

N 71 12024

CR ~~111404~~

Library
K55-10

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1498

*High-Impact Dynamic Response Analysis
of Nonlinear Structures*

K. K. Gupta

T-71-04958

CASE FILE
COPY

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

November 15, 1970

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1498

*High-Impact Dynamic Response Analysis
of Nonlinear Structures*

K. K. Gupta

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

November 15, 1970

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

Preface

The work described in this report was performed by the Engineering Mechanics Division of the Jet Propulsion Laboratory.

Acknowledgment

The author wishes to acknowledge the assistance of Dr. S. Utku, Mr. R. E. Oliver, and Dr. E. Heer in the preparation of this work. Thanks are due to Dr. R. Lutwack and Mr. R. S. Banes for providing continued support to this work program and to Mr. D. R. Maple for editing assistance. A grant from the National Research Council supported this work.

Contents

I. Introduction	1
II. Description of Analysis Procedure	2
A. Structural Idealization by Finite Element Method	2
B. Method of Solution	2
C. Numerical Scheme for the Step-by-Step Integration Procedure	3
D. Limitations	5
E. Conclusions	5
III. Preparation of Input and Details of Output	5
A. Coordinate System and Units	5
B. Zero Deformations	5
C. List of Variables	5
D. List of Arrays	6
E. Input of Data	7
F. Description of Output	8
IV. Description of Program Elements	8
A. Main Program IMAN	8
1. Input of data	8
2. General computations	9
3. Runge-Kutta routine	9
B. Subroutines of IMAN	9
1. Subroutine ACC	9
2. Subroutine ADM	10
3. Subroutine ASSEMB	10
4. Subroutine CONOJ	10
5. Subroutine COORD	10
6. Subroutine CORT	10
7. Subroutine DCS	10
8. Subroutine DFORCE	10
9. Subroutine DIEL	10
10. Subroutine DIRCOS	10

Contents (contd)

11. Subroutine ELEM T	10
12. Subroutine ELIM	10
13. Subroutine INV	11
14. Subroutine MELCON	11
15. Subroutine MMULT	11
16. Subroutine NIEL	11
17. Subroutine NQII	11
18. Subroutine PREST	11
19. Subroutine PROMAT	11
20. Subroutine QCA	11
21. Subroutine QDR	12
22. Subroutine SCAN	12
23. Subroutine SO5	12
24. Subroutine SO7	12
25. Subroutine S11	12
26. Subroutine STIMAT	12
27. Subroutine STRA	12
28. Subroutine STRESS	12
29. Subroutine TRANSP	13
30. Subroutine TRAN	13
31. Subroutine TRIM	13
32. Subroutine TRM	13
33. Subroutine UNIT	13
34. Subroutine VECT	13
V. Sample Problems	13
A. Impact Analysis of a PPO Container	13
B. Impact Analysis of a Silver–Zinc Negative Plate	14
Appendix A. Flowcharts	16
Appendix B. Source Program Listings	33
References	45

Contents (contd)

Tables

B-1. Source program listing of main program IMAN	34
B-2. Source program listing of subroutine ACC	36
B-3. Source program listing of subroutine ADM	36
B-4. Source program listing of subroutine ASSEMB	36
B-5. Source program listing of subroutine CONOJ	36
B-6. Source program listing of subroutine COORD	37
B-7. Source program listing of subroutine CORT	37
B-8. Source program listing of subroutine DCS	37
B-9. Source program listing of subroutine DFORCE	37
B-10. Source program listing of subroutine DIEL	37
B-11. Source program listing of subroutine DIRCOS	37
B-12. Source program listing of subroutine ELEM T	38
B-13. Source program listing of subroutine ELIM	38
B-14. Source program listing of subroutine INV	38
B-15. Source program listing of subroutine MELCON	39
B-16. Source program listing of subroutine MMULT	39
B-17. Source program listing of subroutine NIEL	39
B-18. Source program listing of subroutine NQII	39
B-19. Source program listing of subroutine PREST	39
B-20. Source program listing of subroutine PROMAT	40
B-21. Source program listing of subroutine QCA	40
B-22. Source program listing of subroutine QDR	40
B-23. Source program listing of subroutine SCAN	41
B-24. Source program listing of subroutine SO5	41
B-25. Source program listing of subroutine SO7	41
B-26. Source program listing of subroutine S11	41
B-27. Source program listing of subroutine STIMAT	42
B-28. Source program listing of subroutine STRA	42
B-29. Source program listing of subroutine STRESS	43
B-30. Source program listing of subroutine TRANSP	44
B-31. Source program listing of subroutine TRAN	44
B-32. Source program listing of subroutine TRIM	44

Contents (contd)

Tables (contd)

B-33. Source program listing of subroutine TRM	44
B-34. Source program listing of subroutine UNIT	44
B-35. Source program listing of subroutine VECT	44

