	WY	WXY
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	1	0	1	0
5	1	1	1	1
6	0	1	0	1
7	0	0	1	0
8	0	0	1	1
9	0	0	0	1

FUTILE WAS TROUBLE FREE

EIGENVALUES

1.0129484E 01 3.5518636E 00 1.2180987E 00

THE BUCKLING STRESSES ARE

NXZD	NYZD	NXYZD
-0.98721706E-01	0.	0.

THIS IS A PLASTIC BUCKLING RUN

KLU= 1 KLU2= 0 KLU3= 1 KLU4= 0 KLU5= 0 LOT= 4

ANU= 0.50000000E 00 E= 0.09999999E 08 H= 0.11201899E 01 NCO= 10 S7= 0.99999999E 05

XSZ= 10.0 YSZ= 10.0 XN1= 2.0 YN1= 2.0 NJTS= 9 NMEM= 4

JOINT	X	Y
1	0.	0.
2	5.00	0.
3	10.00	0.
4	0.	5.00
5	5.00	5.00
6	10.00	5.00
7	0.	10.00
8	5.00	10.00
9	10.00	10.00

JII ARRAY

1	2	4	5
2	3	5	6
4	5	7	8
5	6	8	9

APH=-1.0 BTA= 0. GAM= 0. EN=106460.00 DEN= 10.000 TOL= 0.00010

NODE	W	WX	WY	WXY
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	1	0	1	0
5	1	1	1	1
6	0	1	0	1
7	0	0	1	0
8	0	0	1	1
9	0	0	0	1

FUTILE WAS TROUBLE FREE

EIGENVALUES

1.3053704E 03	2.4160286E 02	1.7423310E 02	8.5972297E 01	6.6189570E 01	4.1069916E 01
3.0263184E 01	2.9204427E 01	2.7927527E 01	1.7193346E 01	1.2199934E 01	1.0595114E 01
9.1236345E 00	5.0895141E 00	1.9436174E 00	9.9893939E-01		

EIGENVALUES

1.3057284E 03	2.4166914E 02	1.7428118E 02	8.5996869E 01	6.6210299E 01	4.1081478E 01
3.0274059E 01	2.9212467E 01	2.7936182E 01	1.7200641E 01	1.2205857E 01	1.0599017E 01
9.1264366E 00	5.0919093E 00	1.9444493E 00	9.9924612E-01		

EIGENVALUES

1.3060864E 03	2.4173541E 02	1.7432925E 02	8.6021439E 01	6.6231028E 01	4.1093040E 01
3.0284938E 01	2.9220507E 01	2.7944837E 01	1.7207936E 01	1.2211783E 01	1.0602921E 01
9.1292397E 00	5.0943071E 00	1.9452812E 00	9.9955231E-01		

EIGENVALUES

1.3064443E 03	2.4180169E 02	1.7437732E 02	8.6046011E 01	6.6251756E 01	4.1104603E 01
3.0295818E 01	2.9228546E 01	2.7953492E 01	1.7215232E 01	1.2217710E 01	1.0605824E 01
9.1320426E 00	5.0967013E 00	1.9461101E 00	9.9985754E-01		

EIGENVALUES

1.3068022E 03	2.4186797E 02	1.7442539E 02	8.6070583E 01	6.6272490E 01	4.1115165E 01
3.0306703E 01	2.9236585E 01	2.7962148E 01	1.7222531E 01	1.2223641E 01	1.0610728E 01
9.1348459E 00	5.0991008E 00	1.9469423E 00	1.0001634E 00		

EIGENVALUES

1.3066233E 03	2.4183484E 02	1.7440136E 02	8.6058298E 01	6.6262124E 01	4.1110384E 01
3.0301261E 01	2.9232567E 01	2.7957821E 01	1.7218882E 01	1.2220474E 01	1.0608775E 01
9.1334435E 00	5.0979007E 00	1.9465264E 00	1.0000106E 00		

EIGENVALUES

1.3064443E 03	2.4180170E 02	1.7437733E 02	8.6046013E 01	6.6251758E 01	4.1104603E 01
3.0295820E 01	2.9228547E 01	2.7953492E 01	1.7215232E 01	1.2217711E 01	1.0605823E 01
9.1320432E 00	5.0967022E 00	1.9461107E 00	9.9985886E-01		

EIGENVALUES

1.3065338E 03	2.4181826E 02	1.7438934E 02	8.6052157E 01	6.6256964E 01	4.1107493E 01
3.0298541E 01	2.9230553E 01	2.7955657E 01	1.7217056E 01	1.2219193E 01	1.0607799E 01
9.1327424E 00	5.0973023E 00	1.9463167E 00	9.9993253E-01		

ERROR

-6.7465007E-05

THE BUCKLING STRESSES ARE

NX	NY	NXY
-0.10642750E 06	0.	0.

THIS IS A PLASTIC BUCKLING RUN

KLU= 1 KLU2= 0 KLU3= 1 KLU4= 0 KLU5= 0 LOT= 6

ANU= 0.50000000E 00 E= 0.09999999E 08 H= 0.11201899E 01 NCO= 10 S7= 0.99999999E 05

XSZ= 10.0 YSZ= 10.0 XN1= 2.0 YN1= 2.0 NJTS= 9 NMEM= 4

JOINT	X	Y
1	0.	0.
2	5.00	0.
3	10.00	0.
4	0.	5.00
5	5.00	5.00
6	10.00	5.00
7	0.	10.00
8	5.00	10.00
9	10.00	10.00

JII ARRAY

1	2	4	5
2	3	5	6
4	5	7	8
5	6	8	9

APH=1.0000 BTA=0. GAM=0. EN= 0.0600 DEN= 0.0010

NODE	W	WX	WY	WXY
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	1	0	1	0
5	1	1	1	1
6	0	1	0	1
7	0	0	1	0
8	0	0	1	1
9	0	0	0	1

CYCLE	DETERMINANT	KEY	KAY	VALUE
1	0.87529608E-19	0	-48	0.24637394E-04
2	0.50803776E-19	0	-48	0.14299992E-04
3	0.28017684E-19	0	-48	0.78862769E-05
4	0.14565523E-19	0	-48	0.40998304E-05
5	0.70474748E-20	0	-48	0.19836878E-05
6	0.31018300E-20	0	-48	0.87308754E-06
7	0.11830544E-20	0	-52	0.53280036E-05

8	0.33947504E-21	0	-52	0.15288597E-05
9	0.21750833E-22	0	-52	0.97957044E-07
10	-0.65090466E-22	0	-52	-0.29314139E-06

THE SIGN OF DET C HAS CHANGED

11	-0.38901468E-22	0	-52	-0.17519663E-06

THE SIGN OF DET C HAS CHANGED

12	-0.14012186E-22	0	-52	-0.63105279E-07

THE SIGN OF DET C HAS CHANGED

13	0.23451925E-23	0	-52	0.10561808E-07
14	-0.14012023E-22	0	-52	-0.63104543E-07

THE SIGN OF DET C HAS CHANGED

15	-0.61929072E-23	0	-52	-0.27890375E-07

THE SIGN OF DET C HAS CHANGED

16	-0.20173320E-23	0	-52	-0.90852559E-08

THE SIGN OF DET C HAS CHANGED

17	0.14067376E-24	0	-52	0.63353832E-09
18	-0.20175067E-23	0	-52	-0.90860426E-08

THE SIGN OF DET C HAS CHANGED

19	-0.94315836E-24	0	-52	-0.42476076E-08

THE SIGN OF DET C HAS CHANGED

20	-0.40198591E-24	0	-52	-0.18103836E-08

THE SIGN OF DET C HAS CHANGED

21	-0.13123325E-24	0	-52	-0.59102203E-09

THE SIGN OF DET C HAS CHANGED

22	0.56805052E-26	0	-52	0.25582721E-10
23	-0.13014973E-24	0	-52	-0.58614229E-09

THE SIGN OF DET C HAS CHANGED

BUCKLING STRESSES AR

NX	NY	NXY
NX= -0.10642680E 06	NY= 0.	NXY= 0.

List of Subroutines

The following subroutines are called from the main program.

BIGSYM	computes eigenvalues of a real symmetric matrix
BSTIF	computes element bending stiffness matrix
DAGGER	Performs the operation, $[M_k]\ [H_k]\ [M_k^T]$ (see section labeled, Solution of the Eigenvalue Problem Det $([A] - \lambda[B]) = 0$)
FUTILE	forms lower triangular matrix, [L], from positive definite matrix such that, $[B] = [L][L]^T$
IMEQF	computes the determinant of a matrix, or solves a system of linear equations
MSTIF	computes element initial stress stiffness matrix
SCRIBE	prints C-array
STACK	assembles element stiffness matrices into over-all stiffness matrix in C-array
SWITCH	permutes the elements of the C-array from a partial row arrangement to a partial column arrangement

Time Estimate

Any estimate of program running time is dependent on the number of determinant or eigenvalue calculations necessary to arrive at the proper buckling stresses, and the size of the stiffness matrices associated with the plate idealization. The number of eigenvalue or determinant calculations depends on the initial estimate of the buckling stress state, the stress increments selected and the desired accuracy of the final solution. Computing time for eigenvalue or determinant calculations increases in a

nonlinear fashion as the order of the stiffness matrix increases. If the initial estimate of the buckling stresses is far from the actual buckling stresses, and the stress increments selected are small, program running times for large matrices may be excessive. However, if the estimate of buckling stresses is chosen carefully, the running times for plastic buckling calculations will be significantly reduced. Results for some plastic buckling calculations using the general eigenvalue program are shown below:

Matrix Size	Number of Eigenvalue Calculations	Time (min.)
35 x 35	10	1.52
95 x 95	7	3.78
130 x 130	5	5.90

As can be seen, the computing time increases rapidly as the size of the matrices is increased.

Solution of the Eigenvalue Problem Det $([A] - \lambda[B]) = 0$

The algorithm used by the program requires that $[A]$ and $[B]$ be symmetric and $[B]$ positive definite. If $[A]$ is positive definite, but not $[B]$, a slight variation of the procedure can be used. The calculation takes place in two principal stages:

1) Formation of a real symmetric matrix, $[C]$, where eigenvalues are the roots of $\det([A] - \lambda[B])$.

2) Solution of the standard eigenvalue problem for $[C]$.

The second stage employs a technique which has become almost standard. Specifically, $[C]$ is reduced to a tridiagonal symmetric matrix, $[T]$, by orthogonal similarity transformations

(based on Householder matrices), and the eigenvalues are then located by exploiting a Sturm-sequence property. An excellent account of the procedure is given by Wilkinson,[3] so that we shall not pursue the details here, except to remark that the eigenvalue program accepts the matrix, $[C]$, in triangular form in order to make possible the processing of fairly large matrices (maximum order in excess of 200 on the IBM 7094) without resort to tapes or other secondary storage devices. This computation is performed by subroutine BIGSYM.

The first stage of the algorithm is not so widely known, and we propose to describe it somewhat more closely. The equation, $\det([A] - \lambda[B]) = 0$, is satisfied if and only if there is a non-zero vector, $\{x\}$, for which $[A]\{x\} = \lambda[B]\{x\}$. Because $[B]$ is assumed positive definite, it is certainly nonsingular and the condition given is equivalent to $[B]^{-1}[A]\{x\} = \lambda\{x\}$. One might, therefore, attempt the solution of the standard eigenvalue problem for the matrix, $[B]^{-1}[A]$, but a more convenient alternative is available. The positive definite character of $[B]$ ensures the existence of a nonsingular (real) lower triangular matrix, $[L]$, such that $[B] = [L][L^T]$. Obviously $[B]^{-1}[A]$ is given by $[L]^{-T}[L]^{-1}[A]$, and if $[C]$ is defined to be the matrix, $[L]^{-1}[A][L]^{-T}$, $[C]$ is symmetric and $[C] = [L]^T[B]^{-1}[A][L]^{-T}$, so that $[C]$ is similar to $[B]^{-1}[A]$. It follows that the roots of $\det([A] - \lambda[B])$ are the eigenvalues of $[C]$.

The formation of $[C]$ as an intermediate step of the calculation has much to recommend it. First of all, $[C]$ is obtained more economically than $[B]^{-1}[A]$ itself. Given $[L]$ and $[A]$, $[C]$ can be formed with the use of only $n^3/2$ multiplications,[4]

[3] Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965, pp. 290-315.

[4] All matrices considered here are square, of order n.

whereas about n^3 multiplications seem to be needed to produce $[B]^{-1}[A]$, on the assumption that $[L]$ is obtained from the Choleski algorithm, and the columns of $[B]^{-1}[A]$ are then obtained by solving the triangular systems, $[L]\{z\} = \{y\}$ and $[L]^T\{x\} = \{z\}$, with the role of y taken by successive columns of $[A]$. Other methods for calculating $[B]^{-1}[A]$ appear to be still more expensive. Secondly, the symmetry of $[C]$ makes it possible to store roughly half its elements, while $[B]^{-1}[A]$ requires a full square array. Finally, the solution of the eigenvalue problem is considerably simplified if the matrix is known to be symmetric. The more prominent benefits conferred by symmetry include the following:

> The eigenvalues are known to be real numbers; this contrasts with the situation for general real matrices, which may have complex eigenvalues.
>
> The problem is necessarily well-conditioned; small errors in the matrix cannot lead to large errors in the eigenvalues. Moreover, there are stable, accurate algorithms for computing the roots. The Householder technique is probably the most efficient of these algorithms.
>
> The eigenvalues of a symmetric matrix can be calculated more rapidly than those of a general matrix. A factor of 2 is a conservative estimate of the gain to be expected.

The lower triangular matrix, $[L]$, is derived from $[B]$ by application of the Choleski algorithm, using only about $n^3/6$ multiplications. One assumes the existence of an $[L]$ satisfying $[L][L]^T = [B]$, and determines its elements successively (a row or a column at a time) to satisfy the equations, $\sum_{j=1}^{k} \ell_{ij}\ell_{kj} = b_{ik}$, for $i \geq k$. It is not difficult to show that the stipulation,

$\ell_{kk} > 0$, (for all k) leads to a unique solution of these equations. It is convenient to store each element of [L] over the corresponding row of [B]. Subroutine FUTILE carries out the Choleski decomposition of [B] in this fashion.

We note that any lower triangular matrix, [L], has the factorization, $[L] = [L_1][L_2] \ldots [L_n]$, where each $[L_k]$ is a lower triangular matrix differing from the identity only in the first k elements of row k which are given by ℓ_{ik} for $i = 1, 2, \ldots, k$. The inverse, $[M_k]$, of $[L_k]$, has the same form, with $m_{kk} = 1/\ell_{kk}$ and $m_{ki} = -\ell_{ki} m_{kk}$ for $i < k$. In order to determine [C] we define $[H_o] = [A]$ and $[H_k]$ to be the matrix, $[M_k][H_{k-1}][M_k]^T$. It is easy to establish that $[C] = [H_n]$. Each $[H_k]$ is symmetric and requires roughly nk multiplications. If [L] is placed on a secondary storage device, replaced in main storage by [A], and then reintroduced one row at a time, the formation of [C] can be accomplished with the use of one large (triangular) array and a few extra vectors. Row k of [L] is brought into core, changed to row k of $[M_k]$ and then used to generate $[H_k]$ from $[H_{k-1}]$. The subroutine DAGGER performs these calculations and is entered once for each k to produce $[H_k]$, given $[H_{k-1}]$ and the nontrivial elements, m_{ik}, in M_k.

After calculating the elements of [C], but before invocation of the eigenvalue routine, the program must alter the triangular array containing [C] in order to store [C] in the form expected by the eigenvalue routine. For the Choleski decomposition and the formation of [C], it seems most convenient to store all the lower triangular arrays in consecutive (partial) row order, illustrated on page 42 for a matrix of order 5. Unfortunately, this arrangement appears to be somewhat awkward for the execution

of the Householder algorithm, and we found it expedient to employ a subroutine which permutes the elements of the C array so as to leave the array in consecutive (partial) column order, also shown for a 5 x 5 matrix. Of course, it is desirable to accomplish the transformation within the triangular array (except perhaps for a few additional storage locations) and to carry it out rapidly enough to avoid any serious effect on over-all execution time. The design of an appropriate procedure represented an interesting and fairly challenging programming problem. A reasonably satisfactory solution was reached. It consists of first reversing the triangular array and then reflecting it about the reverse main diagonal. We show the intermediate arrangement for order 5. It is not known whether the problem has been previously considered, or whether our solution is near-optimal; it achieves the order in about $n^2/2$ interchanges (transpositions in the terminology of permutation theory). The rearrangement of the triangular array is done by subroutine SWITCH.

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

1
2 6
3 7 10
4 8 11 13
5 9 12 14 15

15
14 13
12 11 10
 9  8  7  6
 5  4  3  2  1
```

The main lines of the procedure for roots of det ([A] - λ[B]) are set forth by Wilkinson.[5] The entire Wilkinson text is a first-rate discussion of most of the commonly used algorithms for the determination of eigenvalues and eigenvectors. The interested reader is strongly advised to consult this reference.

[5]Ibid., pp. 337-339.

Discussion of Determinant Calculation

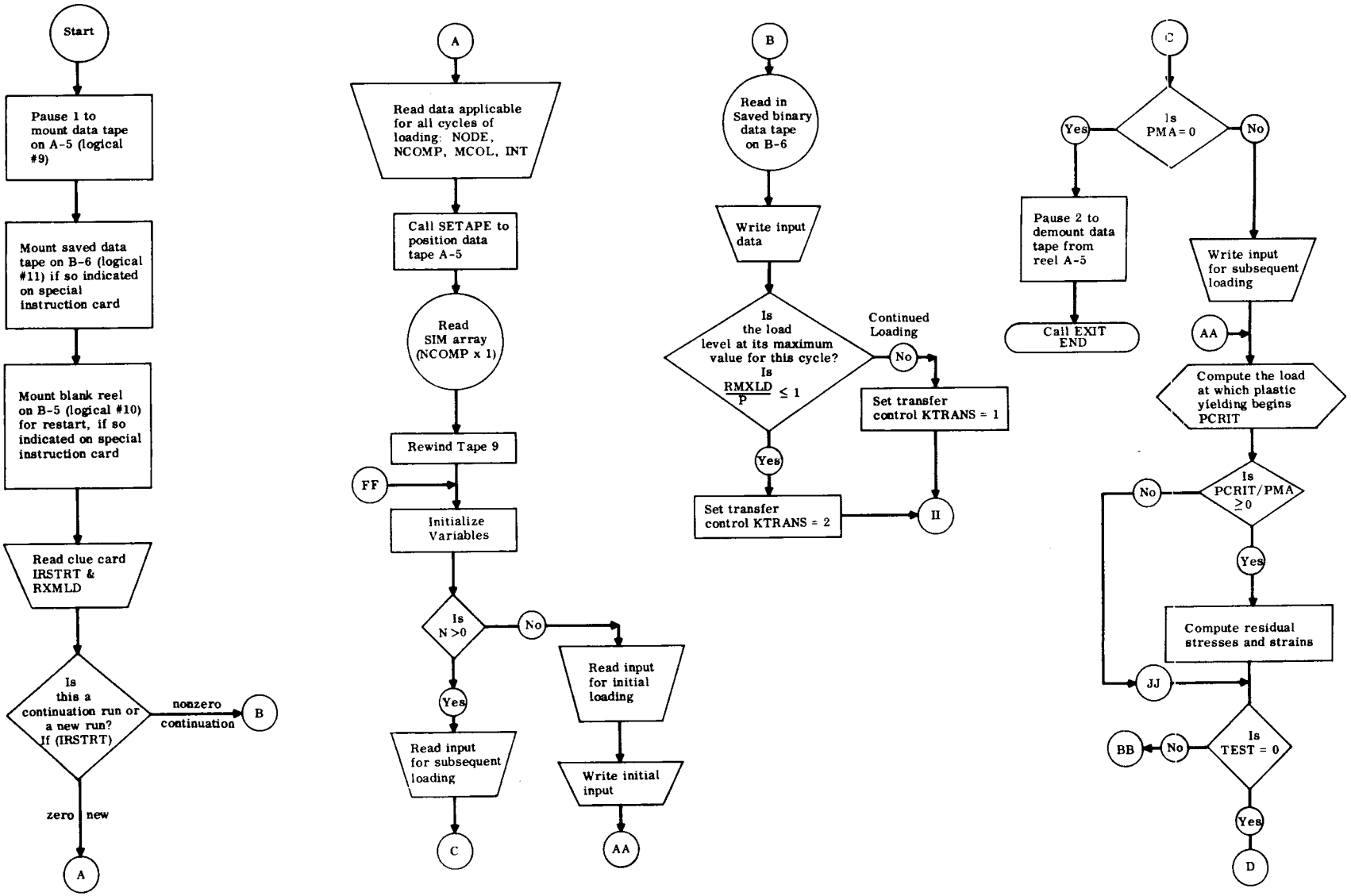
The determinant evaluation is performed by the subprogram, IMEQF, which is a standard routine for the solution of linear algebraic equations. The algorithm may be described as triangular decomposition with scaling, partial pivoting, and double precision inner product accumulation. After the matrix has been scaled so that each row and column has its largest element near 1 in magnitude, the calculation differs very little from the formulation given by Wilkinson.[6] Essentially, one writes $[A] = [L][U]$, where $[L]$ is unit lower triangular (i.e., $\ell_{kk} = 1$ for all k) and $[U]$ is upper triangular. The determinant is then obtained as the product of the diagonal elements of $[U]$. In computing the determinants of matrices of high order, some care must be taken to guard against floating-point spill, since such determinants may have extremely large or small values. The version of IMEQF used here has been modified to monitor the formation of the successive partial products rather closely, so that spill can be detected almost immediately. If the determinant is too large or too small to be stored as a conventional floating-point number, it is then returned as a pair, (p, x), which indicates a value of $2^p x$. On the IBM 7094, p may be any integer in the range 2^{-35} to 2^{35}, so that virtually any real number can be represented in this form.

[6] Ibid., pp. 225-229.

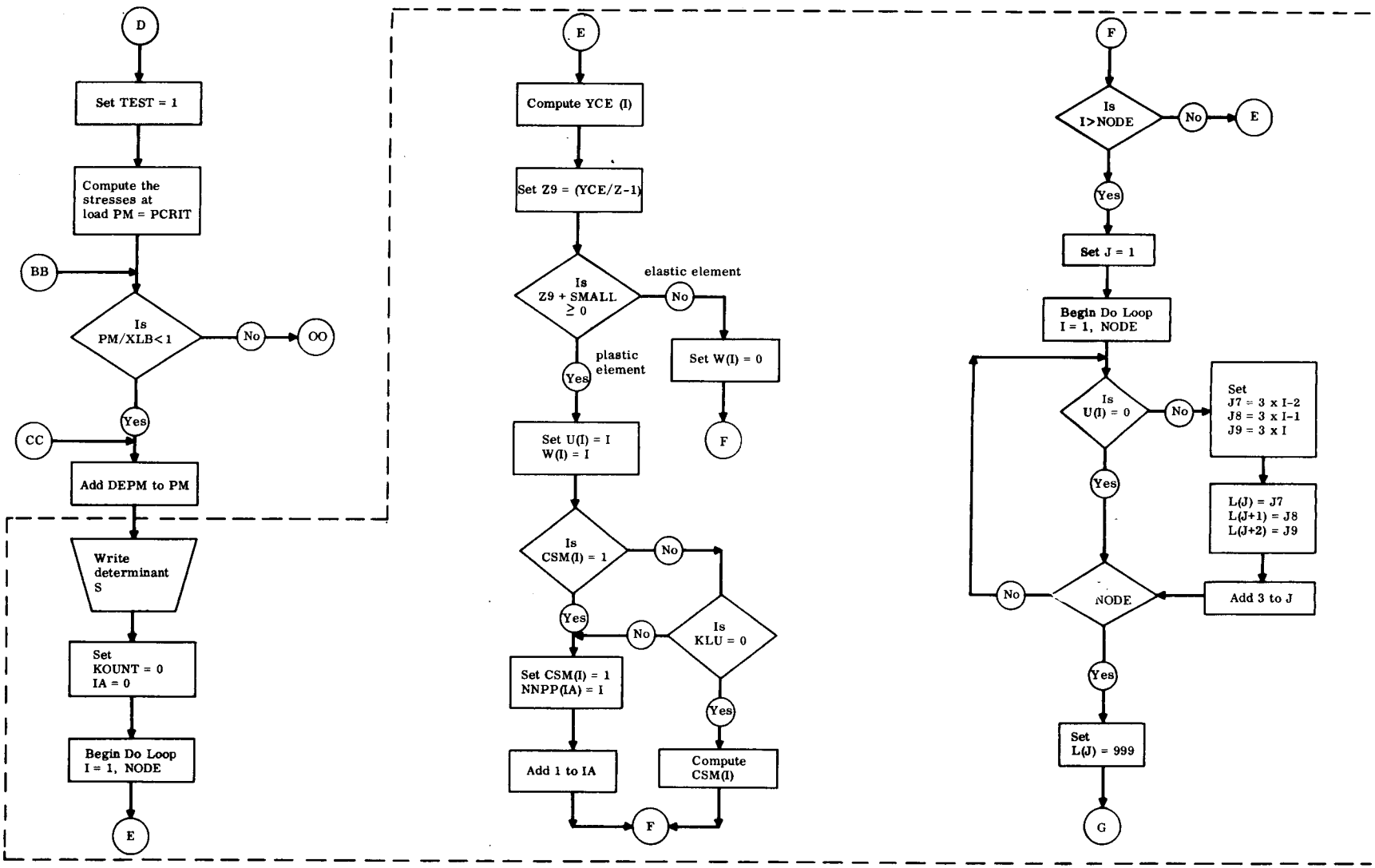
APPENDIX A

FLOW CHARTS AND LISTINGS FOR THE PLASTIC ANALYSIS OF STABLE STRUCTURES

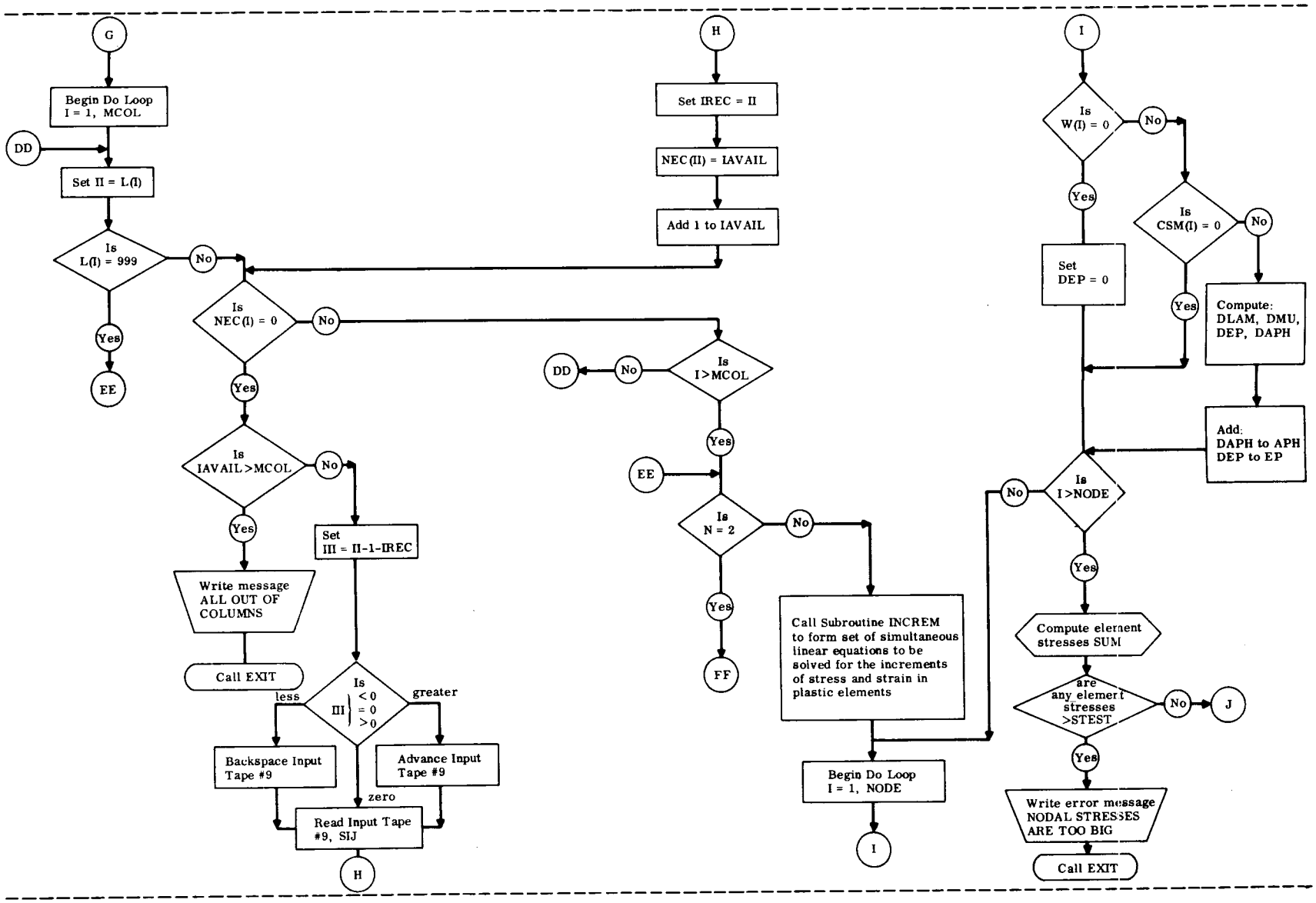
STRESS PROGRAM



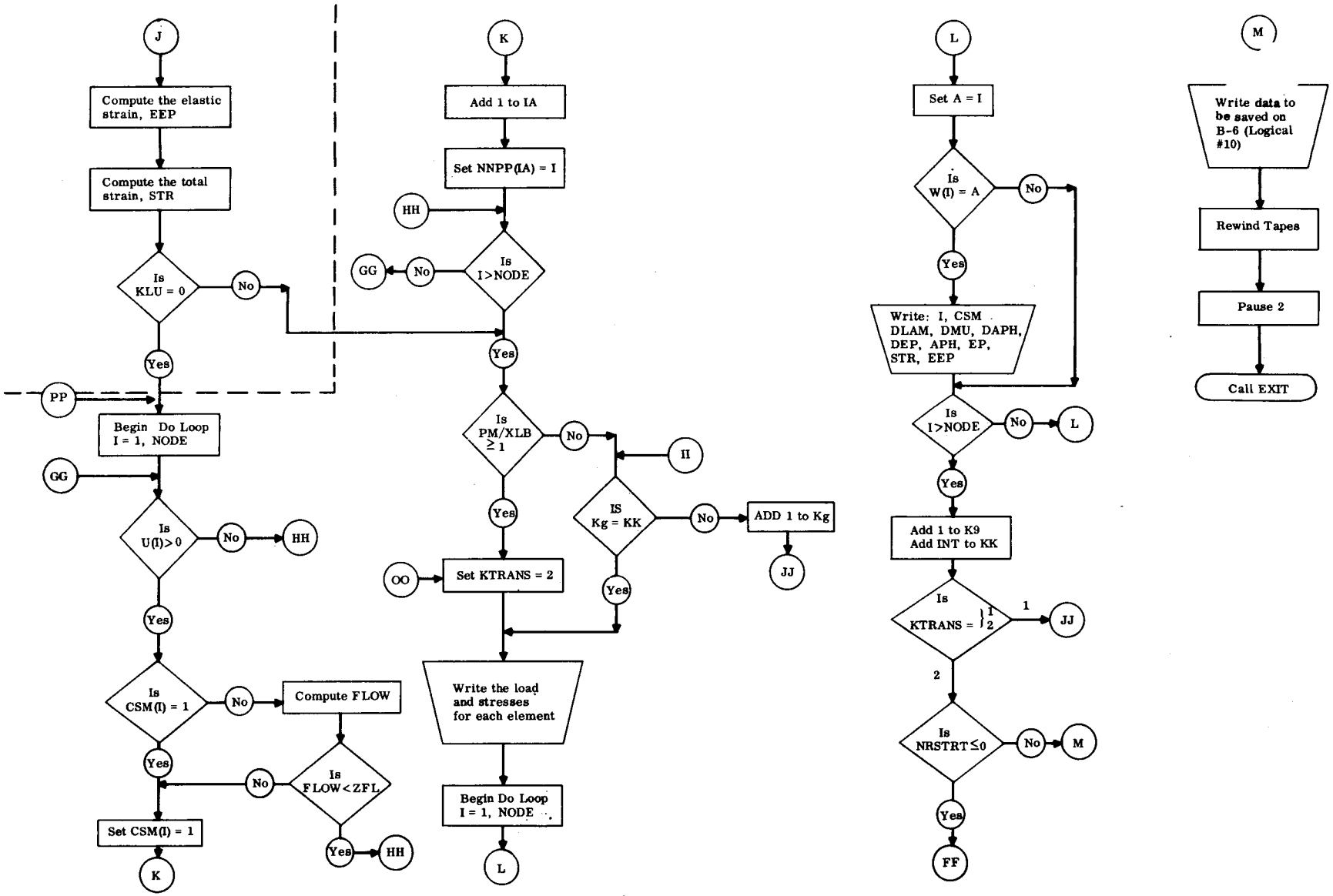
STRESS PROGRAM



STRESS PROGRAM



STRESS PROGRAM



```
$ EXEC U    TE        IBJOB
$ IBJO B              FIOCS,MAP
$ IBFT C     STRESS   LIST,REF
            DIMENSION APH(165)
            DIMENSION BA(60)
            DIMENSION BAPH(165)
            DIMENSION CA(3,3)
            DIMENSION CSM(55)
            DIMENSION DAPH(165)
            DIMENSION DEP(165)
            DIMENSION DLAM(55)
            DIMENSION DMEW(55)
            DIMENSION DSUM(165)
            DIMENSION EEP(165)
            DIMENSION EP(165)
            DIMENSION E1(55)
            DIMENSION E2(55)
            DIMENSION E9(60)
            DIMENSION L(61)
            DIMENSION NEC(165)
            DIMENSION NNPP(55)
            DIMENSION PSIM(60)
            DIMENSION SIG(165)
            DIMENSION SIJ(165,60)
            DIMENSION SIM(165)
            DIMENSION SQ(165)
            DIMENSION STR(165)
            DIMENSION SUM(165)
            DIMENSION U(55)
            DIMENSION W(55)
            DIMENSION YCE(55)
            DIMENSION ZMTRX(60,60)
            COMMON  APH    , BA     , BAPH   , CA     , CSM    , DAPH
            COMMON  DEP    , DEPM   , DLAM   , DMEW   , DSUM   , EFP
            COMMON  EP     , E9     , KOUNT  , KLU    , L      , MCOL
            COMMON  NEC    , NNPP   , NODE   , NP     , N3NO   , PSIM
            COMMON  S      , SIG    , SIJ    , SIM    , SQ     , STR
            COMMON  SUM    , U      , W      , YCE    , ZMTRX
C
C           STRESS METHOD - STEPWISE LINEARIZATION PROCEDURE
C
            PAUSE 1
            CONST=1.0E+06
            REWIND 9
            READ(5,319) IRSTRT,RMLDX
            XLB=RMLDX
            IF(IRSTRT)60,60,61
   61       READ(11) APH,BA,BAPH,CA,CSM,DAPH,DEP,DLAM,DMEW,DSUM,EEP,EP,E1,F2,
     *                E9,L,NEC,NNPP,PSIM,SIG,SIJ,SIM,SQ,STR,SUM,U,W,YCE,ZMTRX,
     *                IAVAIL,INT,IREC,KLU,MCOL,NCOMP,NODE,PM,PCRIT,SIGO,T,Z,
     4                ZFL,DEPM,S,TEST,G,MCOUNT,GNU,E,D
            NP=MCOL
            N3NO=NCOMP
            KK=INT
            IREC=0
            K9=KK
            N=1
            WRITE(6,320)
            WRITE(6,321) INT,SIGO,E,GNU,DEPM,D,T,XLB
            IF(XLB/PM -1.0)62,62,63
```

```
62      KTRANS=2
        GO TO 232
63      KTRANS=1
        GO TO 232
60      READ(5,322) NODE,NCOMP,MCOL,INT
        KTRANS=1
        N3NO=NCOMP
        NP=MCOL
        READ(5,323) SIGO,E,GNU,ZFL
        N=0
        CALL SETAPE (9,2)
        READ (9)(SIM(I),I=1,N3NO)
        REWIND 9
        ALPH=0.
        DO 90 I=1,NODE
        APH(3*I-2)=ALPH
90      APH(3*I-1)=ALPH
        IAVAIL=1
        IREC=0
        DO 91 K=1,NODE
        U(K)=0.
        DLAM(K)=0.
        DMEW(K)=0.
91      YCE(K)=0.
        MCPL1=MCOL+1
        DO 92 I=1,MCPL1
92      L(I)=0
        DO 93 J=1,N3NO
        APH(J)=0.
        BAPH(J)=0.
        DAPH(J)=0.
        DEP(J)=0.
        DSUM(J)=0.
        EEP(J)=0.
        EP(J)=0.
        SIG(J)=0.
        SQ(J)=0.
        STR(J)=0.
        SUM(J)=0.
93      NEC(J)=0
40      TEST=0.
        PM=0.0
        KK=1
        K9=1
        DO 94 I=1,NODE
        CSM(I)=0.0
        NNPP(I)=0
94      W(I)=0.0
        IF(N)65,65,66
65      N=1
        READ(5,324) PMA,DEPM,D,T,XLB,KLU,NRSTRT
        WRITE(6,320)
        WRITE(6,313) INT,SIGO,E,GNU,DEPM,D,T,XLB
        DO 95 J=1,NODE
        N1=3*J-2
        N2=3*J-1
        N3=3*J
        ZIB1=SIM(N1)*PMA
        ZIB2=SIM(N2)*PMA
        ZIB3=SIM(N3)*PMA
```

```
95      YCE(J)=(ZIB1 -ALPH)**2-(ZIB1-ALPH)*(ZIB2-ALPH)+(ZIB2-ALPH)**2+3.0*
   1    (ZIB3)**2
        R=0.0
        DO 96 I=1,NODE
96      R=AMAX1(YCE(I),R)
        DO 97 I=1,NODE
        IF(YCE(I)-R)97,20,97
97      CONTINUE
20      N1=I
        IF(KLU)67,67,68
68      CSM(I)=1.
        NNPP(1)=I
67      BZ=(ALPH-SQ(3*N1-2))*(2.0*SIM(3*N1-2)-SIM(3*N1-1))
        BY=(ALPH-SQ(3*N1-1))*(2.0*SIM(3*N1-1)-SIM(3*N1-2))
        BX=6.0*SIM(3*N1)*(-1.0*SQ(3*N1))
        BV=(-1.0)*(BZ+BY+BX)
        BA(1)=SIM(3*N1-2)**2
        BB=SIM(3*N1-2)*SIM(3*N1-1)
        BC=SIM(3*N1-1)**2
        BD=3.0*(SIM(3*N1)**2)
        BE=BA(1)-BB+BC+BD
        B=BV/BE
        C1=((ALPH-SQ(3*N1-2))**2+(ALPH-SQ(3*N1-1))**2-(ALPH-SQ(3*N1-1))*(A
   1    LPH-SQ(3*N1-2))+3.*(-1.*SQ(3*N1))**2-SIG0**2)/BE
        IF(PMA)69,69,70
70      PCRIT=((-1.0*B)+SQRT(B**2-4.0*C1))/2.0
        GO TO 71
69      PCRIT=((-1.0*B)-SQRT(B**2-4.0*C1))/2.0
        GO TO 71
66      READ(5,324) PMA,DEPM,D,T,XLB,KLU,NRSTRT
        GO TO (33,34) , KTRANS
34      DO 124 I=1,NODE
        W(I)=0.
        CSM(I)=0.
        NNPP(I)=0
        BAPH(3*I-2) = APH(3*I-2)
        BAPH(3*I-1) =  APH(3*I-1)
124     BAPH(3*I) = APH(3*I)
        KQ=KK
        KTRANS=1
33      N=1
        IF(PMA)72,73,72
73      REWIND 9
        PAUSE 2
        CALL EXIT
72      WRITE(6,312)
        WRITE(6,320)
        WRITE(6,313) INT,SIG0,E,GNU,DEPM,D,T,XLB
        DO 98 I=1,N3NO
98      SIG(I)=SIM(I)*PMA+SQ(I)
        DO116 I=1,NODE
116     YCE (I)=((SIG(3*I-1)-BAPH(3*I-1))**2)-((SIG(3*I-1)-BAPH(3*I-1))*(S
   1    IG(3*I-2)-BAPH(3*I-2)))+((SIG(3*I-2)-BAPH(3*I-2))**2)+(3.0*((SIG(3
   2    *I)-BAPH(3*I))**2))
        R=0.0
        DO 99 I=1,NODE
99      R=AMAX1(YCE(I),R)
        DO 100 I=1,NODE
        IF(YCE(I)-R)100,74,100
100     CONTINUE
```

```
 74     N1=I
        IF(KLU)75,75,76
 76     CSM(I)=1.
        NNPP(1)=I
 75     BZ=(BAPH(3*N1-2)-SQ(3*N1-2))*(2.0*SIM(3*N1-2)-SIM(3*N1-1))
        BY=(BAPH(3*N1-1)-SQ(3*N1-1))*(2.0*SIM(3*N1-1)-SIM(3*N1-2))
        BX=(6.0*SIM(3*N1)*(BAPH(3*N1)-SQ(3*N1)))
        BV=(-1.0)*(BZ+BY+BX)
        BA(1)=SIM(3*N1-2)**2
        BB=SIM(3*N1-2)*SIM(3*N1-1)
        BC=SIM(3*N1-1)**2
        BD=3.0*(SIM(3*N1)**2)
        BE=BA(1)-BB+BC+BD
        B=BV/BE
        C1=(((BAPH(3*N1-2)-SQ(3*N1-2))**2+(BAPH(3*N1-1)-SQ(3*N1-1))**2)-((
     1  BAPH(3*N1-1)-SQ(3*N1-1))*(BAPH(3*N1-2)-SQ(3*N1-2)))+((3.0*(BAPH(3*
     2  N1)-SQ(3*N1))**2)-SIG0**2)) /((SIM(3*N1-2)**2)-((SIM(3*N1-2))*(SIM
     3  (3*N1-1)))+(SIM(3*N1-1)**2)+(3.0*SIM(3*N1)**2))
        IF(PMA)77,77,78
 78     PCRIT=((-1.0*B)+SQRT(B**2-4.0*C1))/2.0
        GO TO 71
 77     PCRIT=((-1.0*B)-SQRT(B**2-4.0*C1))/2.0
 71     IF(PCRIT/PMA)79,80,80
 80     IF(TEST) 79,79,37
 37     WRITE(6,308)
        WRITE(6,309)
        DO 101 J=1,NODE
        REST1=(SQ(3*J-2)-GNU*SQ(3*J-1))/E+EP(3*J-2)
        REST2=(SQ(3*J-1)-GNU*SQ(3*J-2))/E+EP(3*J-1)
        REST3=2.0*(1.0+GNU)*SQ(3*J)/E+EP(3*J)
        WRITE(6,310)J,SQ(3*J-2),SQ(3*J-1),SQ(3*J),REST1,REST2,REST3
101     CONTINUE
 79     IF(XLB)41,73,41
 41     TEST=1.
        PM=PCRIT
        DO 102 J=1,N3NO
        SUM(J)=SIM(J)*PM+SQ(J)
102     SIG(J)=SUM(J)
        Z=SIG0**2
        SMALL=CONST/Z
        D2=D/SQRT(3.0)
 83     PM=PM+DEPM
        WRITE(6,326) S
        IA=0
        KOUNT=0
        DO 103 I=1,NODE
        YCE(I)=((SIG (3*I-2)-APH(3*I-2))**2)-((SIG (3*I-2)-APH(3*I-2))*(SI
     1  G (3*I-1)-APH(3*I-1)))+((SIG (3*I-1)-APH(3*I-1))**2)+3.0*((SIG (3*
     2  I)-APH(3*I))**2)
        Z9= (YCE(I)/Z-1.0)
        IF (Z9 + SMALL )85,86,86
 86     U(I) = I
        W(I)=I
        IF(CSM(I) - 1.0) 87,88,87
 87     IF(KLU) 88,89,88
 88     CSM(I) = 1.0
        IA=IA+1
        NNPP(IA)=I
        GO TO 103
 89     KOUNT = KOUNT + 1
```

```
      S1=T*((ABS(SUM (3*I-2)-BAPH(3*I-2))/D)**(T-1.0))/D
      S2=T*((ABS(SUM (3*I-1)-BAPH(3*I-1))/D)**(T-1.0))/D
      S3=T*((ABS(SUM (3*I)-BAPH(3*I))/D2)**(T-1.0))/D2
      QZIG=(((SUM (3*I-1)-APH(3*I-1))**2)+((SUM (3*I-2)-APH(3*I-2))**2)+
     1 ((SUM (3*I)-APH(3*I)) **2))**.5
      V1       =(SUM (3*I-2)-APH(3*I-2))/QZIG
      V2       =(SUM (3*I-1)-APH(3*I-1))/QZIG
      V3     =(SUM (3*I)-APH(3*I))/QZIG
      CSM(I)=((V1**2)*S1)+((V2**2)*S2)+(( V3**2)*S3)
      GO TO 103
 85   W(I) = 0.0
103   CONTINUE
      J=1
      DO 109 I=1,NODE
      IF(U(I))26,109,26
 26   J7=3*I-2
      J8=3*I-1
      J9=3*I
      L(J)=J7
      L(J+1)=J8
      L(J+2)=J9
      J=J+3
109   CONTINUE
      L(J)=999
      DO 110 I=1,MCOL
      II=L(I)
      IF(L(I) - 999 ) 211,209,211
211   IF(NEC(II))213,214,213
214   IF(IAVAIL-MCOL)215,215,204
215   III=II-1-IREC
      IF(III)216,217,218
218   DO 122 JJ=1,III
122   READ (9)
      GO TO 217
216   III=(-1)*III
      DO 123 JJ=1,III
123   BACKSPACE 9
217   READ (9) (SIJ(JJ,IAVAIL),JJ=1,N3NO)
      IREC=II
      NEC(II)=IAVAIL
      IAVAIL=IAVAIL+1
      GO TO 211
213   NN=NEC(II)
110   CONTINUE
204   WRITE(6,311)
      CALL EXIT
209   CALL INCREM