Figures

1. Quadrilateral shell finite element with four subdivisions of triangular elements	10
2. Battery cell container with finite element idealization	13
3. Support reactions in cell container	14
4. Nodal deformations in cell container	14
5. Stress propagation in typical finite elements of cell container	14
6. Finite element idealization of cell negative plate	14
7. Support reactions in a negative plate	15
8. Nodal deformations in a negative plate	15
9. Deformed configuration of unsupported lower edge of a negative plate at various time steps	15
10. Stress propagation in typical finite elements of a negative plate	15
A-1. Flowchart of main program IMAN	17
A-2. Flowchart of subroutine ACC	23
A-3. Flowchart of subroutine ASSEMB	24
A-4. Flowchart of subroutine CONOJ	24
A-5. Flowchart of subroutine COORD	24
A-6. Flowchart of subroutine DCS	25
A-7. Flowchart of subroutine DFORCE	25
A-8. Flowchart of subroutine DIEL	25
A-9. Flowchart of subroutine DIRCOS	26
A-10. Flowchart of subroutine ELEMT	26
A-11. Flowchart of subroutine ELIM	26
A-12. Flowchart of subroutine MELCON	27
A-13. Flowchart of subroutine MMULT	27
A-14. Flowchart of subroutine NIEL	27
A-15. Flowchart of subroutine NQII	28

Contents (contd)

Figures (contd)

A-16. Flowchart of subroutine PREST	28
A-17. Flowchart of subroutine PROMAT	29
A-18. Flowchart of subroutine QCA	29
A-19. Flowchart of subroutine QDR	29
A-20. Flowchart of subroutine SCAN	30
A-21. Flowchart of subroutine STIMAT	31
A-22. Flowchart of subroutine STRESS	31
A-23. Flowchart of subroutine TRANSP	32
A-24. Flowchart of subroutine UNIT	32
A-25. Flowchart of subroutine VECT	32

Abstract

An efficient and generalized digital computer method is presented for determining the dynamic response of nonlinear structures of simple geometric configuration subjected to high-impact loading. A finite element matrix displacement approach utilizing quadrilateral shell elements, in conjunction with a step-by-step integration procedure employing Runge-Kutta extrapolation techniques, has been adopted for the present analysis.

The related procedure yields reliable results for simple structural forms, e.g., rectangular and cylindrical panels, and requires only moderate computer storage and solution time. Numerical results are presented for the analysis of relevant structures subjected to high impact, simulating free landing on a planetary surface.

The program is written in FORTRAN V language to run on the UNIVAC 1108 computer under the EXEC 8 operating system; the source deck consists of about 1800 cards. The physical program, IMAN (IMPact ANalysis), is available from the Computer Software Management and Information Center (COSMIC), the NASA agency for the distribution of computer programs.

High-Impact Dynamic Response Analysis of Nonlinear Structures

I. Introduction

The development of electronic equipment capable of hard-landing survival is of primary interest in connection with the unmanned exploration of the planets. In this regard, considerable effort has been expended on the development of a battery power supply required for a proposed hard landing on Mars in 1973 (Ref. 1). Such cell structures are expected to be subjected to a shock level of about 4000 g (square pulse for 1 msec) that results from an impact velocity of 120 ft/s, the data being calculated from available information pertaining to gravitational and atmospheric conditions on Mars. Under such high impact the battery cell structure is expected to undergo large deformations, thereby introducing structural nonlinearities into the system.

A vital preliminary to the formal analysis of an equivalent linear problem is the determination of the eigenvalues and associated modes. Efficient numerical procedures are available (Refs. 2 and 3) for such solutions. However, for nonlinear structures it is necessary to adopt an alternative recurrence formulation which involves step-by-step incremental procedures and allows for structural and damping nonlinearities.

The incremental procedure has been adopted earlier (Ref. 4) for the solution of nonlinear structural problems

in which the accelerations are usually assumed to vary linearly during each limited time increment. The set of dynamic equations may then be transformed into a convenient pseudostatic form, the solution of which yields the incremental results. Thus the procedure involves the repeated solution of a set of equations of order equal to the number of degrees of freedom of the structure and requires large computer storage and problem solution time. The present analysis is based on a nodewise predictor-corrector approach and requires only moderate computer storage for most problems.

Structural nonlinearities, in general, may be classified into two distinct groups. The physical nonlinearities are related to the material properties, which include nonlinear elastic stress-strain relations and also the elastoplastic behavior of the material. Geometrical nonlinearities, on the other hand, are due to large deformations which cause gross changes in the geometrical configuration of the structure. These nonlinearities may be conveniently subclassified as follows:

- (1) Large rotation, small strain (also referred to as the "prestress" effect).
- (2) Small rotation, large strain.
- (3) Combinations of (1) and (2).