      COMPUTE YIELD CONDITION EQUATION

      DO 104  I=1,NODE
      IF(W(I)) 104,104,221
221   IF(CSM(I) - 1.0) 222,104,222
222   F=(((SUM(3*I-2)-APH(3*I-2))*(DSUM(3*I-2)-(DSUM(3*I-1))/2.0))+((SUM
     1 (3*I-1)-APH(3*I-1))*(DSUM(3*I-1)-(DSUM(3*I-2))/2.0))+(3.0*(SUM(3*I
     2 )-APH(3*I))*(DSUM(3*I))))
      DLAM(I)=(2.0*CSM(I))*F/((5.0*(SUM(3*I-2)-APH(3*I-2)) **2)-(8.0*(SU
     1 M(3*I-2)-APH(3*I-2))*(SUM(3*I-1)-APH(3*I-1)))+(5.0*(SUM(3*I-1)-APH
     2 (3*I-1))**2)+(36.0*(SUM(3*I)-APH(3*I))**2))
      DMEW(I)=F /(((SUM(3*I-2)-APH(3*I-2))**2)-((SUM( 3*I-2)-APH(3*I-2))
```

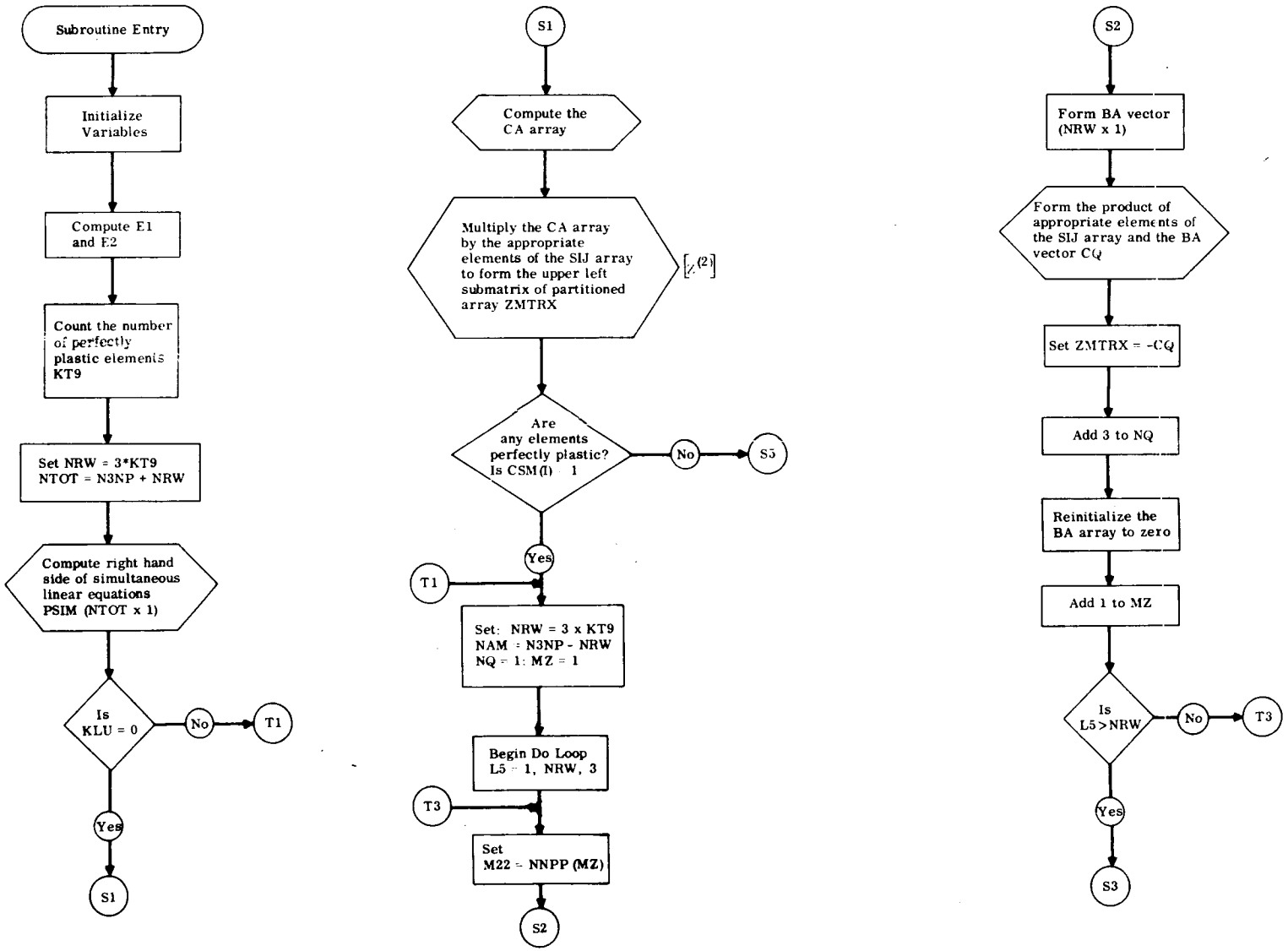
```
     1  *(SUM(3*I-1)-APH(3*I-1)))+((SUM(3*I-1)-APH(3*I-1))**2)+(3.0*(SUM(3
     2  *I)-APH(3*I))**2))
        IF(DLAM(I)) 223,224,224
C
C       COMPUTE THE DELTA EPSILONS
C
  224   DEP(3*I-2)=2.0*((SUM(3*I-2)-APH(3*I-2))-(SUM(3*I-1)-APH(3*I-1))/2.
     1  0)*DLAM(I)
        DEP(3*I-1)=2.0*((SUM(3*I-1)-APH(3*I-1))-(SUM(3*I-2)-APH(3*I-2))/2.
     1  0)*DLAM(I)
        DEP(3*I)=6.0*((SUM(3*I)-APH(3*I)))*DLAM(I)
C
C       COMPUTE THE DELTA ALPHAS
C
        DAPH(3*I-2)=(SUM(3*I-2)-APH(3*I-2))*DMEW(I)
        DAPH(3*I-1)=(SUM(3*I-1)-APH(3*I-1))*DMEW(I)
        DAPH(3*I)=(SUM(3*I)-APH(3*I))*DMEW(I)
C
C       CALCULATE ALPHA SUMS
C
        APH(3*I-2)=APH(3*I-2)+DAPH(3*I-2)
        APH(3*I-1)=APH(3*I-1)+DAPH(3*I-1)
        APH(3*I)=APH(3*I)+DAPH(3*I)
C
C       CALCULATE EPSILON SUMS
C
        EP(3*I-2)=EP(3*I-2)+DEP(3*I-2)
        EP(3*I-1)=EP(3*I-1)+DEP(3*I-1)
        EP(3*I)=EP(3*I)+DEP(3*I)
        GO TO 104
  223   DEP(3*I-2) = 0.0
        DEP(3*I-1)=0.0
        DEP(3*I)=0.0
  104   CONTINUE
        DO 105 K= 1,N3NO
        SJ=0.0
        DO 106 I= 1,N3NO
        NT=(I-1)/3+1
        IF(U(NT)) 225,106,225
  225   K1=NEC(I)
        SJ=SJ+SIJ(K,K1)*EP(I)
  106   CONTINUE
  105   SQ(K) = SJ
        DO 107 LX=1,N3NO
        SUM(LX)=SIM(LX)*PM+SQ(LX)
  107   SIG(LX) = SUM(LX)
        STEST = 10**10
        DO 108 LX=1,N3NO
        IF(SUM(LX) - STEST) 108,108,226
  226   WRITE(6,317)
        CALL EXIT
  108   CONTINUE
        DO 114 I=1,NODE
        EEP(3*I-2)=(SUM(3*I-2)-GNU*SUM(3*I-1))/E
        EEP(3*I-1)=(SUM(3*I-1)-GNU*SUM(3*I-2))/E
  114   EEP(3*I)= 2.0* (1.+ GNU) * SUM(3*I) /E
        DO 115 J=1,N3NO
  115   STR(J) = EP(J) + EEP(J)
C
C       THE TEST FOR PERFECT PLASTICITY
```

```
C
      IF ( ZFL ) 35,36,36
   36 DO 111 I=1,NODE
      IF (U(I)) 111,111,227
  227 IF (KLU) 111,228,111
  228 IF (CSM(I) - 1.0) 230,229,230
  230 FLOW=(APH(3*I-2)-BAPH(3*I-2))**2-(APH(3*I-2)-BAPH(3*I-2))*(APH(3*I
     1 -1)-BAPH(3*I-1))+(APH(3*I-1)-BAPH(3*I-1))**2+(3.0*(APH(3*I)-BAPH(3
     2 *I))**2)
      IF (ZFL - FLOW) 229,229,111
  229 CSM(I) = 1.0
      IA=IA+1
      NNPP(IA)=I
  111 CONTINUE
   95 IF ( PM/XLB - 1.0 ) 231,232,232
  231 IF ( K9 - KK ) 233,235,233
  233 K9=K9+1
      GO TO 83
  232 KTRANS=2
      WRITE(6,301)
      WRITE(6,302)
      WRITE(6,300)(J,SUM (3*J-2),SUM (3*J-1),SUM (3*J),J=1,NODE)
      DO 113 I=1,NODE
      A=I
      IF (W(I)-A) 113,234,113
  234 WRITE(6,314)
      WRITE(6,315) I,CSM(I),DLAM(I),DMEW(I)
      WRITE(6,306)
      WRITE (6,305) DAPH(3*I-2),DAPH(3*I-1),DAPH(3*I),DEP(3*I-2),DEP(3*I
     1 -1),DEP(3*I)
      WRITE(6,307)
      WRITE (6,305) APH(3*I-2),APH(3*I-1),APH(3*I),EP(3*I-2), EP(3*I-1),
     1 EP(3*I)
      WRITE(6,316)
      WRITE (6,305)  STR(3*I-2),STR(3*I-1),STR(3*I),EEP(3*I-2),EEP(3*I-1
     1 ),EEP(3*I)
  113 CONTINUE
      K9=K9+1
      KK=KK+INT
      GO TO (83,42) , KTRANS
   42 IF (NRSTRT) 66,66,236
  236 WRITE(10) APH,BA,BAPH,CA,CSM,DAPH,DEP,DLAM,DMEW,DSUM,EEP,EP,E1,E2,
     *          E9,L,NEC,NNPP,PSIM,SIG,SIJ,SIM,SQ,STR,SUM,U,W,YCE,ZMTRX,
     *          IAVAIL,INT,IREC,KLU,MCOL,NCOMP,NODE,PM,PCRIT,SIGO,T,Z,
     *          ZFL,DEPM,S,TEST,G,MCOUNT,GNU,E,D
      REWIND 10
      REWIND 9
      PAUSE 2
      CALL EXIT
  300 FORMAT(1H ,3X,I3,7X,E16.8,3X,E16.8,3X,E16.8)
  301 FORMAT(/ 21X,17H ELEMENT STRESSES, / )
  302 FORMAT(12HOELEMENT NO.,7X,7HSIGMA-X,14X,7HSIGMA-Y,12X,7HTAU-X,Y)
  305 FORMAT(8X,E16.8,3X,E16.8,3X,E16.8,3X,E16.8,3X,E16.8,3X,E16.8)
  306 FORMAT(1H0,9X,13HDELTA ALPHA-X,6X,13HDELTA ALPHA-Y,6X,12HDELTA GAM
     1 -XY,7X,10HDELTA EP-X,9X,10HDELTA EP-Y,9X,12HDELTA GAM-XY)
  307 FORMAT(1H0,9X,11HTOT ALPHA-X,8X,11HTOT ALPHA-Y,8X,12HTOT ALPHA-XY,
     1 7X,12HPLASTIC EP-X,7X,12HPLASTIC EP-Y,7X,14HPLASTIC GAM-XY)
  308 FORMAT(20X,17HRESIDUAL STRESSES,41X,16HRESIDUAL STRAINS)
  309 FORMAT(8HOELEMENT,6X,1HX,18X,1HY,18X,2HXY,23X,1HX,18X,1HY,18X,2HXY
     1 )
```

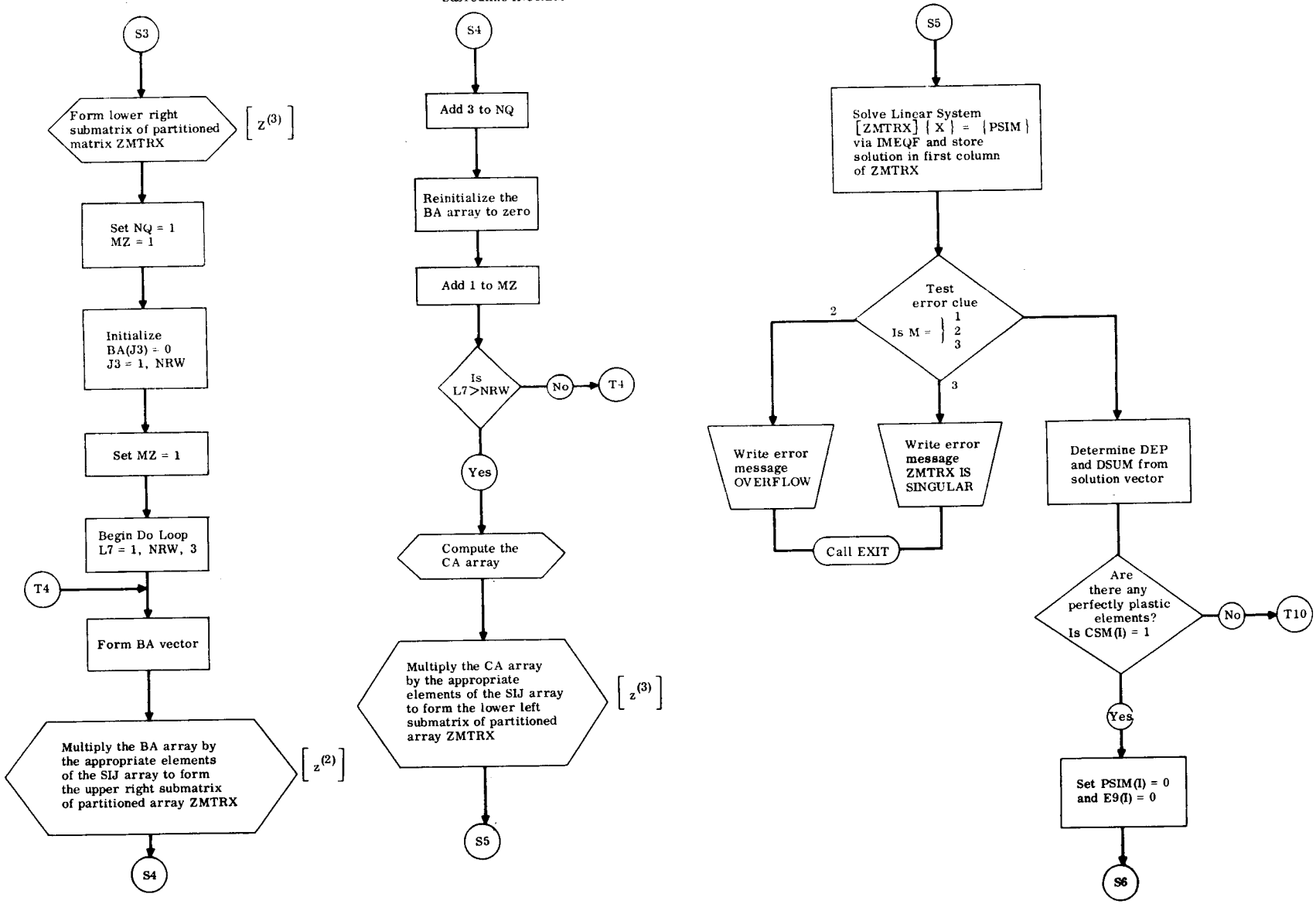
```
310     FORMAT(1H0,I2,3E19.8,3X,3E19.8)
311     FORMAT(19H1ALL OUT OF COLUMNS)
312     FORMAT(1H0,22HUNLOADING CYCLE BEGINS)
313     FORMAT(1H0(I10,8E15.6))
314     FORMAT(13H0 ELEMENT NO.,9X,1HC,17X,7HDLAMBDA,16X,3HDMU)
315     FORMAT(5X,I2,8X,E16.8,5X,E16.8,5X,E16.8)
316     FORMAT(1H0,10X,10HSTRAIN - X,9X,10HSTRAIN - Y,9X,11HSTRAIN - XY,7X
    1   ,12HELASTIC EP-X,7X,12HELASTIC EP-Y,7X,14HELASTIC GAM-XY)
317     FORMAT(1H0,30HTHE NODAL STRESSES ARE TOO BIG)
319     FORMAT(I5,E15.7)
320     FORMAT(6H1INPUT,/ 7X,3HINT,11X,4HSIGO,14X,1HE,13X,2HNU,11X,4HDELP,
    1   14X,1HD,14X,1HT,10X,5HRMXLD)
321     FORMAT(1H0(I10,8E15.6))
322     FORMAT(5I10)
323     FORMAT(5E15.7)
324     FORMAT(5E15.7,I2,I3)
325     FORMAT(17H0 LOAD LEVEL ,P =,F10.0,)
326     FORMAT(14H0 DETERMINANT=,E16.8)
          END


        SUBROUTINE SETAPE(NUNIT,NFILE)
        DATA SIGNAL /O111111111111 /
        REWIND 9
        NSET=NFILE-1
        IF(NSET)3,3,4
  4     DO 1 I=1,NSET
  2     READ (9) X
        IF(X-SIGNAL)2, 1,2
  1     CONTINUE
  3     RETURN
        END
```

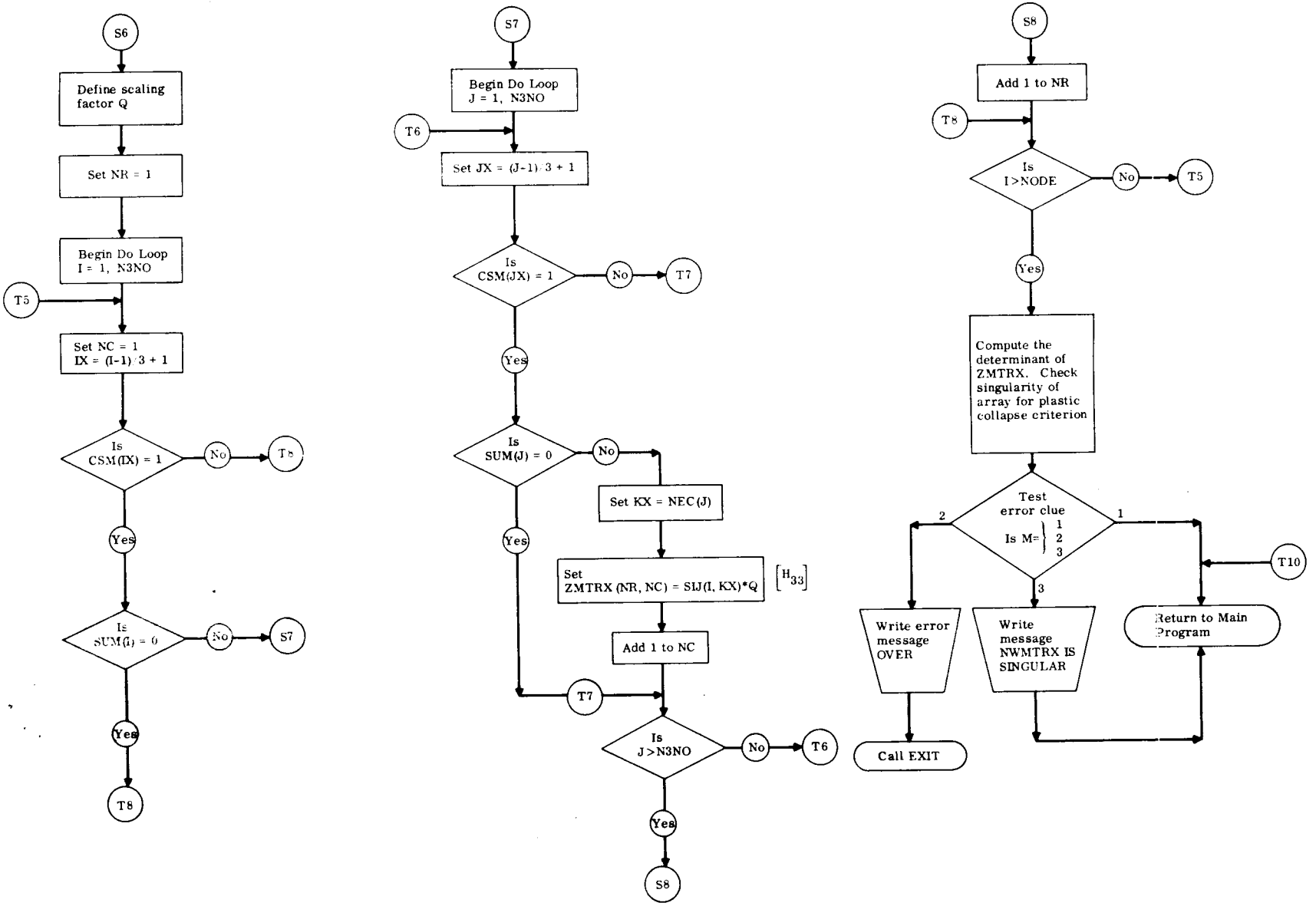
STRESS PROGRAM
Subroutine INCREM



STRESS PROGRAM
Subroutine INCREM



STRESS PROGRAM
Subroutine INCREM



SUBROUTINE INCREM

```
$ IBFT C     INCREM  LIST,REF
           SUBROUTINE INCREM
           DIMENSION APH(165)
           DIMENSION BA(60)
           DIMENSION BAPH(165)
           DIMENSION CA(3,3)
           DIMENSION CSM(55)
           DIMENSION DAPH(165)
           DIMENSION DEP(165)
           DIMENSION DLAM(55)
           DIMENSION DMEW(55)
           DIMENSION DSUM(165)
           DIMENSION EEP(165)
           DIMENSION EP(165)
           DIMENSION E1(55)
           DIMENSION E2(55)
           DIMENSION E9(60)
           DIMENSION L(61)
           DIMENSION NEC(165)
           DIMENSION NNPP(55)
           DIMENSION PSIM(60)
           DIMENSION SIG(165)
           DIMENSION SIJ(165,60)
           DIMENSION SIM(165)
           DIMENSION SQ(165)
           DIMENSION STR(165)
           DIMENSION SUM(165)
           DIMENSION U(55)
           DIMENSION W(55)
           DIMENSION YCE(55)
           DIMENSION ZMTRX(60,60)
           COMMON  APH     , BA      , BAPH    , CA      , CSM     , DAPH
           COMMON  DEP     , DEPM    , DLAM    , DMEW    , DSUM    , EEP
           COMMON  EP      , E9      , KOUNT   , KLU     , L       , MCOL
           COMMON  NEC     , NNPP    , NODE    , NP      , N3NO    , PSIM
           COMMON  S       , SIG     , SIJ     , SIM     , SQ      , STR
           COMMON  SUM     , U       , W       , YCE     , ZMTRX
           N3NP=3*KOUNT
           KT9=0
           DO 102 J=1,NODE
           N3=3*J
           N2=N3-1
           N1=N3-2
           IF(CSM(J)-1.0)102,103,102
  103      E1(J)=((SUM (N2)-APH(N2))-(SUM (N1)-APH(N1))/2.0)/((SUM (N1)-APH(N
     1     1))-(SUM (N2)-APH(N2))/2.0)
           E2(J)=(3.0*(SUM (N3)-APH(N3)))/((SUM (N1)-APH(N1))-(SUM (N2)-APH(N
     1     2))/2.0)
           KT9=KT9+1
  102      CONTINUE
           NRW=3*KT9
           NTOT=N3NP+NRW
           DO 104 I=1,NP
           E9(I)=0.0
           PSIM(I)=0.0
           DO 104 J=1,NP
  104      ZMTRX(I,J)=0.0
C          FORMATION OF THE PSIM MATRIX
           MK=1
           DO 1 I=1,NODE
```

```
      IF (W(I)) 1,1,2
    2 IF (CSM(I)-1.0) 3,1,3
    3 L1=3*I-2
      PSIM(MK)=SIM(L1)*DEPM
      PSIM(MK+1)=SIM(L1+1)*DEPM
      PSIM(MK+2)=SIM(L1+2)*DEPM
      MK=MK+3
    1 CONTINUE
      DO 100 I=1,NODE
      IF (CSM(I)-1.0) 100,101,100
  101 L1=3*I-2
      PSIM(MK)=SIM(L1)*DEPM
      PSIM(MK+1)=SIM(L1+1)*DEPM
      PSIM(MK+2)=SIM(L1+2)*DEPM
      MK=MK+3
  100 CONTINUE
      IF(KLU)16,105,16
C     FORMATION OF THE Z MATRIX (UPPER LEFT)
  105 NQ=1
      DO 4 J=1,NODE
      IF (W(J)) 4,4,5
    5 IF (CSM(J)-1.0) 6,4,6
    6 L2=3*J-2
      L3=L2+1
      L4=L2+2
      KY=NEC(L2)
      D=(5.0*(SUM(L2)-APH(L2))**2  -8.0*(SUM(L2)-APH(L2))*(SUM(L3)-APH(L
     1 3))+5.0*(SUM(L3)-APH(L3))**2  +36.0*(SUM(L4)-APH(L4))**2  )/(4.0*C
     2 SM(J))
      CA(1,1)=((SUM(L2)-APH(L2))-.5*(SUM(L3)-APH(L3)))**2/D
      CA(2,2)=((SUM(L3)-APH(L3))-.5*(SUM(L2)-APH(L2)))**2/D
      CA(3,3)=(3.0*(SUM(L4)-APH(L4)))**2/D
      CA(1,2)=(((SUM(L2)-APH(L2))-.5*(SUM(L3)-APH(L3)))*((SUM(L3)-APH(L3
     1 ))-.5*(SUM(L2)-APH(L2))))/D
      CA(1,3)=(((SUM(L2)-APH(L2))-.5*(SUM(L3)-APH(L3)))*(3.0*(SUM(L4)-AP
     1 H(L4))))/D
      CA(2,1)=CA(1,2)
      CA(2,3)=(((SUM(L3)-APH(L3))-.5*(SUM(L2)-APH(L2)))*(3.0*(SUM(L4)-AP
     1 H(L4))))/D
      CA(3,1)=CA(1,3)
      CA(3,2)=CA(2,3)
      DO 7 NZ=1,3
      K6=0
      DO 8 K=1,N3NO
      IX=(K-1)/3+1
      IF (W(IX)) 8,8,9
    9 IF (CSM(IX)-1.0) 10,8,10
   10 S9=0.0
      K6=K6+1
      KY=NEC(L2)
      DO 11 KK=1,3
      S9=S9+SIJ(K,KY)*CA(KK,NZ)
   11 KY=KY+1
      ZMTRX(K6,NQ)=-1.0*S9
    8 CONTINUE
      NQ=NQ+1
    7 CONTINUE
    4 CONTINUE
      DO 12 IY=1,N3NP
   12 ZMTRX(IY,IY)=1.0+ZMTRX(IY,IY)
```

```
C           FORMATION OF THE Z MATRIX (LOWER RIGHT)
            DO 14 I=1,NODE
            IF(CSM(I)-1.0)14,16,14
   14       CONTINUE
            GO TO 38
   16       NRW=3*KT9
            NAM=N3NP-NRW
            NQ=1
            MZ=1
            DO 17 L5=1,NRW,3
            M22=NNPP(MZ)
            BA(L5)=1.0
            BA(L5+1)=E1(M22)
            BA(L5+2)=E2(M22)
            NX=0
            DO 18 I=1,N3NO
            IX=(I-1)/3+1
            IF (CSM(IX)-1.0) 18,19,18
   19       NZ=1
            CQ=0.0
            DO 20 LY=1,N3NO
            J=(LY-1)/3+1
            IF (CSM(J)-1.0) 20,21,20
   21       KZ=NEC(LY)
            CQ=CQ+SIJ(I,KZ)*BA(NZ)
            NZ=NZ+1
   20       CONTINUE
            NX=NX+1
            N1=NX+N3NP
            N2=NQ+N3NP
            ZMTRX(N1,N2)=-CQ
   18       CONTINUE
            NQ=NQ+3
            DO 65 J3=1,NRW
   65       BA(J3)=0.0
   17       MZ=MZ+1
            MB=1
            DO 22 KRW=1,NRW,3
            M23=NNPP(MB)
            MRW=KRW+N3NP
            ZMTRX(MRW+1,MRW+1)=1.0
            ZMTRX(MRW+2,MRW+2)=1.0
            ZMTRX(MRW,MRW+1)=-E1(M23)
            ZMTRX(MRW,MRW+2)=-E2(M23)
   22       MB=MB+1
C           FORMATION OF Z MATRIX (UPPER RIGHT)
            NQ=1
            MZ=1
            DO 23 J3=1,NRW
   23       BA(J3)=0.0
            DO 24 L7=1,NRW,3
            M22=NNPP(MZ)
            BA(L7)=1.0
            BA(L7+1)=E1(M22)
            BA(L7+2)=E2(M22)
            NX=0
            DO 25 I=1,N3NO
            IX=(I-1)/3+1
            IF(W(IX))25,25,26
   26       IF(CSM(IX)-1.0)27,25,27
```

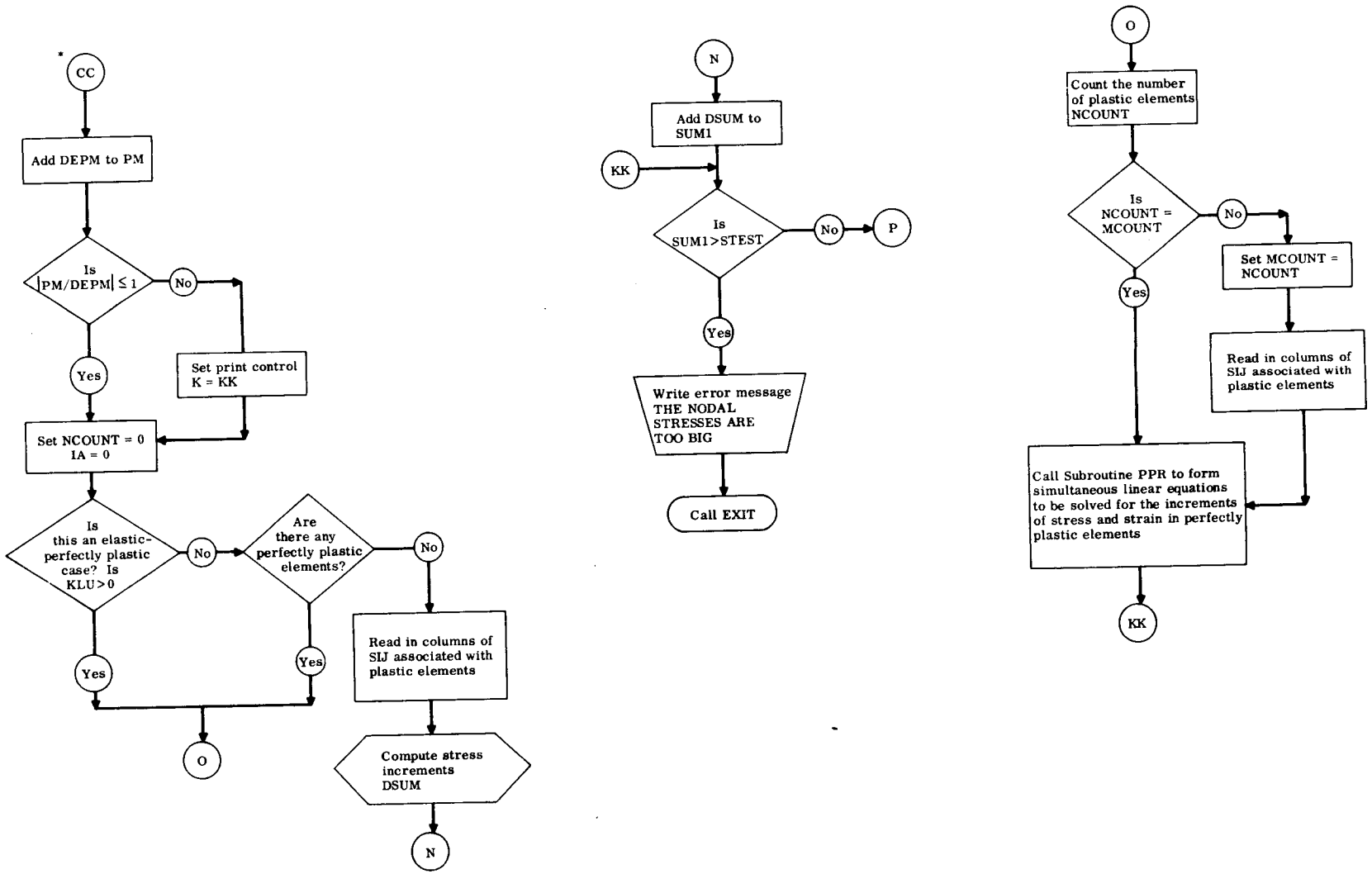
```
27      NZ=1
        CQ=0.0
        DO 28 LY=1,N3NO
        J=(LY-1)/3+1
        IF(CSM(J)-1.0)28,29,28
29      KZ=NEC(LY)
        CQ=CQ+SIJ(I,KZ)*BA(NZ)
        NZ=NZ+1
28      CONTINUE
        NX=NX+1
        N2=NQ+N3NP
        ZMTRX(NX,N2)=-CQ
25      CONTINUE
        NQ=NQ+3
        DO 30 J3=1,NRW
30      BA(J3)=0.0
24      MZ=MZ+1
C       FORMATION OF Z MATRIX (LOWER LEFT)
        NQ=1
        DO 31 J=1,NODE
        IF(W(J))31,31,32
32      IF(CSM(J)-1.0)33,31,33
33      L2=3*J-2
        L3=L2+1
        L4=L2+2
        KZ=NEC(L2)
        D=(5.0*(SUM(L2)-APH(L2))**2  -8.0*(SUM(L2)-APH(L2))*(SUM(L3)-APH(L
     1  3))+5.0*(SUM(L3)-APH(L3))**2  +36.0*(SUM(L4)-APH(L4))**2  )/(4.0*C
     2  SM(J))
        CA(1,1)=((SUM(L2)-APH(L2))-.5*(SUM(L3)-APH(L3)))**2/D
        CA(2,2)=((SUM(L3)-APH(L3))-.5*(SUM(L2)-APH(L2)))**2/D
        CA(3,3)=(3.0*(SUM(L4)-APH(L4)))**2/D
        CA(1,2)=(((SUM(L2)-APH(L2))-.5*(SUM(L3)-APH(L3)))*((SUM(L3)-APH(L3
     1  ))-.5*(SUM(L2)-APH(L2))))/D
        CA(1,3)=(((SUM(L2)-APH(L2))-.5*(SUM(L3)-APH(L3)))*(3.0*(SUM(L4)-AP
     1  H(L4))))/D
        CA(2,3)=(((SUM(L3)-APH(L3))-.5*(SUM(L2)-APH(L2)))*(3.0*(SUM(L4)-AP
     1  H(L4))))/D
        CA(2,1)=CA(1,2)
        CA(3,1)=CA(1,3)
        CA(3,2)=CA(2,3)
        DO 34 NZ=1,3
        K6=0
        DO 35 I=1,N3NO
        IX=(I-1)/3+1
        IF(CSM(IX)-1.0)35,36,35
36      CQ=0.0
        K6=K6+1
        KZ=NEC(L2)
        DO 37 KK=1,3
        CQ=CQ+SIJ(I,KZ)*CA(KK,NZ)
37      KZ=KZ+1
        N3=K6+N3NP
        ZMTRX(N3,NQ)=-CQ
35      CONTINUE
        NQ=NQ+1
34      CONTINUE
31      CONTINUE
C       SOLUTION OF THE SIMULTANEOUS LINEAR EQUATIONS
38      S=0.0
```

```
      M=IMEQF(NP,NTOT,1,ZMTRX,PSIM,S,E9)
      IF(M-2)41,40,39
39    WRITE (6,42)
      CALL EXIT
40    WRITE (6,43)
      CALL EXIT
42    FORMAT(18H ZMTRX IS SINGULAR)
43    FORMAT(9H OVERFLOW)
41    MN=1
      DO 44 K=1,NODE
      IF(W(K))44,44,45
45    IF(CSM(K)-1.0)46,44,46
46    DSUM(3*K-2)=ZMTRX(MN,1)
      DSUM(3*K-1)=ZMTRX(MN+1,1)
      DSUM(3*K)=ZMTRX(MN+2,1)
      MN=MN+3
44    CONTINUE
      DO 47 K=1,NODE
      IF(CSM(K)-1.0)47,48,47
48    DEP(3*K-2)=ZMTRX(MN,1)
      DSUM(3*K-1)=ZMTRX(MN+1,1)
      DSUM(3*K)=ZMTRX(MN+2,1)
      MN=MN+3
47    CONTINUE
      DO 49 I=1,NODE
      IF(CSM(I)-1.0)49,50,49
50    IC=I
      DEP(3*IC-1)=E1(IC)*DEP(3*IC-2)
      DEP(3*IC)=E2(IC)*DEP(3*IC-2)
      EP(3*IC-2)=EP(3*IC-2)+DEP(3*IC-2)
      EP(3*IC-1)=EP(3*IC-1)+DEP(3*IC-1)
      EP(3*IC)=EP(3*IC)+DEP(3*IC)
49    CONTINUE
      DO 56 I=1,NODE
      IF(CSM(I)-1.0)56,57,56
56    CONTINUE
      GO TO 58
57    DO 63 I=1,NP
      PSIM(I)=0.0
      E9(I)=0.0
      DO 63 J=1,NP
63    ZMTRX(I,J)=0.0
      Q=524288.0
      Q=1.0/Q
      NR=1
      DO 52 I=1,N3NO
      NC=1
      IX=(I-1)/3+1
      IF(CSM(IX)-1.0)52,53,52
53    IF(SUM(I)) 64,52,64
64    DO 54 J=1,N3NO
      JX=(J-1)/3+1
      IF(CSM(JX)-1.0)54,55,54
55    IF(SUM(J)) 66,54,66
66    KX=NEC(J)
      ZMTRX(NR,NC)=SIJ(I,KX)*Q
      NC=NC+1
54    CONTINUE
      NR=NR+1
52    CONTINUE
```

```
      NRW1=NR-1
      S=1.0
      M=IMEQF(NP,NRW1,1,ZMTRX,PSIM,S,E9)
      IF(M-2)58,59,60
59    WRITE (6,61)
61    FORMAT(5H OVER)
60    WRITE (6,62)
62    FORMAT(19H NWMTRX IS SINGULAR)
58    CONTINUE
      RETURN
      END
```

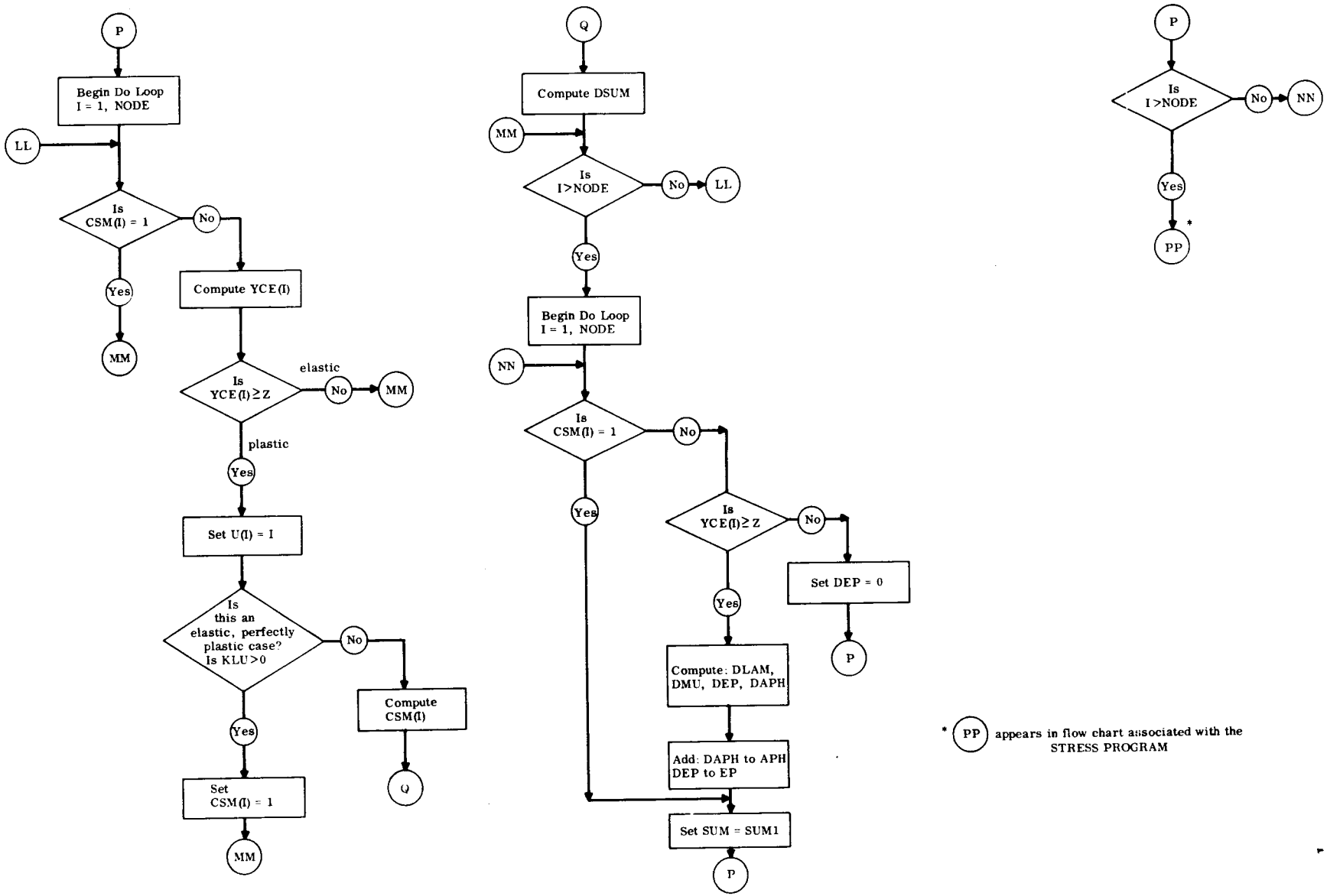
STREPP PROGRAM

Flow of computations similar to that shown for the STRESS PROGRAM except for section enclosed by dashed line. This section is replaced by the following-



* CC appears in flow chart associated with the STRESS PROGRAM

STREPP PROGRAM



STREPP PROGRAM

```
$ IBFT C     STREPP    LIST,REF
C
C            STRESS METHOD - PREDICTOR PROCEDURE
C
             DIMENSION APH(165)
             DIMENSION BA(60)
             DIMENSION BAPH(165)
             DIMENSION CA(60,60)
             DIMENSION CAPH(165)
             DIMENSION CSM(55)
             DIMENSION DEP(165)
             DIMENSION DLAM(55)
             DIMENSION DMEW(55)
             DIMENSION DSUM(165)
             DIMENSION E1(55)
             DIMENSION E2(55)
             DIMENSION E9(60)
             DIMENSION EP(165)
             DIMENSION L(165)
             DIMENSION  NEC(165)
             DIMENSION NNPP(55)
             DIMENSION SIM(165)
             DIMENSION SIJ(165,60)
             DIMENSION SMC(60)
             DIMENSION SMC3(60)
             DIMENSION SQ(165)
             DIMENSION SUM(165)
             DIMENSION SUM1(165)
             DIMENSION X(60)
             DIMENSION YCE(55)
             DIMENSION U(55)
             COMMON  CA
             COMMON  N3NO    , NODE    , MCOL    , CSM     , APH     , DEPM
             COMMON  SIM     , SIJ     , DEP     , SUM1    , NNPP    , L
             COMMON  NEC     , U       , EP      , SQ      , DSUM    , BAPH
             COMMON  YCE     , DLAM    , DMEW    , SUM     , E1      , E2
             COMMON SMC, SMC3, E9, INT, XLB
             COMMON  NCOUNT , CAPH    , BA      , S       , NP      , KLU
             EQUIVALENCE (CA,X)
             PAUSE 1
             REWIND 9
             READ(5,319) IRSTRT,RMLDX
             XLB=RMLDX
             IF(IRSTRT)60,60,61
   61        READ(11) APH,BA,BAPH,CA,CAPH,CSM,DEP,DLAM,DMEW,DSUM,E1,E2,E9,EP,L,
      1                NEC,NNPP,SIM,SIJ,SMC,SMC3,SQ,SUM,SUM1,X,  U,D,E,GNU,
      2                 IAVAIL,INT,IREC,KLU,MCOL,NCOMP,NODE,PM,PCRIT,SIGO,T,Z,
      3               ZFL,DEPM,S,TEST,G,MCOUNT,YCE
             NP=MCOL
             N3NO=NCOMP
             KK=INT
             IREC=0
             K=KK
             N=1
             WRITE(6,320)
             WRITE(6,321) INT,SIGO,E,GNU,DEPM,D,T,XLB
             IF(XLB/PM -1.0)62,62,63
   62        KTRANS=2
             GO TO 232
   63        KTRANS=1
```

```
      GO TO 232
60    READ(5,322) NODE,NCOMP,MCOL,INT
      KTRANS=1
      N3NO=NCOMP
      NP=MCOL
      READ(5,323) SIGO,E,GNU,ZFL
      N=0
      CALL SETAPE (9,2)
      READ (9)(SIM(I),I=1,N3NO)
      REWIND 9
      ALPH=0.
      DO 90 I=1,NODE
      APH(3*I-2)=ALPH
90    APH(3*I-1)=ALPH
      IAVAIL=1
      IREC=0
      DO 91 K=1,NODE
      U(K)=0.
      DLAM(K)=0.
      DMFW(K)=0.
91    YCF(K)=0.
      MCPL1=MCOL+1
      DO 92 I=1,MCPL1
92    L(I)=0
      DO 93 J=1,N3NO
      APH(J)=0.
      BAPH(J)=0.
      DEP(J)=0.
      DSUM(J)=0.
      FP(J)=0.
      SQ(J)=0.
      SUM(J)=0.
93    NFC(J)=0
40    TEST=0.
      PM=0.0
      K=1
      DO 94 I=1,NODE
      NNPP(I)=0
      CSM(I)=0.0
      DEP(3*I-2)=0.0
      DEP(3*I-1)=0.0
      DEP(3*I)=0.0
94    U(I)=0.
      IF(N)65,65,66
65    N=1
      READ(5,324) PMA,DEPM,D,T,XLB,KLU,NRSTRT
      D2=D/SQRT(3.0)
      WRITE(6,320)
      WRITE(6,313) INT,SIGO,E,GNU,DEPM,D,T,XLB
      DO 95 J=1,NODE
      N3=3*J
      N2=N3-1
      N1=N3-2
      ZIG2=SIM(N1)*PMA
      ZIG3=SIM(N2)*PMA
      ZIG4=SIM(N3)*PMA
95    YCF(J)=((ZIG2-ALPH)**2)-((ZIG2-ALPH)*(ZIG3-ALPH))+((ZIG3-ALPH)**2)
    1 +(3.0*(ZIG4**2))
      R=0.0
      DO 96 I=1,NODE
```

```
 96     R=AMAX1(YCE(I),R)
        DO 97 I=1,NODE
        IF(YCE(I)-R)97,20,97
 97     CONTINUE
 20     N1=I
        U(I)=I
        IF(KLU)67,67,68
 68     CSM(I)=1.
        NNPP(1)=I
 67     B=-1.0*((((ALPH-SQ(3*N1-2))*(2.0*SIM(3*N1-2)-SIM(3*N1-1)))+((ALPH-
    1   SQ(3*N1-1))*(2.0*SIM(3*N1-1)-SIM(3*N1-2)))+(6.0*SIM(3*N1)*(-1.0*SQ
    2   (3*N1))))/((SIM(3*N1-2)**2)-(SIM(3*N1-2)*SIM(3*N1-1))+(SIM(3*N1-1)
    3   **2)+(3.0*(SIM(3*N1)**2))))
        C1=((ALPH-SQ(3*N1-2))**2+(ALPH-SQ(3*N1-1))**2-(ALPH-SQ(3*N1-1))*(A
    1   LPH-SQ(3*N1-2))+3.0*(-1.0*SQ(3*N1))**2-SIG0**2)/((SIM(3*N1-2)**2)-
    2   ((SIM(3*N1-2))*(SIM(3*N1-1)))+(SIM(3*N1-1)**2)+(3.0*SIM(3*N1)**2))
        IF(PMA)69,69,70
 70     PCRIT=((-1.*B)+SQRT(B**2-4.0*C1))/2.0
        GO TO 71
 69     PCRIT=((-1.*B)-SQRT(B**2-4.0*C1))/2.0
        GO TO 71
 66     READ(5,324) PMA,DEPM,D,T,XLB,KLU,NRSTRT
        GO TO (33,34) ,KTRANS
 34     DO 116 II=1,N3NO
116     BAPH(II)= APH(II)
        DO 117 I=1,N3NO
        SG= 0.0
        DO 118 J=1,N3NO
        NT=(J-1)/3+1
        IF(U(NT))31,118,31
 31     KQ=NEC(J)
        SG=SG+ SIJ(I,KQ)*EP(J)
118     CONTINUE
117     SQ(I)=SG
        DO 124 IK=1,NODE
        CSM(IK)=0.
        U(IK)=0.
124     NNPP(IK)=0
        MCOUNT=0
        KTRANS = 1
        K = KK
 33     N=1
        IF(PMA)72,73,72
 73     REWIND 9
        PAUSE 2
        CALL EXIT
 72     WRITE(6,312)
        WRITE(6,320)
        WRITE(6,313) INT,SIG0,E,GNU,DEPM,D,T,XLB
        DO 98 I=1,NODE
        N3=3*I
        N2=N3-1
        N1=N3-2
        SIG1=SIM(N1)*PMA+SQ(N1)
        SIG2=SIM(N2)*PMA+SQ(N2)
        SIG3=SIM(N3)*PMA+SQ(N3)
        BAPH1=APH(N1)
        BAPH2=APH(N2)
        BAPH3=APH(N3)
 98     YCE(I)=((SIG2-BAPH2)**2)-((SIG2-BAPH2)*(SIG1-BAPH1))+((SIG1-BAPH1)
```

```
     1  **2)+(3.0*((SIG3-BAPH3)**2))
        R=0.0
        DO 99 I=1,NODE
 99     R=AMAX1(YCE(I),R)
        DO 100 I=1,NODE
        IF(YCE(I)-R)100,74,100
100     CONTINUE
 74     N1=I
        U(I)=I
        IF(KLU)75,75,76
 76     CSM(I)=1.
        NNPP(1)=I
 75     N2=3*N1
        N3=N2-1
        N4=N2-2
        BAPH1=APH(N4)
        BAPH2=APH(N3)
        BAPH3=APH(N2)
        B=-1.0*((((BAPH1 -SQ(3*N1-2))*(2.0*SIM(3*N1-2)-SIM(3*N1-1)))+((BAP
     1  H2 -SQ(3*N1-1))*(2.0*SIM(3*N1-1)-SIM(3*N1-2)))+(6.0*SIM(3*N1)*(BAP
     2  H3 -SQ(3*N1))))/((SIM(3*N1-2)**2)-(SIM(3*N1-2)*SIM(3*N1-1))+(SIM(3
     3  *N1-1)**2)+(3.0*(SIM(3*N1)**2))))
        C1=(((BAPH1 -SQ(3*N1-2))**2+(BAPH2 -SQ(3*N1-1))**2)-((BAPH2 -SQ(3*
     1  N1-1))*(BAPH1 -SQ(3*N1-2)))+((3.0*(BAPH3 -SQ(3*N1))**2)-SIG0**2 ))
     2  /((SIM(3*N1-2)**2)-((SIM(3*N1-2))*(SIM(3*N1-1)))+(SIM(3*N1-1)**2)+
     3  (3.0*SIM(3*N1)**2))
        IF(PMA)77,77,78
 78     PCRIT=((-1.*B)+SQRT(B**2-4.0*C1))/2.0
        GO TO 71
 77     PCRIT=((-1.*B)-SQRT(B**2-4.0*C1))/2.0
 71     IF(PCRIT/PMA) 79,80,80
 80     IF(TEST) 79,79,81
 81     WRITE(6,308)
        WRITE(6,309)
        DO 101 J=1,NODE
        REST1=(SQ(3*J-2)-GNU*SQ(3*J-1))/E+EP(3*J-2)
        REST2=(SQ(3*J-1)-GNU*SQ(3*J-2))/E+EP(3*J-1)
        REST3=2.0*(1.0+GNU)*SQ(3*J)/E+EP(3*J)
        WRITE(6,310) J,SQ(3*J-1),SQ(3*J-1),SQ(3*J),REST1,REST2,REST3
101     CONTINUE
 79     IF(XLB)41,73,41
 41     TEST=1.0
        PM=PCRIT
        DO 102 J=1,N3NO
        SUM(J)=SIM(J)*PM+SQ(J)
        SQ(J)=0.0
102     SUM1(J)=SUM(J)
        Z=SIG0**2
        G=E/3.0
 83     PM=PM+DEPM
        IF(ABS(PM/DEPM) - 1.0)85,85,86
 85     K=KK
 86     NCOUNT=0
        IA=0
        IF(KLU)87,87,22
 87     DO 103 I=1,NODE
        IF(NNPP(I))22,103,22
103     CONTINUE
C
C       CALC. NODAL STRESSES FROM INPUT MATRICES SIM AND SIG
```