This paper is concerned with the detailed analysis relating to the geometrical nonlinearities only; the associated generalized computer program, written in FORTRAN V and utilizing a UNIVAC 1108, has the following features:

- (1) Structural idealizations are based on the finite element method employing curved quadrilateral shell elements, the stiffness matrix of the entire structure being assembled by means of the matrix displacement method employing the direct stiffness approach.
- (2) The masses are assumed to be lumped at the nodes, each having six degrees of freedom, i.e., three translations and three rotations defined in a right-handed orthogonal coordinate system.
- (3) The Runge-Kutta method has been employed for numerical extrapolation of nodal accelerations and deformations at the end of each small time increment of the general step-by-step incremental time procedure adopted for the present analysis, it being assumed that the structure behaves linearly within each small time increment.
- (4) Related dynamic equations are formed nodewise for each incremental time step, and the relevant solutions are added to their respective existing values at the beginning of the step to yield the corresponding updated values.

The merit of these techniques is that considerably less computer storage and solution time is required for the dynamic response analysis of nonlinear structures in comparison to other known methods of solution (Ref. 4). Also, the present analysis may be conveniently extended to include physical nonlinearities.

The program computes the propagation of nodal deformations and velocities as well as element stresses and support reactions throughout the impact contact time period.

II. Description of Analysis Procedure

It is intended here to compute the high-impact dynamic response of nonlinear structural systems, with particular reference to batteries. The associated computer program allows for various structural nonlinearities and is based on the finite element method in conjunction with a step-by-step incremental time (loading) procedure.

The Runge-Kutta method is used for numerical extrapolation of accelerations and deformations at the end of each small time increment from known corresponding values at the beginning of the step.

A. Structural Idealization by Finite Element Method

A refined quadrilateral shell element has been generated and utilized to idealize practical structures. The quadrilateral element is divided into four triangular elements with a common apex at the center of area of the original quadrilateral element. The stiffness matrices of the four triangular elements (Ref. 5) are then assembled, and the apex point is finally eliminated to obtain the quadrilateral element stiffness matrix of order (20, 20) in its local coordinate system. When transformed into the global coordinate system, the final stiffness matrix is of order (24, 24).

B. Method of Solution

Equilibrium equations are first formed for the free-free structure. At the beginning of the first time interval, the velocities and displacements corresponding to any assumed node are set equal to zero to simulate the first touchdown condition of the structure striking a rigid surface. The response is computed for that particular time interval, and the displacements of all nodes are checked. If any node deflects into the rigid impact surface, such deflections and corresponding velocities are set to zero and are substituted into the equilibrium equations for the beginning of the next time interval, the subsequent response being computed. This process is repeated until the time interval of interest has been covered.

As the structure strikes an impact surface, a compressive wave is generated which starts propagating toward the other free end. Upon reaching the free end, the compressive dynamic wave is reflected back as a tension wave, and the structure may separate from the impact surface when the reflected wave reaches that surface. The structure may then strike the impact surface again, in which case the process of wave propagation will be repeated. This program computes the propagation of associated forces and deformations in a structure subjected to high-impact force.

The equation of motion of a structural system may be written in the following form:

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{P(t)\} \quad (1)$$

where

$[M]$ = mass matrix (assumed to be lumped at the nodes)

$[C]$ = viscous damping matrix

$[K]$ = stiffness matrix

$\{q\}$ = deformation vector

$\{P(t)\}$ = time-dependent forcing function vector

Equation (1) is applicable to a nonlinear system when it is modified to represent the linear conditions which are assumed to exist in each limited time increment, i.e., by adopting a step-by-step procedure. Thus, when the dynamic equation is assumed to be valid for a particular time interval $h(n, n+1)$, i.e., the time interval between time $h(n)$ and time $h(n+1)$, it takes the form

$$[M]\{\delta\ddot{q}\}_{n,n+1} + [C]\{\delta\dot{q}\}_{n,n+1} + [K]\{\delta q\}_{n,n+1} = \{\delta P\}_{n,n+1} \quad (2)$$

The present problem involves a large number of degrees of freedom; therefore, a nodewise predictor-corrector procedure, based on the Runge-Kutta formulation of $O(h^5)$ truncation error, is used for every node in definite sequence, avoiding the usual matrix operations with matrices of very large order. This incremental procedure also enables one to take into account nonlinearities associated with the stiffness and damping matrices.

In this work, the damping coefficient c is assumed to be a property of the material only. The damping force at a node corresponding to any particular degree of freedom is thus obtained by the summation of such values calculated by multiplying the appropriate damping constant and the difference in relevant velocity between the node under consideration and any other connecting node.

C. Numerical Scheme for the Step-by-Step Integration Procedure

Within a small time interval $h(n, n+1)$, the following typical primary computational steps are performed, corresponding to a degree of freedom associated with any node (say the p th degree of freedom in totality).

Step 1. The acceleration value at the beginning of time step $n+1$ is computed as follows (superscript n

denoting any particular time step and T denoting matrix transposition):

$$(\ddot{q})^{P,n} = \frac{1}{(M)^{P,n}} [(P)^{P,n} - (S)^{P,n} - \{c^T\}^P \{\delta\dot{q}\}^{P,n}] \quad (3)$$

in which $\{S\}^n$ is the cumulative nodal forces vector, and

$$\{K1\}^n = h\{\ddot{q}\}^n \quad (4)$$

is stored on completion of step 1 for all values of p .