```
C
      J=1
      DO 104 I=1,NODE
      IF(U(I)) 23,104,23
 23   J7=3*I-2
      J8=J7+1
      J9=J7+2
      L(J)=J7
      L(J+1)=J8
      L(J+2)=J9
      J=J+3
104   CONTINUE
      L(J)=999
      DO 105 J=1,N3NO
      SG=0.0
      DO 106 I=1,MCOL
      II=L(I)
      IF(L(I)-999)200,105,200
200   IF(NEC(II))201,202,201
202   IF(IAVAIL-MCOL)203,203,204
203   III=II-1-IREC
      IF(III)205,206,207
207   DO 120 JJ=1,III
120   READ (9)
      GO TO 206
205   III=(-1)*III
      DO 121 JJ=1,III
121   BACKSPACE 9
206   READ (9) (SIJ(JJ,IAVAIL),JJ=1,N3NO)
      IREC=II
      NEC(II)=IAVAIL
      IAVAIL=IAVAIL+1
      GO TO 200
201   NN=NEC(II)
106   SG=SG+SIJ(J,NN)*DEP(II)
105   SQ(J)=SG
      STEST=10**10
      DO 107 LX=1,N3NO
      DSUM(LX)=SQ(LX)+SIM(LX)*DEPM
      SUM1(LX)=SUM1(LX)+DSUM(LX)
      IF(SUM1(LX)-STEST)107,107,24
 24   WRITE(6,317)
      CALL EXIT
107   CONTINUE
      GO TO 199
204   WRITE(6,311)
      CALL EXIT
 22   DO 108 IQ=1,NODE
      IF(U(IQ))25,108,25
 25   NCOUNT=NCOUNT+1
108   CONTINUE
      IF(NCOUNT-MCOUNT)208,209,208
208   MCOUNT=NCOUNT
210   J=1
      DO 109 I=1,NODE
      IF(U(I))26,109,26
 26   J7=3*I-2
      J8=3*I-1
      J9=3*I
      L(J)=J7
```

```
      L(J+1)=J8
      L(J+2)=J9
      J=J+3
109   CONTINUE
      L(J)=999
      DO 110 I=1,MCOL
      II=L(I)
      IF(L(I)-999) 211,209,211
211   IF(NEC(II))213,214,213
214   IF(IAVAIL-MCOL)215,215,204
215   III=II-1-IREC
      IF(III)216,217,218
218   DO 122 JJ=1,III
122   READ (9)
      GO TO 217
216   III=(-1)*III
      DO 123 JJ=1,III
123   BACKSPACE 9
217   READ (9) (SIJ(JJ,IAVAIL),JJ=1,N3NO)
      IREC=II
      NEC(II)=IAVAIL
      IAVAIL=IAVAIL+1
      GO TO 211
213   NN=NEC(II)
110   CONTINUE
209   CALL PPR
      STEST=10**10
      DO 111 JX=1,N3NO
      IF(SUM1(JX)-STEST)111,111,27
 27   WRITE(6,317)
      CALL EXIT
111   CONTINUE
C
C     COMPUTE YIELD CONDITION EQUATION
C
199   DO 112 I=1,NODE
      IF(CSM(I)-1.)28,112,28
 28   N3=3*I
      N2=N3-1
      N1=N3-2
      YCE(I)=((SUM1(N1)-APH(N1))**2)-((SUM1(N1)-APH(N1))*(SUM1(N2)-APH(N
     1 2)))+((SUM1(N2)-APH(N2))**2)+3.0*((SUM1(N3)-APH(N3))**2)
C
C     YIELD CONDITION EQUATION TEST
C
      IF(YCE(I)-Z)112,220,220
220   U(I)=I
      IF(CSM(I)-1.0)221,112,221
221   IF(KLU)222,222,223
223   CSM(I)=1.
      GO TO 112
222   S1=T*((ABS(SUM1(3*I-2)-BAPH(3*I-2))/D)**(T-1.0))/D
      S2=T*((ABS(SUM1(3*I-1)-BAPH(3*I-1))/D)**(T-1.0))/D
      S3=T*((ABS(SUM1(3*I)-BAPH(3*I))/D2)**(T-1.0))/D2
      QZIG=SQRT(((SUM1(N2)-APH(N2)  )**2)+((SUM1(N1)-APH(N1)  )**2)+((SUM1
     1 (N3)-APH(N3))**2))
      V1=(SUM1(N1)-APH(N1)       )/QZIG
      V2=(SUM1(N2)-APH(N2)       )/QZIG
      V3=(SUM1(N3)-APH(N3))/QZIG
      C=((V1**2)*S1)+((V2**2)*S2)+((V3**2)*S3)
```

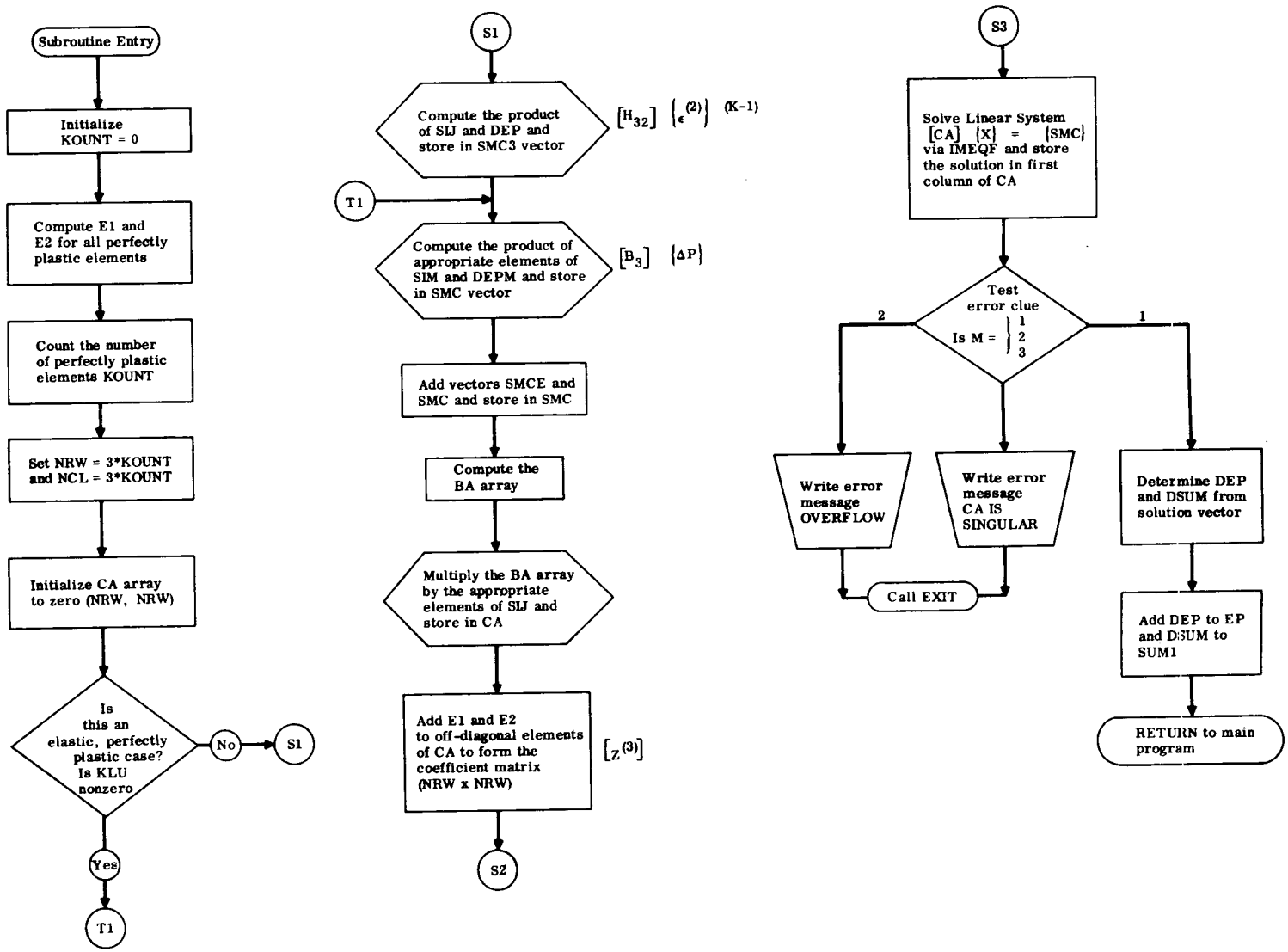
```
      CSM(I)=C
      DSUM(N1)=SUM1(N1)-SUM(N1)
      DSUM(N2)=SUM1(N2)-SUM(N2)
      DSUM(N3)=SUM1(N3)-SUM(N3)
112   CONTINUE
      DO 113 I=1,NODE
      NN=3*I
      NNN=NN-1
      NNK=NN-2
      IF(CSM(I)-1.0)224,225,224
224   IF(YCF(I)-Z)226,227,227
226   DEP(NNK)=0.0
      DEP(NNN)=0.0
      DEP(NN)=0.0
      GO TO 113
227   F=(((SUM(NNK)-APH(NNK))*(DSUM(NNK)-(DSUM(NNN))/2.0))+((SUM(NNN)-AP
     1 H(NNN))*(DSUM(NNN)-(DSUM(NNK))/2.0))+(3.0*(SUM(NN)-APH(NN))*(DSUM(
     2 NN))))
      DLAM(I)=(2.0*CSM(I))*F/((5.0*(SUM(NNK)-APH(NNK))**2)-(8.0*(SUM(NNK
     1 )-APH(NNK))*(SUM(NNN)-APH(NNN) ))+(5.0*(SUM(NNN)-APH(NNN))**2)+(36
     2 .0*(SUM(NN)-APH(NN))**2))
      DMEW(I)=F/(((SUM(NNK)-APH(NNK))**2)-((SUM(NNK)-APH(NNK))*(SUM(NNN)
     1 -APH(NNN)))+((SUM(NNN)-APH(NNN))**2)+(3.0*(SUM(NN)-APH(NN))**2))
      IF(DLAM(I))226,226,228
C
C     COMPUTE THE DELTA EPSILONS
C
228   DEP(NNK)=2.0*((SUM(NNK)-APH(NNK))-(SUM(NNN)-APH(NNN))/2.0)*DLAM(I)
      DEP(NNN)=2.0*((SUM(NNN)-APH(NNN))-(SUM(NNK)-APH(NNK))/2.0)*DLAM(I)
      DEP(NN)=6.0*((SUM(NN)-APH(NN)))*DLAM(I)
C
C     COMPUTE THE DELTA ALPHAS
C
      DAPH1=(SUM(NNK)-APH(NNK))*DMEW(I)
      DAPH2=(SUM(NNN)-APH(NNN))*DMEW(I)
      DAPH3=(SUM(NN)-APH(NN))*DMEW(I)
C
C     CALCULATE ALPHA SUMS
C
      APH(NNK)=APH(NNK)+DAPH1
      APH(NNN)=APH(NNN)+DAPH2
      APH(NN)=APH(NN)+DAPH3
C     CALCULATE EPSILON SUMS
C
      EP(NNK)=EP(NNK)+DEP(NNK)
      EP(NNN)=EP(NNN)+DEP(NNN)
      EP(NN)=EP(NN)+DEP(NN)
225   SUM(NNK)=SUM1(NNK)
      SUM(NNN)=SUM1(NNN)
      SUM(NN)=SUM1(NN)
113   CONTINUE
      DO 114 J=1,NODE
      N3=3*J
      N2=N3-1
      N1=N3-2
      IF(U(I))114,29,114
 29   SUM(N1)=SUM1(N1)
      SUM(N2)=SUM1(N2)
      SUM(N3)=SUM1(N3)
114   CONTINUE
```

```
C
C          THE TEST FOR PERFECT PLASTICITY
C
           IF ( ZFL ) 35,36,36
 36        DO 115 I=1,NODE
           IF(U(I))115,115,229
229        IF(KLU)230,230,231
230        FLOW=(APH(3*I-2)-BAPH(3*I-2))**2-(APH(3*I-2)-BAPH(3*I-2))*(APH(3*I
     1     -1)-BAPH(3*I-1))+(APH(3*I-1)-BAPH(3*I-1))**2+(3.0*(APH(3*I)-BAPH(3
     2     *I))**2)
           IF(ZFL-FLOW)231,231,115
231        CSM(I)=1.
           IA=IA+1
           NNPP(IA)=I
115        CONTINUE
 35        IF( PM/XLB - 1.0 ) 232,233,233
232        IF( K-KK ) 234,235,234
234        K=K+1
           GO TO 83
233        KTRANS=2
235        WRITE(6,325) PM, S
           WRITE(6,301)
           WRITE(6,302)
           WRITE(6,300) (J,SUM1(3*J-2),SUM1(3*J-1),SUM1(3*J),J=1,NODE)
           DO 119 I=1,NODE
           A=I
           IF(U(I)-A)119,32,119
 32        WRITE(6,314)
           WRITE(6,315) I,CSM(I),DLAM(I),DMEW(I)
           WRITE(6,306)
           N3=3*I
           N2=N3-1
           N1=N3-2
           DAPH1=(SUM(N1)-APH(N1))*DMEW(I)
           DAPH2=(SUM(N2)-APH(N2))*DMEW(I)
           DAPH3=(SUM(N3)-APH(N3))*DMEW(I)
           WRITE (6,305) DAPH1,DAPH2,DAPH3,DEP(N1),DEP(N2),DEP(N3)
           WRITE(6,307)
           WRITE(6,305) APH(N1),APH(N2),APH(N3),EP(N1),EP(N2),EP(N3)
           WRITE(6,316)
           EEP1=((SUM1(N1)-(SUM1(N2))/3.0)/E)
           EEP2=((SUM1(N2)-(SUM1(N1))/3.0)/E)
           EEP3=3.0*(SUM1(N3))/E
           STR1=EP(N1)+EEP1
           STR2=EP(N2)+EEP2
           STR3=EP(N3)+EEP3
           WRITE (6,305)  STR1,STR2,STR3,EEP1,EEP2,EEP3
119        CONTINUE
           K=K+1
           KK=KK+INT
           GO TO (83,42) , KTRANS
 42        IF(NRSTRT) 66,66,44
 44        WRITE(10) APH,BA,BAPH,CA,CAPH,CSM,DEP,DLAM,DMEW,DSUM,E1,E2,E9,EP,
     1               L,NEC,NNPP,SIM,SIJ,SMC,SMC3,SQ,SUM,SUM1,X,  U,D,F,GNU,
     2               IAVAIL,INT,IREC,KLU,MCOL,NCOMP,NODE,PM,PCRIT,SIG0,T,Z,
     3             ZFL,DEPM,S,TEST,G,MCOUNT,YCF
           REWIND 10
           REWIND  9
           PAUSE 2
           CALL EXIT
```

```
300     FORMAT(1H ,6X,I3,2X,E16.8,3X,E16.8,3X,F16.8)
301     FORMAT(/ 21X,17H ELEMENT STRESSES, / )
302     FORMAT(12H0ELEMENT NO.,7X,7HSIGMA-X,14X,7HSIGMA-Y,12X,7HTAU-X,Y)
305     FORMAT(8X,E16.8,3X,E16.8,3X,E16.8,3X,E16.8,3X,F16.8,3X,E16.8)
306     FORMAT(1H0,9X,13HDELTA ALPHA-X,6X,13HDELTA ALPHA-Y,6X,12HDELTA GAM
    1   -XY,7X,10HDELTA EP-X,9X,10HDELTA EP-Y,9X,12HDELTA GAM-XY)
307     FORMAT(1H0,9X,11HTOT ALPHA-X,8X,11HTOT ALPHA-Y,8X,12HTOT ALPHA-XY,
    1   7X,12HPLASTIC EP-X,7X,12HPLASTIC EP-Y,7X,14HPLASTIC GAM-XY)
308     FORMAT(20X,17HRESIDUAL STRESSES,41X,16HRESIDUAL STRAINS)
309     FORMAT(8H0ELEMENT,6X,1HX,18X,1HY,18X,2HXY,23X,1HX,18X,1HY,18X,2HXY
    1   )
310     FORMAT(1H0,I2,3E19.8,3X,3E19.8)
311     FORMAT(19H1ALL OUT OF COLUMNS)
312     FORMAT(1H0,22HUNLOADING CYCLE BEGINS)
313     FORMAT(1H0(I10,8E15.6))
314     FORMAT(13H0 ELEMENT NO.,9X,1HC,19X,7HDLAM    ,16X,3HDMU)
315     FORMAT(5X,I2,8X,E16.8,5X,E16.8,5X,E16.8)
316     FORMAT(1H0,10X,10HSTRAIN - X,9X,10HSTRAIN - Y,9X,11HSTRAIN - XY,7X
    1   ,12HELASTIC EP-X,7X,12HELASTIC EP-Y,7X,14HELASTIC GAM-XY)
317     FORMAT(47H0 ELEMENT STRESSES ARE TOO BIG. DISCONTINUE JOB)
319     FORMAT(I5,E15.7)
320     FORMAT(6H1INPUT,/ 7X,3HINT,11X,4HSIGO,14X,1HE,13X,2HNU,11X,4HDELP,
    1   14X,1HD,14X,1HT,10X,5HRMXLD)
321     FORMAT(1H0(I10,8E15.6))
322     FORMAT(5I10)
323     FORMAT(5F15.7)
324     FORMAT(5E15.7,I2,I3)
325     FORMAT(17H0 LOAD LEVEL ,P =,F10.0,5X,12HDETERMINANT=,E16.8)
        END
```

```
        SUBROUTINE SETAPE(NUNIT,NFILE)
        DATA SIGNAL /O111111111111 /
        REWIND 9
        NSET=NFILE-1
        IF(NSET)3,3,4
    4   DO 1 I=1,NSET
    2   READ (9) X
        IF(X-SIGNAL)2, 1,2
    1   CONTINUE
    3   RETURN
        END
```

STREPP PROGRAM
Subroutine PPR



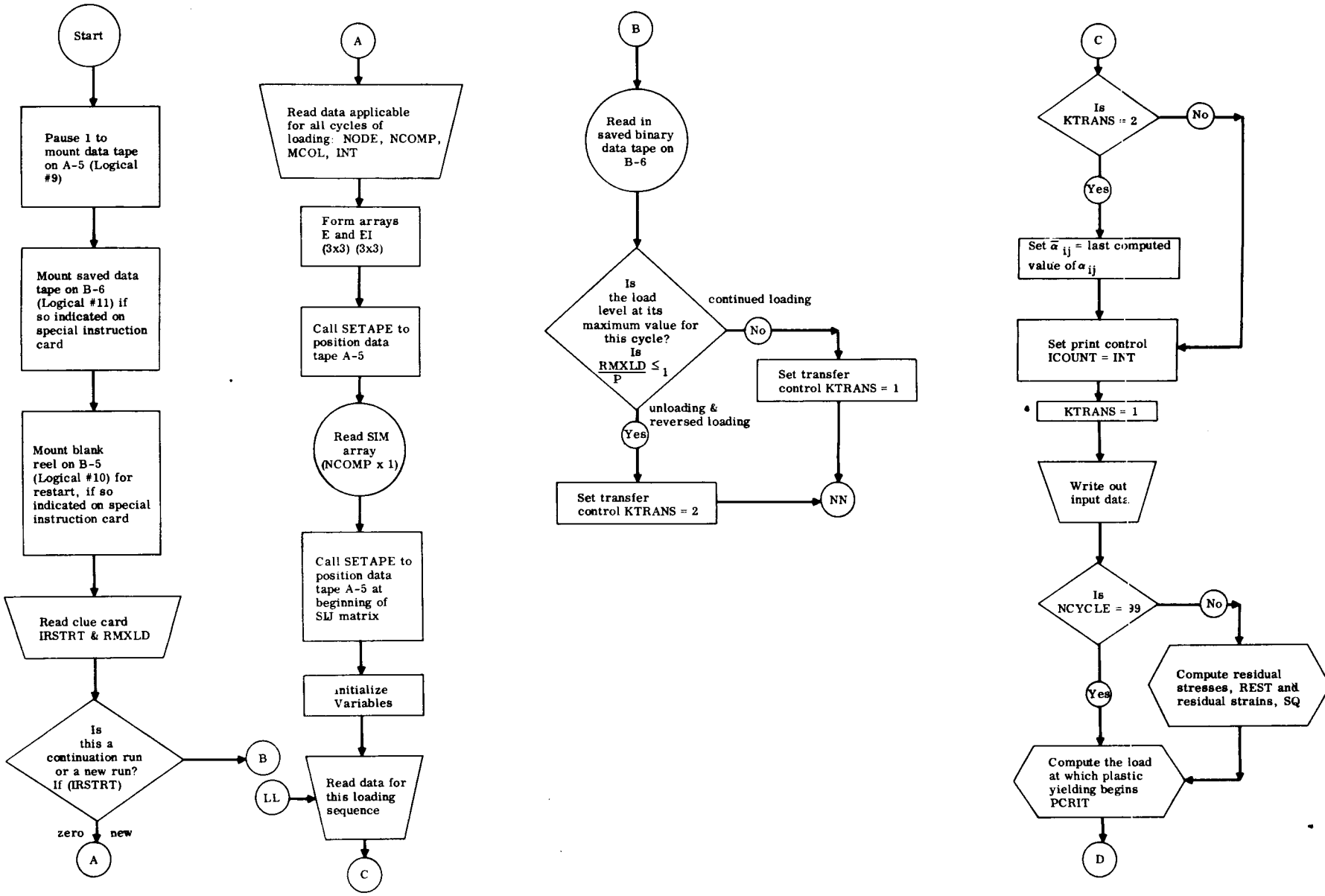
SUBROUTINE PPR

```
$ IBFT C     PPR      LIST,REF
             SUBROUTINE PPR
             DIMENSION APH(165)
             DIMENSION BA(60)
             DIMENSION BAPH(165)
             DIMENSION CA(60,60)
             DIMENSION CAPH(165)
             DIMENSION CSM(55)
             DIMENSION DEP(165)
             DIMENSION DLAM(55)
             DIMENSION DMEW(55)
             DIMENSION DSUM(165)
             DIMENSION E1(55)
             DIMENSION E2(55)
             DIMENSION E9(60)
             DIMENSION EP(165)
             DIMENSION L(165)
             DIMENSION  NEC(165)
             DIMENSION NNPP(55)
             DIMENSION SIM(165)
             DIMENSION SIJ(165,60)
             DIMENSION SMC(60)
             DIMENSION SMC3(60)
             DIMENSION SQ(165)
             DIMENSION SUM(165)
             DIMENSION SUM1(165)
             DIMENSION X(60)
             DIMENSION YCE(55)
             DIMENSION U(55)
             COMMON  CA
             COMMON  N3NO    , NODE    , MCOL    , CSM     , APH     , DEPM
             COMMON  SIM     , SIJ     , DEP     , SUM1    , NNPP    , L
             COMMON  NEC     , U       , EP      , SQ      , DSUM    , BAPH
             COMMON  YCE     , DLAM    , DMEW    , SUM     , E1      , E2
             COMMON SMC, SMC3, E9, INT, XLB
             COMMON  NCOUNT , CAPH    , BA      , S       , NP      , KLU
             EQUIVALENCE (CA,X)
             KOUNT=0
             DO 605 J=1,NODE
             N3=3*J
             N2=N3-1
             N1=N3-2
             IF(CSM(J)-1.0)605,606,605
   606       E1(J)=((SUM1(N2)-APH(N2))-(SUM1(N1)-APH(N1))/2.0)/((SUM1(N1)-APH(N
       1     1))-(SUM1(N2)-APH(N2))/2.0)
             E2(J)=(3.0*(SUM1(N3)-APH(N3)))/((SUM1(N1)-APH(N1))-(SUM1(N2)-APH(N
       1     2))/2.0)
             KOUNT=KOUNT+1
   605       CONTINUE
             NRW=3*KOUNT
             NCL=3*KOUNT
             DO 603 I=1,NRW
             DO 603 J=1,NRW
   603       CA(I,J)=0.0
             IF(KLU)110,111,110
   111       NY=1
             DO 104 I=1,N3NO
             J=(I-1)/3+1
             IF(CSM(J))105,104,105
   105       IF(CSM(J)-1.0)104,106,104
```

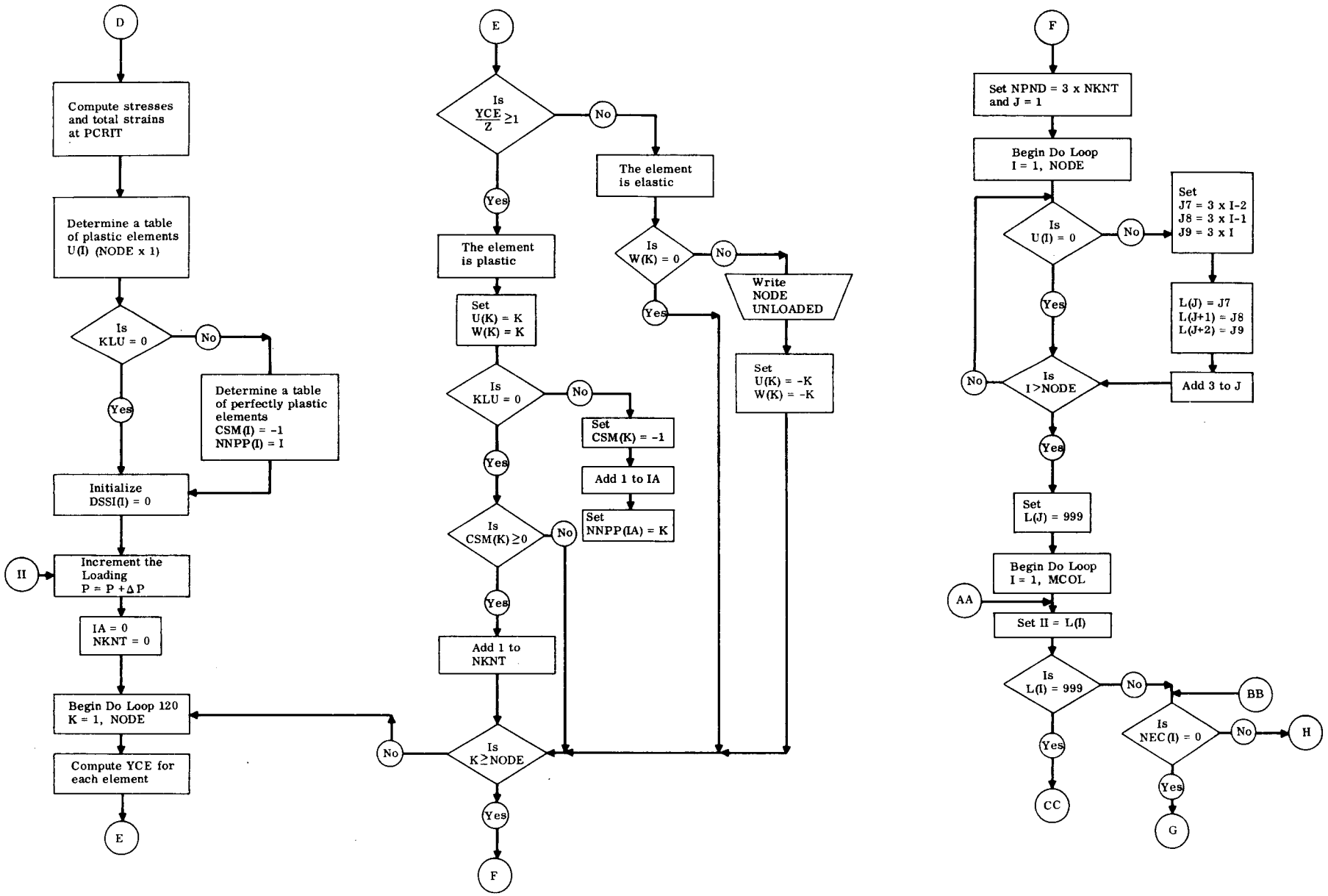
```
106     SMC2=0.0
        DO 107 LY=1,N3NO
        J=(LY-1)/3+1
        IF(CSM(J))108,107,108
108     IF(CSM(J)-1.0)109,107,109
109     KW=NEC(LY)
        SMC2=SMC2+SIJ( I,KW)*DEP(LY)
107     CONTINUE
        SMC3(NY)=SMC2
        NY=NY+1
104     CONTINUE
C
C       FORMATION OF THE SIM COLUMN MATRIX
C
110     MK=1
        DO 700 J=1,NODE
        IF(CSM(J)-1.0)700,701,700
701     NAN=3*J-2
        SMC(MK)=SIM(NAN)*DEPM
        SMC(MK+1)=SIM(NAN+1)*DEPM
        SMC(MK+2)=SIM(NAN+2)*DEPM
        MK=MK+3
700     CONTINUE
C
C       ADDITION OF THE SMC1 AND THE SMC3 MATRICES
C
        DO 773 K1=1,NRW
773     SMC(K1)=SMC(K1)+SMC3(K1)
        NQ=1
        MZ=1
        DO 630 L2=1,NRW,3
        M22=NNPP(MZ)
        BA(L2)=1.0
        BA(L2+1)=E1(M22)
        BA(L2+2)=E2(M22)
        NX=1
        XIDNT=-7.0
        DO 101 I=1,N3NO
        IX=(I-1)/3+1
        IF(CSM(IX)-1.0)101,102,101
102     NZ=1
        SMC4=0.0
        DO 201 LY=1,N3NO
        J=(LY-1)/3+1
        IF(CSM(J)-1.0)201,202,201
202     KZ=NEC(LY)
        SMC4=SMC4+SIJ(I,KZ)*BA(NZ)
        NZ=NZ+1
201     CONTINUE
        CA(NX,NQ)=-1.0*SMC4*(10.0**XIDNT)
        NX=NX+1
101     CONTINUE
        NQ=NQ+3
        DO 779 J3=1,NRW
779     BA(J3)=0.0
630     MZ=MZ+1
        MB=1
        DO 631 KRW=1,NRW,3
        M23=NNPP(MB)
        CA(KRW+1,KRW+1)=1.0
```

```
      CA(KRW+2,KRW+2)=1.0
      CA(KRW,KRW+1)=-1.0*F1(M23)
      CA(KRW,KRW+2)=-1.0*F2(M23)
  631 MB=MB+1
C
C     SOLUTION OF THE SIMULTANEOUS LINEAR EQUATIONS
C
      S=1.0
      M=IMEQF(NP,NRW,1,CA,SMC,S,E9)
      GO TO (20,13,15), M
   13 WRITE (6,18)
      CALL EXIT
   18 FORMAT(9H OVERFLOW)
   15 WRITE (6,19)
      CALL EXIT
   19 FORMAT(15H CA IS SINGULAR)
   20 DO 250 K6=1,NRW,3
  250 X(K6)=X(K6)*(10.0**XIDNT)
      DO 777 J3=1,KOUNT
      IC=NNPP(J3)
      DEP(3*IC-2)=X(3*J3-2)
      DEP(3*IC-1)=E1(IC)*DEP(3*IC-2)
      DEP(3*IC)=E2(IC)*DEP(3*IC-2)
      EP(3*IC-2)=EP(3*IC-2)+DEP(3*IC-2)
      EP(3*IC-1)=EP(3*IC-1)+DEP(3*IC-1)
  777 EP(3*IC)=EP(3*IC)+DEP(3*IC)
      DO 73 I=1,N3NO
      SG=0.0
      DO 74 J=1,N3NO
      N=(J-1)/3+1
      IF (CSM(N)) 77,74,77
   77 KQ=NEC(J)
      SG=SG+SIJ(I,KQ)*DEP(J)
   74 CONTINUE
   73 SQ(I)=SG
      DO75 LX=1,N3NO
      DSUM(LX)=SQ(LX)+SIM(LX)*DEPM
      SUM1(LX)=SUM1(LX)+DSUM(LX)
   75 CONTINUE
      RETURN
       END
```

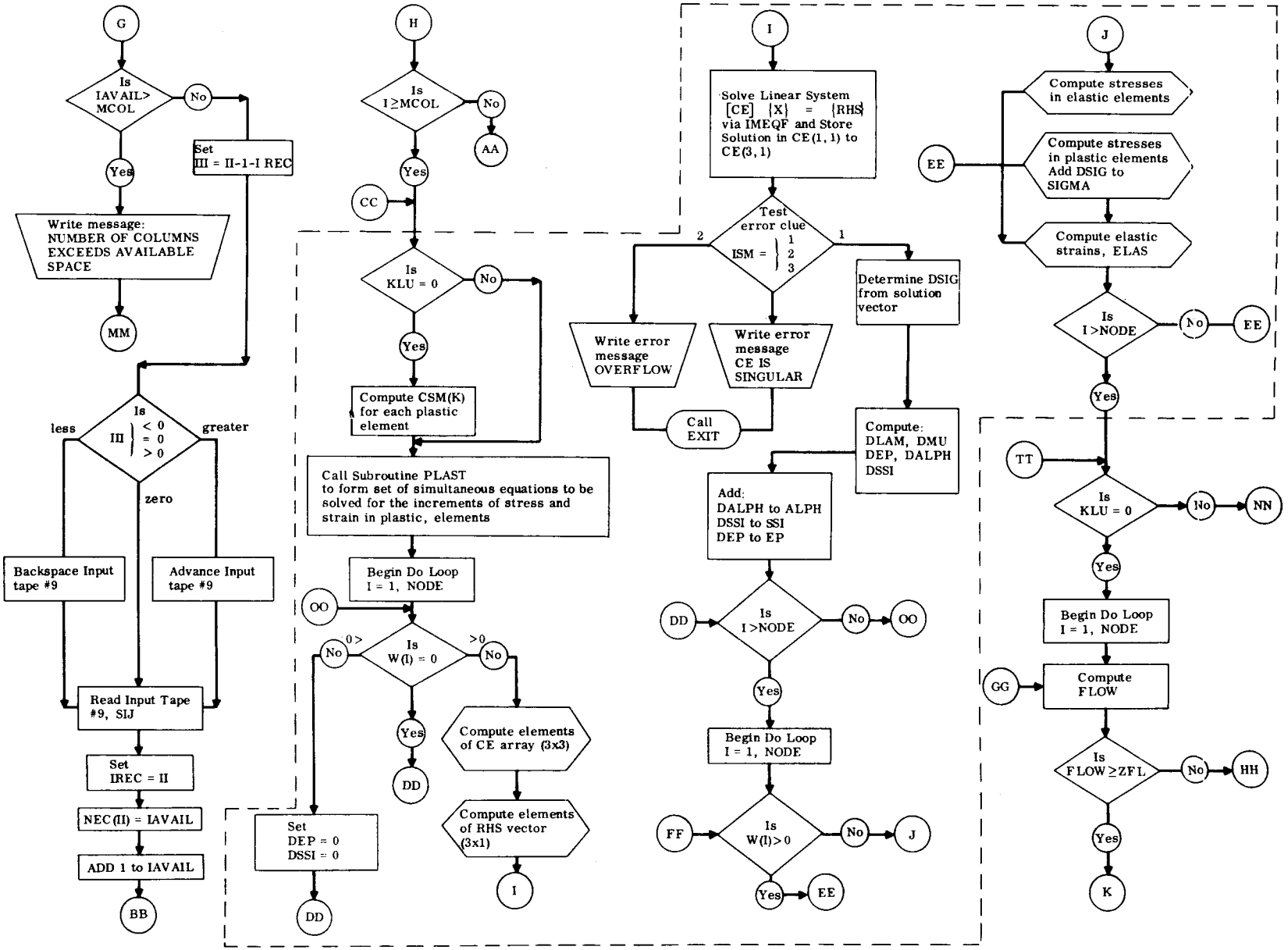
STRAIN PROGRAM



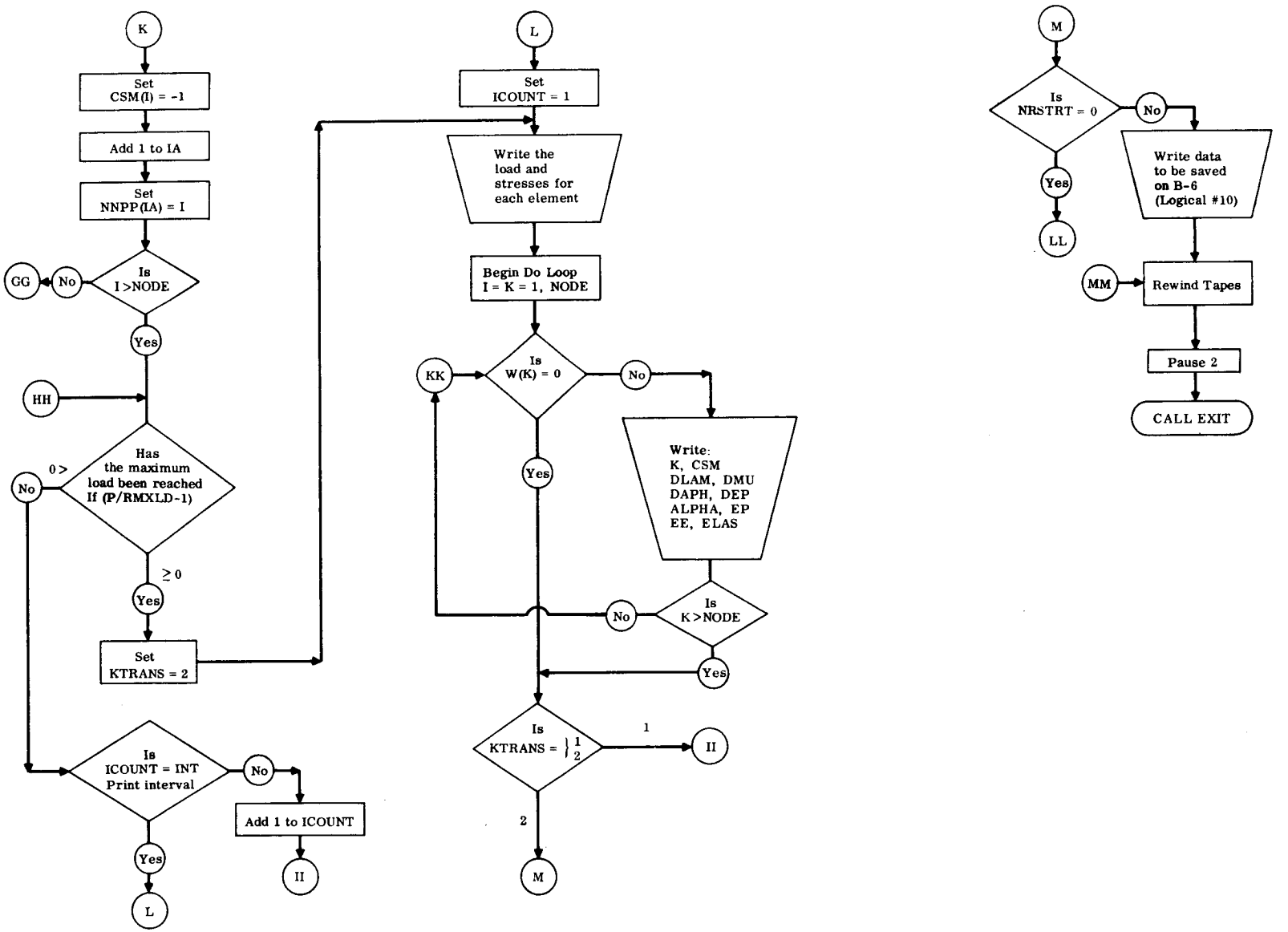
STRAIN PROGRAM



STRAIN PROGRAM



STRAIN PROGRAM



```
$ IBFT C     STRAIN  LIST,REF
          PAUSE 1
          REWIND 9
          INTEGER U,W
          COMMON ALPHA(165)
          COMMON CSM(55)
          COMMON DE(165)
          COMMON DELP
          COMMON DEP(165)
          COMMON DSIG(165)
          COMMON DSSI(165)
          COMMON E(3,3)
          COMMON E1
          COMMON EI(3,3)
          COMMON GNU
          COMMON KLU
          COMMON MCOL
          COMMON NCOMP
          COMMON NEC(165)
          COMMON NNPP(57)
          COMMON NODE
          COMMON NPND
          COMMON SIGMA(165)
          COMMON SIJ(165,57)
          COMMON SIM(165)
          COMMON U(55)
          COMMON W(55)
          DIMENSION ALBAR(165)
          DIMENSION ARG7(3)
          DIMENSION CE(3,3)
          DIMENSION DALPH(165)
          DIMENSION DLAM(55)
          DIMENSION DMU(55)
          DIMENSION EE(165)
          DIMENSION ELAS(165)
          DIMENSION EP(165)
          DIMENSION GIM(165)
          DIMENSION L(99)
          DIMENSION REST(165)
          DIMENSION RHS(3)
          DIMENSION SQ(165)
          DIMENSION SSI(165)
          CONST=1.0E+06
          READ(5,3) IRSTRT,RMXLD
    3     FORMAT(I5,E15.7)
          IF(IRSTRT) 60,60,61
   61     CONTINUE
          READ(11) ALBAR,ALPHA,CSM,D,D1,DALPH,DE,DEP,DELP,DLAM,DMU,DSIG,DSSI
     *             ,E,E1,EE,EI,ELAS,EP,G,GIM,GNU,GNU1,IAVAIL,INT,IREC,KLU,L,
     *             MCOL,NCOMP,NEC,NNPP,NODE,NPND,P,PCRIT,REST,SIGO,SIGMA,SIJ
     *             ,SIM,SMALL,SQ,SSI,T,U,W,Z,ZFL
          IREC = 0
          ICOUNT = INT
          NCYCLE = 99
          WRITE(6,2041)
          WRITE(6,2022) INT,SIGO,E1,GNU,DELP,D,T,RMXLD
          IF(RMXLD/P - 1.0)62,62,63
   62     KTRANS = 2
          GO TO 132
   63     KTRANS = 1
```

```
      GO TO 132
60    CONTINUE
      KTRANS = 1
      READ(5,1)NODE,NCOMP,MCOL,INT
      ICOUNT = INT
1     FORMAT(5I10)
      READ(5,2) SIG0,E1,GNU,ZFL
2     FORMAT(5E15.7,I2,I3)
      GNU1 = E1/(1.0-GNU**2)
      F(1,1)= GNU1
      F(1,2)= GNU*F(1,1)
      F(1,3)= 0.0
      F(2,1) = E(1,2)
      F(2,2) = E(1,1)
      E(2,3)= 0.0
      E(3,1)= 0.0
      E(3,2)= 0.0
      E(3,3) = E1/(2.0*(1.0+GNU))
      EI(1,1)=1.0/E1
      EI(1,2)=-GNU/E1
      EI(1,3)=0.0
      EI(2,1)=EI(1,2)
      EI(2,2)=EI(1,1)
      EI(2,3)=0.0
      EI(3,1)=0.0
      EI(3,2)=0.0
      EI(3,3)=2.0*(1.0+GNU)/E1
C
C     C O   MPUTE GIM MATRIX
C
      CALL SETAPE(9,2)
      READ(9) SIM
      DO 15 I2=1,NCOMP,3
      DO 15 I =1,3
      IN = I2 +I-1
      GIM(IN)=0.0
      DO 15 J=1,3
      IN2 = I2+J-1
15    GIM(IN)= GIM(IN)+ E(I,J)*SIM(IN2)
C
C     IN I   TIALIZE
C
      CALL SETAPE(9,1)
      IAVAIL=1
      IREC = 0
      DO 4 I=1,MCOL
4     L(I)=0
      DO 200 K=1,NCOMP
5     NEC(K)=0
      DSSI(K)=0.0
      ALBAR(K) = 0.0
      FE(K)=0.0
      SIGMA(K)=0.0
      ALPHA(K)=0.0
      SSI(K)=0.0
      EP(K)=0.0
200   CONTINUE
1110  DO 6 J=1,NODE
      CSM(J)=0.0
      W(J) = 0
```

```
    6       U(J) = 0
C
C   RE A    D INPUT FOR NEXT  CYCLE
C
    7       READ(5,2) PMA,DELP,D,T,RMXLD,KLU,NRSTRT
            GO TO (777,787),KTRANS
  787       DO 127 I=1,NCOMP
  127       ALBAR(I) = ALPHA(I)
  777       CONTINUE
            ICOUNT=INT
            KTRANS = 1
            WRITE(6,2041)
 2041       FORMAT(6H1INPUT,/ 7X,3HINT,11X,4HSIG0,14X,1HE,13X,2HNU,11X,4HDELP,
     *            14X,1HD,14X,1HT,10X,5HRMXLD)
            WRITE(6,2022)INT,SIG0,E1,GNU,DELP,D,T,RMXLD
 2022       FORMAT(1H0(I10,8E15.6) )
            IF(NCYCLE-99) 8,11,8
    8       NCYCLE = 99
            YCEMAX = 0.0
            DO 12 I=1,NODE
            Z1= GIM(3*I-2) * PMA
            Z2 = GIM(3*I-1) * PMA
            Z3 = GIM(3*I) * PMA
            YCE = Z1**2 - Z1*Z2 + Z2**2 + 3.0*Z3**2
            IF(YCE-YCEMAX)12,13,13
   13       N1 = I
            YCEMAX = YCE
   12       CONTINUE
            U(N1)=N1
            W(N1) = N1
            IF(KLU)51,50,51
   51       CSM(N1)=-1.0
            NNPP(1) = N1
   50       CONTINUE
            PCRIT=SQRT((SIG0**2)/((GIM(3*N1-2)**2)-(GIM(3*N1-2)*GIM(3*N1-1)) +
     *            (GIM(3*N1-1)**2)+(3.0*(GIM(3*N1)**2))))
            P=PCRIT
            D1 = D/1.732051
            Z = SIG0**2
            SMALL=CONST/Z
            G =E1/3.0
            DO 16 K=1,NCOMP
            EE(K) = SIM(K) * PCRIT
   16       SIGMA(K) = GIM(K) * PCRIT
            GO TO 41
C
C   ST A    RTING PROCEDURE FOR ALL CYCLES BUT  THE FIRST
C
   11       CONTINUE
            DO 30 I=1,NCOMP
            SG=0.0
            DO 31 J=1,NCOMP
            N = (J-1)/3 + 1
            IF(U(N))32,31,32
   32       KQ = NEC(J)
            SG = SG + SIJ(I,KQ) * SSI(J)
   31       CONTINUE
   30       REST(I) = SG
            WRITE(6,2034)
 2034       FORMAT(1H1,7HNODE NO,21X,15HRESIDUAL STRESS,42X,15HRESIDUAL STRAIN
```

```
     *    , //)
          DO 33 I=1,NODE
          NN=3*I
          NNN=3*I-2
          NNK=3*I-1
          SQ(NNN)= GNU1*(REST(NNN)+GNU*REST(NNK))+SSI(NNN)
          SQ(NNK)= GNU1*(REST(NNK)+GNU*REST(NNN))+SSI(NNK)
          SQ(NN) = E1/(2.0*(1.0+GNU))*(REST(NN))+ SSI(NN)
          WRITE(6,2032)(I,SQ(NNN),SQ(NNK),SQ(NN),REST(NNN),REST(NNK),
     *                  REST(NN))
 2032     FORMAT(3X,I3,2X,6E19.8/)
   33     CONTINUE
          IF(PMA)10,9,10
   10     YCEMAX = 0.0
          DO 34 I=1,NODE
          NN=3*I
          NNN=3*I-2
          NNK=3*I-1
          Z1=GIM(NNN)*PMA +SQ(NNN)
          Z2=GIM(NNK)*PMA +SQ(NNK)
          Z3=GIM(NN) *PMA +SQ(NN)
          YCE = ( Z1-ALBAR(NNN))**2 - (Z1-ALBAR(NNN))*(Z2-ALBAR(NNK)) + (Z2-
     *          ALBAR(NNK))**2 + 3.0*(Z3-ALBAR(NN))**2
          IF(YCE-YCEMAX)34,36,36
   36     N1 = I
          YCEMAX = YCE
   34     CONTINUE
          NNN= 3*N1-2
          NNK= 3*N1-1
          NN = 3*N1
          DNOM=GIM(NNN)**2-GIM(NNN)*GIM(NNK)+GIM(NNK)**2+3.0*(GIM(NN)**2).
          B=-((ALBAR(NNN)-SQ(NNN))*(2.0*GIM(NNN)-GIM(NNK))+(ALBAR(NNK)-SQ(NN
     *    K))*(2.0*GIM(NNK)-GIM(NNN))+(6.0*GIM(NN)*(ALBAR(NN)-SQ(NN))))/DNOM
          C=((ALBAR(NNN)-SQ(NNN))**2 -((ALBAR(NNN)-SQ(NNN))*(ALBAR(NNK)-SQ
     *      (NNK)))+((ALBAR(NNK)-SQ(NNK))**2) +(3.0*(ALBAR(NN)-SQ(NN))**2)
     *      -Z)/DNOM
          IF(PMA)37,37,39
   37     PCRIT=(-B-SQRT(B**2-4.0*C))/2.0
          GO TO 40
   39     PCRIT=(-B+SQRT(B**2-4.0*C))/2.0
   40     P=PCRIT
          DO 265 I=1,NCOMP
          SIGMA(I) = GIM(I)*P + SQ(I)
          EE(I) = SIM(I)*P + REST(I)
  265     CONTINUE
          DO 250 K=1,NODE
          W(K)= 0
  250     CSM(K)=0.0
          U(N1)=N1
          W(N1) = N1
          IF(KLU)53,52,53
   53     CSM(N1) = -1.0
          NNPP(1) = N1
   52     CONTINUE
          DO 270 K=1,NCOMP
  270     DSSI(K) = 0.0
   41     P=P+DELP
          IA=0
          NKNT=0
          DO 120 K=1,NODE
```

```
150       NNN=3*K-2
          NNK  = 3*K-1
          NN   = 3*K
          YCE  = (SIGMA( NNN )-ALPHA(NNN  ))**2 -((SIGMA(NNN  )-ALPHA(NNN  ))
     *           *(SIGMA(NNK ) -ALPHA(NNK  )]) + 3.0*(SIGMA(NN )-ALPHA(NN ))
     *           **2 +(SIGMA(NNK)-ALPHA(NNK))**2
          Z9 = (YCE   /Z-1.0)
          IF(Z9+SMALL)121,122,122
122       U(K)=K
          W(K) = K
          IF(KLU)54,55,54
 55       IF(CSM(K))120,128,128
128       NKNT = NKNT + 1
          GO TO 120
 54       CSM(K) = -1.0
          IA = IA+1
          NNPP(IA) = K
          GO TO 120
121       IF(W(K))123,120,123
123       WRITE(6,124) K
124       FORMAT(1H1, 5H NODE, I6,9H UNLOADED, //)
          U(K)=-K
          W(K)=-K
120       CONTINUE
          NPND=3*NKNT
C
C          COLUMN INPUT ROUTINE
C
1000      J=1
          DO 8940 I=1,NODE
          IF(U(I))8930,8940,8930
8930      J7=3*I-2
          J8=3*I-1
          J9=3*I
          L(J)=J7
          L(J+1)=J8
          L(J+2)=J9
          J=J+3
8940      CONTINUE
          L(J)=999
          DO 740 I=1,MCOL
          II=L(I)
          IF(L(I)-999)6015,601,6015
6015      IF(NEC(II))740,6017,740
6017      IF(IAVAIL-MCOL)6006,6006,657
6006      III=I -1-IREC
          IF(III)6003,6005,6001
6001      DO 6002 JJ=1,III
6002      READ(9)
          GO TO 6005
6003      III=(-1)*III
          DO 6004 JJ=1,III
6004      BACKSPACE 9
6005      READ(9) (SIJ(JJ,IAVAIL),JJ=1,NCOMP)
          IREC=II
          NEC(II)=IAVAIL
          IAVAIL=IAVAIL+1
          GO TO 6015
 740      CONTINUE
 601      CONTINUE
```

```
      IF(KLU) 56,57,56
 57   CONTINUE
      DO 142 K=1,NODE
      IF(W(K))142,142,144
144   IF(CSM(K))142,151,151
151   NNN=3*K-2
      NNK = 3*K-1
      NN = 3*K
      S1= ((ABS(SIGMA(NNN  )- ALBAR(NNN)  )/D)**(T-1.0))*(T/D)
      S2= ((ABS(SIGMA(NNK  )- ALBAR(NNK)  )/D)**(T-1.0))*(T/D)
      S3=((ABS(SIGMA(NN)-ALBAR(NN))/D1)**(T-1.0))*(T/D1)
      S4=SQRT(((SIGMA( NNN )-ALPHA( NNN ))**2)+((SIGMA( NNK )-ALPHA(NNK)
     *     )**2)+((SIGMA(NN)-ALPHA(NN))**2))
      V1= (SIGMA( NNN )-ALPHA( NNN ))/S4
      V2= (SIGMA( NNK )-ALPHA(NNK))/S4
      V3= (SIGMA(NN )-ALPHA(NN ))/S4
      CSM(K) = (V1**2*S1) + (V2**2*S2)+ (V3**2*S3)
142   CONTINUE
 56   CONTINUE
C
C     COMPUTE INCREMENTS OF TOTAL STRAIN AND SUM
C
      CALL PLAST
C
      DO 100 I=1,NODE
      NNN=3*I-2
      NNK=3*I-1
      NN =3*I
      IF(W(I))140,100,102
C
C     COMPUTE THESE VALUES FOR ELASTIC NODES
C
140   DEP(NNN)=0.0
      DEP(NNK)=0.0
      DEP(NN )=0.0
      DSSI(NNN)=0.0
      DSSI(NNK)=0.0
      DSSI(NN) =0.0
      GO TO 100
C
C    C O   MPUTE THESE VARIABLES FOR PLASTIC NODES
C
C
102   IF(CSM(I))153,100,152
C
C     SOLVE FOR STRESS INCREMENTS
C
152   SMA2=SIGMA(NNN)-ALPHA(NNN)
      SMA1= SIGMA( NNK )-ALPHA( NNK )
      SMA = SIGMA(NN) -ALPHA(NN )
      DB    =(5.0 * SMA2**2 - 8.0 * SMA2 * SMA1 + 5.0* SMA1**2 + 36.0 *
     *          SMA**2)/(4.0*(CSM(I)))
      CE(1,1) =  ( SMA2 - 0.5* SMA1)**2 /DB + (1.0/E1)
      CE(1,2) =  ( SMA2 - 0.5* SMA1)*(SMA1-0.5*SMA2)/DB-GNU/E1
      CE(1,3) =  ( SMA2-0.5*SMA1)*(3.0*SMA)/DB
      CE(2,1) =  CE(1,2)
      CE(3,1) =  CE(1,3)
      CE(2,2) = (SMA1-0.5*SMA2)**2/DB + (1.0/E1)
      CE(2,3) =  (SMA1-0.5*SMA2)*(3.0*SMA)/DB
      CE(3,2) =  CE(2,3)
```

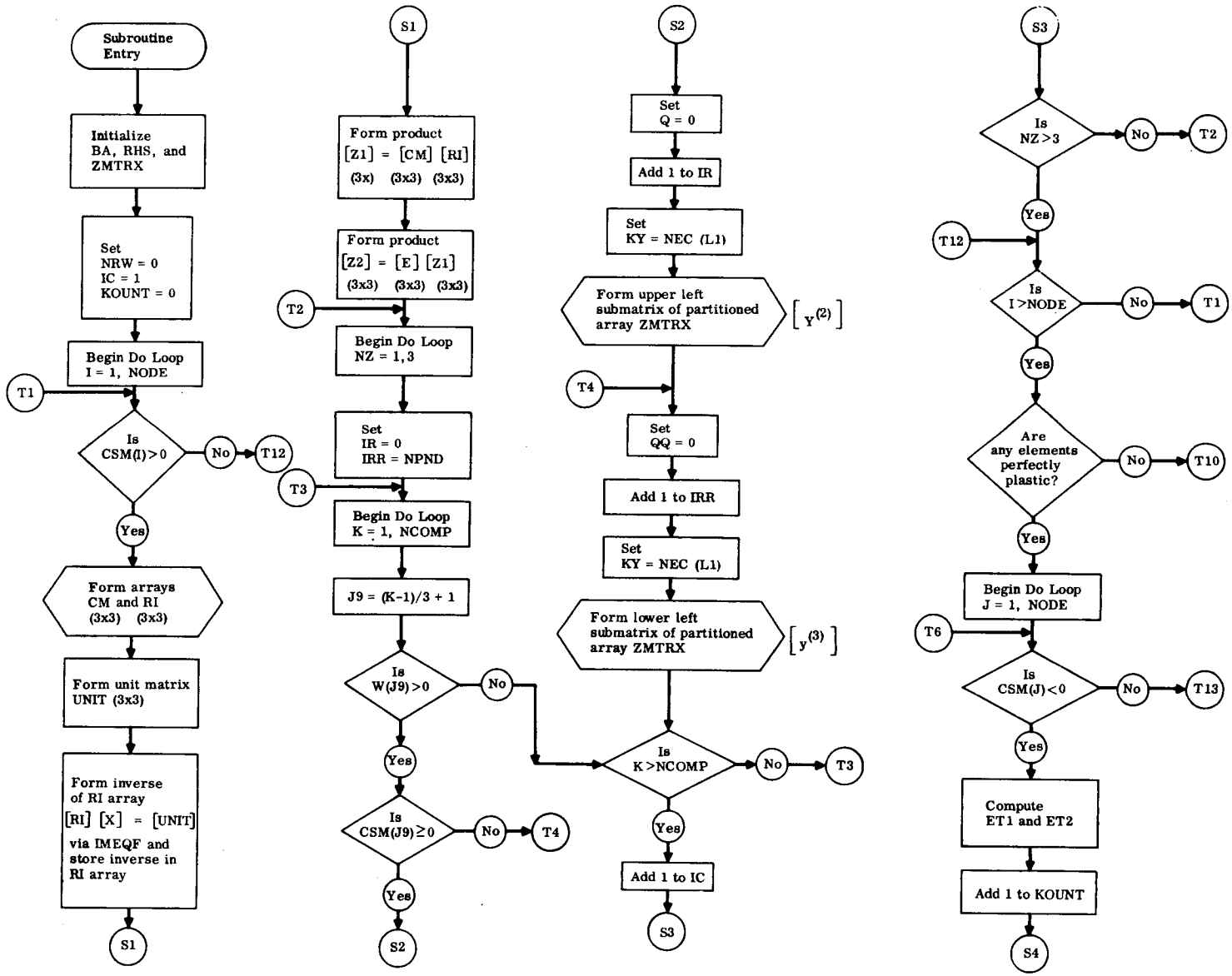
```
      CE(3,3) = (3.0*SMA)**2/DB + (2.0*(1.0+GNU))/E1
      RHS(1)= DE(NNN)
      RHS(2)= DE(NNK)
      RHS(3)= DE(NN)
      ARG6 =0.0
      M=IMEQF(3,3,1,CE,RHS,ARG6,ARG7)
      GO TO (110,111,112),M
112   WRITE(6,113)
113   FORMAT(15H CE IS SINGULAR)
      CALL EXIT
111   WRITE(6,114)
114   FORMAT( 9H OVERFLOW)
      CALL EXIT
110   CONTINUE
C
C     COMPUTE INCREMENTS OF PLASTIC STRAIN, TRANSLATION, AND INITIAL
C                                STRESS
C
      DSIG(NNN)=CE(1,1)
      DSIG(NNK)=CE(2,1)
      DSIG(NN) =CE(3,1)
      ANUM = (SIGMA(NNN)-ALPHA(NNN))*(DSIG(NNN)-0.5*DSIG(NNK))+(SIGMA
     *       (NNK)-ALPHA(NNK))*(DSIG(NNK)-0.5*DSIG(NNN))+3.0*(SIGMA(NN)
     *       -ALPHA(NN))*(DSIG(NN))
      DLAM(I)=ANUM/ DB
      DMU(I)=ANUM/(((SIGMA(NNN)-ALPHA(NNN))**2)-((SIGMA(NNN)-ALPHA(NNN))
     *      *(SIGMA(NNK)-ALPHA(NNK)))+((SIGMA(NNK)-ALPHA(NNK))**2) +
     *      (3.0*(SIGMA(NN)-ALPHA(NN))**2))
      DEP(NNN)=((SIGMA(NNN)-ALPHA(NNN))-0.5*(SIGMA(NNK)-ALPHA(NNK))) *
     *         DLAM(I)
      DEP(NNK)=((SIGMA(NNK)-ALPHA(NNK))-0.5*(SIGMA(NNN)-ALPHA(NNN))) *
     *         DLAM(I)
      DEP(NN)=3.0*(SIGMA(NN)-ALPHA(NN))*DLAM(I)
      DALPH(NNN)=(SIGMA(NNN)-ALPHA(NNN))*DMU(I)
      DALPH(NNK)=(SIGMA(NNK)-ALPHA(NNK))*DMU(I)
      DALPH(NN)=(SIGMA(NN)-ALPHA(NN))*DMU(I)
      DSSI(NNN)=-GNU1*(DEP(NNN)+GNU*DEP(NNK))
      DSSI(NNK)=-GNU1*(DEP(NNK)+GNU*DEP(NNN))
      DSSI(NN) = -E1/(2.0*(1.0+GNU))*DEP(NN)
      ALPHA(NNN)=ALPHA(NNN)+DALPH(NNN)
      ALPHA(NNK)=ALPHA(NNK)+DALPH(NNK)
      ALPHA(NN)=ALPHA(NN)+DALPH(NN)
153   SSI(NNN)=SSI(NNN)+DSSI(NNN)
      SSI(NNK)=SSI(NNK)+DSSI(NNK)
      SSI(NN)=SSI(NN)+DSSI(NN)
      EP(NNN)=EP(NNN)+DEP(NNN)
      EP(NNK)=EP(NNK)+DEP(NNK)
      EP(NN)=EP(NN)+DEP(NN)
100   CONTINUE
      DO 210 I=1,NCOMP
      SG=0.0
      DO 212 J=1,NCOMP
      J1 = (J-1)/3 + 1
      IF(U(J1)) 212,212,213
213   KY = NEC(J)
      SG=SG+SIJ(I,KY)*DSSI(J)
212   CONTINUE
      DE(I)=SG+SIM(I)*DELP
      EE(I) = EE(I) + DE(I)
210   CONTINUE
```

```
      DO 1101 I=1,NODE
      NNN=3*I-2
      NNK=3*I-1
      NN=3*I
      IF(W(I))251,251,252
  251  DSIG(NNN) = E(1,1)*DE(NNN) + E(1,2)*DE(NNK)
       DSIG(NNK) = E(2,1)*DE(NNN) + E(2,2)*DE(NNK)
       DSIG(NN) = E(3,3)*DE(NN)
  252 SIGMA(NNN)=SIGMA(NNN)+DSIG(NNN)
      SIGMA(NNK)=SIGMA(NNK)+DSIG(NNK)
      SIGMA(NN)=SIGMA(NN)+DSIG(NN)
  253 ELAS(NNN)=(1.0/E1)*(SIGMA(NNN)-GNU*SIGMA(NNK))
      ELAS(NNK)=(1.0/E1)*(SIGMA(NNK)-GNU*SIGMA(NNN))
      ELAS(NN)=((2.0*(1.0+GNU))/E1)* SIGMA(NN)
 1101 CONTINUE
C
C TES T     FOR  PERFECT  PLASTICITY
C
      IF(KLU)58,70,58
   70 IF(ZFL)58,59,59
   59 CONTINUE
      DO 117 I=1,NODE
      N1=3*I-2
      N2=3*I-1
      N3=3*I
      AL1=(ALPHA(N1)-ALBAR(N1))
      AL2=(ALPHA(N2)-ALBAR(N2))
      AL3=(ALPHA(N3)-ALBAR(N3))
      FLOW = AL1**2 - AL1*AL2 + AL2**2 + 3.0*AL3**2
      IF(ZFL-FLOW)118,118,117
  118 CSM(I)=-1.0
      IA=IA+1
      NNPP(IA)=I
  117 CONTINUE
   58 CONTINUE
      IF(P/RMXLD - 1.0)132,125,125
  125 CONTINUE
      KTRANS = 2
      GO TO 126
  132 IF(ICOUNT-INT)130,131,130
  130 ICOUNT = ICOUNT + 1
      GO TO 41
  131 CONTINUE
      ICOUNT = 1
  126 WRITE(6,2012) P
      WRITE(6,2000)
 2000 FORMAT(/ 21X,16HELEMENT STRESSES, /)
 2003 FORMAT(1H0,7HNODE NO,4X,7HSIGMA-X,12X,7HSIGMA-Y,13X,7HTAU-X,Y)
      WRITE(6,2003)
      DO 2001 I=1,NODE
      NNN=3*I-2
      NNK=3*I-1
      NN = 3*I
      WRITE(6,2002) I,SIGMA(NNN),SIGMA(NNK),SIGMA(NN)
 2002 FORMAT(3X,I3,2X,E16.8,3X,E16.8 ,3X,E16.8)
 2001 CONTINUE
      DO 2020 K=1,NODE
      IF(W(K))2021,2020,2021
 2021 NNN= 3*K-2
      NNK= 3*K-1
```

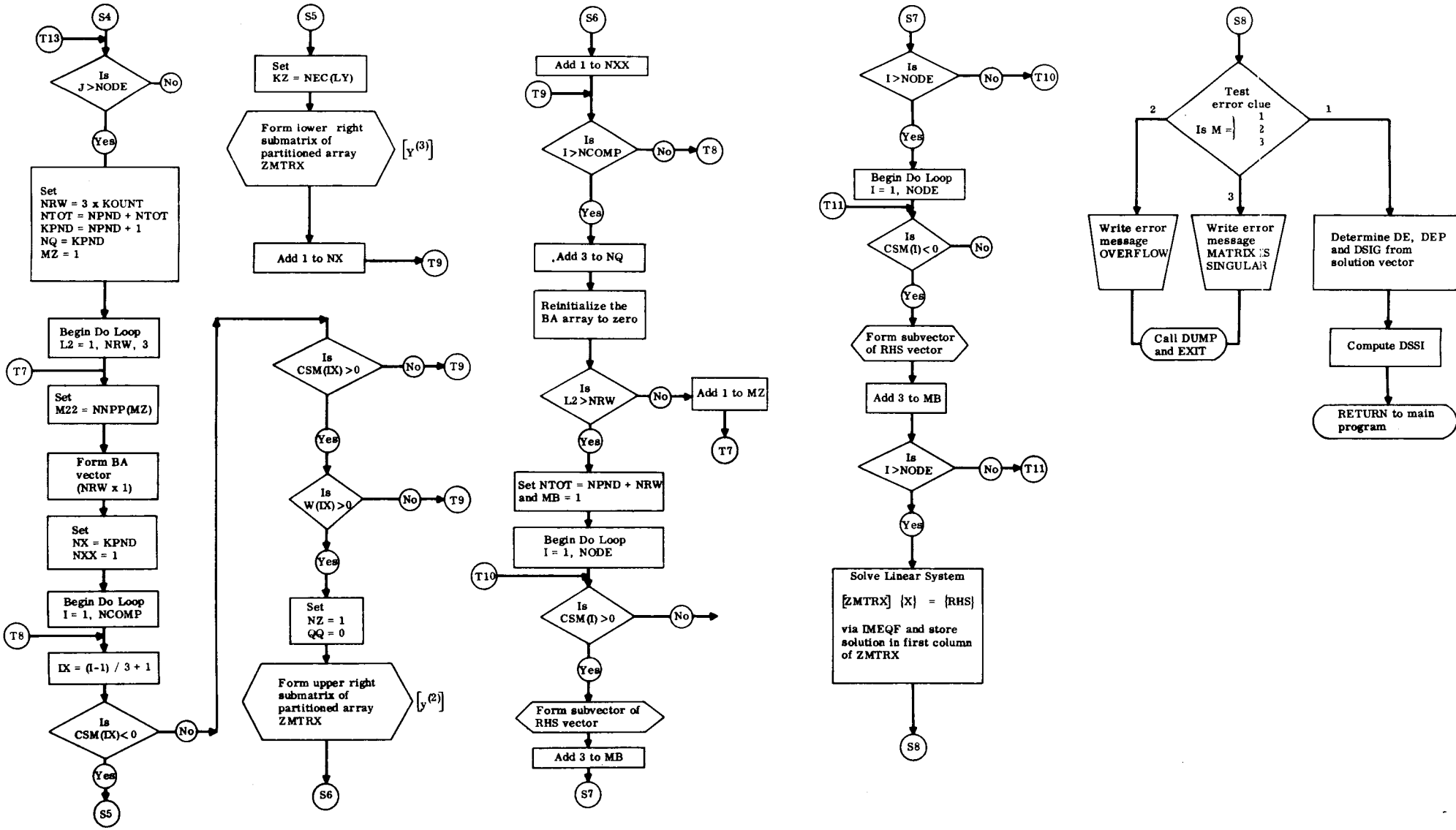
```
      NN = 3*K
      WRITE(6,2006)
      WRITE(6,2007) K,CSM(K),DLAM(K),DMU(K)
      WRITE(6,2008)
      WRITE(6,2009)   DALPH(NNN),DALPH(NNK),DALPH(NN),DEP(NNN),DEP(NNK),
     *  DEP(NN)
      WRITE(6,2050)
2050  FORMAT( 1H0)
      WRITE(6,2010)
      WRITE(6,2009) ALPHA(NNN),ALPHA(NNK),ALPHA(NN),EP(NNN),EP(NNK),EP(N
     *N)
      WRITE(6,2011)
      WRITE(6,2009) EE(NNN),EE(NNK),EE(NN),ELAS(NNN),ELAS(NNK),ELAS(NN)
2006  FORMAT(1H0,10HELEMENT NO,19X,1HC,16X,4HDLAM,17X,3HDMU   )
2007  FORMAT(1H0,I10,6E20.6)
2008  FORMAT(1H0,16X,14H DELTA ALPHA-X,6X,14H DELTA ALPHA-Y,5X,15H DELTA
     *  ALPHA-XY,9X,11H DELTA EP-X, 9X,11H DELTA EP-Y,
     *  7X,13H DELTA GAM-XY)
2009  FORMAT(10X,6E20.6)
2010  FORMAT(18X,12H TOT ALPHA-X, 8X,12H TOT ALPHA-Y,7X,13H TOT ALPHA-XY
     *,7X,13H PLASTIC EP-X,7X,13H PLASTIC EP-Y,5X,15H PLASTIC GAM-XY)
2011  FORMAT(1H0,21X, 9H STRAIN-X,10X, 9H STRAIN-Y,10X,10H STRAIN-XY, 7X
     *,13H ELASTIC EP-X,7X,13H ELASTIC EP-Y,5X,15H ELASTIC GAM-XY)
2012  FORMAT(1H0,3X,17H LOAD LEVEL, P = , F8.0 )
2020  CONTINUE
      GO TO(41,65),KTRANS
  65  IF(NRSTRT) 7,7,66
  66  WRITE(10) ALBAR,ALPHA,CSM,D,D1,DALPH,DE,DEP,DELP,DLAM,DMU,DSIG,
     *          DSSI,E,E1,EE,EI,ELAS,EP,G,GIM,GNU,GNU1,IAVAIL,INT,IREC,
     *          KLU,L,MCOL,NCOMP,NEC,NNPP,NODE,NPND,P,PCRIT,REST,SIGO,
     *          SIGMA,SIJ,SIM,SMALL,SQ,SSI,T,U,W,Z,ZFL
      GO TO 9
 657  WRITE(6,2030)
2030  FORMAT( 42H NUMBER OF COLUMNS EXCEEDS AVAILABLE SPACE )
   9  CONTINUE
      REWIND 10
      REWIND 9
      PAUSE 2
      CALL EXIT
      END


      SUBROUTINE SETAPE(NUNIT,NFILE)
      DATA SIGNAL /0111111111111 /
      REWIND 9
      NSET=NFILE-1
      IF(NSET)3,3,4
   4  DO 1 I=1,NSET
   2  READ (9) X
      IF(X-SIGNAL)2, 1,2
   1  CONTINUE
   3  RETURN
      END
```

STRAIN PROGRAM
Subroutine PLAST



STRAIN PROGRAM
Subroutine PLAST



```
$ IBFT C     PLAST    LIST,REF
             SUBROUTINE PLAST
             INTEGER U,W
             COMMON ALPHA(165)
             COMMON CSM(55)
             COMMON DE(165)
             COMMON DELP
             COMMON DEP(165)
             COMMON DSIG(165)
             COMMON DSSI(165)
             COMMON E(3,3)
             COMMON E1
             COMMON EI(3,3)
             COMMON GNU
             COMMON KLU
             COMMON MCOL
             COMMON NCOMP
             COMMON NEC(165)
             COMMON NNPP(57)
             COMMON NODE
             COMMON NPND
             COMMON SIGMA(165)
             COMMON SIJ(165,57)
             COMMON SIM(165)
             COMMON U(55)
             COMMON W(55)
             DIMENSION ARG9(57)
             DIMENSION BA(57)
             DIMENSION CM(3,3)
             DIMENSION ET1(55)
             DIMENSION ET2(55)
             DIMENSION RHS(57)
             DIMENSION RI(3,3)
             DIMENSION UNIT(3,3)
             DIMENSION Z1(3,3)
             DIMENSION Z2(3,3)
             DIMENSION ZMTRX(57,57)
             DO 200 I=1,MCOL
             BA(I) = 0.0
             RHS(I) = 0.0
             DO 200 J=1,MCOL
             ZMTRX(I,J) = 0.0
      200    CONTINUE
             NRW=0
             IC=1
             KOUNT = 0
             DO 1 I=1,NODE
             IF(CSM(I))1,1,6
        6    L1=3*I-2
             L2=3*I-1
             L3=3*I
             SIGB1= SIGMA(L1)-ALPHA(L1)
             SIGB2= SIGMA(L2)-ALPHA(L2)
             SIGB3= SIGMA(L3)-ALPHA(L3)
             DB = (5.0*SIGB1**2 -8.0 * SIGB1* SIGB2 + 5.0*SIGB2**2 + 36.0*SIGB3
        *         **2)/(4.0*CSM(I))
             CM(1,1)=(SIGB1-0.5*SIGB2)**2/DB
             CM(1,2)=(SIGB1-0.5*SIGB2)*(SIGB2-0.5*SIGB1)/DB
             CM(1,3)=(SIGB1-0.5*SIGB2)*(3.0*SIGB3)/DB
             CM(2,1)= CM(1,2)
```

```
      CM(2,2)=(SIGB2-0.5*SIGB1)**2/DB
      CM(2,3)=(SIGB2-0.5*SIGB1)*(3.0*SIGB3)/DB
      CM(3,1)=CM(1,3)
      CM(3,2)=CM(2,3)
      CM(3,3)=(3.0*SIGB3)**2 /DB
      RI(1,1)= CM(1,1) + (1.0/E1)
      RI(1,2)= CM(1,2) - (GNU/E1)
      RI(1,3)= CM(1,3)
      RI(2,1)= RI(1,2)
      RI(2,2)= CM(2,2) + (1.0/E1)
      RI(2,3)= CM(2,3)
      RI(3,1)= CM(1,3)
      RI(3,2) = CM(2,3)
      RI(3,3) = CM(3,3)+ 2.0* (1.0 + GNU)/E1
      DO 11 II=1,3
      DO 11 JJ=1,3
 11   UNIT(II,JJ)=0.0
      DO 12 II=1,3
 12   UNIT(II,II)= 1.0
      ARG6=0.0
      M=IMEQF(3,3,3,RI,UNIT,ARG6,ARG9)
      DO 2 J1=1,3
      DO 2 J2=1,3
      Q=0.0
      DO 3 J3=1,3
  3   Q= Q+ CM(J1,J3)*RI(J3,J2)
  2   Z1(J1,J2) = Q
      DO 4 J1=1,3
      DO 4 J2=1,3
      Q=0.0
      DO 5 J3=1,3
  5   Q=Q+E (J1,J3)*Z1(J3,J2)
  4   Z2(J1,J2)=Q
      DO 10 NZ=1,3
      IR=0
      IRR=NPND
      DO 7 K=1,NCOMP
      J9=(K-1)/3 +1
      IF(W(J9))7,7,28
 28   IF(CSM(J9))29,8,8
  8   Q=0.0
      IR=IR+1
      KY = NEC(L1)
      DO 9 KK=1,3
      Q=Q+ SIJ(K,KY) * Z2(KK,NZ)
  9   KY=KY+1
      ZMTRX(IR,IC) = Q
      GO TO 7
 29   QQ = 0.0
      IRR = IRR + 1
      KY = NEC(L1)
      DO 30 KQ=1,3
      QQ = QQ + SIJ(K,KY) * Z2(KQ,NZ)
 30   KY=KY+1
      ZMTRX(IRR,IC) = QQ
  7   CONTINUE
      IC=IC+1
 10   CONTINUE
  1   CONTINUE
      IF(KLU)60,61,60
```

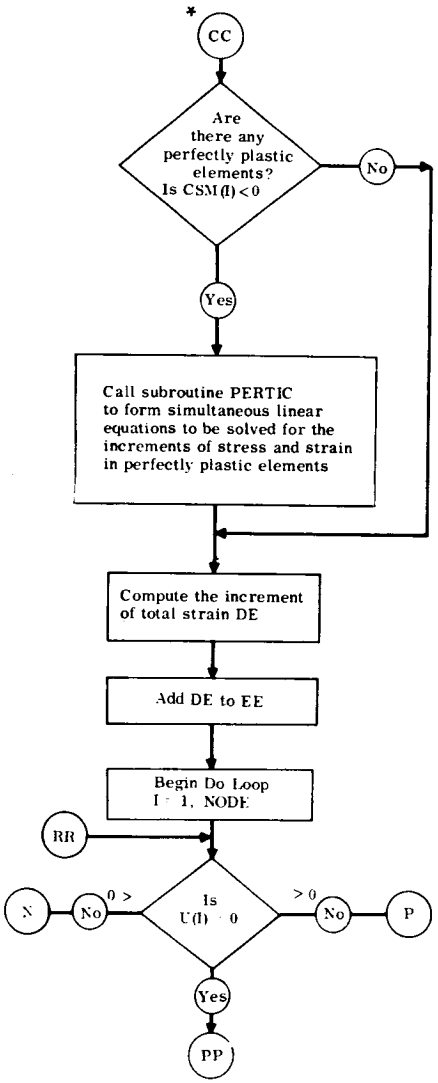
```
61      CONTINUE
        DO 20 I=1,IR
20      ZMTRX(I,I) = ZMTRX(I,I) + 1.0
60      CONTINUE
        DO 21 I=1,NODE
        IF(CSM(I))22,21,21
21      CONTINUE
        GO TO 100
22      DO 31 J=1,NODE
        IF(CSM(J))32,31,31
32      N1=3*J-2
        N2=3*J-1
        N3=3*J
        SB1=(SIGMA(N1)-ALPHA(N1))-(SIGMA(N2)-ALPHA(N2))/2.0
        SB2=(SIGMA(N2)-ALPHA(N2))-(SIGMA(N1)-ALPHA(N1))/2.0
        SB3=3.0 *(SIGMA(N3)-ALPHA(N3))
        ET1(J) = SB2/SB1
        ET2(J) = SB3/SB1
        KOUNT = KOUNT + 1
31      CONTINUE
        NRW = 3* KOUNT
        NTOT = NPND + NRW
        KPND = NPND + 1
        NQ = KPND
        MZ = 1
        DO 34 L2=1,NRW,3
        M22=NNPP(MZ)
        BA(L2) = E(1,1) + E(1,2)*ET1(M22)
        BA(L2+1) = E(2,1) + E(2,2)*ET1(M22)
        BA(L2+2) = E(3,3) * ET2(M22)
        NX = KPND
        NXX=1
        DO 35 I=1,NCOMP
        IX = (I-1)/3 + 1
        IF(CSM(IX))36,35,37
36      NZ=1
        Q=0.0
        DO 38 LY=1,NCOMP
        J=(LY-1)/3 + 1
        IF(CSM(J))39,38,38
39      KZ=NEC(LY)
        Q=Q+SIJ(I,KZ)*BA(NZ)
        NZ=NZ+1
38      CONTINUE
        ZMTRX(NX,NQ) = Q
        NX=NX+1
        GO TO 35
37      IF(W(IX))35,35,40
40      NZ=1
        QQ=0.0
        DO 41 LQ=1,NCOMP
        JQ = (LQ-1)/3 + 1
        IF(CSM(JQ))42,41,41
42      JJ = NEC(LQ)
        QQ = QQ + SIJ(I,JJ)*BA(NZ)
        NZ=NZ+1
41      CONTINUE
        ZMTRX(NXX,NQ)=QQ
        NXX = NXX + 1
35      CONTINUE
```

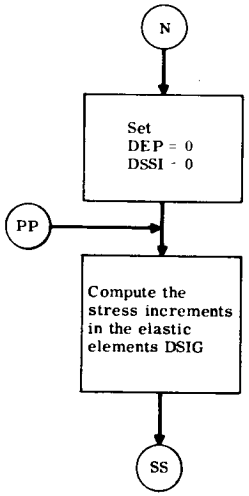
```
      NQ = NQ + 3
      DO 43 IK=1,NRW
43    BA(IK)=0.0
34    MZ = MZ + 1
      MX = 1
      MK = KPND
      DO 44 I=KPND,NTOT,3
      M22 = NNPP(MX)
      ZMTRX(MK,I) = ZMTRX(MK,I) + 1.0
      ZMTRX(MK+1,I) = ZMTRX(MK+1,I) + ET1(M22)
      ZMTRX(MK+2,I) = ZMTRX(MK+2,I) + ET2(M22)
      MK = MK+3
44    MX=MX+1
      MB=1
      DO 45 I=KPND,NTOT,3
      MQ = NNPP(MB)
      I1=I+1
      I2=I+2
      ZMTRX(I,I1) = -EI(1,1) * ET1(MQ) + EI(1,2)
      ZMTRX(I,I2) = -EI(1,1)*ET2(MQ)
      ZMTRX(I1,I1)= -EI(2,1) * ET1(MQ) + EI(2,2)
      ZMTRX(I1,I2)= -EI(2,1) * ET2(MQ)
      ZMTRX(I2,I1)= 0.0
      ZMTRX(I2,I2)= EI(3,3)
45    MB=MB+1
C
C     F   ORMATION OF THE RIGHT  HAND  SIDE
C
100   NTOT = NPND + NRW
      MB=1
      DO 50 I=1,NODE
      IF(CSM(I))50,50,51
51    N1=3*I-2
      N2=3*I-1
      N3=3*I
      RHS(MB)=SIM(N1)*DELP
      RHS(MB+1) = SIM(N2)*DELP
      RHS(MB+2) = SIM(N3)*DELP
      MB=MB+3
50    CONTINUE
      DO 52 I=1,NODE
      IF(CSM(I))53,52,52
53    N1=3*I-2
      N2=3*I-1
      N3=3*I
      RHS(MB)=SIM(N1)*DELP
      RHS(MB+1)=SIM(N2)*DELP
      RHS(MB+2)=SIM(N3)*DELP
      MB=MB+3
52    CONTINUE
      ARG6 = 0.0
      M=IMEQF(MCOL,NTOT,1,ZMTRX,RHS,ARG6,ARG9)
      GO TO (54,55,56),M
56    WRITE(6,57)
57    FORMAT(19H MATRIX IS SINGULAR )
      CALL DUMP
55    WRITE(6,58)
58    FORMAT( 9H OVERFLOW)
      CALL DUMP
C
```

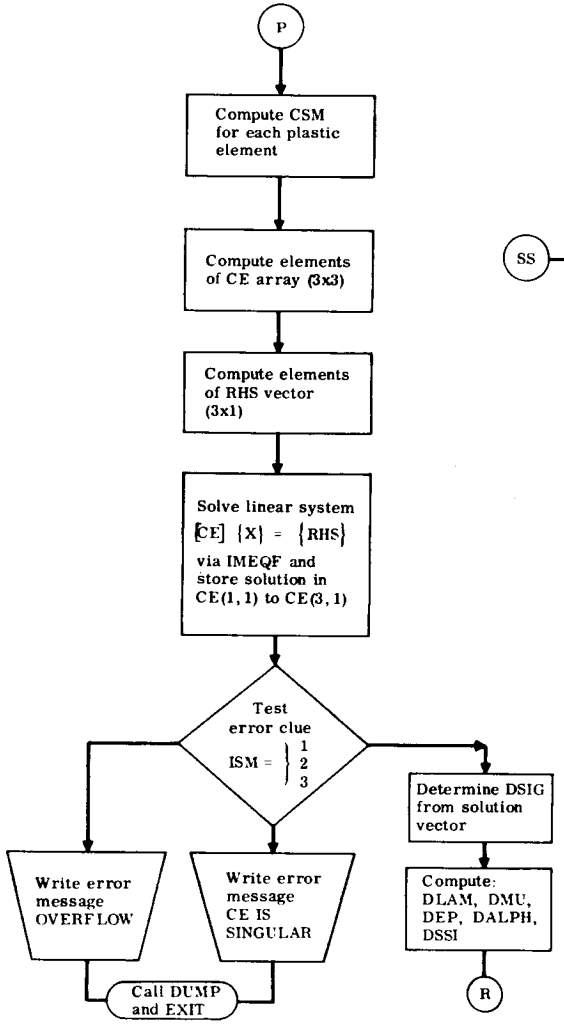
```
C       F   ORMATION  OF  THE  SOLUTION  VECTOR
C
 54     MB=1
        DO 46 I=1,NODE
        IF(CSM(I))46,46,47
 47     N1 = 3*I-2
        N2 = 3*I-1
        N3 = 3*I
        DE(N1) = ZMTRX(MB,1)
        DE(N2) = ZMTRX(MB+1,1)
        DE(N3) = ZMTRX(MB+2,1)
        MB = MB + 3
 46     CONTINUE
        DO 48 I=1,NODE
        IF(CSM(I))49,48,48
 49     N1=3*I-2
        N2=3*I-1
        N3=3*I
        DEP(N1)=ZMTRX(MB,1)
        DEP(N2)=ET1(I)*DEP(N1)
        DEP(N3)=ET2(I)*DEP(N1)
        DSIG(N2)=ZMTRX(MB+1,1)
        DSIG(N3) = ZMTRX(MB+2,1)
        DSIG(N1) = -ET1(I)*DSIG(N2)-ET2(I)*DSIG(N3)
        DSSI(N1) = -(E(1,1)*DEP(N1) + E(1,2)*DEP(N2))
        DSSI(N2) = -(E(2,1)*DEP(N1) + E(2,2)*DEP(N2))
        DSSI(N3) = -E(3,3)*DEP(N3)
        MB = MB + 3
 48     CONTINUE
        RETURN
        END
```

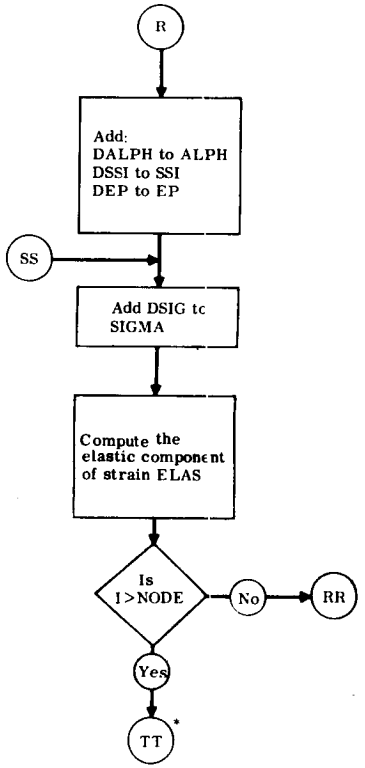
STRAPP PROGRAM

Flow of computations similar to that shown for the STRAIN PROGRAM except for section enclosed by dashed line. This section is replaced by the following-









* CC and TT appear in flow chart associated with the STRAIN PROGRAM

STRAPP PROGRAM

```
$ IBFT C     STRAPP  LIST,REF
             PAUSE 1
             REWIND 9
             COMMON ALPHA(165)
             COMMON CSM( 55)
             COMMON DELP
             COMMON DEP(165)
             COMMON DSIG (165)
             COMMON DSSI(165)
             COMMON E(3,3)
             COMMON EI(3,3)
             COMMON KLU
             COMMON MCOL
             COMMON NCOMP
             COMMON NEC(165)
             COMMON NNPP(57)
             COMMON NODE
             COMMON SIGMA(165)
             COMMON SIJ(165,57)
             COMMON SIM(165)
             DIMENSION ALBAR(165)
             DIMENSION ARG7(3)
             DIMENSION CE(3,3)
             DIMENSION DALPH(165)
             DIMENSION DE(165)
             DIMENSION DLAM(55)
             DIMENSION DMU(55)
             DIMENSION EE(165)
             DIMENSION ELAS(165)
             DIMENSION EP(165)
             DIMENSION GIM(165)
             DIMENSION L(166)
             DIMENSION REST(165)
             DIMENSION RHS(3)
             DIMENSION SQ(165)
             DIMENSION SSI(165)
             DIMENSION U(55)
             CONST=1.0E+06
             READ(5,3) IRSTRT,RMXLD
       3     FORMAT(I5,E15.7)
             IF(IRSTRT) 60,60,61
      61     CONTINUE
             READ(11)ALBAR,ALPHA,CSM,D,D1,DALPH,DE,DEP,DELP,DLAM,DMU,DSIG,DSSI,
        *            E,E1,EE,EI,ELAS,EP,G,GIM,GNU,GNU1,IAVAIL,INT,IREC,KLU,L,
        *            MCOL,NCOMP,NEC,NNPP,NODE,NPND,P,PCRIT,REST,SIGO,SIGMA,SIJ,
        *            SIM,SMALL,SQ,SSI,T,U,Z,ZFL
             REWIND 11
             IREC = 0
             ICOUNT = INT
             NCYCLE = 99
             WRITE(6,2041)
             WRITE(6,2022) INT,SIGO,E1,GNU,DELP,D,T,RMXLD
             IF(RMXLD/P-1.0)62,62,63
      62     KTRANS = 2
             GO TO 132
      63     KTRANS = 1
             GO TO 132
      60     CONTINUE
             KTRANS = 1
             READ(5,1)NODE,NCOMP,MCOL,INT
```

```
      1     FORMAT(5I10)
            WRITE(6,1) MCOL
            READ(5,2) SIGO,E1,GNU,ZFL
      2     FORMAT(5E15.7,I2,I3)
            GNU1 = E1/(1.0-GNU**2)
            E(1,1)= GNU1
            E(1,2)= GNU*E(1,1)
            E(2,1) = E(1,2)
            E(2,2) = E(1,1)
            E(3,3) = E1/(2.0*(1.0+GNU))
            E(1,3)= 0.0
            E(2,3)= 0.0
            E(3,1)= 0.0
            E(3,2)= 0.0
            EI(1,1)= 1.0/E1
            EI(1,2)=-GNU/E1
            EI(1,3)=0.0
            EI(2,1)=EI(1,2)
            EI(2,2)=EI(1,1)
            EI(2,3)=0.0
            EI(3,1)=0.0
            EI(3,2)=0.0
            EI(3,3)=2.0*(1.0+GNU)/E1
C
C     C O   MPUTE GIM MATRIX
C
            CALL SETAPE(9,2)
            READ(9) SIM
            DO 15 I2=1,NCOMP,3
            DO 15 I =1,3
            IN = I2 +I-1
            GIM(IN)=0.0
            DO 15 J=1,3
            IN2 = I2+J-1
     15     GIM(IN)= GIM(IN)+ E(I,J)*SIM(IN2)
C
C    IN I   TIALIZE
C
            CALL SETAPE(9,1)
            IAVAIL=1
            IREC = 0
            DO 4 I=1,MCOL
            NNPP(I)=0
      4     L(I)=0
            DO 200 K=1,NCOMP
            NEC(K)=0
            DSSI(K)=0.0
            ALBAR(K) = 0.0
            EE(K)=0.0
            SIGMA(K)=0.0
            ALPHA(K)=0.0
            SSI(K)=0.0
            EP(K)=0.0
    200     CONTINUE
   1110     DO 6 J=1,NODE
            CSM(J)=0.0
      6     U(J) = 0.0
C
C    RE A   D INPUT FOR NEXT  CYCLE
C
```

```
    7       READ(5,2) PMA,DELP,D,T,RMXLD,KLU,NRSTRT
            GO TO (777,787),KTRANS
  787       DO 127 I=1,NCOMP
            DSSI(I) = 0.0
  127       ALBAR(I) = ALPHA(I)
  777       CONTINUE
            ICOUNT = INT
            KTRANS = 1
            WRITE(6,2041)
 2041       FORMAT(6H1INPUT,/ 7X,3HINT,11X,4HSIGO,14X,1HE,13X,2HNU,11X,4HDELP,
     *            14X,1HD,14X,1HT,10X,5HRMXLD)
            WRITE(6,2022)INT,SIGO,E1,GNU,DELP,D,T,RMXLD
 2022       FORMAT(1H0(I10,8E15.6) )
            IF(NCYCLE-99) 8,11,8
    8       NCYCLE = 99
            YCEMAX = 0.0
            DO 12 I=1,NODE
            Z1= GIM(3*I-2) * PMA
            Z2 = GIM(3*I-1) * PMA
            Z3 = GIM(3*I) * PMA
            YCE = Z1**2 - Z1*Z2 + Z2**2 + 3.0*Z3**2
            IF(YCE-YCEMAX)12,13,13
   13       N1 = I
            YCEMAX = YCE
   12       CONTINUE
            U(N1)=N1
            IF(KLU) 20,21,20
   20       CSM(N1) = -1.0
            NNPP(1) = N1
   21       CONTINUE
            PCRIT=SQRT((SIGO**2)/((GIM(3*N1-2)**2)-(GIM(3*N1-2)*GIM(3*N1-1)) +
     *            (GIM(3*N1-1)**2)+(3.0*(GIM(3*N1)**2))))
            P=PCRIT
            D1 = D/1.732051
            Z = SIGO**2
            SMALL=CONST/Z
            G =E1/3.0
            DO 16 K=1,NCOMP
            EE(K) = SIM(K) * PCRIT
   16       SIGMA(K) = GIM(K) * PCRIT
            GO TO 41
C
C   ST A    RTING PROCEDURE FOR ALL CYCLES BUT  THE FIRST
C
   11       CONTINUE
            DO 30 I=1,NCOMP
            SG=0.0
            DO 31 J=1,NCOMP
            N = (J-1)/3 + 1
            IF(U(N))32,31,32
   32       KQ = NEC(J)
            SG = SG + SIJ(I,KQ) * SSI(J)
   31       CONTINUE
   30       REST(I) = SG
            WRITE(6,2034)
 2034       FORMAT(1H1,7HNODE NO,21X,15HRESIDUAL STRESS,42X,15HRESIDUAL STRAIN
     *      , //)
            DO 33 I=1,NODE
            NN=3*I
            NNN=3*I-2
```

```
      NNK=3*I-1
      SQ(NNN)= GNU1*(REST(NNN)+GNU*REST(NNK))+SSI(NNN)
      SQ(NNK)= GNU1*(REST(NNK)+GNU*REST(NNN))+SSI(NNK)
      SQ(NN) = E1/(2.0*(1.0+GNU))*(REST(NN))+ SSI(NN)
      WRITE(6,2032)(I,SQ(NNN),SQ(NNK),SQ(NN),REST(NNN),REST(NNK),
     *                REST(NN))
 2032 FORMAT(3X,I3,2X,6E19.8/)
   33 CONTINUE
      IF(PMA)10,9,10
   10 YCEMAX = 0.0
      DO 34 I=1,NODE
      NN=3*I
      NNN=3*I-2
      NNK=3*I-1
      Z1=GIM(NNN)*PMA +SQ(NNN)
      Z2=GIM(NNK)*PMA +SQ(NNK)
      Z3=GIM(NN) *PMA +SQ(NN)
      YCE = ( Z1-ALBAR(NNN))**2 - (Z1-ALBAR(NNN))*(Z2-ALBAR(NNK)) + (Z2-
     *      ALBAR(NNK))**2 + 3.0*(Z3-ALBAR(NN))**2
      IF(YCE-YCEMAX)34,36,36
   36 N1 = I
      YCEMAX = YCE
   34 CONTINUE
      NNN= 3*N1-2
      NNK= 3*N1-1
      NN = 3*N1
      DNOM=GIM(NNN)**2-GIM(NNN)*GIM(NNK)+GIM(NNK)**2+3.0*(GIM(NN)**2)
      B=-((ALBAR(NNN)-SQ(NNN))*(2.0*GIM(NNN)-GIM(NNK))+(ALBAR(NNK)-SQ(NN
     *K))*(2.0*GIM(NNK)-GIM(NNN))+(6.0*GIM(NN)*(ALBAR(NN)-SQ(NN))))/DNOM
      C=((ALBAR(NNN)-SQ(NNN))**2 -((ALBAR(NNN)-SQ(NNN))*(ALBAR(NNK)-SQ
     *  (NNK)))+((ALBAR(NNK)-SQ(NNK))**2) +(3.0*(ALBAR(NN)-SQ(NN))**2)
     *  -Z)/DNOM
      IF(PMA)37,37,39
   37 PCRIT=(-B-SQRT(B**2-4.0*C))/2.0
      GO TO 40
   39 PCRIT=(-B+SQRT(B**2-4.0*C))/2.0
   40 P=PCRIT
      DO 265 I=1,NCOMP
      SIGMA(I) = GIM(I)*P + SQ(I)
      EE(I) = SIM(I)*P + REST(I)
  265 CONTINUE
      DO 51 I=1,MCOL
   51 NNPP(I)=0
      DO50 J=1,NODE
      CSM(J) = 0.0
   50 U(J) = 0.0
      U(N1) = N1
      IF(KLU) 22,23,22
   22 CSM(N1) = -1.0
      NNPP(1) = N1
   23 CONTINUE
   41 P=P+DELP
      IA=0
      DO 120 K=1,NODE
      NNN = 3*K-2
      NNK = 3*K-1
      NN  = 3*K
      YCE = (SIGMA( NNN )-ALPHA(NNN  ))**2 -((SIGMA(NNN  )-ALPHA(NNN  ))
     *      *(SIGMA(NNK ) -ALPHA(NNK  ))) + 3.0*(SIGMA(NN )-ALPHA(NN ))
     *      **2 +(SIGMA(NNK)-ALPHA(NNK))**2
```

```
      Z9 = (YCE   /Z-1.0)
      IF(Z9+SMALL)121,122,122
122   U(K)=K
      IF(KLU)24,120,24
 24   CSM(K) = -1.0
      IA = IA+1
      NNPP(IA) = K
      GO TO 120
121   IF(U(K))123,120,123
123   WRITE(6,124) K
124   FORMAT(1H1, 5H NODE, I6,9H UNLOADED, //)
      U(K)=-K
120   CONTINUE
C
C      COLUMN INPUT ROUTINE
C
1000  J=1
      DO 8940 I=1,NODE
      IF(U(I))8930,8940,8930
8930  J7=3*I-2
      J8=3*I-1
      J9=3*I
      L(J)=J7
      L(J+1)=J8
      L(J+2)=J9
      J=J+3
8940  CONTINUE
      L(J1)=999
      DO 740 I=1,MCOL
      II=L(I)
      IF(L(I)-999)6015,604,6015
6015  IF(NEC(II))740,6017,740
6017  IF(IAVAIL-MCOL)6006,6006,657
6006  III=II-1-IREC
      IF(III)6003,6005,6001
6001  DO 6002 JJ=1,III
6002  READ(9)
      GO TO 6005
6003  III=(-1)*III
      DO 6004 JJ=1,III
6004  BACKSPACE 9
6005  READ(9) (SIJ(JJ,IAVAIL),JJ=1,NCOMP)
      IREC=II
      NEC(II)=IAVAIL
      IAVAIL=IAVAIL+1
      GO TO 6015
 740  CONTINUE
 604  DO 601 I=1,NODE
      IF(CSM(I))602,601,601
 601  CONTINUE
      GO TO 603
 602  CALL PERTIC
 603  CONTINUE
C     COMPUTE INCREMENTS OF TOTAL STRAIN AND SUM
C
      DO 103 I1=1,NCOMP
      SG=0.0
      DO 104 J1=1,NCOMP
      N=(J1-1)/3 +1
      IF(U(N))104,104,105
```