Step 2. The values of incremental deformations and velocities at the half-time interval are extrapolated to yield the corresponding acceleration

$$(\ddot{q})^{P,n+\frac{1}{2}} = \frac{1}{(M)^{P,n+\frac{1}{2}}} [(P)^{P,n+\frac{1}{2}} - (S)^{P,n+\frac{1}{2}} - \{K^T\}^{P,n} \{\delta q\}^{P,n+\frac{1}{2}} - \{c^T\}^P \{\delta\dot{q}\}^{P,n+\frac{1}{2}}] \quad (5)$$

in which the extrapolated values are

$$\{\delta q\}^{P,n+\frac{1}{2}} = \frac{h}{2} \{\dot{q}\}^{P,n} + \frac{h}{8} \{K1\}^{P,n} \quad (6)$$

$$\{\delta\dot{q}\}^{P,n+\frac{1}{2}} = \{\delta\dot{q}\}^{P,n} + \frac{1}{2} \{\delta K1\}^{P,n} \quad (7)$$

and

$$\{K2\}^{n+\frac{1}{2}} = h\{\ddot{q}\}^{n+\frac{1}{2}} \quad (8)$$

is stored.

Step 3. The revised acceleration value at the half-time step is computed from extrapolated deformations and velocities as follows (subscript R denoting the revised step):

$$(\ddot{q})_{(R)}^{P,n+\frac{1}{2}} = \frac{1}{(M)_{(R)}^{P,n+\frac{1}{2}}} [(P)_{(R)}^{P,n+\frac{1}{2}} - (S)_{(R)}^{P,n+\frac{1}{2}} - \{K^T\}^{P,n} \{\delta q\}_{(R)}^{P,n+\frac{1}{2}} - \{C^T\}^P \{\delta\dot{q}\}_{(R)}^{P,n+\frac{1}{2}}] \quad (9)$$

where

$$\{\delta q\}_{(R)}^{P,n+1/2} = \frac{h}{2} \{\dot{q}\}^{P,n} + \frac{h}{8} \{K2\}^{P,n+1/2} \quad (10)$$

$$\{\delta \dot{q}\}_{(R)}^{P,n+1/2} = \{\delta \dot{q}\}^{P,n} + \frac{1}{2} \{\delta K2\}^{P,n+1/2} \quad (11)$$

Further,

$$\{K3\}_{(R)}^{n+1/2} = h\{\ddot{q}\}_{(R)}^{n+1/2}$$

$$\{\ddot{q}\}^{P,n+1} = \frac{1}{(M)^{P,n+1}} [(P)^{P,n+1} - (S)^{P,n+1} - \{K^T\}_{(R)}^{P,n+1/2} \{\delta q\}^{P,n+1} - \{C^T\}^P \{\delta \dot{q}\}^{P,n+1}] \quad (13)$$

in which

$$\{\delta q\}^{P,n+1} = h\{\dot{q}\}^{P,n} + \frac{h}{2} \{K3\}_{(R)}^{n+1/2} \quad (14)$$

and

$$\{\delta \dot{q}\}^{P,n+1} = \{\delta \dot{q}\}^{P,n} + \{\delta K3\}_{(R)}^{P,n+1/2} \quad (15)$$

Also,

$$\{K4\}^{n+1} = h\{\ddot{q}\}^{n+1} \quad (16)$$

is stored.

Step 5. The deformation and velocity files are finally updated according to the following schemes:

$$\{q\}^{n+1} = \{q\}^n + h \left[\{\dot{q}\}^n + \frac{1}{6} (\{K1\} + \{K2\} + \{K3\}) \right] \quad (17)$$

$$= \{q\}^n + \{qI\}^{n+1} \quad (17a)$$

$$\{\dot{q}\}^{n+1} = \{\dot{q}\}^n + \frac{1}{6} [\{K1\} + 2\{K2\} + 2\{K3\} + \{K4\}] \quad (18)$$

The nodal forces are also updated, incorporating changes due to increments in nodal deformation and also to large rotation (the prestress effect), as follows:

$$\{S\}^{n+1} = \{S\}^n + \{\delta S\}_{n,n+1} + [K]_{(R)}^{n+1/2} \{qI\}^{n+1} \quad (19)$$

and

$$\{\delta q\}_{(R)}^{n+1/2} \quad (12)$$

are stored.

Step 4. The revised stiffness matrix $\{K\}_{(R)}^{n+1/2}$ based on the $\{q\}_{(R)}^{n+1/2}$ configuration is formed and the acceleration at the end of the $n + 1$ step is computed from extrapolated values of deformations and velocities as follows:

where the term $\{S\}^n + \{\delta S\}_{n,n+1}$ is computed as the appropriate summation of $[DIR]^T \{s^*\}^n$ terms in which $[DIR]$ and $\{s^*\}^n$ are the element direction cosine matrix and such nodal forces in the local coordinate system, respectively.