```
105     KQ=NEC(J1)
        SG=SG + SIJ(I1,KQ)* DSSI(J1)
104     CONTINUE
        DE(I1) = SG + SIM(I1)*DELP
103     EE(I1) = EE(I1) + DE(I1)
        DO 100 I=1,NODE
        NNN=3*I-2
        NNK=3*I-1
        NN =3*I
        IF(U(I))140,101,116
116     IF(CSM(I))115,102,102
C
C       COMPUTE THESE VALUES FOR ELASTIC NODES
C
140     DEP(NNN)=0.0
        DEP(NNK)=0.0
        DEP(NN )=0.0
        DSSI(NNN)=0.0
        DSSI(NNK)=0.0
        DSSI(NN) =0.0
101     DSIG(NNN)= GNU1*(DE(NNN)+GNU*DE(NNK))
        DSIG(NNK)= GNU1*(DE(NNK)+GNU*DE(NNN))
        DSIG(NN)=(E1/(2.0*(1.0+GNU)))*DE(NN)
        GO TO 1101
C
C    C O   MPUTE THESE VARIABLES FOR PLASTIC NODES
C
C
102     CONTINUE
        S1= ((ABS(SIGMA(NNN  )- ALBAR(NNN)  )/D)**(T-1.0))*(T/D)
        S2= ((ABS(SIGMA(NNK  )- ALBAR(NNK)  )/D)**(T-1.0))*(T/D)
        S3=((ABS(SIGMA(NN)-ALBAR(NN))/D1)**(T-1.0))*(T/D1)
        S4=SQRT(((SIGMA( NNN )-ALPHA( NNN ))**2)+((SIGMA( NNK )-ALPHA(NNK)
     *      )**2)+((SIGMA(NN)-ALPHA(NN))**2))
        V1= (SIGMA( NNN )-ALPHA( NNN ))/S4
        V2= (SIGMA( NNK )-ALPHA(NNK))/S4
        V3= (SIGMA(NN )-ALPHA(NN ))/S4
        CSM(I) = (V1**2*S1) + (V2**2*S2)+ (V3**2*S3)
C
C       SOLVE FOR STRESS INCREMENTS
C
        SMA2= SIGMA( NNN )-ALPHA( NNN )
        SMA1= SIGMA( NNK )-ALPHA( NNK )
        SMA = SIGMA(NN) -ALPHA(NN )
        DB    =(5.0 * SMA2**2 - 8.0 * SMA2 * SMA1 + 5.0* SMA1**2 + 36.0 *
     *          SMA**2)/(4.0*(CSM(I)))
        CE(1,1) =  ( SMA2 - 0.5* SMA1)**2 /DB + (1.0/E1)
        CE(1,2) =  ( SMA2 - 0.5* SMA1)*(SMA1-0.5*SMA2)/DB-GNU/E1
        CE(1,3) =  ( SMA2-0.5*SMA1)*(3.0*SMA)/DB
        CE(2,1) =  CE(1,2)
        CE(3,1) =  CE(1,3)
        CE(2,2) = (SMA1-0.5*SMA2)**2/DB + (1.0/E1)
        CE(2,3) =  (SMA1-0.5*SMA2)*(3.0*SMA)/DB
        CE(3,2) =  CE(2,3)
        CE(3,3) = (3.0*SMA)**2/DB + (2.0*(1.0+GNU))/E1
        RHS(1)= DE(NNN)
        RHS(2)= DE(NNK)
        RHS(3)= DE(NN)
        ARG6 =0.0
        M=IMEQF(3,3,1,CE,RHS,ARG6,ARG7)
```

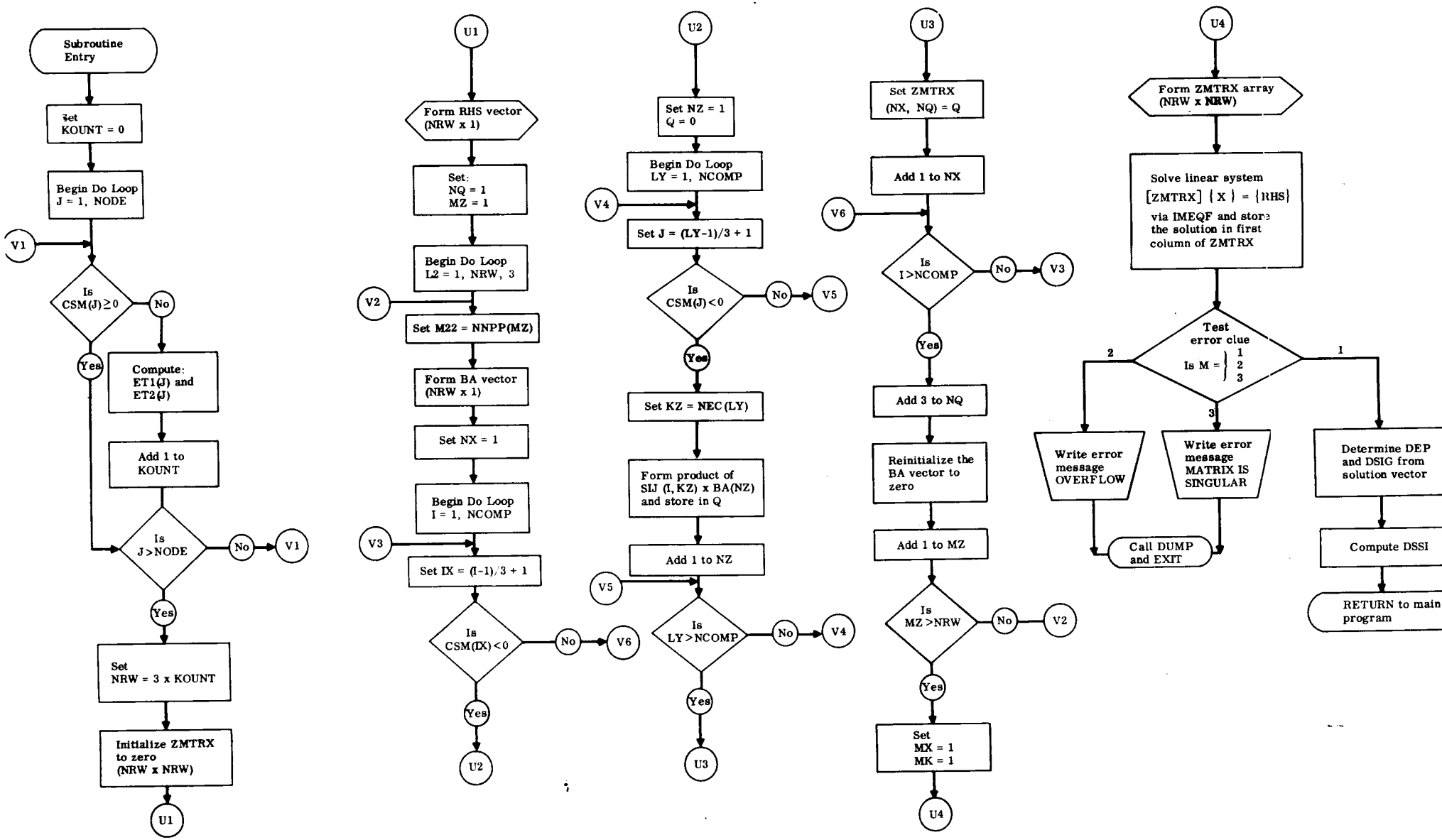
```
      GO TO (110,111,112),M
112   WRITE(6,113)
113   FORMAT(15H CE IS SINGULAR)
      CALL EXIT
111   WRITE(6,114)
114   FORMAT( 9H OVERFLOW)
      CALL EXIT
110   CONTINUE
C
C     COMPUTE INCREMENTS OF PLASTIC STRAIN, TRANSLATION, AND INITIAL
C                                STRESS
C
      DSIG(NNN)=CE(1,1)
      DSIG(NNK)=CE(2,1)
      DSIG(NN) =CE(3,1)
      ANUM = (SIGMA(NNN)-ALPHA(NNN))*(DSIG(NNN)-0.5*DSIG(NNK))+(SIGMA
     *       (NNK)-ALPHA(NNK))*(DSIG(NNK)-0.5*DSIG(NNN))+3.0*(SIGMA(NN)
     *       -ALPHA(NN))*(DSIG(NN))
      DLAM(I)=ANUM/ DB
      DMU(I)=ANUM/(((SIGMA(NNN)-ALPHA(NNN))**2)-((SIGMA(NNN)-ALPHA(NNN))
     *      *(SIGMA(NNK)-ALPHA(NNK)))+((SIGMA(NNK)-ALPHA(NNK))**2) +
     *      (3.0*(SIGMA(NN)-ALPHA(NN))**2))
      DEP(NNN)=((SIGMA(NNN)-ALPHA(NNN))-0.5*(SIGMA(NNK)-ALPHA(NNK))) *
     *         DLAM(I)
      DEP(NNK)=((SIGMA(NNK)-ALPHA(NNK))-0.5*(SIGMA(NNN)-ALPHA(NNN))) *
     *         DLAM(I)
      DEP(NN)=3.0*(SIGMA(NN)-ALPHA(NN))*DLAM(I)
      DALPH(NNN)=(SIGMA(NNN)-ALPHA(NNN))*DMU(I)
      DALPH(NNK)=(SIGMA(NNK)-ALPHA(NNK))*DMU(I)
      DALPH(NN)=(SIGMA(NN)-ALPHA(NN))*DMU(I)
      DSSI(NNN)=-GNU1*(DEP(NNN)+GNU*DEP(NNK))
      DSSI(NNK)=-GNU1*(DEP(NNK)+GNU*DEP(NNN))
      DSSI(NN) = -E1/(2.0*(1.0+GNU))*DEP(NN)
      ALPHA(NNN)=ALPHA(NNN)+DALPH(NNN)
      ALPHA(NNK)=ALPHA(NNK)+DALPH(NNK)
      ALPHA(NN)=ALPHA(NN)+DALPH(NN)
115   SSI(NNN) = SSI(NNN) + DSSI(NNN)
      SSI(NNK)=SSI(NNK)+DSSI(NNK)
      SSI(NN)=SSI(NN)+DSSI(NN)
      EP(NNN)=EP(NNN)+DEP(NNN)
      EP(NNK)=EP(NNK)+DEP(NNK)
      EP(NN)=EP(NN)+DEP(NN)
C
C  CO M PUTE THESE VARIABLES FOR ALL NODES
C
1101  SIGMA(NNN)=SIGMA(NNN)+DSIG(NNN)
      SIGMA(NNK)=SIGMA(NNK)+DSIG(NNK)
      SIGMA(NN)=SIGMA(NN)+DSIG(NN)
      ELAS(NNN)=(1.0/E1)*(SIGMA(NNN)-GNU*SIGMA(NNK))
      ELAS(NNK)=(1.0/E1)*(SIGMA(NNK)-GNU*SIGMA(NNN))
      ELAS(NN)=((2.0*(1.0+GNU))/E1)* SIGMA(NN)
100   CONTINUE
C
C  TES T   FOR  PERFECT  PLASTICITY
C
      IF(KLU)26,70,26
70    IF(ZFL)26,27,27
27    CONTINUE
      DO 117 I=1,NODE
      N1=3*I-2
```

```
      N2=3*I-1
      N3=3*I
      AL1=(ALPHA(N1)-ALBAR(N1))
      AL2=(ALPHA(N2)-ALBAR(N2))
      AL3=(ALPHA(N3)-ALBAR(N3))
      FLOW = AL1**2 - AL1*AL2 + AL2**2 + 3.0*AL3**2
      IF(ZFL-FLOW)118,118,117
  118 CSM(I)=-1.0
      IA=IA+1
      NNPP(IA)=I
  117 CONTINUE
   26 CONTINUE
      IF(P/RMXLD - 1.0)132,125,125
  125 CONTINUE
      KTRANS = 2
      GO TO 126
  132 IF(ICOUNT-INT)130,131,131
  130 ICOUNT = ICOUNT + 1
      GO TO 41
  131 CONTINUE
      ICOUNT = 1
  126 WRITE(6,2012) P
      WRITE(6,2000)
 2000 FORMAT(/ 21X,16HELEMENT STRESSES, /)
 2003 FORMAT(1H0,7HNODE NO,4X,7HSIGMA-X,12X,7HSIGMA-Y,13X,7HTAU-X,Y)
      WRITE(6,2003)
      DO 2001 I=1,NODE
      NNN=3*I-2
      NNK=3*I-1
      NN = 3*I
      WRITE(6,2002) I,SIGMA(NNN),SIGMA(NNK),SIGMA(NN)
 2002 FORMAT(3X,I3,2X,E16.8,3X,E16.8,3X,E16.8)
 2001 CONTINUE
      DO 2020 K=1,NODE
      IF(U(K))2021,2020,2021
 2021 NNN= 3*K-2
      NNK= 3*K-1
      NN = 3*K
      WRITE(6,2006)
      WRITE(6,2007) K,CSM(K),DLAM(K),DMU(K)
      WRITE(6,2008)
      WRITE(6,2009)   DALPH(NNN),DALPH(NNK),DALPH(NN),DEP(NNN),DEP(NNK),
     *  DEP(NN)
      WRITE(6,2050)
 2050 FORMAT( 1H0)
      WRITE(6,2010)
      WRITE(6,2009) ALPHA(NNN),ALPHA(NNK),ALPHA(NN),EP(NNN),EP(NNK),EP(N
     *  N)
      WRITE(6,2011)
      WRITE(6,2009) EE(NNN),EE(NNK),EE(NN),ELAS(NNN),ELAS(NNK),ELAS(NN)
 2006 FORMAT(1H0,10HELEMENT NO,19X,1HC,16X,4HDLAM,17X,3HDMU   )
 2007 FORMAT(1H0,I10,6E20.6)
 2008 FORMAT(1H0,16X,14H DELTA ALPHA-X,6X,14H DELTA ALPHA-Y,5X,15H DELTA
     *  ALPHA-XY,9X,11H DELTA EP-X, 9X,11H DELTA EP-Y,
     *  7X,13H DELTA GAM-XY)
 2009 FORMAT(10X,6E20.6)
 2010 FORMAT(18X,12H TOT ALPHA-X, 8X,12H TOT ALPHA-Y,7X,13H TOT ALPHA-XY
     *  ,7X,13H PLASTIC EP-X,7X,13H PLASTIC EP-Y,5X,15H PLASTIC GAM-XY)
 2011 FORMAT(1H0,21X, 9H STRAIN-X,10X, 9H STRAIN-Y,10X,10H STRAIN-XY, 7X
     *  ,13H ELASTIC EP-X,7X,13H ELASTIC EP-Y,5X,15H ELASTIC GAM-XY)
```

```
2012    FORMAT(1H0,3X,17H LOAD LEVEL, P = , F8.0 )
2020    CONTINUE
        GO TO(41,65),KTRANS
  65    IF(NRSTRT) 7,7,66
  66    WRITE (10)ALBAR,ALPHA,CSM,D,D1,DALPH,DE,DEP,DELP,DLAM,DMU,DSIG,
     *          DSSI,E,E1,EE,EI,ELAS,EP,G,GIM,GNU,GNU1,IAVAIL,INT,IREC,
     *          KLU,L,MCOL,NCOMP,NEC,NNPP,NODE,NPND,P,PCRIT,REST,SIGO,
     *          SIGMA,SIJ,SIM,SMALL,SQ,SSI,T,U,Z,ZFL
        REWIND 10
        GO TO 9
 657    WRITE(6,2030)
2030    FORMAT( 42H NUMBER OF COLUMNS EXCEEDS AVAILABLE SPACE )
   9    REWIND 9
        PAUSE 2
        CALL EXIT
        END
```

```
        SUBROUTINE SETAPE(NUNIT,NFILE)
        DATA SIGNAL /O111111111111 /
        REWIND 9
        NSET=NFILE-1
        IF(NSET)3,3,4
   4    DO 1 I=1,NSET
   2    READ (9) X
        IF(X-SIGNAL)2, 1,2
   1    CONTINUE
   3    RETURN
        END
```

STRAPP PROGRAM
Subroutine PERTIC



SUBROUTINE PERTIC

```
$ IBFT C     PERTIC  LIST,REF
            SUBROUTINE PERTIC
            COMMON ALPHA(165)
            COMMON CSM( 55)
            COMMON DELP
            COMMON DEP(165)
            COMMON DSIG  (165)
            COMMON DSSI(165)
            COMMON E(3,3)
            COMMON EI(3,3)
            COMMON KLU
            COMMON MCOL
            COMMON NCOMP
            COMMON NEC(165)
            COMMON NNPP(57)
            COMMON NODE
            COMMON SIGMA(165)
            COMMON SIJ(165,57)
            COMMON SIM(165)
            DIMENSION ARG7(57)
            DIMENSION BA(57)
            DIMENSION ET1(55)
            DIMENSION ET2(55)
            DIMENSION RHS(57)
            DIMENSION ZMTRX(57,57)
            KOUNT=0
            DO 1 J=1,NODE
            N1=3*J-2
            N2=3*J-1
            N3=3*J
            IF(CSM(J))2,1,1
    2       SB1=(SIGMA(N1)-ALPHA(N1))-(SIGMA(N2)-ALPHA(N2))/2.
            SB2=(SIGMA(N2)-ALPHA(N2))-(SIGMA(N1)-ALPHA(N1))/2.
            SB3=3.*(SIGMA(N3)-ALPHA(N3))
            ET1(J)=SB2/SB1
            ET2(J)=SB3/SB1
            KOUNT=KOUNT+1
    1       CONTINUE
            NRW=3*KOUNT
            DO 3 I=1,NRW
            RHS(I)=0.0
            DO 3 J=1,NRW
    3       ZMTRX(I,J)=0.
C           FORMATION OF RIGHT HAND SIDE
            NY=1
            IF(KLU) 30,31,30
   31       CONTINUE
            DO 4 I=1,NCOMP
            J= (I-1)/3+1
            IF(CSM(J))5,4,4
    5       Q=0.
            DO 6 IY=1,NCOMP
            JY= (IY-1)/3+1
            IF(CSM(JY))6,6,7
    7       KY=NEC(IY)
            Q=Q+SIJ(I,KY)*DSSI(IY)
    6       CONTINUE
            RHS(NY)=Q
            NY=NY+1
    4       CONTINUE
```

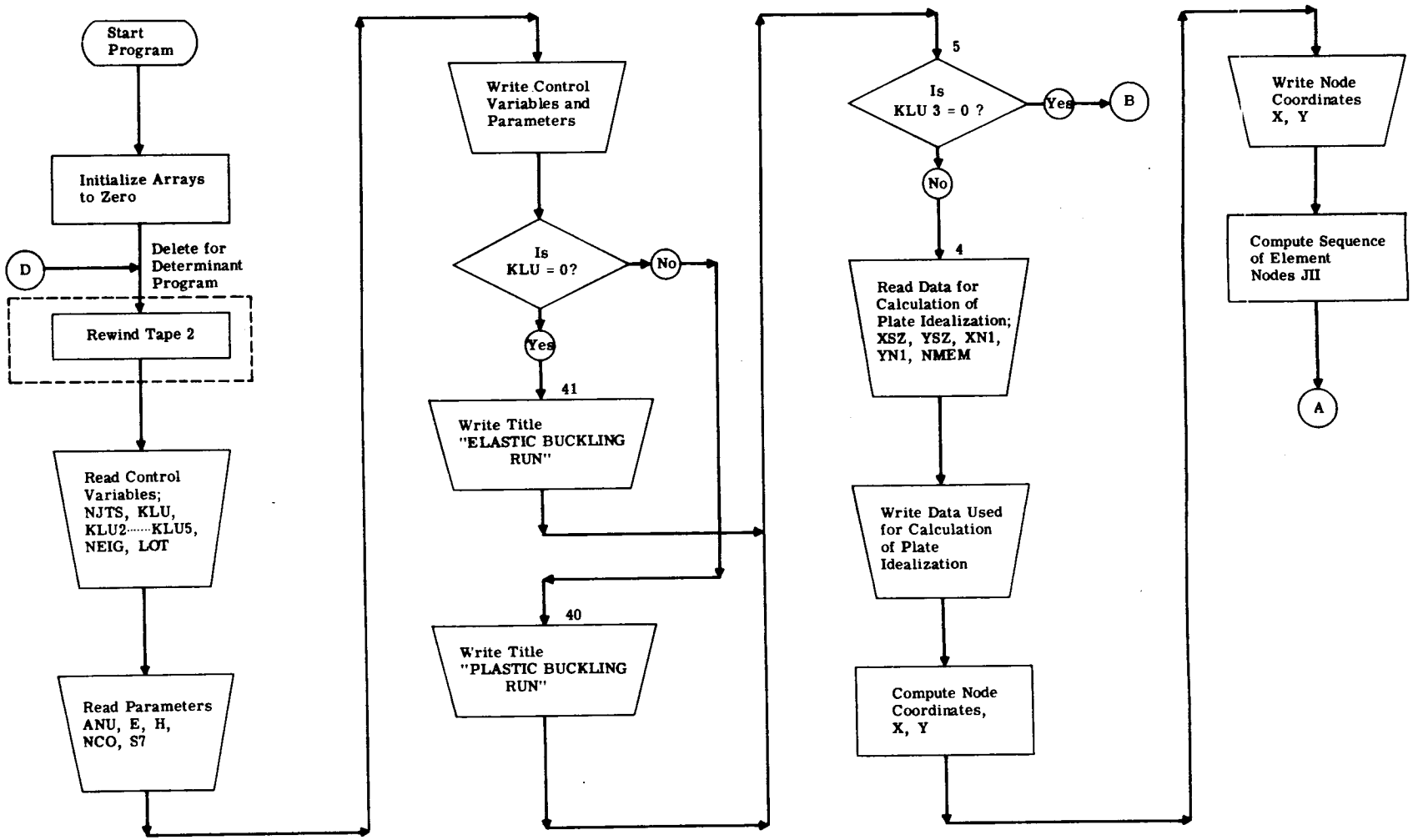
```
30      CONTINUE
        MK=1
        DO 8 J=1,NODE
        IF (CSM(J))9,8,8
 9      N1=3*J-2
        N2=3*J-1
        N3=3*J
        RHS(MK)=RHS(MK)+SIM(N1)*DELP
        RHS(MK+1)=RHS(MK+1)+SIM(N2)*DELP
        RHS(MK+2)=RHS(MK+2)+SIM(N3)*DELP
        MK=MK+3
 8      CONTINUE
C       FORMATION OF COEFFICIENT MATRIX
        NQ=1
        MZ=1
        DO 10 L2=1,NRW,3
        M22=NNPP(MZ)
        BA(L2)=E(1,1)+E(1,2)*ET1(M22)
        BA(L2+1)=E(2,1)+E(2,2)*ET1(M22)
        BA(L2+2)=E(3,3)*ET2(M22)
        NX=1
        DO 11 I=1,NCOMP
        IX=(I-1)/3+1
        IF(CSM(IX))12,11,11
 12     NZ=1
        Q=0.
        DO 14 LY=1,NCOMP
        J=(LY-1)/3+1
        IF(CSM(J))15,14,14
 15     KZ=NEC(LY)
        Q=Q+SIJ(I,KZ)*BA(NZ)
        NZ=NZ+1
 14     CONTINUE
        ZMTRX(NX,NQ)=Q
        NX=NX+1
 11     CONTINUE
        NQ=NQ+3
        DO 16 IK=1,NRW
 16     BA(IK)=0.
 10     MZ=MZ+1
        MX=1
        MK=1
        DO 17 I=1,NRW,3
        M22=NNPP(MX)
        ZMTRX(MK,I)=ZMTRX(MK,I)+1.
        ZMTRX(MK+1,I)=ZMTRX(MK+1,I)+ET1(M22)
        ZMTRX(MK+2,I)=ZMTRX(MK+2,I)+ET2(M22)
        MK=MK+3
 17     MX=MX+1
        MB=1
        DO 18 I=1,NRW,3
        MQ=NNPP(MB)
        I1=I+1
        I2=I+2
        ZMTRX(I,I1)=-EI(1,1)*ET1(MQ)+EI(1,2)
        ZMTRX(I,I2)=-EI(1,1)*ET2(MQ)
        ZMTRX(I1,I1)=-EI(2,1)*ET1(MQ)+EI(2,2)
        ZMTRX(I1,I2)=-EI(2,1)*ET2(MQ)
        ZMTRX(I2,I1)=0.0
        ZMTRX(I2,I2)=EI(3,3)
```

```
 18     MB=MB+1
C
C SOL U TION OF THE SIMULTANEOUS LINEAR EQUATIONS
C
        ARG6 =0.0
        M=IMEQF(MCOL,NRW,1,ZMTRX,RHS,ARG6,ARG7)
        GO TO (21,20,19),M
 19     WRITE(6,22)
 22     FORMAT(19H MATRIX IS SINGULAR )
        CALL DUMP
 20     WRITE(6,23)
 23     FORMAT( 9H OVERFLOW )
        CALL DUMP
 21     MB=1
        DO 24 I=1,NODE
        IF(CSM(I))25,24,24
 25     N1=3*I-2
        N2=3*I-1
        N3=3*I
        DEP(N1)=ZMTRX(MB,1)
        DEP(N2)=ET1(I)*DEP(N1)
        DEP(N3)=ET2(I)*DEP(N1)
        DSIG  (N2)=ZMTRX(MB+1,1)
        DSIG  (N3)=ZMTRX(MB+2,1)
        DSIG  (N1)=-ET1(I)*DSIG (N2)-ET2(I)*DSIG  (N3)
        DSSI(N1)=-(E(1,1)*DEP(N1)+E(1,2)*DEP(N2))
        DSSI(N2)=-(E(2,1)*DEP(N1)+E(2,2)*DEP(N2))
        DSSI(N3)=-E(3,3)*DEP(N3)
        MB=MB+3
 24     CONTINUE
        RETURN
        END
```

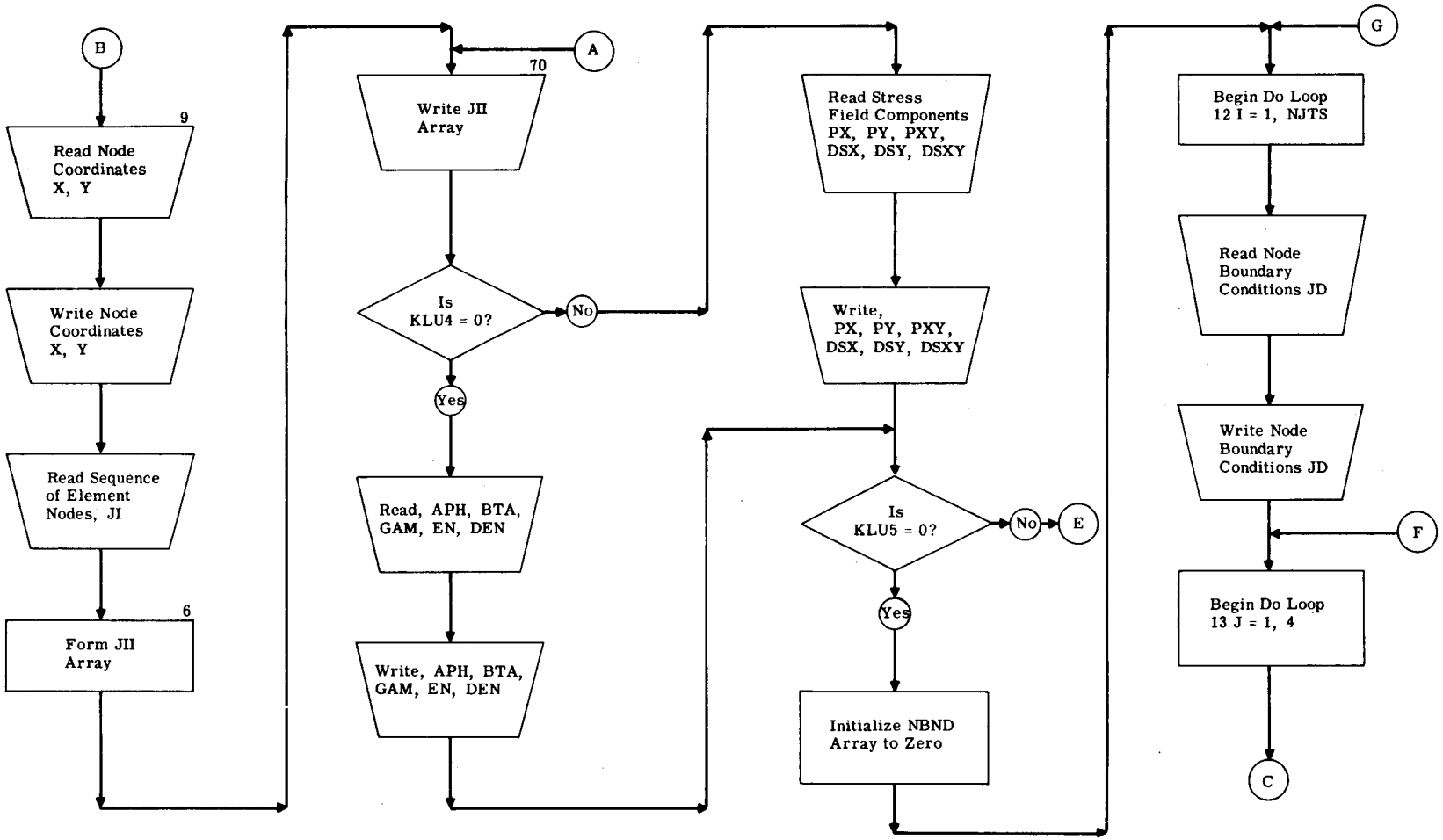
APPENDIX B
FLOW CHARTS AND LISTINGS FOR PLASTIC BUCKLING OF RECTANGULAR PLATES

FLOW CHARTS

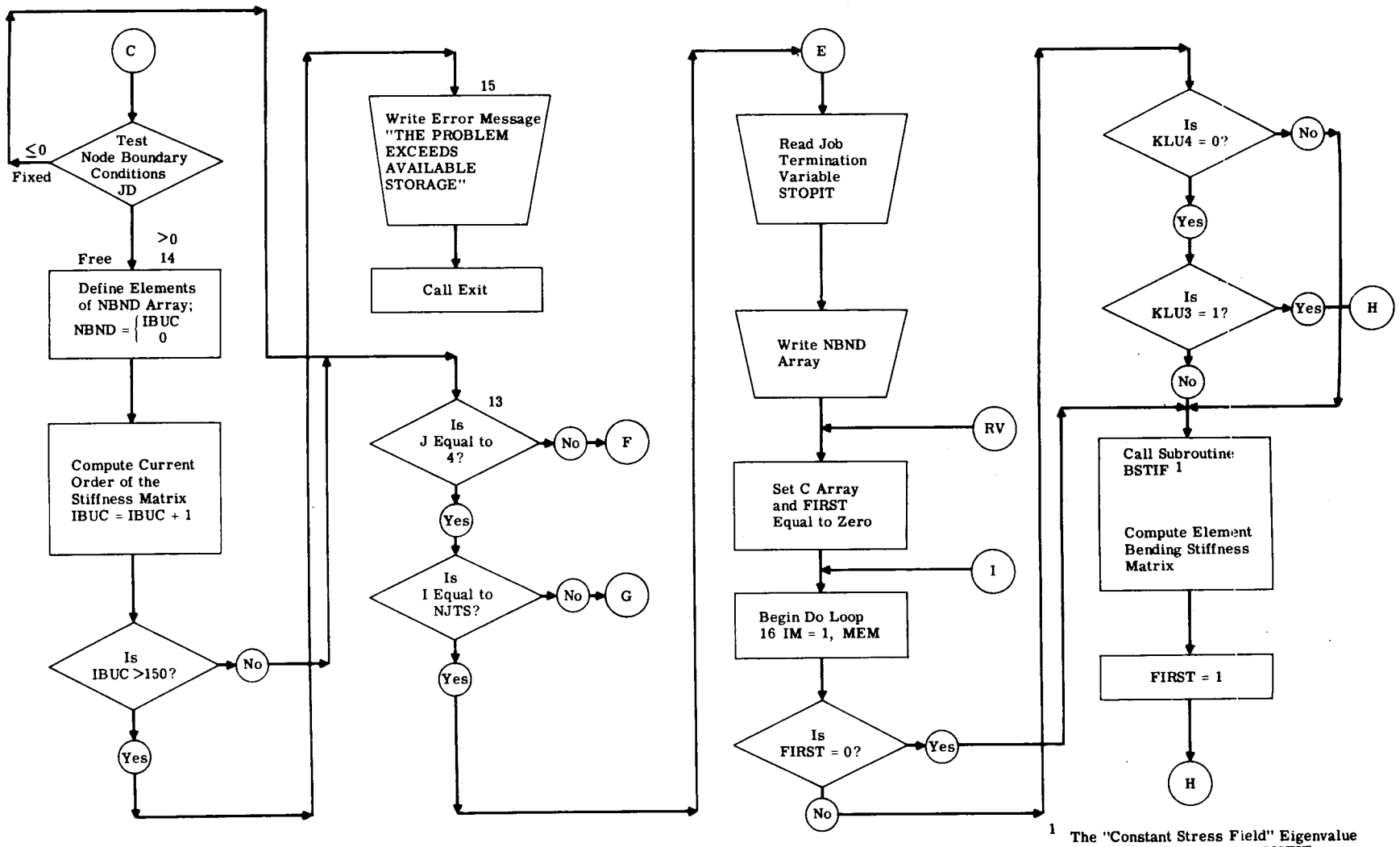
Eigenvalue and Determinant Program



Eigenvalue and Determinant Program

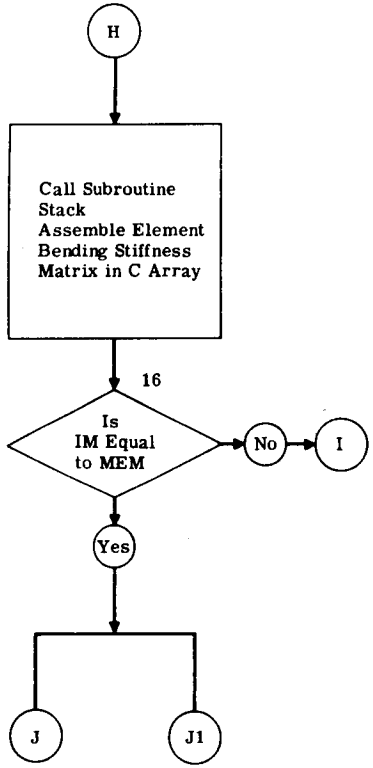


Eigenvalue and Determinant Program



1 The "Constant Stress Field" Eigenvalue Program Calls Subroutine MSTIF at this Location

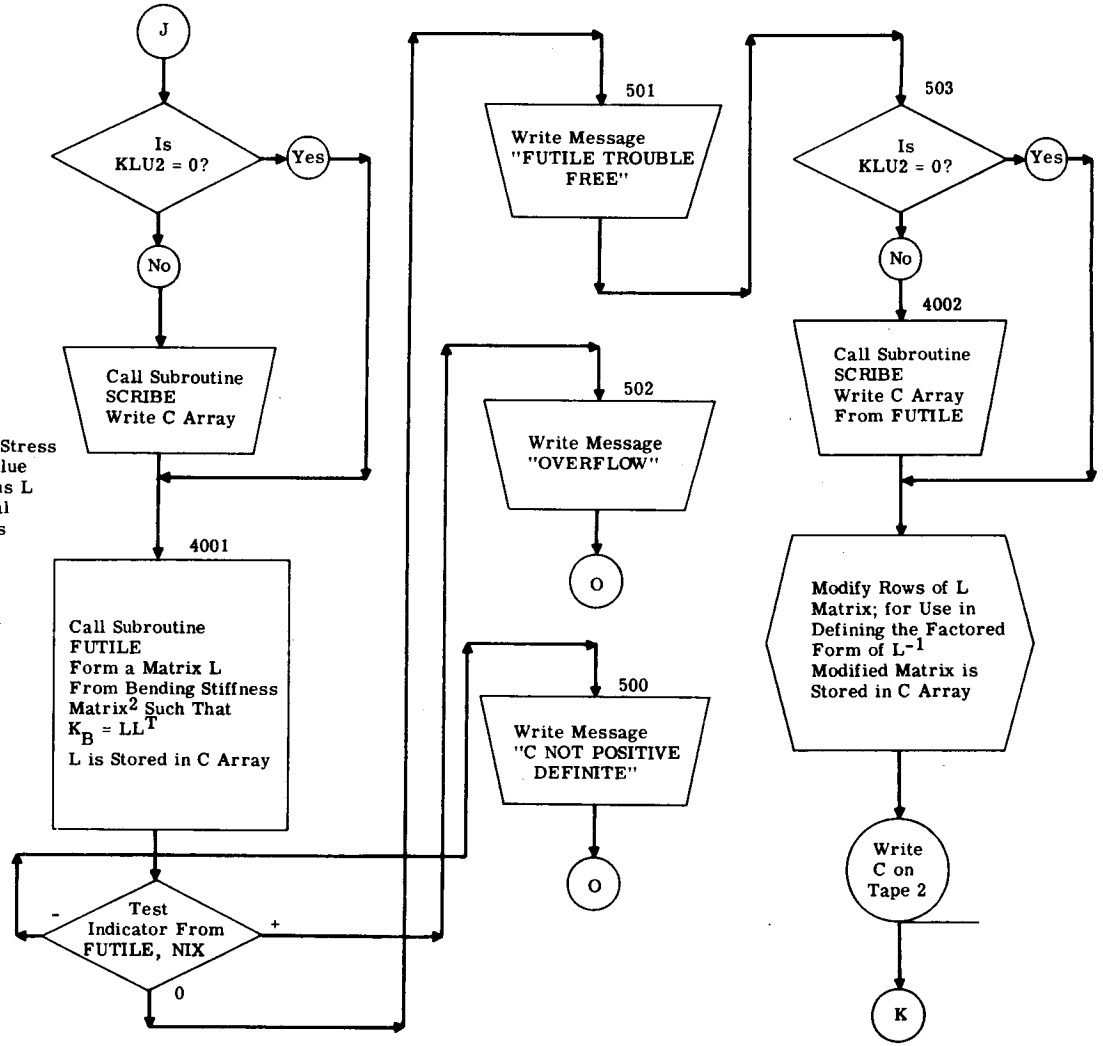
Eigenvalue and Determinant Program



Statement (J) Proceeds with the Eigenvalue Solution

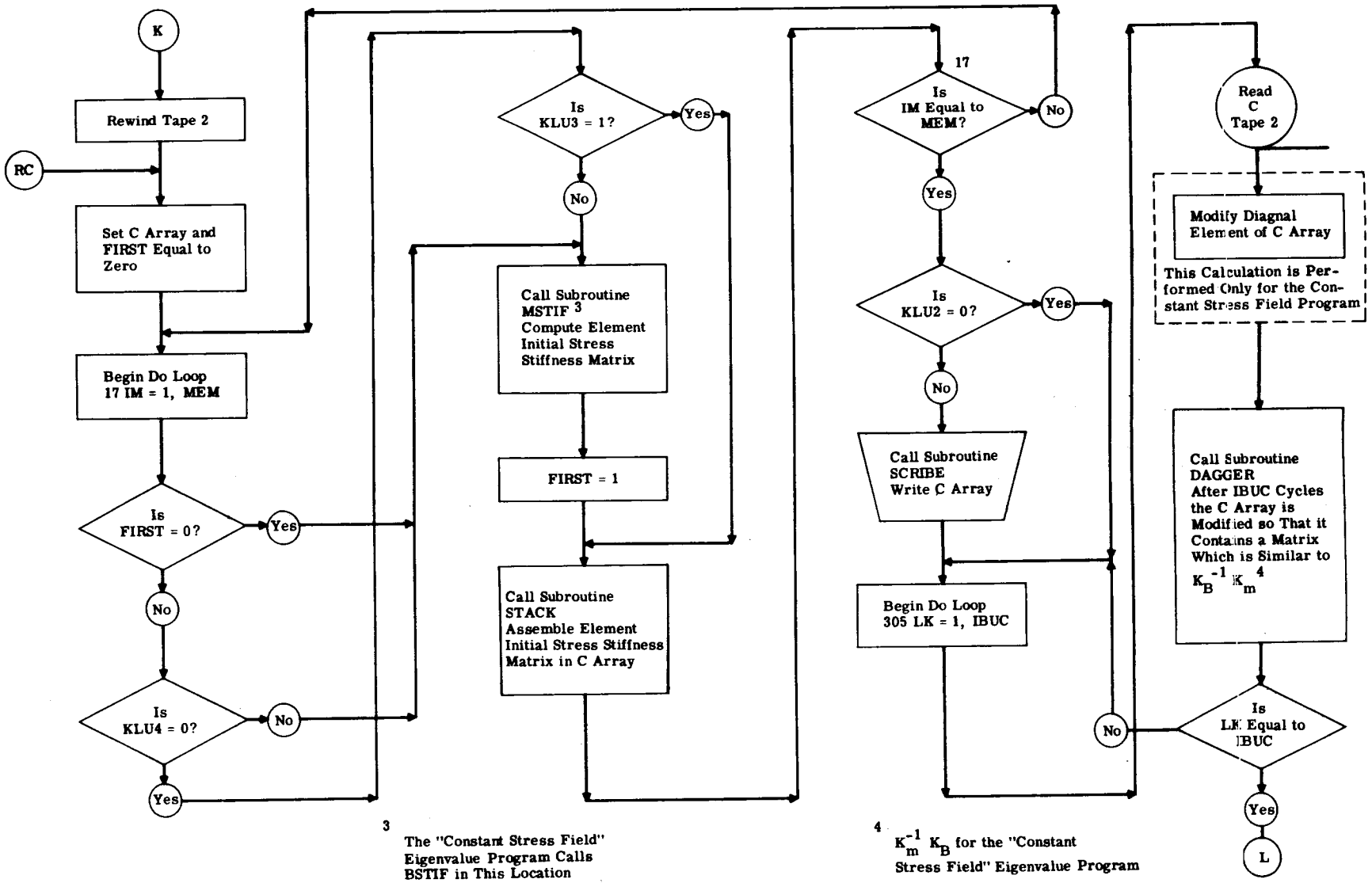
Statement (J1) Proceeds with the Determinant Solution on Page

Eigenvalue Program



[2] The "Constant Stress Field" Eigenvalue Program Forms L From the Initial Stress Stiffness Matrix, K_m

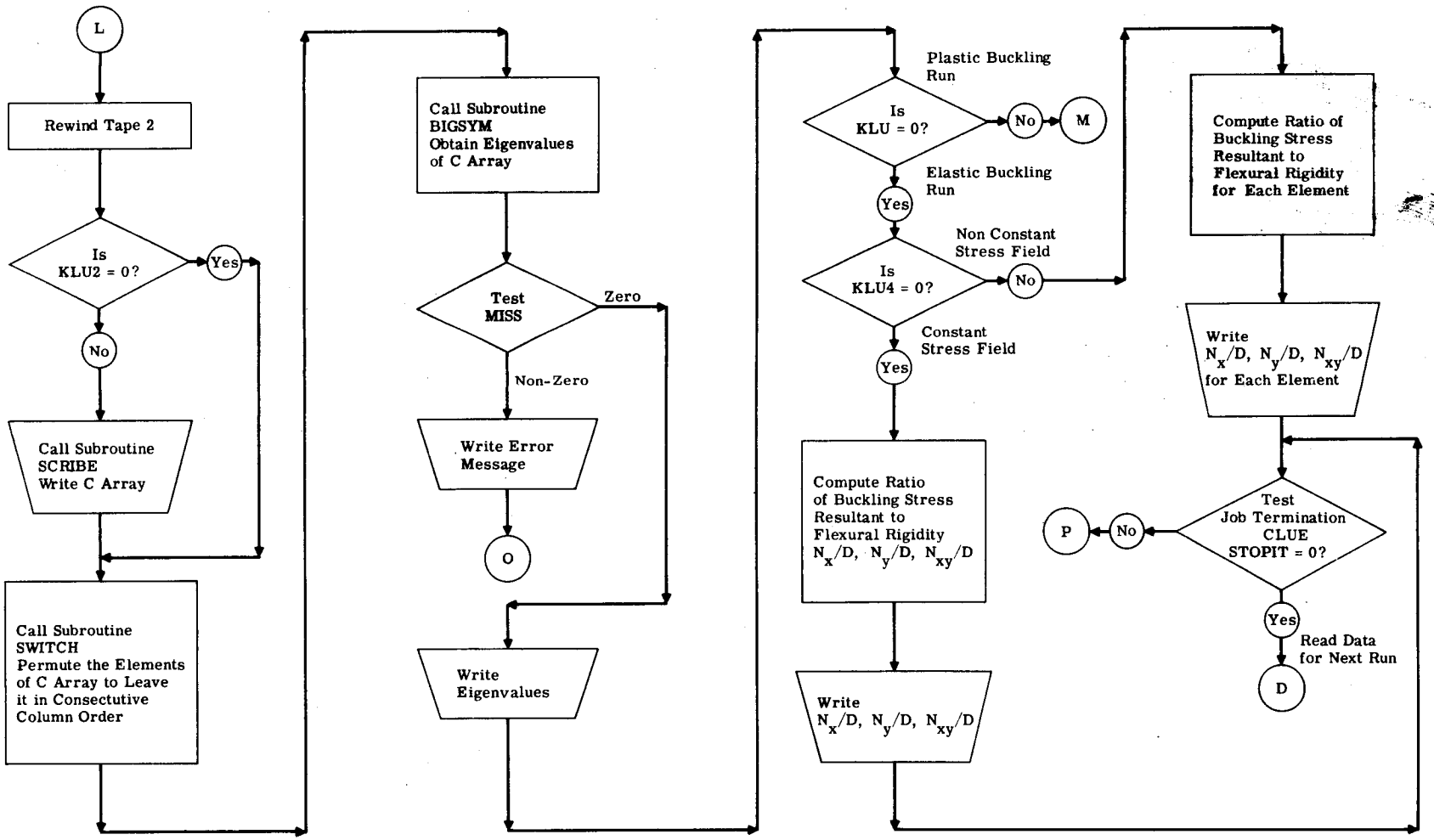
Eigenvalue Program



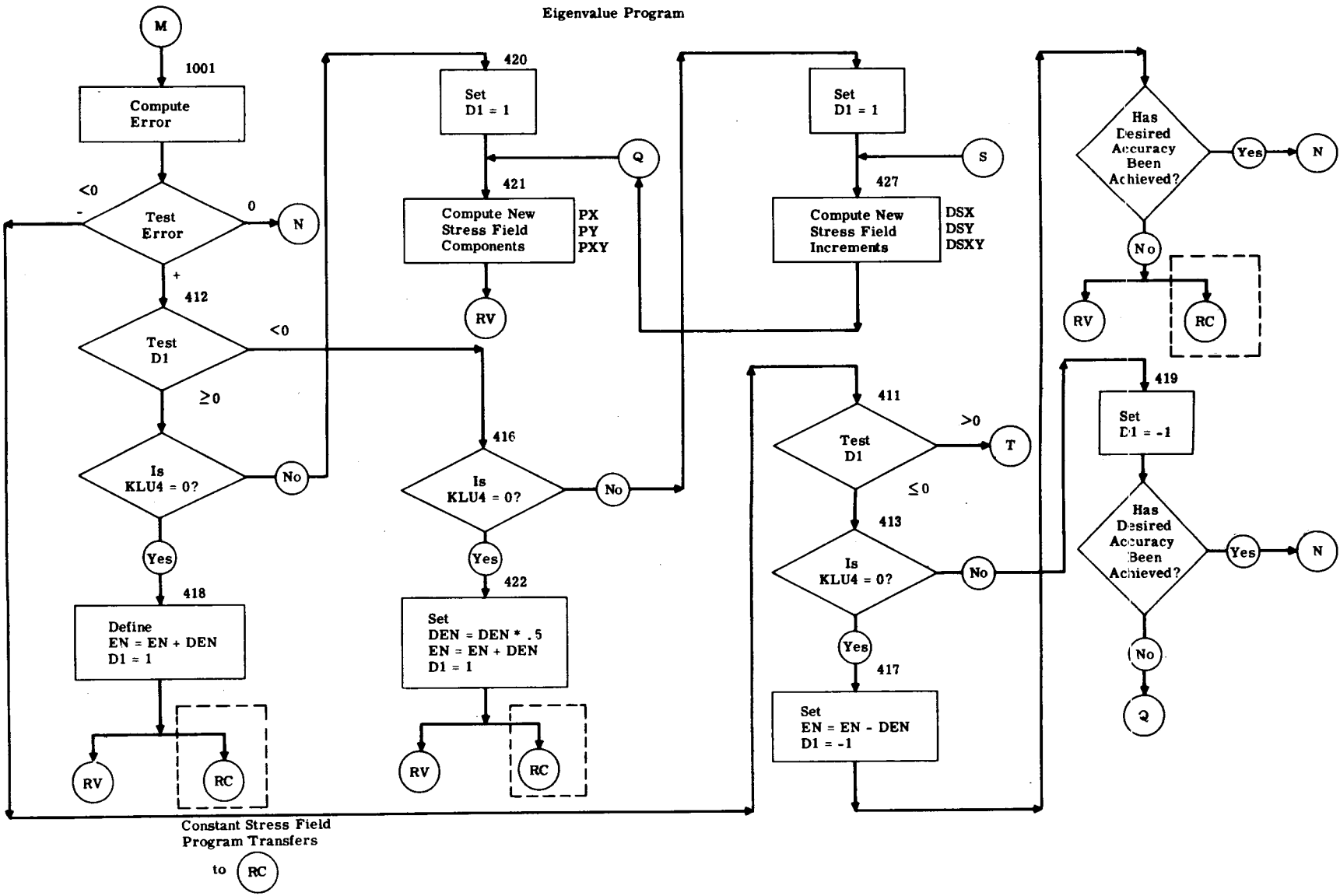
3 The "Constant Stress Field" Eigenvalue Program Calls BSTIF in This Location

4 $K_m^{-1} K_B$ for the "Constant Stress Field" Eigenvalue Program

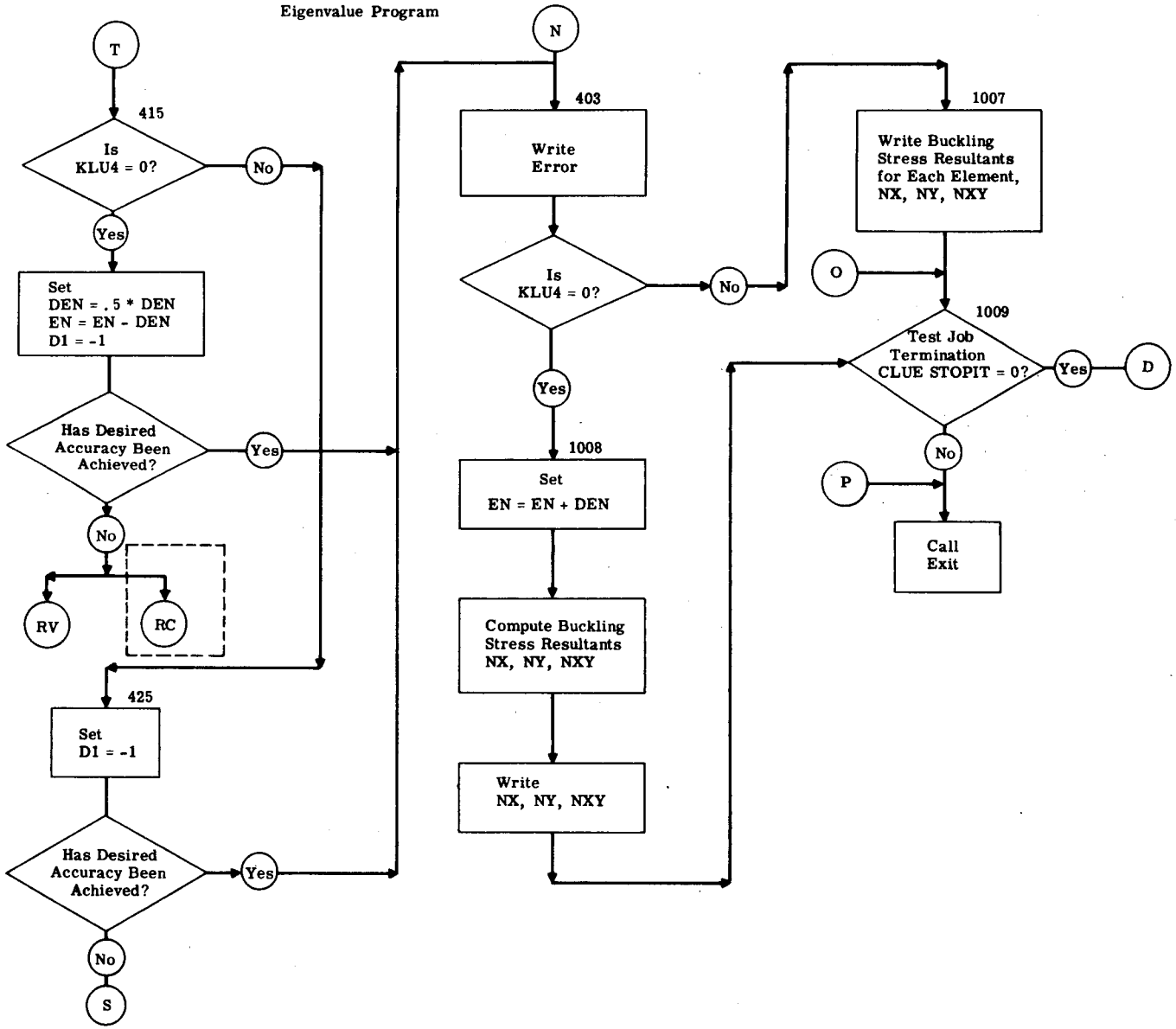
Eigenvalue Program



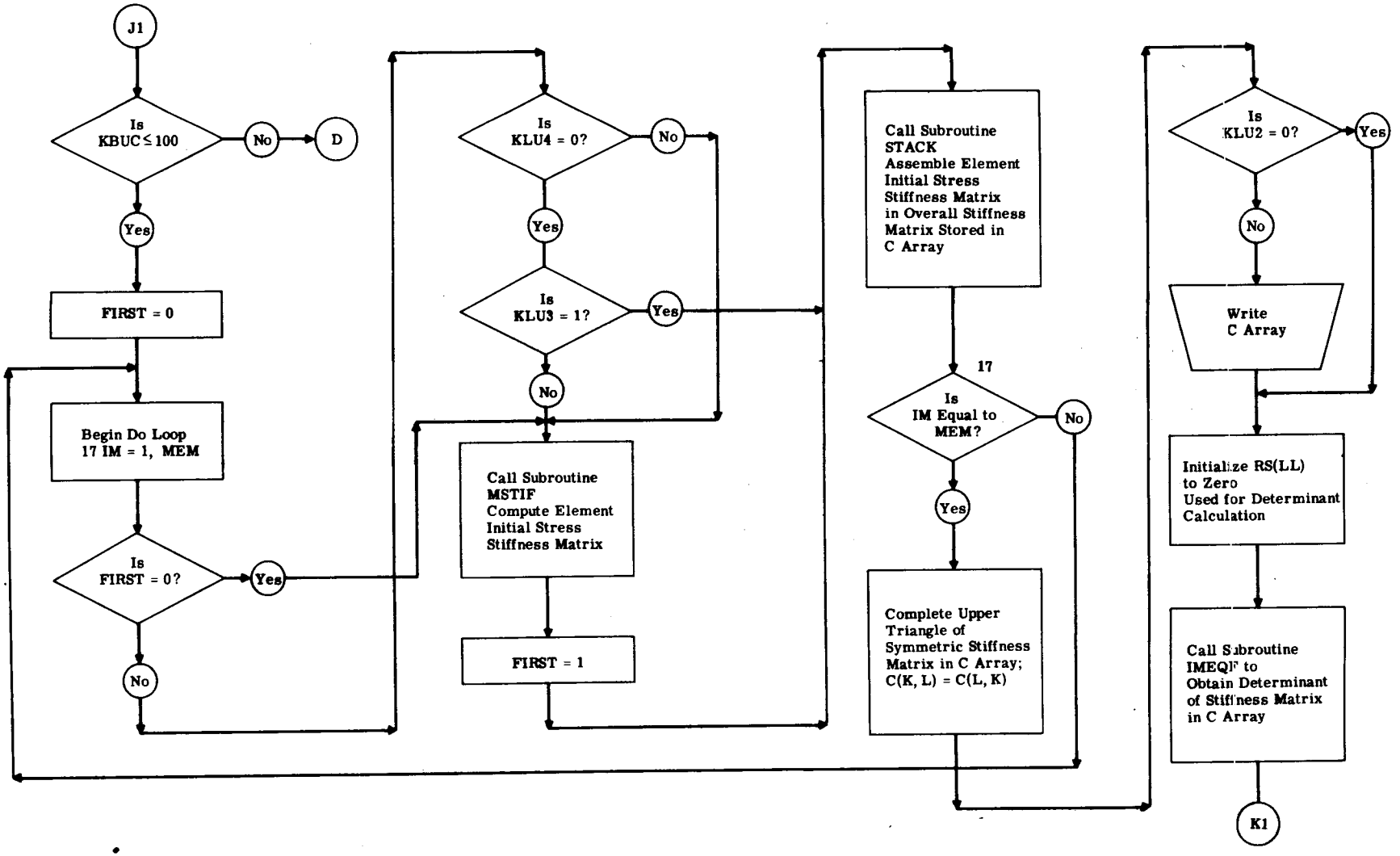
Eigenvalue Program



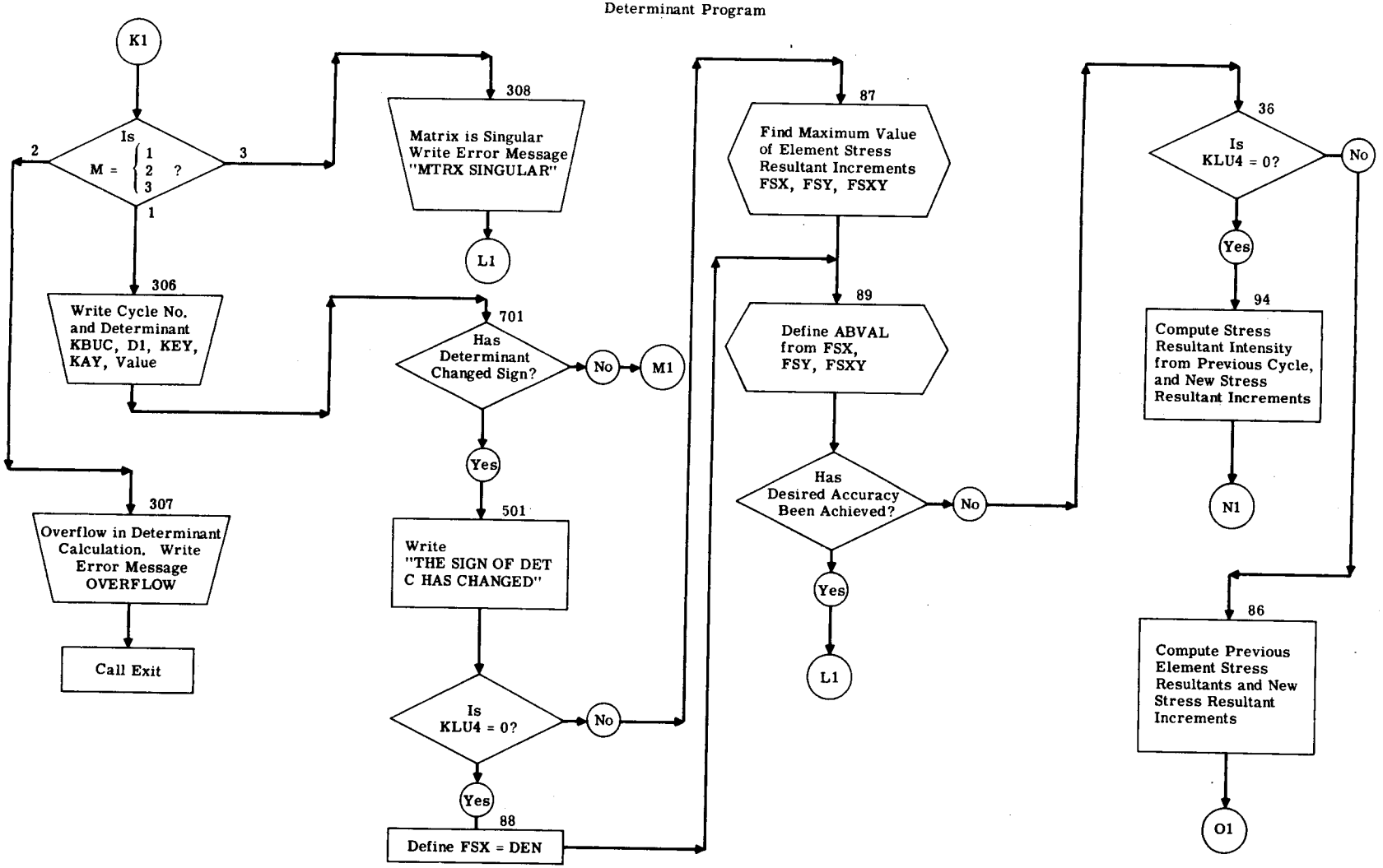
Eigenvalue Program

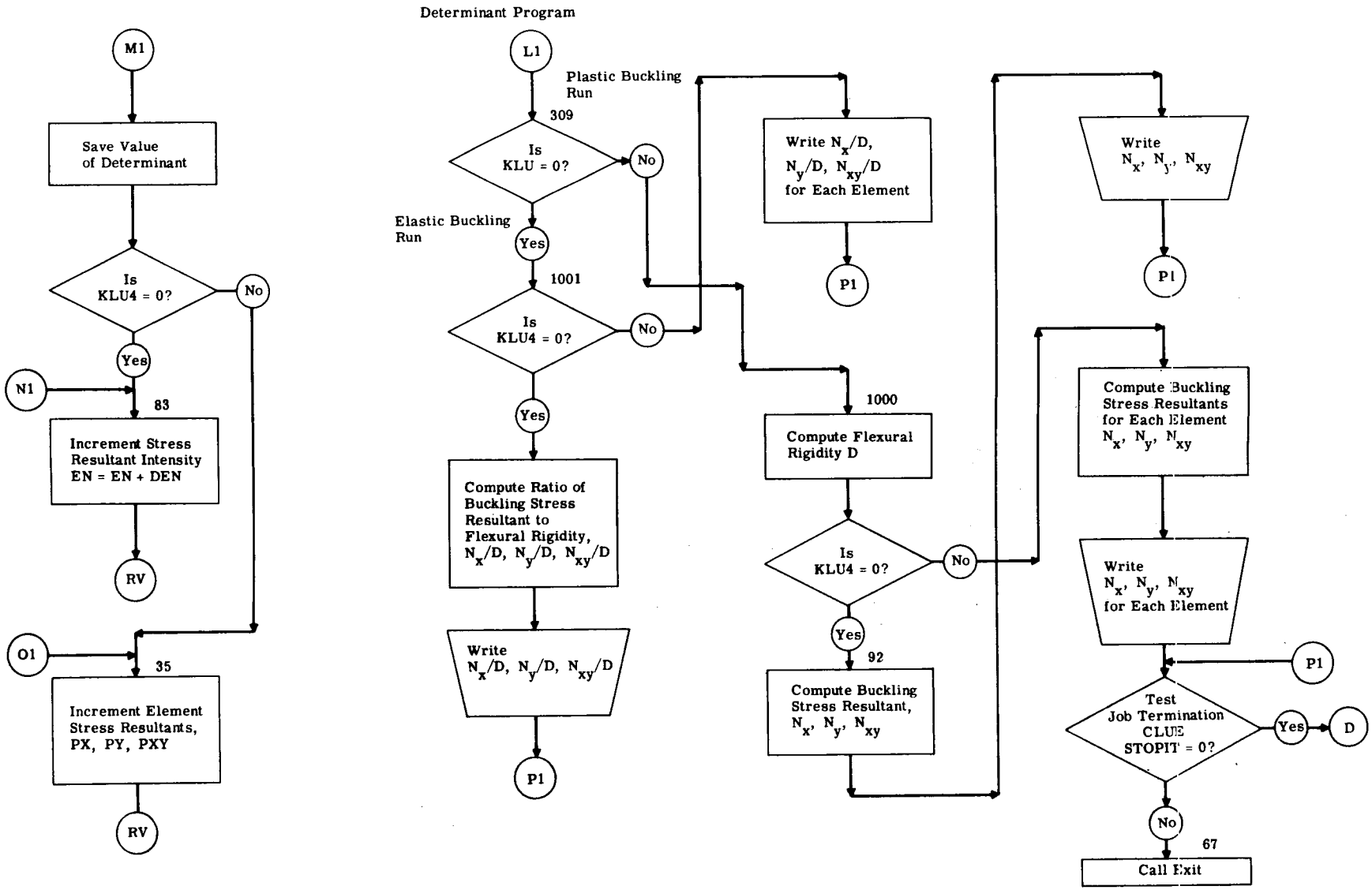


Determinant Program



Determinant Program




Determinant Program
M1
Save Value of Determinant
Is KLU4 = 0?
No
Yes
N1
83
Increment Stress Resultant Intensity EN = EN + DEN
RV
O1
35
Increment Element Stress Resultants, PX, PY, PXY
RV
L1
Plastic Buckling Run
309
Is KLU = 0?
No
Elastic Buckling Run
Yes
1001
Is KLU4 = 0?
No
Yes
Compute Ratio of Buckling Stress Resultant to Flexural Rigidity, N_x/D, N_y/D, N_{xy}/D
Write N_x/D, N_y/D, N_{xy}/D
P1
Write N_x/D, N_y/D, N_{xy}/D for Each Element
P1
1000
Compute Flexural Rigidity D
Is KLU4 = 0?
No
Yes
92
Compute Buckling Stress Resultant, N_x, N_y, N_{xy}
Write N_x, N_y, N_{xy}
P1
Compute Buckling Stress Resultants for Each Element N_x, N_y, N_{xy}
Write N_x, N_y, N_{xy} for Each Element
P1
Test Job Termination CLUE STOPIT = 0?
Yes
D
No
67
Call Exit

LISTING FOR EIGENVALUE PROGRAMS