Since velocity values computed from Eq. (18) differ slightly from those given in step 4, the acceleration values are recalculated at the beginning of the next time step as shown in step 1. Also, stresses are computed in two triangular parts of the quadrilateral shell element, and an average value based on relative areas of the triangles is accepted as the final element stress assigned to the centroid of the quadrilateral.

The computations are repeated for each time step, and updated relevant force and deformation vectors are obtained as accumulated sums at the end of each time step. Owing to rather small changes in their values, the P , M , and S vectors considered in this program and used throughout steps 1 to 4 refer to their initial values at the beginning of a particular time step.

The reactive forces at any subtime interval r of the $n + 1$ time step may be calculated as

$$(RF)^{P,n+r} = - \frac{(\ddot{q})^{P,n+r}}{(M)^{P,n+r}} \quad (20)$$

the corresponding acceleration $(\ddot{q})^{P,n+r}$ being subsequently equated to zero. However, when the tension wave reaches the impact surface, the reaction may become negative, indicating a "liftoff" of the structure, in which case the acceleration value is computed as usual.

Thus the present numerical scheme computes the propagation of nodal deformations, velocities, and accelera-

tions, as well as element stresses and support reactions throughout the impact time period.

D. Limitations

So far, the program has been utilized to analyze only structures of simple geometric form, e.g., rectangular and cylindrical panels; therefore, it should be used with caution for structures of complex geometry. Also, the program is presently limited to the use of quadrilateral shell elements, but the scope of the program may be easily extended to its more general form, incorporating other types of elements.

E. Conclusions

The present analysis is a nodewise predictor-corrector method and has several advantages over the usual recurrence formula procedure (Ref. 4), which involves repeated solutions of sets of simultaneous equations. In general, the following important conclusions may be drawn from the results of the present analysis:

- (1) The analysis is independent of the form of the damping matrix $[C]$, which may indeed be nonlinear.
- (2) The stiffness matrix $[K]$ may be nonlinear.
- (3) No strict node numbering scheme is necessary.
- (4) For short impact contact time, for which the present method appears to be stable, its use may be preferred over the usual method, which requires repeated solution of simultaneous equations.
- (5) Much less computer storage is required for the present method when compared with that required for the usual methods. In fact, about 10 times the total number of degrees of freedom is the approximate maximum storage requirement, in addition to that required to store the input data.
- (6) There is a significant reduction in solution time, the computations being simple and repetitive in nature. An estimate of typical run time varies between 0.4 to 0.6s/node for each time step.
- (7) The quadrilateral finite element used in the analysis is much refined in nature when compared to, say, an equivalent triangular element.
- (8) Very little storage is necessary for the stiffness matrix since computations are performed nodewise, the total storage involved being an array of dimension (6,90).
- (9) For very small impact contact time, as the case often is for high-impact problems, the damping has very little effect on the general solution.

III. Preparation of Input and Details of Output

A. Coordinate System and Units

The nodal deformations along with other related sets of results are obtained in a right-handed, orthogonal, rectangular, global coordinate system, whereas element nodal forces are computed in a similar but local member coordinate system. The global coordinate system (X,Y,Z) refers to the complete structure and normally will conform in direction to the major dimensions of the structure. A member coordinate system (x,y,z) refers to an individual member in which the local x axis is assumed to be coincident with the element edge connecting the first two of the four nodes used in defining the member (Fig. 1).

Joint numbering is usually done in a counterclockwise sequence. Further, the program does not perform any conversion of units. Therefore, the input data must be consistent in units.

B. Zero Deformations

The zero-displacements are introduced at the nodes in contact with the impact surface and also at nodes lying on planes of symmetry when only the symmetric part of a structure is analyzed. Such nodes, along with their characteristic modes, are the associated input data. The characteristic zero-deformation modes numbered 1 to 6 are associated respectively with the degrees of freedom δ_x , δ_y , δ_z , θ_x , θ_y , and θ_z .

C. List of Variables

The important variables used in the present computer program are as follows:

- E Young's modulus
- μ Poisson's ratio
- c damping coefficient for the material

- IPRINT index for printout of results
= 0, then only final results at end of each time step are printed
= 1, then detailed results at end of each subtime step are printed

NJ	total number of nodes	AK12(20,5)	matrices which are portions of the
NEL	total number of elements	AK21(5,20)	rearranged A matrix utilized to
NMAT	number of material types, the latter being dependent on the three variable E , μ , and c	AK22(5,5)	eliminate the fifth central node of the quadrilateral
NZD	number of zero deformations, including such deformations at nodes along planes of symmetry	B(20,20)	quadrilateral element stiffness matrix in the local coordinate system derived from a matrix by eliminating the fifth central node
NET	number of element types, which is determined by element thickness and dimensions in the local coordinate system	B1(300*NET)	file storing the symmetric part of the B matrix pertaining to the differing element types and also storing zero elements to conform to the final AK1 matrix
NDF	total number of degrees of freedom ($= 6 \times NJ$)	C(15,15)	triangular shell element stiffness matrix in the local coordinate system
IUPDAT	index for updating nodal coordinates = 0, then stiffness matrices are computed based on the geometry of the structure at the start of the time step = 1, then stiffness matrices are computed based on the revised geometry of the structure at the half-time step	C1,C2,C3,C4,C5 C6,C7,C8	C1 to C7 are of order (3,3), whereas C8 is of order (3,4); they are used for general computational work
ITIME	index for nature of incremental time intervals = 0, then uniform time interval TI is read = 1, then NIN number of time intervals are read	CMASS(NDF)	file storing lumped nodal masses
TI	incremental time step	CNJ(15,3)	file storing coordinates of nodes connected to any particular node being considered
NIN	total number of such increments	DAMP(15)	damping coefficients file for a node connected to a maximum of 14 other nodes
VIN	initial nodal velocity prior to impact	D33(3,3)	elastic stress-strain matrix for plate bending case
IDFV	particular degree of freedom associated with the initial velocity	DIR(3,3)	direction cosine matrix of an element

D. List of Arrays

The important arrays used in the computer program are listed as follows:

A(25,25)	stiffness matrix of a quadrilateral shell element in the local coordinate system in which the center of area is considered as the fifth node	DM(8)	damping coefficients of a maximum number of 8 elements connecting a node
AK(6,90)	stiffness matrix in the global coordinate system pertaining to a node	DP(6)	damping force at a node corresponding to its 6 degrees of freedom
AK1(24,24)	final stiffness matrix of a quadrilateral element in the global coordinate system	E22(3,3)	elastic stress-strain matrix for plane stress case
		FEL(NEL*24)	element nodal forces file in its local coordinate system
		IC(8)	file storing the element numbers connected to a node
		IEL(8,4)	file storing element nodal numbers connected to a node

<i>IELCON</i> (<i>NJ</i> ,4)	file storing element nodal numbers (a maximum of 8 elements stored in 4 words in pairs as <i>IC</i> (1)*1000+ <i>IC</i> (2), etc.) connecting each node	<i>Q22</i> (<i>NDF</i>)	file storing <i>Q2</i> for use in the Runge-Kutta operations at subtime steps
<i>IMT</i> (<i>NEL</i>)	element material type indicator file	<i>RF</i> (<i>NZD</i>)	file storing reactions at contact points as well as at points of symmetry, i.e., reactions at zero-deformation nodes
<i>INLT</i> (<i>NEL</i>)	element type indicator file	<i>S</i> (<i>NDF</i>)	nodal forces file
<i>INODE</i> (15)	file containing nodal numbers connecting a node	<i>ST</i> (6* <i>NEL</i>)	file storing element stresses δ_x , δ_y , and τ_{xy} at the top and bottom layer of the cross section, at the center of area of the element, and expressed in element local coordinate system
<i>IZD</i> (<i>NZD</i> ,2)	zero-deformation file storing the nodal number and the mode of the associated zero deformation	<i>TEL</i> (<i>NEL</i>)	element thickness file
<i>M</i> (8)	file storing element numbers connecting a node	<i>TINT</i> (<i>NIN</i>)	file storing incremental time information
<i>NLT</i> (<i>NET</i> ,5)	file storing element nodal numbers and its material type	<i>TNLT</i> (<i>NET</i>)	file storing thickness of various element types
<i>NNEL</i> (<i>NEL</i> ,4)	file storing 4 nodal numbers for each element	<i>V1,V2,V3,V4,V10</i>	files of order (3), excepting for <i>V10</i> , which is of order (4), used for storage purposes during various computations
<i>PM</i> (<i>NMAT</i> ,3)	material properties file containing Young's modulus <i>E</i> , Poisson's ratio μ , and damping coefficient <i>c</i> values for each material type	<i>X,Y,Z</i>	files of order (3) storing element nodal coordinates and used throughout the computations
<i>P</i> (<i>NDF</i>)	external loading file	<i>XD,YD,ZD</i>	files of order (2) storing element side lengths during various computations
<i>Q</i> (<i>NDF</i>)	nodal deformation file, the first three columns of which represent current nodal coordinates		
<i>QI</i> (<i>NDF</i>)	incremental nodal deformation file pertaining to each incremental subtime step of the Runge-Kutta procedure		
<i>QII</i> (90)	incremental nodal deformations pertaining to the nodes connecting the particular node under consideration		
<i>QQI</i> (<i>NDF</i>)	accumulating nodal incremental deformations during a time interval of the Runge-Kutta step		
<i>QIR</i> (<i>NDF</i>)	incremental revised nodal deformation <i>QI</i> at half-time interval		
<i>Q1</i> (<i>NDF</i>)	nodal velocity file		
<i>QII</i> (<i>NDF</i>)	accumulating nodal incremental velocity file during a time interval of the Runge-Kutta step		
<i>Q2</i> (<i>NDF</i>)	nodal acceleration file		

E. Input of Data

The program IMAN is stored on tape; the control cards are those usually associated with the UNIVAC 1108. Data cards for a particular job should be prepared as follows:

- Replace the first two PARAMETER cards in the main program with data relevant to the particular problem. The following cards, with appropriate data, are to be provided:

```
@FOR,S IMAN,IMAN,IMAN
-10,11
PARAMETER NJ= , NEL= , NZD= ,
NMAT= , NET= , NIN=
PARAMETER ITIME= , IPRINT=
```
- Title of job:FORMAT (13A6), i.e., a maximum of 78 words.