```
C     BUCKLING OF RECTANGULAR PLATES        GENERAL EIGENVALUE PROGRAM
      COMMON JI,JT,X,Y,AD,BD,JD,LOCOL,C,AA
      COMMON JII,NBND,G,AKG,IM,NJTS,ANU,LIN
      COMMON E,H,ES,ET,KLU,APH,BTA,GAM,EN
      COMMON DEN,NCO,S7,IBUC,KLU4
      COMMON PX,PY,PXY,DSX,DSY,DSXY,SMAX
      DIMENSION JI(4),JT(81),X(81),Y(81),JII(64,4)
      DIMENSION NBND(81,4),LOCOL(16),G(16,16)
      DIMENSION C(11325),AA(4,3),AD(3),BD(3)
      DIMENSION JD(4),AKG(16,16),U(150),V(150)
      DIMENSION CA(150),BX(150),W(150),Q(150)
      DIMENSION PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      DOUBLE PRECISION P(150)
      DO 42 I=1,81
      X(I)=0.
      Y(I)=0.
   42 JT(I)=0
      DO 43 J=1,4
      JI(J)=0
   43 JD(J)=0
      DO 44 K=1,3
      AD(K)=0.
   44 BD(K)=0.
      DO 45 L=1,16
      LOCOL(L)=0
      DO 45 M=1,16
      G(L,M)=0.
   45 AKG(L,M)=0.
      DO 46 N=1,4
      DO 46 II=1,64
   46 JII(II,N)=0
      DO 47 JJ=1,4
      DO 47 KK=1,81
   47 NBND(KK,JJ)=0
      DO 48 LL=1,150
      U(LL)=0.
      V(LL)=0.
      CA(LL)=0.
      BX(LL)=0.
      W(LL)=0.
   48 Q(LL)=0.
      DO 49 N=1,50
      DSX(N)=0.
      DSY(N)=0.
      DSXY(N)=0.
      PX(N)=0.
      PY(N)=0.
   49 PXY(N)=0.
 2000 REWIND 2
      D1=0.
      BK=1.
      READ (5,99) NJTS,KLU,KLU2,KLU3,KLU4,KLU5,NEIG,LOT

      IF(KLU )40,41,40
```

```
41 WRITE (6,208)
   GO TO 5
40 WRITE(6,209)
 5 TOL=1. * 10.**(-LOT)
   READ(5,98) ANU,E,H,NCO,S7
   WRITE(6,27) KLU,KLU2,KLU3,KLU4,KLU5,NEIG,LOT
   WRITE(6,28) ANU,E,H,NCO,S7
   IF(KLU3)4,9,4
 4 READ (5,60) XSZ,YSZ,XN1,YN1,NMEM
   WRITE(6,61) XSZ,YSZ,XN1,YN1,NJTS,NMEM
   XN2=XN1
   YN2=YN1
   DO 1 I=1,NJTS
   X(I)=((XN1-XN2)/(XN1))*XSZ
   Y(I)=((YN1-YN2)/(YN1))*YSZ
   JT(I)=I
   XN2=XN2-1.
   IF(XN2)3,1,1
 3 XN2=XN1
   YN2=YN2-1.
 1 CONTINUE
   WRITE(6,62)
   WRITE(6,63) (JT(I),X(I),Y(I),I=1,NJTS)
   NN1=XN1
   NN2=1
   NN3=XN1+2.
   NN4=XN1
   DO 64 MEM=1,NMEM
   JII(MEM,1)=NN2
   JII(MEM,2)=NN2+1
   JII(MEM,3)=NN3
   JII(MEM,4)=NN3+1
   NN2=NN2+1
   NN3=NN3+1
   NEM=MEM-NN4
   IF(NEM)64,65,65
65 NN2=NN2+1
   NN3=NN3+1
   NN4=NN1+MEM
64 CONTINUE
   GO TO 70
 9 READ(5,100) (JT(I),X(I),Y(I),I=1,NJTS)
   WRITE (6,200)
   WRITE (6,201) (JT(I),X(I),Y(I),I=1,NJTS)
 8 READ(5,101) MEM,NTYPE,JI(1),JI(2),JI(3),JI(4)
   IF(MEM-99) 6,7,6
 6 DO 10 I=1,4
   JII(MEM,I)=JI(I)
10 CONTINUE
   KODE=MEM
   GO TO 8
 7 CONTINUE
   MEM=KODE
70 WRITE(6,66)  ((JII(K,N),N=1,4),K=1, MEM)

   IF(KLU4) 5050,5051,5050
```

```
 5051 READ (5,103) APH,BTA,GAM,EN,DEN
      WRITE(6,109) APH,BTA,GAM,EN,DEN
      GO TO 5052
 5050 WRITE (6,96)
      DO 71 MEM=1,NMEM
      READ(5,103) PX(MEM),DSX(MEM),PY(MEM),DSY(MEM),PXY(MEM),DSXY(MEM)
   71 WRITE(6,97) PX(MEM),DSX(MEM),PY(MEM),DSY(MEM),PXY(MEM),DSXY(MEM)
 5052 IF(KLU5)5060,5061,5060
 5061 IBUC=1
      DO 11 I=1,NJTS
      DO 11 J=1,4
   11 NBND(I,J)=0
      WRITE(6,207)
      DO 12 I=1,NJTS
      READ(5,102) JOINT,JD(1),JD(2),JD(3),JD(4)
      WRITE(6,203) JOINT,JD(1),JD(2),JD(3),JD(4)
      DO 13 J=1,4
      IF(JD(J)) 13,13,14
   14 NBND(JOINT,J)=IBUC
      IBUC=IBUC+1
      IF(IBUC-150)13,13,15
   15 WRITE(6,204)
      CALL EXIT
   13 CONTINUE
   12 CONTINUE
      IBUC=IBUC-1
 5060 READ(5,550) STOPIT
      LBUC=((IBUC+1)*IBUC)/2
      IF(KLU2) 504,402,504
  504 WRITE(6,23)
   23 FORMAT(1H0,14HTHE NBND ARRAY)
      WRITE(6,24)(NBND(I,1),NBND(I,2),NBND(I,3),NBND(I,4),I=1,JOINT)
  402 DO 896 KK=1,LBUC
  896 C(KK)=0.
      FIRST=0.
      DO 16 IM=1,MEM
      IM=IM
      LIN=4
      IF(FIRST) 551,552,551
  551 IF(KLU4) 552,553,552
  553 IF(KLU3-1) 552,554,552
  552 CALL BSTIF
      FIRST=1.
  554 CALL STACK
   16 CONTINUE
      IF(KLU2)4000,4001,4000
 4000 WRITE(6,703)
      CALL SCRIBE
 4001 CALL FUTILE (C,IBUC,NIX)
C
C     FORMATION OF L INVERSE
C
      IF(NIX)500,501,502
  501 WRITE(6,600)
      GO TO 503
  500 WRITE(6,601)
```

```
      GO TO 1009
  502 WRITE(6,602)
      GO TO 1009
  503 IF(KLU2)4002,4003,4002
 4002 WRITE(6,57)
      CALL SCRIBE
 4003 IK=1
      IQ=0
      KKK=IBUC+1
      DO 300 IL=2,KKK
      IF(IK-1)301,301,302
  302 IP=IK-1
      DO 303 J=1,IQ
      C(IP)=-1.*C(IP)/C(IK)
  303 IP=IP-1
  301 C(IK)=1./C(IK)
      IK=IK+IL
  300 IQ=IQ+1
C
C     WRITE L INVERSE ON TAPE 2
C
      KZ=1
      KQ=1
      DO 304 K=1,IBUC
      WRITE (2) (C(KP), KP=KQ,KZ)
      KQ=KQ+K
  304 KZ=KQ+K
      REWIND 2
      DO 897 KK=1,LBUC
  897 C(KK)=0.
      FIRST=0.
      DO 17 IM=1,MEM
      IM=IM
      LIN=4
      IF(FIRST) 555,558,555
  555 IF(KLU4) 556,557,556
  557 IF(KLU3-1) 556,558,556
  556 CALL MSTIF
      FIRST=1.
  558 CALL STACK
   17 CONTINUE
      IF(KLU2)4006,4007,4006
 4006 WRITE(6,704)
      CALL SCRIBE
 4007 DO 305 LK=1,IBUC
      READ (2) (CA(KQ),KQ=1,LK)
      CALL DAGGER(C,IBUC,CA,LK,W)
  305 CONTINUE
      REWIND 2
      IF(KLU2)4008,4009,4008
 4008 WRITE(6,702)
  702 FORMAT(1H0,14HC AFTER DAGGER)
      CALL SCRIBE
 4009 CALL SWITCH(C,IBUC)
      M=11325
      CALL BIGSYM (C,IBUC,NEIG,M,BX,P,Q,U,V,MISS)
```

```
      IF(MISS)306,307,306
  306 WRITE(6,308)
      GO TO 1009
  307 WRITE(6,310) (BX(I),I=1,NEIG)
  310 FORMAT(13H0 EIGENVALUES//(1P6E20.7))
  309 IF(KLU)1000,1000,1001
 1000 WRITE(6,104)
      IF(KLU4)1004,1005,1004
 1005 EXOO=(1./BX(1))*APH
      EYOO=(1./BX(1))*BTA
      EXYOO=(1./BX(1))*GAM
      WRITE(6,105) EXOO,EYOO,EXYOO
      GO TO 1009
 1004 DO 5040 II=1,NMEM
      PX(II)=PX(II)/BX(1)
      PY(II)=PY(II)/BX(1)
 5040 PXY(II)=PXY(II)/BX(1)
      WRITE (6,26) (PX(II),PY(II),PXY(II),II=1,NMEM)
      GO TO 1009
 1001 ERROR =(1./BX(1))-1.
      IF(ERROR) 411,403,412
  412 IF(D1) 416,414,414
  414 IF(KLU4)420,418,420
  418 EN=EN+DEN
      D1=1.
      GO TO 402
  420  D1=1.
  421 DO 35 IN=1,NMEM
      PX(IN)=DSX(IN)*D1+PX(IN)
      PY(IN)=DSY(IN)*D1+PY(IN)
   35 PXY(IN)=DSXY(IN)*D1+PXY(IN)
      GO TO 402
  416 IF(KLU4) 424,422,424
  422 DEN=DEN*.5
      EN=EN+DEN
      D1=1.
      GO TO 402
  424 D1=1.
  427 DO 37 IN=1,NMEM
      DSX(IN)=.5*DSX(IN)
      DSY(IN)=.5*DSY(IN)
   37 DSXY(IN)=.5*DSXY(IN)
      GO TO 421
  411 IF(D1) 413,413,415
  413 IF(KLU4) 419,417,419
  417 EN=EN-DEN
      D1=-1.
      IF(ERROR+TOL) 402,403,403
  419 D1=-1.
      IF(ERROR+TOL)421,403,403
  415 IF(KLU4)425,423,425
  423 DEN=.5*DEN
      EN=EN-DEN
      D1=-1.
      IF(ERROR+TOL)402,403,403
  425 D1=-1.0
```

```
      IF(ERROR+TOL) 427,403,403
  403 WRITE(6,106) ERROR
      WRITE(6,107)
      IF(KLU4)1007,1008,1007
 1008 EN=EN+DEN
      ENX=APH*EN
      ENY=BTA*EN
      ENXY=GAM*EN
      WRITE(6,108) ENX,ENY,ENXY
      GO TO 1009
 1007 WRITE(6,26) (PX(II),PY(II),PXY(II),II=1,NMEM)
 1009 IF(STOPIT)67,2000,67
   67 CALL EXIT
   24 FORMAT(1H0,I5,5X,I5,5X,I5,5X,I5)
   25 FORMAT(1H0,7X,2HNX,14X,2HNY,13X,3HNXY)
   26 FORMAT(1H0,E16.8,5X,E16.8,5X,E16.8)
   27 FORMAT(1H0,4HKLU=,I5,3X,5HKLU2=,I5,3X,5HKLU3=,I5,3X,5HKLU4=,I5,3X,
     15HKLU5=,I5,3X,5HNELU=,I4,3X,4HLJT=,I4)
   28 FORMAT(1H0,4HANU=,E16.8,3X,2HE=,E16.8,3X,2HH=,E16.8,3X,4HNCO=,I5,3
     1X,3HS7=,E16.8)
   57 FORMAT(1H0,17HRESULTS OF FUTILE)
   60 FORMAT(4F6.1,I2)
   61 FORMAT(1H0,4HXSZ=,F6.1,3X,4HYSZ=,F6.1,3X,4HXN1=,F6.1,3X,4HYN1=,F6.
     11,3X,5HNJTS=,I3,3X,5HNMEM=,I3)
   62 FORMAT(1H0,1X,5HJOINT,9X,1HX,9X,1HY)
   63 FORMAT(1H0,I5,7X,F7.2,3X,F7.2)
   66 FORMAT(11H0 JII ARRAY//(4I5))
   96 FORMAT(1H0,5X,2HPX,13X,3HDSX,14X,2HPY,13X,3HDSY,14X,3HPXY,12X,4HDS
     1XY)
   97 FORMAT(1H0,E15.8,1X,E15.8,1X,E15.8,1X,E15.8,1X,E15.8,1X,E15.8)
   98 FORMAT(3E14.8,I7,E10.2)
   99 FORMAT(8I3)
  100 FORMAT(I4,2E12.1)
  101 FORMAT(I3,I2,4I3)
  102 FORMAT(5I3)
  103 FORMAT(6F10.2)
  104 FORMAT(29H0  THE BUCKLING STRESSES ARE /1H0,6X,4HNX/D,17X,4HNY/D,
     117X,5HNXY/D)
  105 FORMAT(1H0,E16.8,5X,E16.8,5X,E16.8)
  106 FORMAT(7H0 ERROR//(1P6E20.7))
  107 FORMAT(29H0  THE BUCKLING STRESSES ARE /1H0,7X,2HNX,19X,2HNY,19X,
     13HNXY)
  108 FORMAT(1H0,E16.8,5X,E16.8,5X,E16.8)
  109 FORMAT(1H0,4HAPH=,F6.2,3X,4HBTA=,F6.2,3X,4HGAM=,F6.2,3X,3HEN=,F9.2
     1,3X,4HDEN=,F9.3)
  200 FORMAT(34X,5HJOINT,9X,1HX,16X,1HY)
  201 FORMAT(33X,I4,5X,2E16.7)
  203 FORMAT(3X,I4,4X,I4,7X,I4,6X,I4,6X,I4)
  204 FORMAT( 37HTHE PROBLEM EXCEEDS AVAILABLE STORAGE )
  207 FORMAT(4X,4HNODE,5X,1HW,9X,3H WX,7X,3H WY,6X,5H  WXY)
  208 FORMAT(1H1,31HTHIS IS AN ELASTIC BUCKLING RUN)
  209 FORMAT(1H1,30HTHIS IS A PLASTIC BUCKLING RUN)
  308 FORMAT(1H0,15HERROR IN BIGSYM)
  550 FORMAT(F5.0)
  600 FORMAT(1H0,23HFUTILE WAS TROUBLE FREE)
  601 FORMAT(1H0,26HC IS NOT POSITIVE DEFINITE)
```

```
602 FORMAT(1H0,8HOVERFLOW)
703 FORMAT(1H0,12HC FROM BSTIF)
704 FORMAT(14H0 C FROM MSTIF)
      END
```

```
C     BUCKLING OF RECTANGULAR PLATES CONSTANT STRESS FIELD EIGENVALUE
C     PROGRAM
      COMMON JI,JT,X,Y,AD,BD,JD,LOCOL,C,AA
      COMMON JII,NBND,G,AKG,IM,NJTS,ANU,LIN
      COMMON E,H,ES,ET,KLU,APH,BTA,GAM,EN
      COMMON DEN,NCO,S7,IBUC
      DIMENSION JI(4),JT(81),X(81),Y(81),JII(64,4)
      DIMENSION NBND(81,4),LOCOL(16),G(16,16)
      DIMENSION C(11325),AA(4,3),AD(3),BD(3)
      DIMENSION JD(4),AKG(16,16),U(150),V(150)
      DIMENSION CA(150),BX(150),W(150),Q(150)
      DOUBLE PRECISION P(150)
      DO 42 I=1,81
      X(I)=0.
      Y(I)=0.
   42 JT(I)=0
      DO 43 J=1,4
      JI(J)=0
   43 JD(J)=0
      DO 44 K=1,3
      AD(K)=0.
   44 BD(K)=0.
      DO 45 L=1,16
      LOCOL(L)=0
      DO 45 M=1,16
      G(L,M)=0.
   45 AKG(L,M)=0.
      DO 46 N=1,4
      DO 46 II=1,64
   46 JII(II,N)=0
      DO 47 JJ=1,4
      DO 47 KK=1,81
   47 NBND(KK,JJ)=0
      DO 48 LL=1,150
      U(LL)=0.
      V(LL)=0.
      CA(LL)=0.
      BX(LL)=0.
      W(LL)=0.
   48 Q(LL)=0.
 2000 REWIND 2
      D1=0.
      BK=1.
      EK=1.
      READ(5,99) NJTS,KLU,KLU2,KLU3,KLU4,KLU5,NEIG,LOT

      IF(KLU )40,41,40
   41 WRITE (6,208)
      GO TO 5
   40 WRITE(6,209)
    5 TOL= 1. * 10.**(-LOT)
      READ(5,98) ANU,E,H,NCO,S7
      WRITE(6,27) KLU,KLU2,KLU3,KLU4,KLU5,LOT
      WRITE(6,28) ANU,E,H,NCO,S7
      IF(KLU3)4,9,4
```

```
    4 READ (5,60) XSZ,YSZ,XN1,YN1,NMEM
      WRITE(6,61) XSZ,YSZ,XN1,YN1,NJTS,NMEM
      XN2=XN1
      YN2=YN1
      DO 1 I=1,NJTS
      X(I)=((XN1-XN2)/(XN1))*XSZ
      Y(I)=((YN1-YN2)/(YN1))*YSZ
      JT(I)=I
      XN2=XN2-1.
      IF(XN2)3,1,1
    3 XN2=XN1
      YN2=YN2-1.
    1 CONTINUE
      WRITE(6,62)
      WRITE(6,63) (JT(I),X(I),Y(I),I=1,NJTS)
      NN1=XN1
      NN2=1
      NN3=XN1+2.
      NN4=XN1
      DO 64 MEM=1,NMEM
      JII(MEM,1)=NN2
      JII(MEM,2)=NN2+1
      JII(MEM,3)=NN3
      JII(MEM,4)=NN3+1
      NN2=NN2+1
      NN3=NN3+1
      NEM=MEM-NN4
      IF(NEM)64,65,65
   65 NN2=NN2+1
      NN3=NN3+1
      NN4=NN1+MEM
   64 CONTINUE
      GO TO 70
    9 READ(5,100) (JT(I),X(I),Y(I),I=1,NJTS)
      WRITE (6,200)
      WRITE (6,201) (JT(I),X(I),Y(I),I=1,NJTS)
    8 READ(5,101) MEM,NTYPE,JI(1),JI(2),JI(3),JI(4)
      IF(MEM-99) 6,7,6
    6 DO 10 I=1,4
      JII(MEM,I)=JI(I)
   10 CONTINUE
      KODE=MEM
      GO TO 8
    7 CONTINUE
      MEM=KODE
   70 WRITE(6,66)  ((JII(K,N),N=1,4),K=1, MEM)

      READ (5,103) APH,BTA,GAM,EN,DEN
      ENI=EN
      WRITE(6,109) APH,BTA,GAM,EN,DEN,TOL
      IF(KLU5)5060,5061,5060
 5061 IBUC=1
      DO 11 I=1,NJTS
      DO 11 J=1,4
   11 NBND(I,J)=0
      WRITE(6,207)
```

```
      DO 12 I=1,NJTS
      READ(5,102) JOINT,JD(1),JD(2),JD(3),JD(4)
      WRITE(6,203) JOINT,JD(1),JD(2),JD(3),JD(4)
      DO 13 J=1,4
      IF(JD(J)) 13,13,14
   14 NBND(JOINT,J)=IBUC
      IBUC=IBUC+1
      IF(IBUC-150)13,13,15
   13 CONTINUE
   12 CONTINUE
      IBUC=IBUC-1
 5060 READ(5,550) STOPIT
      LBUC=((IBUC+1)*IBUC)/2
      IF(KLU2) 504,18,504
  504 WRITE(6,23)
   23 FORMAT(1H0,14HTHE NBND ARRAY)
      WRITE(6,24)(NBND(I,1),NBND(I,2),NBND(I,3),NBND(I,4),I=1,JOINT)
      GO TO 18
   15 WRITE(6,204)
      CALL EXIT
   18 FIRST=0.
      DO 897 II=1,LBUC
  897 C(II)=0.
      DO 16 IM=1,MEM
      IM=IM
      LIN=4
      IF(FIRST) 554,555,554
  554 IF(KLU3-1) 555,556,555
  555 CALL MSTIF
      FIRST=1.
  556 CALL STACK
   16 CONTINUE
      IF(KLU2)4000,4001,4000
 4000 WRITE(6,703)
      CALL SCRIBE
 4001 CALL FUTILE (C,IBUC,NIX)
C
C     FORMATION OF L INVERSE
C
      IF(NIX)500,501,502
  501 WRITE(6,600)
      GO TO 503
  500 WRITE(6,601)
      GO TO 1009
  502 WRITE(6,602)
      GO TO 1009
  503 IF(KLU2)4002,4003,4002
 4002 WRITE(6,57)
      CALL SCRIBE
 4003 IK=1
      IQ=0
      KKK=IBUC+1
      DO 300 IL=2,KKK
      IF(IK-1)301,301,302
  302 IP=IK-1
      DO 303 J=1,IQ
```

```
      C(IP)=-1.*C(IP)/C(IK)
  303 IP=IP-1
  301 C(IK)=1./C(IK)
      IK=IK+IL
  300 IQ=IQ+1
C
C     WRITE L INVERSE ON TAPE 2
C
      KZ=1
      KQ=1
      DO 304 K=1,IBUC
      WRITE (2) (C(KP), KP=KQ,KZ)
      KQ=KQ+K
  304 KZ=KQ+K
  402 REWIND 2
      DO 896 KK=1,11250
  896 C(KK)=0.
      FIRST=0.
      DO 17 IM=1,MEM
      IM=IM
      LIN=4
      IF(FIRST)551,552,551
  551 IF(KLU3-1) 552,553,552
  552 CALL BSTIF
      FIRST=1.0
  553 CALL STACK
   17 CONTINUE
      IF(KLU2)4006,4007,4006
 4006 WRITE(6,704)
      CALL SCRIBE
 4007 DO 305 LK=1,IBUC
      READ (2) (CA(KQ),KQ=1,LK)
      CA(LK)=CA(LK)/EK
      CALL DAGGER(C,IBUC,CA,LK,W)
  305 CONTINUE
      IF(KLU2)4008,4009,4008
 4008 WRITE(6,702)
  702 FORMAT(1H0,14HC AFTER DAGGER)
      CALL SCRIBE
 4009 CALL SWITCH(C,IBUC)
      M=11325
      NEIG=IBUC
      CALL BIGSYM(C, IBUC, IBUC,M,BX,P,Q,U,V,MISS)
      IF(MISS)306,307,306
  306 WRITE(6,308)
      CALL EXIT
  307 WRITE(6,310) (BX(I),I=1,NEIG)
  309 IF(KLU)1000,1000,1001
 1000 WRITE(6,104)
      EXOD=BX(IBUC)*APH
      EYOD=BX(IBUC)*BTA
      EXYOD=BX(IBUC)*GAM
      WRITE(6,105) EXOD,EYOD,EXYOD
      GO TO 1009
 1001 ERROR=BX(IBUC)-1.
      IF(ERROR) 411,403,412
```

```
412 IF(D1) 416,414,414
414 BK=BK+DEN/ENI
    EK=SQRT(BK)
    EN=EN+DEN
    D1=1.
    GO TO 402
416  DEN=.5*DEN
    BK=BK+DEN/ENI
    EK=SQRT(BK)
    EN=EN+DEN
    D1=1.
    GO TO 402
411 IF(D1) 413,413,415
413 BK=BK-DEN/ENI
    EK=SQRT(BK)
    EN=EN-DEN
    D1=-1.
    IF(ERROR+TOL) 402,403,403
415 DEN=.5*DEN
    BK=BK-DEN/ENI
    EK=SQRT(BK)
    EN=EN-DEN
    D1=-1.
    IF(ERROR+TOL) 402,403,403
403 WRITE(6,106) ERROR
    EN=EN+DEN
    WRITE(6,107)
    ENX=APH*EN
    ENY=BTA*EN
    ENXY=GAM*EN
    WRITE(6,108) ENX,ENY,ENXY
1009 IF(STOPIT) 67,2000,67
  67 CALL EXIT
  24 FORMAT(1H0,I5,5X,I5,5X,I5,5X,I5)
  27 FORMAT(1H0,4HKLU=,I5,3X,5HKLU2=,I5,3X,5HKLU3=,I5,3X,5HKLU4=,I5,3X,
    15HKLU5=,I5,3X,4HLOT=,I2)
  28 FORMAT(1H0,4HANU=,E16.8,3X,2HE=,E16.8,3X,2HH=,E16.8,3X,4HNCO=,I5,3
    1X,3HS7=,E16.8)
  57 FORMAT(1H0,17HRESULTS OF FUTILE)
  60 FORMAT(4F6.1,I2)
  61 FORMAT(1H0,4HXSZ=,F6.1,3X,4HYSZ=,F6.1,3X,4HXN1=,F6.1,3X,4HYN1=,F6.
    11,3X,5HNJTS=,I3,3X,5HNMEM=,I3)
  62 FORMAT(1H0,1X,5HJOINT,9X,1HX,9X,1HY)
  63 FORMAT(1H0,I5,7X,F7.2,3X,F7.2)
  66 FORMAT(11H0 JII ARRAY//(4I5))
  97 FORMAT(1H0,E15.8,1X,E15.8,1X,E15.8,1X,E15.8,1X,F15.8,1X,E15.8)
  98 FORMAT(3E14.8,I7,F10.2)
  99 FORMAT(8I3)
 100 FORMAT(I4,2E12.1)
 101 FORMAT(I3,I2,4I3)
 102 FORMAT(5I3)
 103 FORMAT(6F10.2)
 104 FORMAT(29H0  THE BUCKLING STRESSES ARE /1H0,6X,4HNX/D,17X,4HNY/D,
    117X,5HNXY/D)
 105 FORMAT(1H0,E16.8,5X,E16.8,5X,E16.8)
 106 FORMAT(7H0 ERROR//(1P6E20.7))
```

```
107 FORMAT(29H0  THE BUCKLING STRESSES ARE /1H0,7X,2HNX,19X,2HNY,19X,
   13HNXY)
108 FORMAT(1H0,E16.8,5X,E16.8,5X,E16.8)
109 FORMAT(1H0,4HAPH=,F4.1,3X,4HBTA=,F4.1,3X,4HGAM=,F4.1,3X,3HEN=,F9.2
   1,3X,4HDEN=,F9.3,3X,4HTOL=,F9.5)
200 FORMAT(34X,5HJOINT,9X,1HX,16X,1HY)
201 FORMAT(33X,I4,5X,2E16.7)
203 FORMAT(3X,I4,4X,I4,7X,I4,6X,I4,6X,I4)
204 FORMAT( 37HTHE PROBLEM EXCEEDS AVAILABLE STORAGE )
207 FORMAT(4X,4HNODE,6X,1HW,9X,3H WX,7X,3H WY,6X,5H  WXY)
208 FORMAT(1H1,31HTHIS IS AN ELASTIC BUCKLING RUN)
209 FORMAT(1H1,30HTHIS IS A PLASTIC BUCKLING RUN)
308 FORMAT(1H0,15HERROR IN BIGSYM)
310 FORMAT(13H0 EIGENVALUES//(1P6E20.7))
550 FORMAT(F5.0)
600 FORMAT(1H0,23HFUTILE WAS TROUBLE FREE)
601 FORMAT(1H0,26HC IS NOT POSITIVE DEFINITE)
602 FORMAT(1H0,8HOVERFLOW)
703 FORMAT(1H0,12HC FROM BSTIF)
704 FORMAT(14H0 C FROM MSTIF)
      END
```

EIGENVALUE PROGRAM

```
      SUBROUTINE BSTIF
      COMMON JI,JT,X,Y,AD,BD,JD,LOCOL,C,AA
      COMMON JII,NBND,G,AKG,IM,NJTS,ANU,LIN
      COMMON E,H,ES,ET,KLU,APH,BTA,GAM,EN
      COMMON DEN,NCU,S7,IBUC,KLU4
      COMMON PX,PY,PXY,DSX,DSY,DSXY,SMAX
      DIMENSION JI(4),JT(81),X(81),Y(81),JII(64,4)
      DIMENSION NBND(81,4),LOCOL(16),G(16,16)
      DIMENSION C(11325),AA(4,3),AD(3),BD(3)
      DIMENSION JD(4),AKG(16,16)
      DIMENSION PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      DO 20 K  = 1,LIN
      DO 12 I  = 1,NJTS
      IF(JT(I)-JII( IM,K)) 12,14,12
   12 CONTINUE
      WRITE (6,16)
      CALL EXIT
   16 FORMAT (18H NO VALUE IN TABLE)
   14 AA(K,1)  = X(I)
      AA(K,2)  = Y(I)
   20 CONTINUE
   23 DO 25 I  = 1,2
      AD(I)    = AA(2,I) - AA(1,I)
   25 BD(I)    = AA(3,I) - AA(1,I)
      AL12     = (AD(1)**2+AD(2)**2+AD(3)**2)**.5
      AL13     = (BD(1)**2+BD(2)**2+BD(3)**2)**.5
      A=AL12
      B=AL13
      AB=A*B
      AOB=A/B
      AOB2=AOB**2
      AOB3=1./AOB2
      BOA=B/A
      BOA2=BOA**2
      B1=156./35.
      B2=36./25.
      B3=78./35.
      B4=3./25.
      B5=22./35.
      B6=54./35.
      B7=13./35.
      B8=27./35.
      B9=52./35.
      B10=4./25.
      B11=11./35.
      B12=26./35.
      B13=13./70.
      B14=18./35.
      B15=9./35.
      B16=4./35.
      B17=3./35.
      B18=3./70.
      B19=22./105.
      B20=2./35.
```

```
      B21=11./105.
      B22=13./105.
      B23=13./210.
      B24=4./105.
      B25=4./225.
      B26=2./105.
      B27=1./225.
      B28=2./70.
      B29=1./70.
      SODPLA=2.+120.*ANU
      SXDPLA=2.+10.*ANU
      ENDPLA=2.+20.*ANU
      IF(KLU)27,26,27
   26 C1=BOA2
      C2=0.
      C3=1.
      C4=0.
      C5=AOB2
      DS=1./(A*B)
      GO TO 28
   27 IF(KLU4)31,30,31
   30 SIG=((EN**2)/(H**2))*(APH**2+BTA**2-(APH*BTA)+(3.*GAM**2))
      SX=(EN*APH)/H
      SY=(EN*BTA)/H
      SXY=(EN*GAM)/H
      GO TO 32
   31 SX=PX(IM)/H
      SY=PY(IM)/H
      SXY=PXY(IM)/H
      SIG=(SX**2+SY**2-SX*SY+3.*SXY**2)
   32 EFS=SQRT(SIG)
      BN=NCO
      ET=1./(1.+(3./7.)*BN*(EFS/S7)**(NCO-1))
      ES=1./(1.+(3./7.)*(EFS/S7)**(NCO-1))
      DS=(ES*F*H**3)/(12.*(1.-ANU**2)*AB)
      DOD=1.-ET/ES
      C1=BOA2*(1.-.75*DOD*(SX/EFS)**2)
      C2=3.*DOD*(SX*SXY/(EFS**2))*BOA
      C3=1.-.75*((SX*SY+2.*SXY**2)/(EFS**2))*DOD
      C4=3.*DOD*((SY*SXY)/EFS**2)*AOB
      C5= AOB2*(1.-.75*DOD*(SY**2/EFS**2))
      SODPLA=(62.-((3.*SXY**2+91.5*SX*SY)/SIG)*DOD)/C3
      SXDPLA=(7.-((3.*SXY**2+9.*SX*SY)/SIG)*DOD)/C3
      ENDPLA=(12.-((3.*SXY**2+16.5*SX*SY)/SIG)*DOD)/C3
   28 DO 200 I=1,16
      DO 200 J=1,16
  200 G(I,J)=0.0
      G(16,1)=(-B13* C1 + .01* C3 * 2. - B13* C5 )
      G(15,1)=( B7 * C1 - B4 * C3 * 2. + B8 * C5 )
      G(14,1)=(+B8 * C1 - B4 * C3 * 2. + B7 * C5 )
      G(13,1)=(-B6 * C1 + B2 * C3 * 2. - B6 * C5 )
      G(12,1)=(-B13* C1 + ((SXDPLA/100.) * C3) + B11* C5 )
      G(11,1)=(-B7 * C1 + B4 * C3 * 2. + B3 * C5 )
      G(10,1)=(+B8 * C1 - B4 * SXDPLA * C3 - B5 * C5 )
      G(9,1)=( B6 * C1 - B2 * C3 * 2. - B1 * C5 )
      G( 8,1)=( B11* C1 + ((SXDPLA/100.) * C3) - B13* C5 )
```