(3) Incremental time file:

If $ITIME = 0$, read TI , incremental uniform time step: $FORMAT(E11.3)$.

If $ITIME = 1$, read $TINT$, incremental time steps file: $FORMAT(3E11.3)$. Time intervals are to be set such that the numerical speed of wave propagation in the structure caused by impact remains greater than the speed of sound in the structure material.

(4) VIN , $IDFV$: $FORMAT(E11.3,15)$.

(5) Zero-deformation file IZD : $FORMAT(5(215))$.

(6) Nodal coordinate file Q : $FORMAT(3F9.4)$.

(7) Element descriptions:

(a) Element nodal numbers, thickness, material, and element type indicators to be stored in files $NNEL$, TEL , IMT , and $INLT$: $FORMAT(415,F8.4,214)$.

(b) Files NLT and $TNLT$ containing element nodal numbers, material type, and thickness corresponding to each element type: $FORMAT(515,F8.4)$.

(8) Material properties file PM containing E , μ , and c values: $FORMAT(E10.2,F6.2,F8.3)$.

(9) Nodal lumped masses file M containing all nonzero mass values: $FORMAT(6E11.3)$

(10) Nodal external loading file P : $FORMAT(6E11.3)$.

The present program at JPL is stored on tape (No. 30902, in this case), and the exact sequence of control and data cards immediately following the first $@RUN$ card is as follows:

$@ASG,T$ TAPE 1, T, 30902W

$@MSG$ PLEASE MOUNT TAPE REEL 30902W
(KKG)

$@COPIN$ TAPE 1

$@FORS$ IMAN, IMAN, IMAN

-10, 11

PARAMETER $NJ =$, $NEL =$, $NZD =$,
 $NMAT =$, $NET =$, $NIN =$

PARAMETER $ITIME =$, $IPRINT =$

$@REWIND$ TAPE 1.

$@COPOUT$, TAPE 1.

$@FREE$ TAPE 1.

$@PREP$.

$@PRT$, T

$@XQT$

{ data cards

End card.

F. Description of Output

If $IPRINT = 0$, the following results are printed:

(1) Input files IZD , Q , $NNEL$, TEL , IMT , $INLT$, NLT , $TNLT$, PM , $CMASS$, P , and $IELCON$.

(2) File RF at end of each subtime step.

(3) File ST at end of revised halftime step and also at end of each full time step.

(4) Files DIR , S , Q , $Q1$, and $Q2$ at end of each full time step.

If $IPRINT = 1$, then additional printouts of files S , Q , $Q1$, and $Q2$ are obtained.

IV. Description of Program Elements

This section presents detailed descriptions of the individual program elements. Corresponding flowcharts and computer program listings are presented in Appendixes A and B, respectively.

A. Main Program IMAN

The main program IMAN performs the entire set of operations, including calling the subroutines at various stages and coordinating all such operations. The major computational steps are presented in this section.

1. Input of data. This part of the program is primarily concerned with data input for a particular job. Details of input data preparation have been given separately in Section III.

2. General computations. To prepare basic data for Runge-Kutta procedures, the following steps are performed:

- (1) SCAN is called to form file *IELCON* storing element numbers connecting each node.
- (2) Stiffness matrix *B* of each quadrilateral element type is formed using subroutine QDR. Matrix *B* is rearranged into *AK1* and finally stored in *B1* file.

3. Runge-Kutta routine. For numerical extrapolation of accelerations and deformations, the following computational procedure has been adopted for each small time interval (n to $n + 1$):

- (1) To find the acceleration at the beginning of the time step, the following operations steps are performed:
 - (a) DIEL is called to form *DAMP* and *INODE* arrays.
 - (b) Damping force file *DP* is formed utilizing *DAMP* and *Q1ⁿ* files.
 - (c) Acceleration file *Q2ⁿ* is computed from *P*, *S*, *DP* and *CMASS* files.
 - (d) Support reactions files *RF* are computed at boundary and symmetry points from *Q2* and *CMASS* files.
 - (e) *Q2ⁿ* file is copied into *Q22* so as to use it in the next substep.
 - (f) Accumulating incremental velocity and deformation files *QI1* and *QQI* are updated.
- (2) To find the acceleration at the half time step, the following major computations are performed:
 - (a) Extrapolated incremental deformation file *QI^{n+1/2}* is computed using files *Q1ⁿ* and *Q2ⁿ*.
 - (b) ACC is called to compute *Q2^{n+1/2}* and *RF* files.
 - (c) *Q2^{n+1/2}* is stored into *Q22* file. *QI1* and *QQI* files are updated.
- (3) To compute revised sets of results at the half step, the following procedure is used:
 - (a) File *QI^{n+1/2}* is computed from *Q1ⁿ* and *Q2^{n+1/2}* files.
 - (b) ACC is called to compute *Q2^{n+1/2}* and *RF* files.
 - (c) STRESS is called to compute stress increments in the elements due to incremental deformation *QI^{n+1/2}*.