```
G(7,1)=(-B5 * C1 - B4 * SXDPLA * C3 +B8 * C5 )
G(6,1)=(+B3 * C1 + B4 * C3 * 2. - B7 * C5 )
G(5,1)=(-B1 * C1 - B2 * C3 * 2. + B6 * C5 )
G( 4,1)=( B11* C1 + ((ENDPLA/100.) * C3)  + B11* C5)
G(3,1)=( B5 * C1 + B4 * SXDPLA * C3 +B3 * C5 )
G(2,1)=(+B3 * C1 + B4 * SXDPLA * C3 +B5 * C5 )
G(1,1)=( B1 * C1 + B2 * C3 * 2. + B1 * C5 )
G(16,2)=(-B23*C1 - ((C3/300.) * 2.) - B18* C5 )
G(15,2)=( B13* C1 - ((C3/100.) * 2.) + B13* C5 )
G(14,2)=( B15* C1 +((C3/25. ) * 2.) + B17* C5 )
G(13,2)=(-B8 * C1 + B4 * C3 * 2. -B7 * C5 )
G(12,2)=(-B22* C1 + ((C3/75. ) * 2.) + B20*C5 )
G(11,2)=(-B13* C1 + ((SXDPLA/100.) * C3) + B11* C5 )
G(10,2)=( B14* C1 - B10* C3 * 2. -B16* C5 )
G(9,2 )=(+B8 * C1 - B4 * SXDPLA * C3 - B5 * C5 )
G( 8,2)=( B21* C1 - ((SXDPLA/300.) * C3) -B18* C5 )
G( 7,2)=(-B11* C1 - ((SXDPLA/100.) * C3) + B13* C5 )
G( 6,2)=( B12* C1 - ((C3/25. ) * 2.) - B17* C5 )
G(5,2 )=(-B3 * C1 - B4 * C3 * 2. + B7 * C5 )
G( 4,2)=( B19* C1 + ((SXDPLA/75. ) * C3) + B20* C5 )
G( 3,2)=( B11* C1 + ((SUDPLA/100.) * C3) + B11* C5 )
G(2,2 )=( B9 * C1 + B10* C3 * 2. + B16* C5 )
G(16,3)=( -B18* C1 - ((C3/300.) * 2.) - B23* C5 )
G(15,3)=( B17* C1 + ((C3/25.  ) * 2.) + B15* C5)
G(14,3)=( B13* C1 - ((C3/100.)* 2.) + B13* C5 )
G(13,3)=(-B7 * C1 + B4 * C3 * 2. -B8 * C5 )
G(12,3)=(-B18* C1 - ((SXDPLA/300.) * C3) + B21* C5 )
G(11,3)=(-B17* C1 - ((C3/25. ) * 2.) + B12*C5 )
G(10,3)=( B13* C1 - ((SXDPLA/100.) * C3) - B11* C5 )
G(9,3 )=( B7 * C1 - B4 * C3 * 2. -B3 * C5 )
G( 8,3)=( B20* C1 + ((C3/75. ) * 2.) - B22*C5 )
G(7,3 )=(-B16* C1 - B10* C3 * 2. +B14* C5 )
G( 6,3)=( B11* C1 + ((SXDPLA/100.) * C3) - B13* C5)
G(5,3 )=(-B5 * C1 - B4 * SXDPLA * C3 + B8 * C5 )
G( 4,3)=( B20* C1 + ((SXDPLA/75. ) * C3 ) + B19* C5 )
G(3,3 )=( B16* C1 + B10* C3 * 2. +B9 * C5 )
G(16,4)=(-B29* C1 + ((C3/900.) * 2.) - B29* C5 )
G(15,4)=( B18* C1 + ((C3/300.) * 2.) + B23* C5 )
G(14,4)=( B23* C1 + ((C3/300.) * 2.) + B18* C5 )
G(13,4)=(-B13* C1 + ((C3/100.) * 2. )- B13* C5 )
G(12,4)=(-B28* C1 - B27* C3 * 2. + B26 * C5 )
G(11,4)=(-B18* C1 - ((SXDPLA/300.) * C3 ) + B21* C5 )
G(10,4)=( B22* C1 - ((C3/75. ) * 2. ) - B20* C5 )
G( 9,4)=( B13 * C1 - ((SXDPLA/100.) * C3 ) - B11* C5 )
G(8,4 )=( B26* C1 - B27* C3 * 2. - B28 * C5 )
G(7,4)=(-B20*C1-(2./75.)*C3+B22*C5)
G( 6,4)=( B21* C1 - ((SXDPLA/300.) * C3 ) - B18 *C5)
G( 5,4)=(-B11*C1 - ((SXDPLA/100.) * C3 ) + B13*C5 )
G(4,4 )=( B24* C1 + B25* C3 * 2. + B24 * C5 )
G(5,5)=G(1,1)
G(6,5)=-G(2,1)
G(7,5)= G(3,1)
G(8,5)=-G(4,1)
G(6,6)=G(2,2)
G(7,6)=-G(3,2)
G(8,6)= G(4,2)
```

```
G(7,7)= G(3,3)
G(8,7)=-G(4,3)
G(8,8)= G(4,4)
G(9,9)= G(1,1)
G(10,9)=G(2,1)
G(11,9)=-G(3,1)
G(12,9)=-G(4,1)
G(10,10)= G(2,2)
G(11,10)=-G(3,2)
G(12,10)=-G(4,2)
G(11,11)= G(3,3)
G(12,11)= G(4,3)
G(12,12)= G(4,4)
G(13,13)=G(1,1)
G(14,13)=-G(2,1)
G(15,13)=-G(3,1)
G(16,13)= G(4,1)
G(14,14)= G(2,2)
G(15,14)= G(3,2)
G(16,14)=-G(4,2)
G(15,15)= G(3,3)
G(16,15)=-G(4,3)
G(16,16)= G(4,4)
G(9,5)=  G(13,1)
G(10,5)=G(13,2)
G(11,5)=-G(13,3)
G(12,5)=-G(13,4)
G(9,6)=G(14,1)
G(10,6)= G(14,2)
G(11,6)=-G(14,3)
G(12,6)=-G(14,4)
G(9,7)=-G(15,1)
G(10,7)=-G(15,2)
G(11,7)= G(15,3)
G(12,7)=G(15,4)
G(9,8)=-G(16,1)
G(10,8)=-G(16,2)
G(11,8)= G(16,3)
G(12,8)= G(16,4)
G(13,5)= G(9,1)
G(14,5)=-G(9,2)
G(15,5)=-G(9,3)
G(16,5)= G(9,4)
G(13,6)=-G(10,1)
G(14,6)= G(10,2)
G(15,6)= G(10,3)
G(16,6)=-G(10,4)
G(13,7)=-G(11,1)
G(14,7)= G(11,2)
G(15,7)=+G(11,3)
G(16,7)=-G(11,4)
G(13,8)= G(12,1)
G(14,8)=-G(12,2)
G(15,8)=-G(12,3)
G(16,8)= G(12,4)
G(13,9)= G(5,1)
```

```
      G(14,9)= G(6,1)
      G(15,9)=-G(7,1)
      G(16,9)=-G(8,1)
      G(13,10)= G(5,2)
      G(14,10)= G(6,2)
      G(15,10)=-G(7,2)
      G(16,10)=-G(8,2)
      G(13,11)=-G(5,3)
      G(14,11)=-G(6,3)
      G(15,11)= G(7,3)
      G(16,11)= G(8,3)
      G(13,12)=-G(5,4)
      G(14,12)=-G(6,4)
      G(15,12)= G(7,4)
      G(16,12)= G(8,4)
      DO 210 J=1,16
      DO 210 I=J,16
  210 G(J,I)=G(I,J)
  245 DO 250 I=1,16
       DO 250 J=1,16
  250 AKG(I,J)=G(I,J)*DS
      IF(KLU)257,256,257
  257 IF(SXY)258,256,258
  258 DO 350 I=1,16
      DO 350 J=1,16
  350 G(I,J)=0.
      G(2,2)=-C2*.25
      G(3,2)=-.1*(C2+C4)
      G(3,3)=-.25*C4
      G(4,1)= .1*(C2+C4)
      G(5,3)=-.5*C4
      G(5,4)=-.1*(C2+C4)
      G(6,3)=G(4,1)
      G(6,4)=.05*C2+(1./60.)*C4
      G(6,6)=-G(2,2)
      G(7,1)=-G(5,3)
      G(7,2)=G(4,1)
      G(7,6)=G(3,2)
      G(7,7)=-G(3,3)
      G(8,1)=G(5,4)
      G(8,2)=-G(6,4)
      G(8,5)=G(4,1)
      G(9,2)=-.5*C2
      G(9,4)= G(5,4)
      G(9,6)=-G(9,2)
      G(9,7)= G(5,3)
      G(9,8)=G(4,1)
      G(10,1)=-G(9,2)
      G(10,3)=G(4,1)
      G(10,5)= G(9,2)
      G(10,6)= -G(2,2)
      G(10,7)=  G(3,2)
      G(10,8)=  G(6,4)
      G(10,10)= -G(2,2)
      G(11,2)=G(4,1)
      G(11,4)=(1./60.)*C2+.05*C4
```

```
      G(11,5)= -G(5,3)
      G(11,6)= G(3,2)
      G(11,7)= -G(3,3)
      G(11,8)= -G(11,4)
      G(11,10)= G(3,2)
      G(11,11)=-G(3,3)
      G(12,1)= G(5,4)
      G(12,3)=-G(11,4)
      G(12,5)=G(4,1)
      G(12,6)=-G(6,4)
      G(12,7)= G(11,4)
      G(12,8)=-(1./120.)*(C2+C4)
      G(12,9)=G(4,1)
      G(13,2)=-G(9,2)
      G(13,3)=-G(5,3)
      G(13,4)=G(4,1)
      G(13,6)= G(9,2)
      G(13,8)= G(5,4)
      G(13,11)= G(5,3)
      G(13,12)= G(5,4)
      G(14,1)= G(9,2)
      G(14,2)= G(2,2)
      G(14,3)= G(3,2)
      G(14,4)=-G(6,4)
      G(14,5)=-G(9,2)
      G(14,7)=G(4,1)
      G(14,11)=G(4,1)
      G(14,12)= G(6,4)
      G(14,14)= G(2,2)
      G(15,1)= G(5,3)
      G(15,2)= G(3,2)
      G(15,3)= G(3,3)
      G(15,4)=-G(11,4)
      G(15,5)=G(4,1)
      G(15,8)= G(11,4)
      G(15,9)=-G(5,3)
      G(15,10)=G(4,1)
      G(15,14)= G(3,2)
      G(15,15)= G(3,3)
      G(16,1)=G(4,1)
      G(16,2)= G(6,4)
      G(16,3)=G(11,4)
      G(16,4)=-G(12,8)
      G(16,5)= G(5,4)
      G(16,7)=-G(11,4)
      G(16,9)= G(5,4)
      G(16,10)=-G(6,4)
      G(16,13)=G(4,1)
      DO 360 I=1,16
      DO 360 J=I,16
  360 G(I,J)=G(J,I)
      DO 320 I=1,16
      DO 320 J=1,16
  320 AKG(I,J)=AKG(I,J)+DS*G(I,J)
  256 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MSTIF
      COMMON JI,JT,X,Y,AD,BD,JD,LCCOL,C,AA
      COMMON JII,NBND,G,AKG,IM,NJTS,ANU,LIN
      COMMON E,H,ES,ET,KLU,APH,BTA,GAM,EN
      COMMON DEN,NCO,S7,IBUC,KLU4
      COMMON PX,PY,PXY,DSX,DSY,DSXY,SMAX
      DIMENSION JI(4),JT(81),X(81),Y(81),JII(64,4)
      DIMENSION NBND(81,4),LOCOL(16),G(16,16)
      DIMENSION C(11325),AA(4,3),AD(3),BD(3)
      DIMENSION JD(4),AKG(16,16)
      DIMENSION PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
C     TABLE LOOKUP FOLLOWS
      DO 20 K  = 1,LIN
      DO 12 I  = 1,NJTS
      IF(JT(I)-JII( IM,K)) 12,14,12
   12 CONTINUE
      WRITE(6,16)
      CALL EXIT
   16 FORMAT (18H NO VALUE IN TABLE)
   14 AA(K,1)  = X(I)
      AA(K,2)  = Y(I)
   20 CONTINUE
   23 DO 25 I  = 1,2
      AD(I)    = AA(2,I) - AA(1,I)
   25 BD(I)    = AA(3,I) - AA(1,I)
      AL12     = (AD(1)**2+AD(2)**2+AD(3)**2)**.5
      AL13     = (BD(1)**2+BD(2)**2+BD(3)**2)**.5
      A=AL12
      B=AL13
      AB=A*B
      ACB=A/B
      ACB2=ACB**2
      AOB3=1./AOB2
      BCA=B/A
      BOA2=BOA**2
      DO 260 I=1,16
      DO 260 J=1,16
  260 G(I,J)=0.0
      D1=1872.
      D2=156.
      D3=208.
      D4=264.
      D5=22.
      D6=48.
      D7=88./3.
      D8=4.
      D9=16./3.
      D10=52.
      D11=22./3.
      D12=4./3.
      D13=648.
      D14=54.
      D15=13.
      D16=72.
```

```
      D17=52./3.
      D18=36.
      D19=3.
      D20=18.
      D21=13./3.
      D22=1.
      IF(KLU4)41,40,41
   41 IF(KLU)42,43,42
   43 SX=(B/A)*PX(IM)
      SY=(A/B)*PY(IM)
      SXY=PXY(IM)
      GO TO 285
   42 SX=(B/A)*PX(IM)
      SY=(A/B)*PY(IM)
      SXY=PXY(IM)
      GO TO 285
   40 IF(KLU)284,283,284
  283 SX=(B/A)*APH
      SY=(A/B)*BTA
      SXY= GAM
      GO TO 285
  284 SX= (B/A)*APH*EN
      SY=(A/B)*BTA*EN
      SXY=GAM*EN
  285 G(1,1 )=SX * D1 + SY * D1
      G(2,1 )=SX * D2 + SY * D4
      G(2,2 )=SX * D3 + SY * D6
      G(3,1 )=SX * D4 + SY * D2
      G(3,2 )=SX * D5 + SY * D5
      G(3,3 )=SX * D6 + SY * D3
      G(4,1 )=SX * D5 + SY * D5
      G(4,2 )=SX * D7 + SY * D8
      G(4,3)= SX*D8+SY*D7
      G(4,4 )=SX * D9 + SY * D9
      G(5,1 )=SX*(-D1)+ SY * D13
      G(5,2 )=SX*(-D2)+ SY * D2
      G(5,3 )=SX*(-D4)+ SY * D14
      G(5,4 )=SX*(-D5)+ SY * D15
      G(6,1 )=SX * D2 + SY *(-D2)
      G(6 ,2 )=SX*(-D10)+SY *(-D18)
      G(6 ,3 )=SX * D5 + SY *(-D15)
      G(6 ,4 )=SX*(-D11)+SY *(-D19)
      G(7 ,1 )=SX*(-D4)+ SY * D14
      G(7 ,2 )=SX*(-D5)+ SY * D15
      G(7 ,3 )=SX*(-D6)+ SY * D16
      G(7 ,4 )=SX*(-D8)+ SY * D17
      G(8 ,1 )=SX * D5 + SY *(-D15)
      G(8, 2 )=SX*(-D11)+ SY*(-D19)
      G(8 ,3 )=SX * D8 + SY *(-D17)
      G(8 ,4 )=SX*(-D12)+SY *(-D8)
      G(9 ,1 )=SX * D13+ SY *(-D1)
      G(9 ,2 )=SX * D14+ SY *(-D4)
      G(9 ,3 )=SX * D2 + SY *(-D2)
      G(9 ,4 )=SX * D15+ SY *(-D5)
      G(10,1 )=SX * D14+ SY *(-D4)
      G(10,2 )=SX * D16+ SY *(-D6)
```

```
G(10,3 )=SX * D15+ SY *(-D5)
G(10,4 )=SX * D17+ SY *(-D8)
G(11,1 )=SX*(-D2)+ SY * D2
G(11,2 )=SX*(-D15)+SY * D5
G(11,3 )=SX*(-D18)+SY *(-D10)
G(11,4 )=SX*(-D19)+SY *(-D11)
G(12,1 )=SX*(-D15)+SY * D5
G(12,2 )=SX*(-D17)+SY * D8
G(12,3 )=SX*(-D19)+SY *(-D11)
G(12,4 )=SX*(-D8)+ SY *(-D12)
G(13,1 )=SX*(-D13)+SY *(- D13)
G(13,2 )=SX*(-D14)+SY *(-D2)
G(13,3 )=SX*(-D2)+ SY *(-D14)
G(13,4)=-SX*D15-SY*D15
G(14,1 )=SX * D14 + SY * D2
G(14,2 )=SX*(-D20)+ SY * D18
G(14,3 )=SX * D15 + SY * D15
G(14,4)=-SX*D21-SY*D19
G(15,1 )=SX * D2  + SY * D14
G(15,2 )=SX * D15 + SY * D15
G(15,3 )=SX * D18 + SY *(-D20)
G(15,4 )=SX * D19 + SY *(-D21)
G(16,1 )=SX*(-D15)+ SY *(-D15)
G(16,2 )=SX * D21 + SY *(-D19)
G(16,3 )=SX*(-D19)+ SY  *D21
G(16,4 )=SX * D22 + SY * D22
G(5,5)=G(1,1)
G(6,5)=-G(2,1)
G(7,5)= G(3,1)
G(8,5)=-G(4,1)
G(6,6)=G(2,2)
G(7,6)=-G(3,2)
G(8,6)= G(4,2)
G(7,7)= G(3,3)
G(8,7)=-G(4,3)
G(8,8)= G(4,4)
G(9,9)= G(1,1)
G(10,9)=G(2,1)
G(11,9)=-G(3,1)
G(12,9)=-G(4,1)
G(10,10)= G(2,2)
G(11,10)=-G(3,2)
G(12,10)=-G(4,2)
G(11,11)= G(3,3)
G(12,11)= G(4,3)
G(12,12)= G(4,4)
G(13,13)=G(1,1)
G(14,13)=-G(2,1)
G(15,13)=-G(3,1)
G(16,13)= G(4,1)
G(14,14)= G(2,2)
G(15,14)= G(3,2)
G(16,14)=-G(4,2)
G(15,15)= G(3,3)
G(16,15)=-G(4,3)
G(16,16)= G(4,4)
```

```
      G(9,5)=  G(13,1)
      G(10,5)=G(13,2)
      G(11,5)=-G(13,3)
      G(12,5)=-G(13,4)
      G(9,6)=G(14,1)
      G(10,6)= G(14,2)
      G(11,6)=-G(14,3)
      G(12,6)=-G(14,4)
      G(9,7)=-G(15,1)
      G(10,7)=-G(15,2)
      G(11,7)= G(15,3)
      G(12,7)=G(15,4)
      G(9,8)=-G(16,1)
      G(10,8)=-G(16,2)
      G(11,8)= G(16,3)
      G(12,8)= G(16,4)
      G(13,5)= G(9,1)
      G(14,5)=-G(9,2)
      G(15,5)=-G(9,3)
      G(16,5)= G(9,4)
      G(13,6)=-G(10,1)
      G(14,6)= G(10,2)
      G(15,6)= G(10,3)
      G(16,6)=-G(10,4)
      G(13,7)=-G(11,1)
      G(14,7)= G(11,2)
      G(15,7)=+G(11,3)
      G(16,7)=-G(11,4)
      G(13,8)= G(12,1)
      G(14,8)=-G(12,2)
      G(15,8)=-G(12,3)
      G(16,8)= G(12,4)
      G(13,9)= G(5,1)
      G(14,9)= G(6,1)
      G(15,9)=-G(7,1)
      G(16,9)=-G(8,1)
      G(13,10)= G(5,2)
      G(14,10)= G(6,2)
      G(15,10)=-G(7,2)
      G(16,10)=-G(8,2)
      G(13,11)=-G(5,3)
      G(14,11)=-G(6,3)
      G(15,11)= G(7,3)
      G(16,11)= G(8,3)
      G(13,12)=-G(5,4)
      G(14,12)=-G(6,4)
      G(15,12)= G(7,4)
      G(16,12)= G(8,4)
      DC 270 I=1,16
      DC 270 J=I,16
  270 G(I,J)=G(J,I)
       CO 310 I=1,16
       CO 310 J=1,16
  310 AKG(I,J)=-G(I,J)/4200.
      IF(SXY)311,312,311
  311 DC 330 I=1,16
```

```
      DC 330 J=1,16
  330 G(I,J)=0.0
      G(1,1)= .5
      G(3,2)= .02
      G(4,1)=-.02
      G(5,3)= .1
      G(5,4)= .02
      G(5,5)=-.5
      G(6,3)=-.02
      G(6,4)=-(1./300.)
      G(7,1)=-.1
      G(7,2)=-.02
      G(7,6)= .02
      G(8,1)= .02
      G(8,2)=-G(6,4)
      G(8,5)=-.02
      G(9,2)=.1
      G(9,4)=.02
      G(9,5)= .5
      G(9,6)=-.1
      G(9,7)= .1
      G(9,8)=-.02
      G(9,9)=-.5
      G(10,1)=-.1
      G(10,3)=-.02
      G(10,5)= .1
      G(10,6)=-(1./60.)
      G(10,7)= .02
      G(10,8)= G(6,4)
      G(11,2)=-.02
      G(11,4)= G(6,4)
      G(11,5)= -.1
      G(11,6)= .02
      G(11,7)=G(10,6)
      G(11,8)=-G(6,4)
      G(11,10)= .02
      G(12,1)= .02
      G(12,3)=-G(6,4)
      G(12,5)=-.02
      G(12,6)=-G(6,4)
      G(12,7)= G(6,4)
      G(12,8)= 1./1800.
      G(12,9)=-.02
      G(13,1)=-.5
      G(13,2)=-.1
      G(13,3)=-.1
      G(13,4)=-.02
      G(13,6)=.1
      G(13,8)= .02
      G(13,11)= .1
      G(13,12)= .02
      G(13,13)= .5
      G(14,1)= .1
      G(14,2)=-G(10,6)
      G(14,3)= .02
      G(14,4)=-G(6,4)
```

```
      G(14,5)=-.1
      G(14,7)=-.02
      G(14,11)=-.02
      G(14,12)= G(6,4)
      G(15,1)= .1
      G(15,2)= .02
      G(15,3)=-G(10,6)
      G(15,4)=-G(6,4)
      G(15,6)=-.02
      G(15,8)= G(6,4)
      G(15,9)=-.1
      G(15,10)=-.02
      G(15,14)= .02
      G(16,1)=-.02
      G(16,2)= G(6,4)
      G(16,3)= G(6,4)
      G(16,4)=-G(12,8)
      G(16,5)= .02
      G(16,7)=-G(6,4)
      G(16,9)= .02
      G(16,10)=-G(6,4)
      G(16,13)=-.02
      DO 360 I=1,16
      DO 360 J=I,16
  360 G(I,J)=G(J,I)
      DO 320 I=1,16
      DO 320 J=1,16
  320 AKG(I,J)= AKG(I,J)+SXY*G(I,J)
  312 RETURN
      END
```

```
      SUBROUTINE STACK
      COMMON JI,JT,X,Y,AD,BD,JD,LOCOL,C,AA
      COMMON JII,NBND,G,AKG,IM,NJTS,ANU,LIN
      COMMON E,F,ES,ET,KLU,APH,BTA,GAM,EN
      COMMON DEN,NCO,S7,IBUC,KLU4
      COMMON PX,PY,PXY,DSX,DSY,DSXY,SMAX
      DIMENSION JI(4),JT(81),X(81),Y(81),JII(64,4)
      DIMENSION NBND(81,4),LOCOL(16),G(16,16)
      DIMENSION C(11325),AA(4,3),AD(3),BD(3)
      DIMENSION JD(4),AKG(16,16)
      DIMENSION PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      DO 18 I=1,16
      LOCOL(I)=0
   18 CONTINUE
      DO 19 IA=1,LIN
      ID=JII( IM,IA)
      DO 20 IB=1,4
      IC=4*(IA-1)+IB
      LOCOL(IC)=NBND(ID,IB)
   20 CONTINUE
   19 CONTINUE
      DO 21 I=1,16
      DO 21 J=1,I
      IF(LOCOL(J)) 21,21,22
   22 IF(LOCOL(I)) 21,21,23
   23 KROW=MAX0(LOCOL(I),LOCOL(J))
      KCOL=MIN0(LOCOL(I),LOCOL(J))
      INDEX=(KROW**2-KROW)/2+KCOL
      C(INDEX)=C(INDEX)+AKG(I,J)
   21 CONTINUE
      RETURN
       END
```

```
      SUBROUTINE SCRIBE
      COMMON JI,JT,X,Y,AD,BD,JD,LOCOL,C,AA
      COMMON JII,NBND,G,AKG,IM,NJTS,ANU,LIN
      COMMON F,H,FS,ET,KLU,APH,BTA,GAM,EN
      COMMON DEN,NCO,S7,IBUC,KLU4
      COMMON PX,PY,PXY,DSX,DSY,DSXY,SMAX
      DIMENSION JI(4),JT(81),X(81),Y(81),JII(64,4)
      DIMENSION NBND(81,4),LOCOL(16),G(16,16)
      DIMENSION C(11325),AA(4,3),AD(3),BD(3)
      DIMENSION JD(4),AKG(16,16),U(150),V(150)
      DIMENSION CA(150),BX(150),W(150),Q(150)
      DIMENSION PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      LIMA=1
      LIMB=1
      DO 55 I=1,IBUC
      WRITE(6,205) I,(C(J),J=LIMA,LIMB)

      LIMA=LIMA+I
      LIMB=LIMB+I+1
   55 CONTINUE
  205 FORMAT(1X,I3,2X,5E16.8/6X,5E16.8/(6X,2E16.8))
      RETURN
      END
```

```
      SUBROUTINE DAGGER(A,M,P,KO,Q)
      DIMENSION A(1),P(1),Q(1)
      DOUBLE PRECISION SUM
      COMMON /SUMMER/ SUM
      EQUIVALENCE (SUN,SUM)
      K = KO
      K1 = K + 1
      L=1
      LL = 0
      CALL PRELIM(P,1,A,1)
      DO 130  I = 2,K1
      SUM = 0.D0
      CALL DOT(L)
      IF (K - L) 100,120,100
  100 LL = LL + L
      LJ = LL + L
      DO 110  J = I,K
      SUM = SUM + A(LJ)*P(J)
  110 LJ = LJ + J
  120 Q(I-1) = SUN
  130 L = I
  160 SUM = 0.D0
      DO 170  I = 1,K
      SUM = SUM + P(I)*Q(I)
      A(LL+1) = Q(I)
  170 LL = LL + 1
      A(LL) = SUN
      IF (M - K) 140,200,140
  140 CALL PRELIM(P,K,A(LL+1),0)
      DO 190  I = K1,M
      SUM = 0.D0
      LL=LL+I-1
  190 A(LL)=DOT(I)
  200 CONTINUE
      RETURN
      END
```

```
$IBFTC FACTOR  LIST,M94/2,XR7
      SUBROUTINE FUTILE(A,N,SCORE)
      DIMENSION A(1)
      COMMON /OPT1/ DETERM
      COMMON /OVFLOW/ OVFLOW
      DOUBLE PRECISION SUM
      COMMON /SUMMER/ SUM
      DATA CATCH/0100000/
      EQUIVALENCE (SUN,SUM)
      INTEGER SCORE
      SCORE = 0
      K1 = 1
      KK = 0
      DO 210  K = 1,N
      KK = KK + K
      CALL PRELIM(A(K1),K-1,A(K1),0)
      IK = KK
      DO 140  I = K,N
      SUM = -A(IK)
      IF (I - K) 120,100,120
  100 DENOM = -DOT(I)
      DETERM = DENOM * DETERM
      IF (DENOM) 980,980,110
  110 DENOM = -SQRT(DENOM)
      A(IK) = -DENOM
      GO TO 130
  120 CALL DOT(I)
      A(IK) = SUM / DENOM
  130 IK = IK + I
  140 CONTINUE
      K1 = K1 + K
  210 CONTINUE
      IF (AND(OVFLOW,CATCH)) 220,230,220
  220 SCORE = 1
      OVFLOW = 0.
  230 RETURN
  980 SCORE = -1
      RETURN
       END
```

```
      SUBROUTINE SWITCH(A,M)
      DIMENSION A(1)
      N = IABS(M)
      IF (N - 2) 190,190,90
   90 L = (N*(N+1)) / 2
      KEY = 1
      LOCK = N/2 + 1
      IF (M) 100,190,160
  100 IF (N - 3) 110,140,110
  110 KKT = 3
      NKF = N - 1
      IMAGE = L
      INTO = L - 3
      I = 3
      DO 130  K = 2,LOCK
      DO 120  IK = KKT,NKF
      X = A(IK)
      A(IK) = A(INTO)
      A(INTO) = X
      INTO = INTO - I
  120 I = I + 1
      KKT = NKF + K
      NKF = NKF + N - K
      IMAGE = IMAGE - K
      INTO = IMAGE
  130 I = K
  140 IF (KEY) 150,190,150
  150 KEY = 0
  160 LOV2 = L / 2
      K = L - 2
      DO 170  I = 3,LOV2
      X = A(I)
      A(I) = A(K)
      A(K) = X
  170 K = K - 1
      IF (KEY) 180,190,180
  180 KEY = 0
      GO TO 100
  190 RETURN
      END
```

```
$IBFTC TRISYS  M94/2,XR7,NODECK
C HOUSEHOLDER TRI-DIAGONALIZATION ROUTINE FOR REAL SYMMETRIC MATRICES.
C FOLLOWED BY STURM-SEQUENCE COMPUTATION OF THE EIGENVALUES. PROGRAM
C SHOULD WORK FOR ORDERS UP TO 1000 AT LEAST, INCLUDING 1 AND 2, BUT IT
C IS AIMED PRIMARILY AT LARGE ORDERS. AN ATTEMPT IS MADE TO USE SCRATCH
C TAPES ABOUT AS ECONOMICALLY AS POSSIBLE, AND INNER PRODUCTS ARE ACCUM-
C   ULATED IN D.P. TO ACHIEVE HIGH ACCURACY.
C
C
      SUBROUTINE BIGSYM(A,NO,NUMBER,ASIZE,B,P,Q,U,V,MISS)
C
C
      DIMENSION A(1),Q(1),U(1),V(1),B(1),P(1)
      DOUBLE PRECISION P,SUM,PI
      INTEGER ASIZE
      EQUIVALENCE (SUM,SUN)
C
C INITIALIZE.
C
      CALL OVERFL(I)
      N = IABS(NO)
      N1 = N - 1
      N2 = N - 2
C
C TEST TO SEE IF MATRIX IS ALREADY IN CORE.
C
      IF (NUMBER) 70,2000,80
C
C NO. REWIND TAPES, SET THE INDICATORS INCORE AND KEY, AND READ IN THE
C   FIRST ROW OF MATRIX. THE ARRAY A IS USED TO HOLD AS MANY ROWS AS
C   POSSIBLE. AT EACH STEP THE FIRST ROW IN CORE IS ELIMINATED AND ONE
C   OR MORE NEW ROWS BROUGHT IN UNTIL ALL REMAINING ROWS ARE IN CORE.
C
   70 REWIND 2
      REWIND 3
      REWIND 4
      INCORE = 1
      KEY = 0
      READ (2)  (A(I), I = 1,N)
      GO TO 90
C
C YES. SET THE INDICATOR INCORE AND DEFINE THE INDICES WHICH DESIGNATE
C   FIRST AND LAST ELEMENTS OF FIRST ROW IN CORE. THE ARRAY A ALWAYS
C   HOLDS THE ENTIRE MATRIX AND AS EACH ROW IS ELIMINATED, IT IS RE-
C   PLACED BY THE CORRESPONDING V.
   80 INCORE = 0
      K2K2 = 1
      K2N = N
   90 NUMMER = IABS(NUMBER)
C
C DISPOSE OF TRIVIAL CASES.
C
      IF (ASIZE - N - N1) 100,110,110
  100 MISS = 1
      RETURN
  110 IF (N2) 120,125,130
  120 B(1) = A(1)
      GO TO 750
  125 U(2) = 0.
```

```
C
C INCORE = 0 MEANS THE (REMAINING) MATRIX FITS IN CORE. INCORE = 1 MEANS
C   THAT SOME ROWS ARE STILL ON TAPE.
C IF KEY = 0 (K EVEN) READ T2 AND WRITE T3. REVERSE IF KEY = 1 (K ODD).
C M IS NUMBER OF ROWS (STARTING WITH K+1) WHICH FIT IN CORE. LIM IS THE
C   INDEX OF LAST SUCH ROW. EVENTUALLY LIM REACHES N, AND THEREAFTER
C   READING AND WRITING ARE SUPPRESSED.
C
  130 K2 = 1
      K3 = 2
      K1K1 = 1
      NK1 = N
      NEXTM = LIMIT(N,ASIZE)
      LIM = NEXTM
C
C MAJOR LOOP. EXCEPT FOR K = 0 AND K = N-2, THE KTH STEP COMPLETES STAGE
C   K OF THE HOUSEHOLDER ALGORITHM AND BEGINS STAGE K+1.  K = 0 PREPARES
C   FOR STAGE 1, AND K = N-2 (VIRTUALLY) COMPLETES THE ALGORITHM.
C
C IF K IS G.T. 0, THE LOOP STARTS WITH P UPPER K AND V UPPER K STORED IN
C   Q(K+1)...Q(N) AND U(K+1)...U(N). THE 1ST K DIAGONAL ELEMENTS ARE IN
C   Q(1)...Q(K), WITH OFFDIAG ELEMENTS IN U(1)...U(K) AND THEIR SQUARES
C   IN V(1)...V(K). ROWS K+1...N OF A UPPER K ARE STORED IN CORE AND, IF
C   NECESSARY, ON TAPE. IF L IS THE 1ST K-VALUE FOR WHICH ROWS L+1...N
C   FIT IN CORE, THEN A UPPER K STARTS IN A(1) FOR K L.T.E. L, AND IN
C   A((K-L)(N-L) - (K-L)(K-L-1)/2 + 1) FOR K G.T. L.
C
C NIX = 0, BUT A POINTLESS PROVISION OF FORTRAN IV PROHIBITS EXPLICIT 0.
C
      NIX = 0
      DO 690  K = NIX,N2
      K1 = K2
      K2 = K3
      K3 = K + 3
      NK = NK1
      NK1 = NK - 1
      M = NEXTM
C
C TEST TO SEE IF ENTIRE (REMAINING) MATRIX FITS IN CORE
C
      IF (INCORE) 2000,140,150
C
C YES. IF K G.T. 0, FORM TAU, Q, AND NEW A. IF K = 0, SKIP.
C
  140 K1K1 = K2K2
      K1N = K2N
      IF (K) 2000,250,200
C
C NO. TRY AGAIN NEXT TIME
C
  150 INCORE = N - LIM
      K1N = NK
C
C IF K = 0, READ IN (AT LEAST) EARLY ROWS INTO UPPER TRIANGULAR ARRAY.
C
      IF (K) 2000,160,180
  160 II = N + 1
      IN = N + N1
      DO 170 I = 2,LIM
      READ (2) (A(IJ), IJ = II,IN)
```

```
      II = IN + 1
  170 IN = IN + N - I
      GO TO 250
C
C NOW SKIP TO FORMATION OF NEW SUM, V, ALF, AND P.
C
C
C
C IF K IS G.T.0, FORM TAU AND GET Q (FROM OLD P AND V, NOW IN Q AND U.)
C
C IF MATRIX FITS, DON'T FRET ABOUT TAPES 2 AND 3.
C
C OTHERWISE GET SET FOR MORE READING AND WRITING.
C
  180 IF (INCORE) 2000,200,190
  190 REWIND 2
      REWIND 3
  200 SUM = 0.D0
      DO 210  I = K1,N
  210 SUM = SUM + Q(I)*U(I)
      TAU = .5 * ALF * SUM
      DO 220  I = K1,N
  220 Q(I) = Q(I) - TAU*U(I)
C
C CONVERT ROWS K+1...LIM TO ROWS OF A UPPER K+1
C
      II = K1K1
      IN = K1N
      DO 240  I = K1,LIM
      J = I
      DO 230  IJ = II,IN
      A(IJ) = A(IJ) - Q(I)*U(J) - Q(J)*U(I)
  230 J = J + 1
      II = IN + 1
  240 IN = IN + N - I
C
C IF K = N-2, NO NEW SUM, V, ALF, ETC. ARE NEEDED. KICK OUT AND MOP UP.
C
  250 SUM = 0.D0
      K1K2 = K1K1 + 1
      IF (N - K2) 2000,690,260
  260 DO 270  IJ = K1K2,K1N
  270 SUM = SUM + A(IJ)*A(IJ)
      Q(K+1) = A(K1K1)
      V(K+1) = SUN
      U(K+1) = SQRT(SUN)
C
C IF SUM = 0, ROW K+1 ALREADY CONFORMS TO TRI-DIAG FORM. MAKE SPECIAL
C   DEFINITION OF V AND ALF.
C
      IF (SUM) 2000,280,290
  280 V(K+2) = 1.
      ALF = 2.
      GO TO 320
C
C ORDINARY DEFINITION OF V AND ALF.
C
  290 IF (A(K1K2)) 310,310,300
  300 U(K+1) = -U(K+1)
  310 V(K+2) = A(K1K2) - U(K+1)
```

```
      ALF = 1. / (V(K+1) + ABS(A(K1K2)*U(K+1)))
  320 J = K1K1 + 2
      DO 330  I = K3,N
      V(I) = A(J)
  330 J = J + 1
C
C IF MATRIX WAS INITIALLY IN CORE, STORE V UPPER K+1 IN K+1 ST ROW OF A.
C
      IF (NUMBER) 336,2000,333
  333 A(K1K1) = ALF
      A(K1K1+1) = V(K+2)
      GO TO 440
C
C OTHERWISE, WRITE V UPPER K+1 ON TAPE 4.
C
  336 WRITE (4) ALF, (V(I), I = K2,N)
C
C IF MATRIX FAIL TO FIT, ELIMINATE ROW K+1 (NOW SUPERFLUOUS) AND MOVE
C   OTHER ROWS FORWARD TO MAKE ROOM FOR 1 OR MORE NEW ROWS.
C
      IF (INCORE) 2000,440,340
  340 LK1 = NK + 1
      MOVE = II - LK1
      J = LK1
      DO 350  I = 1,MOVE
      A(I) = A(J)
  350 J = J + 1
      II = MOVE + 1
      IN = NK + MOVE - M
C
C UPDATE LIM. BRING IN NEW ROWS AND CONVERT TO ROWS OF A UPPER K+1.
C
      NEXTM = LIMIT(NK1,ASIZE)
      LIMINC = LIM + 1
      LIM = K1 + NEXTM
      DO 430  I = LIMINC,LIM
  360 IF (KEY) 2000,370,380
  370 READ (2) (A(IJ), IJ = II,IN)
      GO TO 390
  380 READ (3) (A(IJ), IJ = II,IN)
  390 IF (K) 2000,420,400
  400 J = I
      DO 410  IJ = II,IN
      A(IJ) = A(IJ) - Q(I)*U(J) - Q(J)*U(I)
  410 J = J + 1
  420 II = IN + 1
  430 IN = IN + N - I
      K2K2 = 1
      K2N = NK1
      GO TO 450
  440 K2K2 = K1N + 1
      K2N = K1N + NK1
C
C COMPUTE POSITIONS K+2...LIM OF AV AND SUM UP TO LIM FOR THE LATE POSI-
C   TIONS, IF ANY. AT STEP I, ITH ELEMENT OF AV IS COMPLETED, AND EFFECT
C   OF COLUMN I ON ELEMENTS I+1...N IS TAKEN INTO ACCOUNT.
C
  450 II = K2K2
      IN = K2N
      DO 460  I = K2,N
```

```
  460 P(I) = 0.D0
      DO 500  I = K2,LIM
      J = I
      DO 470  IJ = II,IN
      P(I) = P(I) + A(IJ)*V(J)
  470 J = J + 1
      IF (N - I) 2000,500,480
  480 II1 = II + 1
      J = I + 1
      DO 490  IJ = II1,IN
      P(J) = P(J) + A(IJ)*V(I)
  490 J = J + 1
      II = IN + 1
  500 IN = IN + N - I
C
C IF SOME ROWS ARE STILL ON TAPE, READ THEM IN, 1 AT A TIME. EACH ROW IS
C   CONVERTED (IF K G.T. 0), THEN USED IN CALCULATION OF AV, THEN PUT ON
C   THE OTHER TAPE TO MAKE ROOM FOR THE NEXT ROW.
C
      IF (N - LIM) 2000,660,510
  510 II = 1
      IN = N - LIM
      LIMINC = LIM + 1
      DO 640  I = LIMINC,N
      IF (KEY) 2000,520,530
  520 READ (2) (B(IJ), IJ = II,IN)
      GO TO 540
  530 READ (3) (B(IJ), IJ = II,IN)
  540 IF (K) 2000,570,550
  550 J = I
      DO 560  IJ = II,IN
      B(IJ) = B(IJ) - Q(I)*U(J) - Q(J)*U(I)
  560 J = J + 1
  570 J = I
      DO 580  IJ = II,IN
      P(I) = P(I) + B(IJ)*V(J)
  580 J = J + 1
      IF (N - I) 2000,610,590
  590 J = I + 1
      II1 = II + 1
      DO 600  IJ = II1,IN
      P(J) = P(J) + B(IJ)*V(I)
  600 J = J + 1
  610 IF (KEY) 2000,620,630
  620 WRITE (3) (B(IJ),IJ = II,IN)
      GO TO 640
  630 WRITE (2) (B(IJ),IJ = II,IN)
  640 IN = IN - 1
  650 KEY = 1 - KEY
  660 CONTINUE
C
C PLACE NEW V AND P IN U AND Q.
C
  670 DO 680  I = K2,N
      U(I) = V(I)
  680 Q(I) = ALF * P(I)
  690 CONTINUE
C
C MOP UP BY COMPLETING ARRAYS OF DIAGONAL, OFF-DIAG AND SQUARE ELEMENTS.
C
```

```
      Q(N-1) = A(K1K1)
      U(N-1) = A(K1K1+1)
      V(N-1) = U(N-1)*U(N-1)
      Q(N) = A(K1K1+2)
  700 IF (NO) 710,2000,720
  710 WRITE (6,10) (Q(I), I = 1,N)
   10 FORMAT (/////18H TRI-DIAGONAL FORM////5X,18H DIAGONAL ELEMENTS//(5
     1X,1P8E15.7))
      WRITE (6,20) (U(I), I = 1,N1)
   20 FORMAT (///5X,22H SUB-DIAGONAL ELEMENTS//(5X,1P8E15.7))
  720 CALL OVERFL(I)
      GO TO (730,740),I
  730 MISS = 2
      GO TO 1000
  740 CALL PROPER(N,NUMMER,Q,U,V,B)
  750 MISS = 0
      GO TO 1000
 2000 MISS = 3
 1000 RETURN
      END
```

```
$IBFTC POLITE  M94/2,XR7,NODECK
      SUBROUTINE PROPER(N,NUMBER,D,OFFD,SEC,SIGMA)
      DIMENSION D(1),OFFD(1),SEC(1),SIGMA(1)
      CALL PREP(N,D,SEC,ROOT,LORD)
      N1 = N - 1
      BOUND = AMAX1(ABS (D(1))+ABS (OFFD(1)),ABS (OFFD(N1))+ABS (D(N)))
      IF (N - 2) 16,200,100
  100 DO  1   I = 2,N1
    1 BOUND = AMAX1(BOUND,ABS (OFFD(I-1)) + ABS (D(I)) + ABS (OFFD(I)))
  200 DO  2   I = 1,N
      SIGMA(I) = -BOUND
    2 OFFD(I) = BOUND
      LORD = 0
      RUTE = 1.0
      L = 0
    3 K = L + 1
      IF (K - NUMBER) 4,4,13
    4 ROOT = .5 * (SIGMA(K) + OFFD(K))
    5 DO 6  I = K,NUMBER
      IF (OFFD(K) - OFFD(I)) 7,6,7
    6 L = I
    7 IF (ROOT - RUTE) 8,3,8
    8 CALL DET
      DO 11  I = K,L
      IF(I -LORD) 9,9,10
    9 SIGMA(I) = ROOT
      GO TO 11
   10 OFFD(I) = ROOT
   11 CONTINUE
      RUTE = ROOT
      IF (ROOT) 4,12,4
   12 KING = LORD
      GO TO 4
   13 IF (KING) 14,16,14
   14 DO 15  I = 1,KING
```

```
   15 SIGMA(I) = OFFD(I)
   16 RETURN
      END
```

```
$IBFTC OUTER   M94/2,XR7,NODECK
      FUNCTION LIMIT(N,M)
      K = N
      L = 0
      DO 100  I = 1,N
      L = L + K
      K = K - 1
      IF (M - L) 120,110,100
  100 CONTINUE
      LIMIT = N
      RETURN
  110 LIMIT = I
      RETURN
  120 LIMIT = I - 1
      RETURN
      END
```

```
$IBMAP WORKER   100,M94/2,NOREF,NODECK
       ENTRY    PREP
       ENTRY    DET
PREP   CLA*     3,4                THIS SECTION IS ENTERED
       ALS      18                 ONCE FOR EACH EXECUTION
       SUB      =32767B17          OF PROPER. IT SETS UP THE
       STD      TEST               ADDRESS REFERENCES FOR DET.
       CLA      4,4
       SUB      =1
       STA      DIAG
       CLA      5,4
       SUB      =1
       STA      SQ
       SUB      =1
       STA      SQ1
       CLA      6,4
       STA      ROOT
       CLA      7,4
       STA      LORD
       TRA      8,4                RETURN FROM PREP
DET    SXA      XR1,1
       SXA      XR2,2              SAVE XR1 AND XR2.
       CLA      =1.
       STO      X                  INITIAL VALUES
       STZ      Y
       AXC      1,1                XR1 IS THE LOOP COUNTER AND
       AXT      0,2                XR2 KEEPS TRACK OF SIGNS.
SQ1    LDQ      0,1                B(I-1)**2
       FMP      Y                  DET(I-2)
       FRN                         SECOND TERM
       STO      Z
```

```
DIAG    CLA     0,1             A(I)
ROOT    FSB     0               ROOT
        FRN
        XCA
        FMP     X               DET(I-1)
        FRN                     FIRST TERM
        FSB     Z
        FRN                     DET(I)
        ZET     OVFLOW          CHECK FOR BAD EXPONENT.
        TRA     BADEXP          BETTER INVESTIGATE.
CHECK   TNZ     NONZER          CHECK FOR ZERO
        TXI     *+1,1,-1        O.K. SKIP A STEP.
SQ      ZET     0,1             BONA FIDE ZERO. CHECK P(I).
        CLA     =1.             EITHER WAY
        STO     X               RE-INITIALIZE X AND Y
        STZ     Y               AND COUNT 1 AGREEMENT.
        TXI     INC,2,1
NONZER  LDQ     X               NORMAL SITUATION. RE-DEFINE
        STQ     Y               X AND Y AND MAKE SIGN
        STO     X               COMPARISON.
        TMI     NEG
        TQP     STEP
        TXI     TEST,1,-1       NO CIGAR. X+ AND Y-.
NEG     TQP     INC             NO CIGAR. X- AND Y+.
STEP    TXI     *+1,2,1         SIGNS AGREE.
INC     TXI     *+1,1,-1        INCREMENT I
TEST    TXH     SQ1,1,0         REPEAT OR EXIT FROM LOOP.
LORD    SXA     0,2             RECORD RESULT.
XR1     AXT     0,1             RESTORE XR1 AND XR2.
XR2     AXT     0,2
        TRA     3,4             RETURN
BADEXP  STO     Z               SAVE.
        CLA     OVFLOW
        ANA     =1B15
        TNZ     OVFLOP          OVERFLOW.
UNFLOW  CLA     OVFLOW          MAKE SURE THAT THE
        ANA     =1B16           UNDERFLOW IS IN AC.
        TNZ     UNDER
        STZ     OVFLOW
        CLA     Z               RESTORE.
        TRA     CHECK
UNDER   CAL     X               DET(I) UNDERFLOWS. MULTIPLY
        ADD     =192B8          X AND Y BY 2**192 AND
        SLW     X               REPEAT ITH STEP.
        CAL     Y
        ADD     =192B8
        SLW     Y
        TRA     RESET
OVFLOP  CAL     X               DET(I) OVERFLOWS. DIVIDE
        SUB     =192B8          X AND Y BY 2**192 AND
        SLW     X               REPEAT ITH STEP.
        CAL     Y
        SUB     =192B8
        SLW     Y
RESET   STZ     OVFLOW          ERASE OVERFLOW INDICATION.
        TRA     SQ1
X       OCT     0
Y       OCT     0
Z       OCT     0
        END
```

```
$IBMAP HASTEN   60,M94/2,NOREF,NODECK
       ENTRY    PRELIM
       ENTRY    DOT
       ENTRY    PROLOG
       ENTRY    ADJ
PRELIM CLA      3,4
       SUB      =1
       STA      LOAD
       CLA*     4,4
       ALS      18
       COM
       STD      LOOP
       CLA      5,4
       SUB      =1
       STA      MULT
       CLA*     6,4
       SUB      =1B20
       ALS      18
       STD      INC
       TRA      1,4
DOT    SXA      XR1,1
       AXC      1,1
LOOP   TXL      INC,1,**
LOAD   LDQ      **,1
MULT   FMP      **,1
       DFAD     SUM
       DST      SUM
       TXI      LOOP,1,-1
INC    TXI      *+1,1,**
       SXD      LOOP,1
       CLA*     3,4
       ADD      MULT
       STA      MULT
       CLA      SUM
       FRN
XR1    AXT      0,1
       TRA      1,4
PROLOG CLA      3,4
       SUB      =1
       STA      ADD
       STA      GET
       CLA*     4,4
       ALS      18
       COM
       STD      LOOR
       CLA      5,4
       SUB      =1
       STA      MPY
       CLA*     6,4
       ADD      =1B20
       ALS      18
       STD      INF
       TRA      1,4
ADJ    SXA      XR1,1
       CLA      MPY
```

```
        SUB*     3,4
        STA      MPY
        LXD      LOOR,1
INF     TXI      *+1,1,**
        SXD      LOOR,1
        AXC      1,1
LOOR    TXL      XR1,1,**
        LDQ      SUM
MPY     FMP      **,1
ADD     FAD      **,1
        FRN
GET     STO      **,1
        TXI      LOOR,1,-1
SUMMER  CONTRL   SUMMER
        USE      SUMMER
SUM     BSS      2
        USE
        END
```

LISTING FOR DETERMINANT PROGRAMS

```
C     BUCKLING OF RECTANGULAR PLATES  DETERMINANT FORMULATION
      COMMON JI,JT,X,Y,JII,NBND,LOCOL,G,C,AA
      COMMON AD,BD,AKG,LIN,E,H,FS,IM,NJTS,ANU
      COMMON KLU,SX,DSX,SXMAX,SY,DSY,SYMAX,SXY,KLU4
      COMMON DSXY,SXYMAX,RS,Q,NCO,S7,JD,PX,PY, PXY
      COMMON APH,BTA,GAM,EN
      COMMON /DOODLE/KEY,KAY,VALUE
      DIMENSION JI(4),JT(65),X(65),Y(65),JII(50,4)
      DIMENSION NBND(65,4),AD(3),BD(3),JD(4),LOCOL(16)
      DIMENSION G(16,16),C(125,125),AA(4,3),AKG(16,16)
      DIMENSION RS(125),Q(125),PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      DO 42 I=1,65
      JT(I)=0
      X(I)=0.
   42 Y(I)=0.
      DO 43 J=1,4
      JI(J)=0
   43 JD(J)=0
      DO 44 K=1,3
      AD(K)=0.
   44 BD(K)=0.
      DO 45 L=1,16
      LOCOL(L)=0
      DO 45 M=1,16
      AKG(L,M)=0.
   45 G(L,M)=0.
      DO 46 N=1,50
      DSXY(N)=0.
      DSY(N)=0.
      DSX(N)=0.
      PXY(N)=0.
      PY(N)=0.
   46 PX(N)=0.
      DO 47 II=1,125
      RS(II)=0.
      Q(II)=0.
      DO 47 JJ=1,125
   47 C(II,JJ)=0.
      DO 48 KK=1,4
      DO 48 LL=1,65
   48 NBND(LL,KK)=0
      DO 49 MM=1,4
      DO 49 NN=1,50
   49 JII(NN,MM)=0
      DO 50 III=1,3
      DO 50 JJJ=1,4
   50 AA(JJJ,III)=0.
  311 KBUC=0
      D2=0.
      READ (5,99) NJTS,KLU,KLU2,KLU3,KLU4,KLU5,NEIG,LOT

      READ (5,98) ANU,E,H,NCO,S7
      IF(KLU )40,41,40
   41 WRITE (6,208)
```

```
      GO TO 5
   40 WRITE(6,209)
    5 TOL=1. * 10.**(-LOT)
      WRITE(6,27)KLU,KLU2,KLU3,KLU4,KLU5,LOT
      WRITE(6,25) ANU,E,H,NCO,S7
      IF(KLU3)4,9,4
    4 READ (5,60) XSZ,YSZ,XN1,YN1,NMEM
      WRITE(6,61) XSZ,YSZ,XN1,YN1,NJTS,NMEM
      XN2=XN1
      YN2=YN1
      DO 1 I=1,NJTS
      X(I)=((XN1-XN2)/(XN1))*XSZ
      Y(I)=((YN1-YN2)/(YN1))*YSZ
      JT(I)=I
      XN2=XN2-1.
      IF(XN2)3,1,1
    3 XN2=XN1
      YN2=YN2-1.
    1 CONTINUE
      WRITE(6,62)
      WRITE(6,63) (JT(I),X(I),Y(I),I=1,NJTS)
      NN1=XN1
      NN2=1
      NN3=XN1+2.
      NN4=XN1
      DO 64 MEM=1,NMEM
      JII(MEM,1)=NN2
      JII(MEM,2)=NN2+1
      JII(MEM,3)=NN3
      JII(MEM,4)=NN3+1
      NN2=NN2+1
      NN3=NN3+1
      NEM=MEM-NN4
      IF(NEM)64,65,65
   65 NN2=NN2+1
      NN3=NN3+1
      NN4=NN1+MEM
   64 CONTINUE
      GO TO 70
    9 READ(5,100) (JT(I),X(I),Y(I),I=1,NJTS)
      WRITE (6,200)
      WRITE (6,201) (JT(I),X(I),Y(I),I=1,NJTS)
      WRITE(6,104)
    8 READ(5,101)MEM,NTYPE,JI(1),JI(2),JI(3),JI(4)
      IF(MEM-99) 6,7,6
    6 DO 10 I=1,4
      JII(MEM,I)=JI(I)
   10 CONTINUE
      KODE=MEM
      GO TO 8
    7 CONTINUE
      NMEM=KODE
   70 WRITE(6,66) ((JII(K,N),N=1,4),K=1,MEM)

      IF(KLU4) 81,80,81
   80 READ(5,106) APH,BTA,GAM,EN,DEN
```

```
      WRITE(6,107) APH,BTA,GAM,EN,DEN
      GO TO 82
   81 WRITE(6,96)
      DO 71 MEM=1,NMEM
      READ(5,103) PX(MEM),DSX(MEM),PY(MEM),DSY(MEM),PXY(MEM),DSXY(MEM)
   71 WRITE(6,97) PX(MEM),DSX(MEM),PY(MEM),DSY(MEM),PXY(MEM),DSXY(MEM)
   82 IF(KLU5) 84,83,84
   83 IBUC=1
      DO 11 I=1,NJTS
      DO 11 J=1,4
   11 NBND(I,J)=0
      WRITE(6,207)
      DO 12 I=1,NJTS
      READ(5,102) JOINT,JD(1),JD(2),JD(3),JD(4)
      WRITE(6,203) JOINT,JD(1),JD(2),JD(3),JD(4)
      DO 13 J=1,4
      IF(JD(J)) 13,13,14
   14 NBND(JOINT,J)=IBUC
      IBUC=IBUC+1
      IF(IBUC-125) 13,13,15
   15 WRITE(6,204)
      CALL EXIT
   13 CONTINUE
   12 CONTINUE
      IBUC=IBUC-1
   84 READ(5,500) STOPIT
      IF(KLU2) 95,85,95
   95 WRITE(6,23)
      WRITE(6,24) (NBND(I,1),NBND(I,2),NBND(I,3),NBND(I,4), I=1,JOINT)
   85 CONTINUE
      WRITE(6,52)
   18 DO 896 LL=1,IBUC
      DO 896 KK=1,IBUC
  896 C(KK,LL)=0.
      FIRST=0.
      DO 16 IM=1,MEM
      IM=IM
      LIN=4
      IF(FIRST) 809,810,809
  809 IF(KLU4) 810,811,810
  811 IF(KLU3-1) 810,814,810
  810 CALL BSTIF
      FIRST=1.
  814 CALL STACK
   16 CONTINUE
      KBUC=KBUC+1
      IF(KBUC-100) 30,311,311
   30 FIRST=0.
      DO 17 IM=1,MEM
      IM=IM
      LIN=4
      IF(FIRST) 815,816,815
  815 IF(KLU4) 816,817,816
  817 IF(KLU3-1) 816,818,816
  816 CALL MSTIF
      FIRST=1.
```

```
  818 CALL STACK
   17 CONTINUE
      DO 57 K=1,IBUC
      DO 57 L=K,IBUC
   57 C(K,L)=C(L,K)
      IF(KLU2) 32,33,32
   32 WRITE(6,206) ((C(I,J),J=1,IBUC),I=1,IBUC)
C
C     FIND THE DETERMINANT OF THE C MATRIX
C

   33 DO 305 LL=1,IBUC
  305 RS(LL)=0.0
      D1=1.
      NP=125
      M=IMECF(NP,IBUC,1,C,RS,D1,Q)
      GO TO (306,307,308),M
  306 WRITE(6,53) KBUC,D1,KFY,KAY,VALUE
      D1=VALUE
      GO TO 701
  307 WRITE (6,54)
   54 FORMAT(9H OVERFLOW)
      CALL EXIT
  308 WRITE(6,700)
      GO TO 309
  700 FORMAT(14H C IS SINGULAR)
  701 TEST=ABS(D1+D2)-ABS(D1-D2)
      IF(TEST)501,502,502
  501 WRITE(6,600)
      IF(KLU4) 87,88,87
   88 FSX=DEN
      GO TO 89
   87 FSX=0.
      FSY=0.
      FSXY=0.
      DO 400 II=1,NMEM
      DPX=ABS(DSX(II))
      DPY=ABS(DSY(II))
      DPXY=ABS(DSXY(II))
      FSX=AMAX1(DPX,FSX)
      FSY=AMAX1(DPY,FSY)
  400 FSXY=AMAX1(DPXY,FSXY)
   89 IF(FSX) 72,73,72
   72 ABVAL=ABS(FSX)
      GO TO 74
   73 IF(FSY)75,76,75
   75 ABVAL=ABS(FSY)
      GO TO 74
   76 IF(FSXY)77,78,77
   78 WRITE(6,105)
      CALL EXIT
   77 ABVAL=ABS(FSXY)
   74 IF(ABVAL- TOL)309,309,36
  502 D2=D1
      IF(KLU4) 35,830,35
  830 FN=EN+DEN
```

```
      GO TO 18
   35 DO 34 IN=1,NMEM
      PX(IN)=PX(IN)+DSX(IN)
      PY(IN)=PY(IN)+DSY(IN)
   34 PXY(IN)=PXY(IN)+DSXY(IN)
      GO TO 18
   36 IF(KLU4) 86,94,86
   94 EN=EN-DEN
      DEN=.5*DEN
      GO TO 830
   86 DO 37 IO=1,NMEM
      PX(IO)=PX(IO)-DSX(IO)
      PY(IO)=PY(IO)-DSY(IO)
      PXY(IO)=PXY(IO)-DSXY(IO)
      DSX(IO)=.5*DSX(IO)
      DSY(IO)=.5*DSY(IO)
   37 DSXY(IO)=.5*DSXY(IO)
      GO TO 35
  309 WRITE(6,29)
      IF(KLU) 1000,1001,1000
 1001 IF(KLU4) 90,890,90
  890 ENX=APH*EN
      ENY=BTA*EN
      ENXY=GAM*EN
      WRITE(6,28) ENX,ENY,ENXY
      GO TO 1009
   90 WRITE(6,108)
      WRITE(6,26)(PX(II),PY(II),PXY(II),II=1,NMEM)
      GO TO 1009
 1000 D=E*H**3/(12.*(1.-ANU**2))
      IF(KLU4)91,92,91
   92 ENX=APH*EN*D
      ENY=BTA*EN*D
      ENXY=GAM*EN*D
      WRITE(6,31)
      WRITE(6,109) ENX,ENY,ENXY
      GO TO 1009
   91 DO 38 JJ=1,NMEM
      PX(JJ)=PX(JJ)*D
      PXY(JJ)=PXY(JJ)*D
   38 PY(JJ)=PY(JJ)*D
      WRITE(6,207)
      WRITE(6,26)(PX(II),PY(II),PXY(II),II=1,NMEM)
 1009 IF(STOPIT) 67,311,67
   67 CALL EXIT
   21 FORMAT(1H0,13HTHE JII ARRAY)
   22 FORMAT(11H0 JII ARRAY//(6I5))
   23 FORMAT(1H0,14HTHE NBND ARRAY)
   24 FORMAT(1H0,I5,    5X,I5,    5X,I5,5X,I5)
   25 FORMAT(1H0,4HANU=,E16.8,3X,2HE=,E16.8,3X,2HH=,E16.8,3X,4HNCO=,I5,3
     1X,3HS7=,E16.8)
   26 FORMAT(1H0,E15.8,5X,E15.8,5X,E15.8)
   27 FORMAT(1H0,4HKLU=,I5,3X,5HKLU2=,I5,3X,5HKLU3=,I5,3X,5HKLU4=,I5,3X,
     15HKLU5=,I5,3X,4HLOT=,I2)
   28 FORMAT(1H0,5HNX/D=E15.8,5X,5HNY/D=E15.8,5X,6HNXY/D=E15.8)
   29 FORMAT(22H0 BUCKLING STRESSES ARE)
```

```
 31 FORMAT(1H0,7X,2HNX,18X,2HNY,18X,3HNXY)
 52 FORMAT(1H0,5X,5HCYCLE,10X,11HDETERMINANT,12X,3HKEY,12X,3HKAY,13X,5
   1HVALUE)
 53 FORMAT(1H0,6X,I3,9X,E16.8,5X,I10,5X,I10,5X,E16.8)
 60 FORMAT(4F6.1, I2)
 61 FORMAT(1H0,4HXSZ=,F6.1,3X,4HYSZ=,F6.1,3X,4HXN1=,F6.1,3X,4HYN1=,F6.
   11,3X,5HNJTS=,I3,3X,5HNMEM=,I3)
 62 FORMAT(1H0,1X,5HJOINT,9X,1HX,9X,1HY)
 63 FORMAT(1H0,I5,7X,F7.2,3X,F7.2)
 66 FORMAT(11H0 JII ARRAY//(4I5))
 96 FORMAT(1H0,5X,2HPX,13X,3HDSX,14X,2HPY,13X,3HDSY,14X,3HPXY,12X,4HDS
   1XY)
 97 FORMAT(1H0,E15.8,1X,E15.8,1X,E15.8,1X,E15.8,1X,E15.8,1X,E15.8)
 98 FORMAT(3E14.8,I7,F9.1)
 99 FORMAT(8I3)
100 FORMAT(I4,2E12.1)
101 FORMAT(I3,I2,4I3)
102 FORMAT(5I3)
103 FORMAT(6F13.8)
104 FORMAT(2X,3HMEM,2X,5HNTYPE,4X,2HJ1,2X,2HJ2,2X,2HJ3,2X,2HJ4,7X,2HPX
   1,10X,2HPY,10X,3HPXY)
105 FORMAT(1H0,35HTHERE ARE NO DSX DSY OR DSXY VALUES)
106 FORMAT(5F10.2)
107 FORMAT(1H0,4HAPH=,F6.4,3X,4HBTA=,F6.4,3X,4HGAM=,F6.4,3X,3HEM=,F9.4
   1,3X,4HDEN=,F9.4)
108 FORMAT(1H0,6X,4HNX/D,16X,4HNY/D,16X,5HNXY/D)
109 FORMAT(1H0,3HNX=,E16.8,5X,3HNY=,E16.8,5X,4HNXY=,E16.8)
200 FORMAT(34X,5HJOINT,9X,1HX,16X,1HY)
201 FORMAT(33X,I4,5X,2E16.7)
203 FORMAT(3X,I4,4X,I4,7X,I4,6X,I4,6X,I4)
204 FORMAT( 37HTHE PROBLEM EXCEEDS AVAILABLE STORAGE )
206 FORMAT(15H0 C AFTER BSTIF//(6E16.8))
207 FORMAT(4X,4HNODE,6X,1HW,9X,3H WX,7X,3H WY,6X,5H  WXY)
208 FORMAT(1H1,31HTHIS IS AN ELASTIC BUCKLING RUN)
209 FORMAT(1H1,30HTHIS IS A PLASTIC BUCKLING RUN)
500 FORMAT(F5.0)
600 FORMAT(1H0,29HTHE SIGN OF DET C HAS CHANGED)
     END
```

DETERMINANT PROGRAM

```
      SUBROUTINE BSTIF
      COMMON JI,JT,X,Y,JII,NBND,LOCOL,G,C,AA
      COMMON AD,BD,AKG,LIN,E,H,ES,IM,NJTS,ANU
      COMMON KLU,SX,DSX,SXMAX,SY,DSY,SYMAX,SXY,KLU4
      COMMON DSXY,SXYMAX,RS,Q,NCO,S7,JD,PX,PY, PXY
      COMMON APH,BTA,GAM,EN
      DIMENSION JI(4),JT(65),X(65),Y(65),JII(50,4)
      DIMENSION NBND(65,4),AD(3),BD(3),JD(4),LOCOL(16)
      DIMENSION G(16,16),C(125,125),AA(4,3),AKG(16,16)
      DIMENSION RS(125),Q(125),PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      DO 20 K  = 1,LIN
      DO 12 I  = 1,NJTS
      IF(JT(I)-JII( IM,K)) 12,14,12
   12 CONTINUE
      WRITE (6,16)
      CALL EXIT
   16 FORMAT (18H NO VALUE IN TABLE)
   14 AA(K,1)  = X(I)
      AA(K,2)  = Y(I)
   20 CONTINUE
   23 DO 25 I  = 1,2
      AD(I)    = AA(2,I) - AA(1,I)
   25 BD(I)    = AA(3,I) - AA(1,I)
      AL12     = (AD(1)**2+AD(2)**2+AD(3)**2)**.5
      AL13     = (BD(1)**2+BD(2)**2+BD(3)**2)**.5
      A=AL12
      B=AL13
      AB=A*B
      AOB=A/B
      AOB2=AOB**2
      AOB3=1./AOB2
      BOA=B/A
      BOA2=BOA**2
      B1=156./35.
      B2=36./25.
      B3=78./35.
      B4=3./25.
      B5=22./35.
      B6=54./35.
      B7=13./35.
      B8=27./35.
      B9=52./35.
      B10=4./25.
      B11=11./35.
      B12=26./35.
      B13=13./70.
      B14=18./35.
      B15=9./35.
      B16=4./35.
      B17=3./35.
      B18=3./70.
      B19=22./105.
      B20=2./35.
      B21=11./105.
```

```
      B22=13./105.
      B23=13./210.
      B24=4./105.
      B25=4./225.
      B26=2./105.
      B27=1./225.
      B28=2./70.
      B29=1./70.
      SODPLA=2.+120.*ANU
      SXDPLA=2.+10.*ANU
      ENDPLA=2.+20.*ANU
      IF(KLU)27,26,27
   26 C1=BOA2
      C2=0.
      C3=1.
      C4=0.
      C5=AOB2
      DS=1./(A*B)
      GO TO 28
   27 D=(E*H**3)/(12.*(1.-ANU**2))
      IF(KLU4) 31,30,31
   30 SX=(APH*EN)*(D/H)
      SY=(BTA*EN)*(D/H)
      SXY=(GAM*EN)*(D/H)
      GO TO 33
   31 SX=PX(IM)*D/H
      SY=PY(IM)*D/H
      SXY=PXY(IM)*D/H
   33 BN=NCO
      SIG=(SX**2+SY**2-SX*SY+3.*SXY**2)
      EFS=SQRT(SIG)
      ET=1./(1.+(3./7.)*BN*(EFS/S7)**(NCO-1))
      ES=1./(1.+(3./7.)*(EFS/S7)**(NCO-1))
      DS=ES/(A*B)
      DOD=1.-ET/ES
      C1=BOA2*(1.-.75*DOD*(SX/EFS)**2)
      C2=3.*DOD*(SX*SXY/(EFS**2))*BOA
      C3=1.-.75*((SX*SY+2.*SXY**2)/(EFS**2))*DOD
      C4=3.*DOD*((SY*SXY)/EFS**2)*AOB
      C5= AOB2*(1.-.75*DOD*(SY**2/EFS**2))
      SODPLA=(62.-((3.*SXY**2+91.5*SX*SY)/SIG)*DOD)/C3
      SXDPLA=(7.-((3.*SXY**2+9.*SX*SY)/SIG)*DOD)/C3
      ENDPLA=(12.-((3.*SXY**2+16.5*SX*SY)/SIG)*DOD)/C3
   28 DO 200 I=1,16
      DO 200 J=1,16
  200 G(I,J)=0.0
      G(16,1)=(-B13* C1 + .01* C3 * 2. - B13* C5 )
      G(15,1)=( B7 * C1 - B4 * C3 * 2. + B8 * C5 )
      G(14,1)=(+B8 * C1 - B4 * C3 * 2. + B7 * C5 )
      G(13,1)=(-B6 * C1 + B2 * C3 * 2. - B6 * C5 )
      G(12,1)=(-B13* C1 + ((SXDPLA/100.) * C3) + B11* C5 )
      G(11,1)=(-B7 * C1 + B4 * C3 * 2. + B3 * C5 )
      G(10,1)=(+B8 * C1 - B4 * SXDPLA * C3 - B5 * C5 )
      G(9,1)=( B6 * C1 - B2 * C3 * 2. - B1 * C5 )
      G( 8,1)=( B11* C1 + ((SXDPLA/100.) * C3) - B13* C5 )
      G(7,1)=(-B5 * C1 - B4 * SXDPLA * C3 +B8 * C5 )
```

```
G(6,1)=(+B3 * C1 + B4 * C3 * 2. - B7 * C5 )
G(5,1)=(-B1 * C1 - B2 * C3 * 2. + B6 * C5 )
G( 4,1)=( B11* C1 + ((ENDPLA/100.) * C3)  + B11* C5)
G(3,1)=( B5 * C1 + B4 * SXDPLA * C3 +B3 * C5 )
G(2,1)=(+B3 * C1 + B4 * SXDPLA * C3 +B5 * C5 )
G(1,1)=( B1 * C1 + B2 * C3 * 2. + B1 * C5 )
G(16,2)=(-B23*C1 - ((C3/300.) * 2.) - B18* C5 )
G(15,2)=( B13* C1 - ((C3/100.) * 2.) + B13* C5 )
G(14,2)=( B15* C1 +((C3/25. ) * 2.) + B17* C5 )
G(13,2)=(-B8 * C1 + B4 * C3 * 2. -B7 * C5 )
G(12,2)=(-B22* C1 + ((C3/75. ) * 2.) + B20*C5 )
G(11,2)=(-B13* C1 + ((SXDPLA/100.) * C3) + B11* C5 )
G(10,2)=( B14* C1 - B10* C3 * 2. -B16* C5 )
G(9,2 )=(+B8 * C1 - B4 * SXDPLA * C3 - B5 * C5 )
G( 8,2)=( B21* C1 - ((SXDPLA/300.) * C3) -B18* C5 )
G( 7,2)=(-B11* C1 - ((SXDPLA/100.) * C3) + B13* C5 )
G( 6,2)=( B12* C1 - ((C3/25. ) * 2.) - B17* C5 )
G(5,2 )=(-B3 * C1 - B4 * C3 * 2. + B7 * C5 )
G( 4,2)=( B19* C1 + ((SXDPLA/75. ) * C3) + B20* C5 )
G( 3,2)=( B11* C1 + ((SODPLA/100.) * C3) + B11* C5 )
G(2,2 )=( B9 * C1 + B10* C3 * 2. + B16* C5 )
G(16,3)=( -B18* C1 - ((C3/300.) * 2.) - B23* C5 )
G(15,3)=( B17* C1 + ((C3/25.  ) * 2.) + B15* C5)
G(14,3)=( B13* C1 - ((C3/100.)* 2.) + B13* C5 )
G(13,3)=(-B7 * C1 + B4 * C3 * 2. -B8 * C5 )
G(12,3)=(-B18* C1 - ((SXDPLA/300.) * C3) + B21* C5 )
G(11,3)=(-B17* C1 - ((C3/25. ) * 2.) + B12*C5 )
G(10,3)=( B13* C1 - ((SXDPLA/100.) * C3) - B11* C5 )
G(9,3 )=( B7 * C1 - B4 * C3 * 2. -B3 * C5 )
G( 8,3)=( B20* C1 + ((C3/75. ) * 2.) - B22*C5 )
G(7,3 )=(-B16* C1 - B10* C3 * 2. +B14* C5 )
G( 6,3)=( B11* C1 + ((SXDPLA/100.) * C3) - B13* C5)
G(5,3 )=(-B5 * C1 - B4 * SXDPLA * C3 + B8 * C5 )
G( 4,3)=( B20* C1 + ((SXDPLA/75. ) * C3 ) + B19* C5 )
G(3,3 )=( B16* C1 + B10* C3 * 2. +B9 * C5 )
G(16,4)=(-B29* C1 + ((C3/900.) * 2.) - B29* C5 )
G(15,4)=( B18* C1 + ((C3/300.) * 2.) + B23* C5 )
G(14,4)=( B23* C1 + ((C3/300.) * 2.) + B18* C5 )
G(13,4)=(-B13* C1 + ((C3/100.) * 2. )- B13* C5 )
G(12,4)=(-B28* C1 - B27* C3 * 2. + B26 * C5 )
G(11,4)=(-B18* C1 - ((SXDPLA/300.) * C3 ) + B21* C5 )
G(10,4)=( B22* C1 - ((C3/75. ) * 2. ) - B20* C5 )
G( 9,4)=( B13 * C1 - ((SXDPLA/100.) * C3 ) - B11* C5 )
G(8,4 )=( B26* C1 - B27* C3 * 2. - B28 * C5 )
G( 7,4)=(-B20* C1 - ((C3/75. ) * 2. ) + B22* C5 )
G( 6,4)=( B21* C1 - ((SXDPLA/300.) * C3 ) - B18 *C5)
G( 5,4)=(-B11*C1 - ((SXDPLA/100.) * C3 ) + B13*C5 )
G(4,4 )=( B24* C1 + B25* C3 * 2. + B24 * C5 )
G(5,5)=G(1,1)
G(6,5)=-G(2,1)
G(7,5)= G(3,1)
G(8,5)=-G(4,1)
G(6,6)=G(2,2)
G(7,6)=-G(3,2)
G(8,6)= G(4,2)
G(7,7)= G(3,3)
```

```
G(8,7)=-G(4,3)
G(8,8)= G(4,4)
G(9,9)= G(1,1)
G(10,9)=G(2,1)
G(11,9)=-G(3,1)
G(12,9)=-G(4,1)
G(10,10)= G(2,2)
G(11,10)=-G(3,2)
G(12,10)=-G(4,2)
G(11,11)= G(3,3)
G(12,11)= G(4,3)
G(12,12)= G(4,4)
G(13,13)=G(1,1)
G(14,13)=-G(2,1)
G(15,13)=-G(3,1)
G(16,13)= G(4,1)
G(14,14)= G(2,2)
G(15,14)= G(3,2)
G(16,14)=-G(4,2)
G(15,15)= G(3,3)
G(16,15)=-G(4,3)
G(16,16)= G(4,4)
G(9,5)=  G(13,1)
G(10,5)=G(13,2)
G(11,5)=-G(13,3)
G(12,5)=-G(13,4)
G(9,6)=G(14,1)
G(10,6)= G(14,2)
G(11,6)=-G(14,3)
G(12,6)=-G(14,4)
G(9,7)=-G(15,1)
G(10,7)=-G(15,2)
G(11,7)= G(15,3)
G(12,7)=G(15,4)
G(9,8)=-G(16,1)
G(10,8)=-G(16,2)
G(11,8)=+G(16,3)
G(12,8)= G(16,4)
G(13,5)= G(9,1)
G(14,5)=-G(9,2)
G(15,5)=-G(9,3)
G(16,5)= G(9,4)
G(13,6)=-G(10,1)
G(14,6)= G(10,2)
G(15,6)= G(10,3)
G(16,6)=-G(10,4)
G(13,7)=-G(11,1)
G(14,7)= G(11,2)
G(15,7)=+G(11,3)
G(16,7)=-G(11,4)
G(13,8)= G(12,1)
G(14,8)=-G(12,2)
G(15,8)=-G(12,3)
G(16,8)= G(12,4)
G(13,9)= G(5,1)
G(14,9)= G(6,1)
```

```
      G(15,9)=-G(7,1)
      G(16,9)=-G(8,1)
      G(13,10)= G(5,2)
      G(14,10)= G(6,2)
      G(15,10)=-G(7,2)
      G(16,10)=-G(8,2)
      G(13,11)=-G(5,3)
      G(14,11)=-G(6,3)
      G(15,11)= G(7,3)
      G(16,11)= G(8,3)
      G(13,12)=-G(5,4)
      G(14,12)=-G(6,4)
      G(15,12)= G(7,4)
      G(16,12)= G(8,4)
      DO 210 J=1,16
      DO 210 I=J,16
  210 G(J,I)=G(I,J)
  245 DO 250 I=1,16
       DO 250 J=1,16
  250 AKG(I,J)=G(I,J)*DS
      IF(KLU)257,256,257
  257 DO 350 I=1,16
      DO 350 J=1,16
  350 G(I,J)=0.
      G(2,2)=-C2*.25
      G(3,2)=-.1*(C2+C4)
      G(3,3)=-.25*C4
      G(4,1)= .1*(C2+C4)
      G(5,3)=-.5*C4
      G(5,4)=-.1*(C2+C4)
      G(6,3)=G(4,1)
      G(6,4)=.05*C2+(1./60.)*C4
      G(6,6)=-G(2,2)
      G(7,1)=-G(5,3)
      G(7,2)=G(4,1)
      G(7,6)=G(3,2)
      G(7,7)=-G(3,3)
      G(8,1)=G(5,4)
      G(8,2)=-G(6,4)
      G(8,5)=G(4,1)
      G(9,2)=-.5*C2
      G(9,4)= G(5,4)
      G(9,6)=-G(9,2)
      G(9,7)= G(5,3)
      G(9,8)=G(4,1)
      G(10,1)=-G(9,2)
      G(10,3)=G(4,1)
      G(10,5)= G(9,2)
      G(10,6)= -G(2,2)
      G(10,7)=  G(3,2)
      G(10,8)=  G(6,4)
      G(10,10)= -G(2,2)
      G(11,2)=G(4,1)
      G(11,4)=(1./60.)*C2+.05*C4
      G(11,5)= -G(5,3)
      G(11,6)= G(3,2)
```

```
      G(11,7)= -G(3,3)
      G(11,8)= -G(11,4)
      G(11,10)= G(3,2)
      G(11,11)=-G(3,3)
      G(12,1)= G(5,4)
      G(12,3)=-G(11,4)
      G(12,5)=G(4,1)
      G(12,6)=-G(6,4)
      G(12,7)= G(11,4)
      G(12,8)=-(1./120.)*(C2+C4)
      G(12,9)=G(4,1)
      G(13,2)=-G(9,2)
      G(13,3)=-G(5,3)
      G(13,4)=G(4,1)
      G(13,6)= G(9,2)
      G(13,8)= G(5,4)
      G(13,11)= G(5,3)
      G(13,12)= G(5,4)
      G(14,1)= G(9,2)
      G(14,2)= G(2,2)
      G(14,3)= G(3,2)
      G(14,4)=-G(6,4)
      G(14,5)=-G(9,2)
      G(14,7)=G(4,1)
      G(14,11)=G(4,1)
      G(14,12)= G(6,4)
      G(14,14)= G(2,2)
      G(15,1)= G(5,3)
      G(15,2)= G(3,2)
      G(15,3)= G(3,3)
      G(15,4)=-G(11,4)
      G(15,6)=G(4,1)
      G(15,8)= G(11,4)
      G(15,9)=-G(5,3)
      G(15,10)=G(4,1)
      G(15,14)= G(3,2)
      G(15,15)= G(3,3)
      G(16,1)=G(4,1)
      G(16,2)= G(6,4)
      G(16,3)=G(11,4)
      G(16,4)=-G(12,8)
      G(16,5)= G(5,4)
      G(16,7)=-G(11,4)
      G(16,9)= G(5,4)
      G(16,10)=-G(6,4)
      G(16,13)= G(4,1)
      DO 360 I=1,16
      DO 360 J=I,16
  360 G(I,J)=G(J,I)
      DO 320 I=1,16
      DO 320 J=1,16
  320 AKG(I,J)=AKG(I,J)+DS*G(I,J)
  256 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MSTIF
      COMMON JI,JT,X,Y,JII,NBND,LOCOL,G,C,AA
      COMMON AD,BD,AKG,LIN,E,H,ES,IM,NJTS,ANU
      COMMON KLU,SX,DSX,SXMAX,SY,DSY,SYMAX,SXY,KLU4
      COMMON DSXY,SXYMAX,RS,Q,NCU,S7,JD,PX,PY, PXY
      COMMON APH,BTA,GAM,EN
      DIMENSION JI(4),JT(65),X(65),Y(65),JII(50,4)
      DIMENSION NBND(65,4),AD(3),BD(3),JD(4),LOCOL(16)
      DIMENSION G(16,16),C(125,125),AA(4,3),AKG(16,16)
      DIMENSION RS(125),Q(125),PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
C     TABLE LOOKUP FOLLOWS
      DO 20 K  = 1,LIN
      DO 12 I  = 1,NJTS
      IF(JT(I)-JII( IM,K)) 12,14,12
   12 CONTINUE
      WRITE(6,16)
      CALL EXIT
   16 FORMAT (18H NO VALUE IN TABLE)
   14 AA(K,1)  = X(I)
      AA(K,2)  = Y(I)
   20 CONTINUE
   23 DO 25 I  = 1,2
      AD(I)    = AA(2,I) - AA(1,I)
   25 BD(I)    = AA(3,I) - AA(1,I)
      AL12     = (AD(1)**2+AD(2)**2+AD(3)**2)**.5
      AL13     = (BD(1)**2+BD(2)**2+BD(3)**2)**.5
      A=AL12
      B=AL13
      AB=A*B
      AOB=A/B
      AOB2=AOB**2
      AOB3=1./AOB2
      BOA=B/A
      BOA2=BOA**2
      DO 260 I=1,16
      DO 260 J=1,16
  260 G(I,J)=0.0
      D1=1872.
      D2=156.
      D3=208.
      D4=264.
      D5=22.
      D6=48.
      D7=88./3.
      D8=4.
      D9=16./3.
      D10=52.
      D11=22./3.
      D12=4./3.
      D13=648.
      D14=54.
      D15=13.
      D16=72.
      D17=52./3.
```

```
      D18=36.
      D19=3.
      D20=18.
      D21=13./3.
      D22=1.
      IF(KLL4) 41,40,41
   40 SX=APH*BOA*EN
      SY=BTA*AOB*EN
      GO TO 285
   41 SX=PX(IM)*BCA
      SY=PY(IM)*ACB
  285 G(1,1 )=SX * D1 + SY * D1
      G(2,1 )=SX * D2 + SY * D4
      G(2,2 )=SX * D3 + SY * D6
      G(3,1 )=SX * D4 + SY * D2
      G(3,2 )=SX * D5 + SY * D5
      G(3,3 )=SX * D6 + SY * D3
      G(4,1 )=SX * D5 + SY * D5
      G(4,2 )=SX * D7 + SY * D8
      G(4,3)=+SX*D8+ SY*D7
      G(4,4 )=SX * D9 + SY * D9
      G(5,1 )=SX*(-D1)+ SY * D13
      G(5,2 )=SX*(-D2)+ SY * D2
      G(5,3 )=SX*(-D4)+ SY * D14
      G(5,4 )=SX*(-D5)+ SY * D15
      G(6,1 )=SX * D2 + SY *(-D2)
      G(6 ,2 )=SX*(-D10)+SY *(-D18)
      G(6 ,3 )=SX * D5 + SY *(-D15)
      G(6 ,4 )=SX*(-D11)+SY *(-D19)
      G(7 ,1 )=SX*(-D4)+ SY * D14
      G(7 ,2 )=SX*(-D5)+ SY * D15
      G(7 ,3 )=SX*(-D6)+ SY * D16
      G(7 ,4 )=SX*(-D8)+ SY * D17
      G(8 ,1 )=SX * D5 + SY *(-D15)
      G(8, 2 )=SX*(-D11)+ SY*(-D19)
      G(8 ,3 )=SX * D8 + SY *(-D17)
      G(8 ,4 )=SX*(-D12)+SY *(-D8)
      G(9 ,1 )=SX * D13+ SY *(-D1)
      G(9 ,2 )=SX * D14+ SY *(-D4)
      G(9 ,3 )=SX * D2 + SY *(-D2)
      G(9 ,4 )=SX * D15+ SY *(-D5)
      G(10,1 )=SX * D14+ SY *(-D4)
      G(10,2 )=SX * D16+ SY *(-D6)
      G(10,3 )=SX * D15+ SY *(-D5)
      G(10,4 )=SX * D17+ SY *(-D8)
      G(11,1 )=SX*(-D2)+ SY * D2
      G(11,2 )=SX*(-D15)+SY * D5
      G(11,3 )=SX*(-D18)+SY *(-D10)
      G(11,4 )=SX*(-D19)+SY *(-D11)
      G(12,1 )=SX*(-D15)+SY * D5
      G(12,2 )=SX*(-D17)+SY * D8
      G(12,3 )=SX*(-D19)+SY *(-D11)
      G(12,4 )=SX*(-D8)+ SY *(-D12)
      G(13,1 )=SX*(-D13)+SY *(- D13)
      G(13,2 )=SX*(-D14)+SY *(-D2)
      G(13,3 )=SX*(-D2)+ SY *(-D14)
```

```
G(13,4)=-SX*D15-SY*D15
G(14,1 )=SX * D14 + SY * D2
G(14,2 )=SX*(-D20)+ SY * D18
G(14,3 )=SX * D15 + SY * D15
G(14,4)=-SX*D21-SY*D19
G(15,1 )=SX * D2  + SY * D14
G(15,2 )=SX * D15 + SY * D15
G(15,3 )=SX * D18 + SY *(-D20)
G(15,4 )=SX * D19 + SY *(-D21)
G(16,1 )=SX*(-D15)+ SY *(-D15)
G(16,2 )=SX * D21 + SY *(-D19)
G(16,3 )=SX*(-D19)+ SY  *D21
G(16,4 )=SX * D22 + SY * D22
G(5,5)=G(1,1)
G(6,5)=-G(2,1)
G(7,5)= G(3,1)
G(8,5)=-G(4,1)
G(6,6)=G(2,2)
G(7,6)=-G(3,2)
G(8,6)= G(4,2)
G(7,7)= G(3,3)
G(8,7)=-G(4,3)
G(8,8)= G(4,4)
G(9,9)= G(1,1)
G(10,9)=G(2,1)
G(11,9)=-G(3,1)
G(12,9)=-G(4,1)
G(10,10)= G(2,2)
G(11,10)=-G(3,2)
G(12,10)=-G(4,2)
G(11,11)= G(3,3)
G(12,11)= G(4,3)
G(12,12)= G(4,4)
G(13,13)=G(1,1)
G(14,13)=-G(2,1)
G(15,13)=-G(3,1)
G(16,13)= G(4,1)
G(14,14)= G(2,2)
G(15,14)= G(3,2)
G(16,14)=-G(4,2)
G(15,15)= G(3,3)
G(16,15)=-G(4,3)
G(16,16)= G(4,4)
G(9,5)=  G(13,1)
G(10,5)=G(13,2)
G(11,5)=-G(13,3)
G(12,5)=-G(13,4)
G(9,6)=G(14,1)
G(10,6)= G(14,2)
G(11,6)=-G(14,3)
G(12,6)=-G(14,4)
G(9,7)=-G(15,1)
G(10,7)=-G(15,2)
G(11,7)= G(15,3)
G(12,7)=G(15,4)
G(9,8)=-G(16,1)
```

```
      G(10,8)=-G(16,2)
      G(11,8)=+G(16,3)
      G(12,8)= G(16,4)
      G(13,5)= G(9,1)
      G(14,5)=-G(9,2)
      G(15,5)=-G(9,3)
      G(16,5)= G(9,4)
      G(13,6)=-G(10,1)
      G(14,6)= G(10,2)
      G(15,6)= G(10,3)
      G(16,6)=-G(10,4)
      G(13,7)=-G(11,1)
      G(14,7)= G(11,2)
      G(15,7)=+G(11,3)
      G(16,7)=-G(11,4)
      G(13,8)= G(12,1)
      G(14,8)=-G(12,2)
      G(15,8)=-G(12,3)
      G(16,8)= G(12,4)
      G(13,9)= G(5,1)
      G(14,9)= G(6,1)
      G(15,9)=-G(7,1)
      G(16,9)=-G(8,1)
      G(13,10)= G(5,2)
      G(14,10)= G(6,2)
      G(15,10)=-G(7,2)
      G(16,10)=-G(8,2)
      G(13,11)=-G(5,3)
      G(14,11)=-G(6,3)
      G(15,11)= G(7,3)
      G(16,11)= G(8,3)
      G(13,12)=-G(5,4)
      G(14,12)=-G(6,4)
      G(15,12)= G(7,4)
      G(16,12)= G(8,4)
      DO 270 I=1,16
      DO 270 J=I,16
  270 G(I,J)=G(J,I)
       DO 310 I=1,16
       DO 310 J=1,16
  310 AKG(I,J)= G(I,J)/4200.
      IF(KLC4) 42,43,42
   43 SXY=GAM*EN
      GO TO 44
   42 SXY=PXY(IM)
   44 CONTINUE
      IF(SXY)30,31,30
   30 DO 330 I=1,16
      DO 330 J=1,16
  330 G(I,J)=0.0
      G(1,1)= .5
      G(3,2)= .02
      G(4,1)=-.02
      G(5,3)= .1
      G(5,4)= .02
      G(5,5)=-.5
```

```
G(6,3)=-.02
G(6,4)=-(1./300.)
G(7,1)=-.1
G(7,2)=-.02
G(7,6)= .02
G(8,1)= .02
G(8,2)=-G(6,4)
G(8,5)=-.02
G(9,2)=.1
G(9,4)=.02
G(9,5)= .5
G(9,6)=-.1
G(9,7)= .1
G(9,8)=-.02
G(9,9)=-.5
G(10,1)=-.1
G(10,3)=-.02
G(10,5)= .1
G(10,6)=-(1./60.)
G(10,7)= .02
G(10,8)= G(6,4)
G(11,2)=-.02
G(11,4)= G(6,4)
G(11,5)= -.1
G(11,6)= .02
G(11,7)=G(10,6)
G(11,8)=-G(6,4)
G(11,10)= .02
G(12,1)= .02
G(12,3)=-G(6,4)
G(12,5)=-.02
G(12,6)=-G(6,4)
G(12,7)= G(6,4)
G(12,8)= 1./1800.
G(12,9)=-.02
G(13,1)=-.5
G(13,2)=-.1
G(13,3)=-.1
G(13,4)=-.02
G(13,6)=.1
G(13,8)= .02
G(13,11)= .1
G(13,12)= .02
G(13,13)= .5
G(14,1)= .1
G(14,2)=-G(10,6)
G(14,3)= .02
G(14,4)=-G(6,4)
G(14,5)=-.1
G(14,7)=-.02
G(14,11)=-.02
G(14,12)= G(6,4)
G(15,1)= .1
G(15,2)= .02
G(15,3)=-G(10,6)
G(15,4)=-G(6,4)
```

```
      G(15,6)=-.02
      G(15,8)= G(6,4)
      G(15,9)=-.1
      G(15,10)=-.02
      G(15,14)= .02
      G(16,1)=-.02
      G(16,2)= G(6,4)
      G(16,3)= G(6,4)
      G(16,4)=-G(12,8)
      G(16,5)= .02
      G(16,7)=-G(6,4)
      G(16,9)= .02
      G(16,10)=-G(6,4)
      G(16,13)=-.02
      DO 360 I=1,16
      DO 360 J=I,16
  360 G(I,J)=G(J,I)
      DO 320 I=1,16
      DO 320 J=1,16
  320 AKG(I,J)= AKG(I,J)+SXY*G(I,J)
   31 RETURN
      END
```

DETERMINANT PROGRAM

```
      SUBROUTINE STACK
      COMMON JI,JT,X,Y,JII,NBND,LOCOL,G,C,AA
      COMMON AD,BD,AKG,LIN,E,H,ES,IM,NJTS,ANU
      COMMON KLU,SX,DSX,SXMAX,SY,DSY,SYMAX,SXY,KLU4
      COMMON DSXY,SXYMAX,RS,Q,NCU,S7,JD,PX,PY, PXY
      COMMON APH,BTA,GAM,EN
      DIMENSION JI(4),JT(65),X(65),Y(65),JII(50,4)
      DIMENSION NBND(65,4),AD(3),BD(3),JD(4),LOCOL(16)
      DIMENSION G(16,16),C(125,125),AA(4,3),AKG(16,16)
      DIMENSION RS(125),Q(125),PX(50),PY(50),PXY(50)
      DIMENSION DSX(50),DSY(50),DSXY(50)
      DO 18 I=1,16
      LOCOL(I)=0
   18 CONTINUE
      DO 19 IA=1,LIN
      ID=JII( IM,IA)
      DO 20 IB=1,4
      IC=4*(IA-1)+IB
      LOCOL(IC)=NBND(ID,IB)
   20 CONTINUE
   19 CONTINUE
      DO 21 I=1,16
      DO 21 J=I,16
      IF(LOCOL(J)) 21,21,22
   22 IF(LOCOL(I)) 21,21,23
   23 KROW=MAX0(LOCOL(I),LOCOL(J))
      KCOL=MIN0(LOCOL(I),LOCOL(J))
      C(KROW,KCOL)=C(KROW,KCOL)+AKG(I,J)
   21 CONTINUE
      RETURN
      END
```

```
      FUNCTION IMEQF(MID,M,N,A,Y,DD,SCALE)
C THIS FORTRAN 4 PROGRAM SOLVES AX = Y BY TRIANGULAR DECOMPOSITION.
      DIMENSION A(MID,1),Y(MID,1),SCALE(1)
      DOUBLE PRECISION SUM
      DATA TOL /5.960464E-8/
      DATA EXCESS,LARGE /1.,134217728/
      EQUIVALENCE (X,IX),(KX,EXCESS),(LARGE,BIG),(KAY,CHE)
      INTEGER SPILL
      COMMON /DOUBLE/ KEY,KAY,D  /OVFLOW/ SPILL
C SET OVERFLOW INDICATOR.
      CALL TAMPER
      KAY = 0
      D = 1.
C SCALE SO EACH ROW AND COL HAVE LARGEST ENTRY BETWEEN .5 AND 1. IN MAG
      DO 120  I = 1,M
      X = 0.
      DO 100  J = 1,M
  100 X = AMAX1(X,ABS(A(I,J)))
C POWER(X) IS THE POWER OF 2 NEXT LARGER THAN X.
      X = POWER(X)
      KAY = KAY + (IX-KX)/LARGE
      X = 1./ X
      DO 110  J = 1,M
  110 A(I,J) = A(I,J) * X
      DO 120  J = 1,N
  120 Y(I,J) = Y(I,J) * X
      DO 140  J = 1,M
      X = 0.
      DO 130  I = 1,M
  130 X = AMAX1(X,ABS(A(I,J)))
      X = POWER(X)
      KAY = KAY + (IX-KX)/LARGE
      X = 1./ X
      SCALE(J) = X
      DO 140  I = 1,M
  140 A(I,J) = A(I,J) * X
C MAJOR LOOP. TRIANGULAR DECOMPOSITION WITH D.P. ACCUM OF INNER PRODUCT
      DO 310  K = 1,M
      K1 = K - 1
      X = 0.
      L = K
      DO 180  I = K,M
      SUM = A(I,K)
      IF (K1) 150,165,150
  150 DO 160  J = 1,K1
  160 SUM = SUM + A(I,J)*A(J,K)
      A(I,K) = SUM
  165 IF (X - ABS(SUM)) 170,180,180
  170 X = ABS(SUM)
      L = I
 180  CONTINUE
C ROW INTERCHANGES TO INSURE LARGE PIVOTS
      IF (L - K) 190,220,190
  190 D = - D
      DO 200 J=1,M
```

```
      X = A(L,J)
      A(L,J) = A(K,J)
  200 A(K,J) = X
      DO 210  J = 1,N
      X = Y(L,J)
      Y(L,J) = Y(K,J)
  210 Y(K,J) = X
  220 X = -A(K,K)
      IF (M - K) 230,275,230
  230 KD = K + 1
      DO 240  I = KD,M
  240 A(I,K) = A(I,K) / X
      IF (K1) 250,300,250
  250 DO 270  L = KD,M
      SUM = A(K,L)
      DO 260  J = 1,K1
  260 SUM = SUM + A(K,J)*A(J,L)
  270 A(K,L) = SUM
  275 DO 290  L = 1,N
      SUM = Y(K,L)
      DO 280  J = 1,K1
  280 SUM = SUM + A(K,J)*Y(J,L)
  290 Y(K,L) = SUM
C UNDULY SMALL PIVOT INDICATES A IS SINGULAR
  300 IF (ABS(X) .GE. TOL) GO TO 310
      IMEQF = 3
      DD = 0.
      D = 0.
      GO TO 500
  310 CONTINUE
C BACK SOLUTION
      I = M
      DO 345 K=1,M
      I1 = I + 1
      DO 340 L=1,N
      SUM = Y(I,L)
      IF (I1 - M) 320,320,340
  320 DO 330  J = I1,M
  330 SUM = SUM - A(I,J)*Y(J,L)
  340 Y(I,L) = SUM / A(I,I)
  345 I = I-1
      DO 350  I = 1,M
      X = 1./ SCALE(I)
      DO 350  J = 1,N
  350 Y(I,J) = Y(I,J) * X
      IMEQF = 1
      IF (SPILL .LE. 1) GO TO 370
  360 IMEQF = 2
  370 SPILL = 0
  380 CONTINUE
      DO 410  I = 1,M
  390 T = D*A(I,I)
      IF (SPILL) 400,410,400
  400 SPILL = 0.
      X = POWER(D)
      D = D / X
```

```
      KAY = KAY + (IX-KX)/LARGE
      GO TO 390
410   D = 1
      DD = 2.**KAY * D * DD
      KEY = SPILL
500   CALL TAMPER
      RETURN
      END
```