(d) *Q2^{n+1/2}* is stored in *Q22* file and *QI1* and *QQI* are updated (to obtain *QQIⁿ⁺¹* file).

- (4) To compute acceleration values at the end of the time step, the primary steps are as follows:
 - (a) *QIⁿ⁺¹* is formed from files *Q1ⁿ* and *Q2^{n+1/2}*.
 - (b) ACC is called, IUPDAT being 1 to compute *Q2ⁿ⁺¹*, *RF*, and nodal forces incremental file *S* (denoted by *S* in ACC) = $K_R^{n+1/2} QI^{n+1}$.
 - (c) STRESS is called to compute incremental stresses in the elements due to *QIⁿ⁺¹*.
 - (d) Element stresses $ST^{n+1} = ST^n + \Delta ST^{n+1}$ are updated.
 - (e) PREST is called to compute updated nodal forces by incorporating "prestress" effect and also to compute updated element nodal forces in the local coordinate system.
 - (f) *QI1* file is updated to obtain *QI1ⁿ⁺¹*.
 - (g) Updated $Q^{n+1} = Q^n + QQI^{n+1}$ and $Q1^{n+1} = Q1^n + QI1^{n+1}$ are next obtained.

With the new updated files, the Runge-Kutta procedure is repeated for the next time step.

B. Subroutines of IMAN

1. Subroutine ACC. Subroutine ACC is called by IMAN to predict the nodal accelerations and support reactions at the end of each Runge-Kutta substep. It also computes the nodal force increments at the end of a full step, based on the accumulated incremental nodal deformations *QQI*. The primary computational steps for each node are as follows:

- (1) Subroutines DIEL, DFORCE, MELCON, and CONOJ are called to compute, respectively, the damping coefficients file *DAMP* and the nodal connection file *INODE*, the damping forces file *DP*, the element connection file *M*, and file *CNJ* having the current coordinates of connecting nodes which are being referred by file *INODE*. If the index IUPDAT $\neq 0$, then the *CNJ* file is updated by adding to it the appropriate elements of the revised half step incremental deformation file *QIR*.
- (2) Subroutine NIEL is called to form *IEL*, which lists nodal numbers of connecting elements of *M*.
- (3) Subroutine STIMAT is called to form the overall stiffness matrix *AK* corresponding to the node.

- (4) Subroutine NQII is called to copy incremental deformation file *QI* and to form such file pertaining to the node.
- (5) Reactive forces at supports and points of symmetry are computed as well as the acceleration values.
- (6) If index $IUPDAT \neq 0$, then NQII is called to copy the accumulating incremental deformation file *QQI* and form *QII* file pertaining to the node.
- (7) Increment is obtained in the nodal forces file *S* based on *QII* (i.e., *QQI*).

2. Subroutine ADM. Subroutine ADM is called by subroutines S05 and S07 to perform the addition of submatrices to form the stiffness matrix of a triangular shell element.

3. Subroutine ASSEMB. Subroutine ASSEMB is called by QDR to assemble the stiffness matrices *C* (15,15) of four triangular shell elements into the quadrilateral element stiffness matrix *A* (25,25).

4. Subroutine CONOJ. Subroutine CONOJ is called primarily by ACC. It forms the *CNJ* file, listing the current coordinates of the nodes associated with the node under consideration by copying the current deformation file *Q*.

5. Subroutine COORD. Subroutine COORD is called by IMAN. It reads the initial nodal coordinates and stores them in the *Q* file.

6. Subroutine CORT. Subroutine CORT is called by QDR and is a modified version of CORT in ELAS (Ref. 6). CORT obtains coordinates of each of the four triangular shell elements in the local coordinate system (x, y, z) of the quadrilateral element (Fig. 1).

7. Subroutine DCS. Subroutine DCS is called by subroutine PREST to compute the direction cosines matrix *DIR* for any element. If $INDEX = 1$, *DIR* is based on $Q^{n+1} = Q^n + QQI^{n+1}$, in which Q^n and QQI^{n+1} are retained in their original forms. If $INDEX = 0$, *DIR* is based on Q^n only.

8. Subroutine DFORCE. Subroutine DFORCE is called by ACC to form the damping forces *DP* at a node. Files *DAMP*, *INODE*, *Q1*, and *Q2* are used.

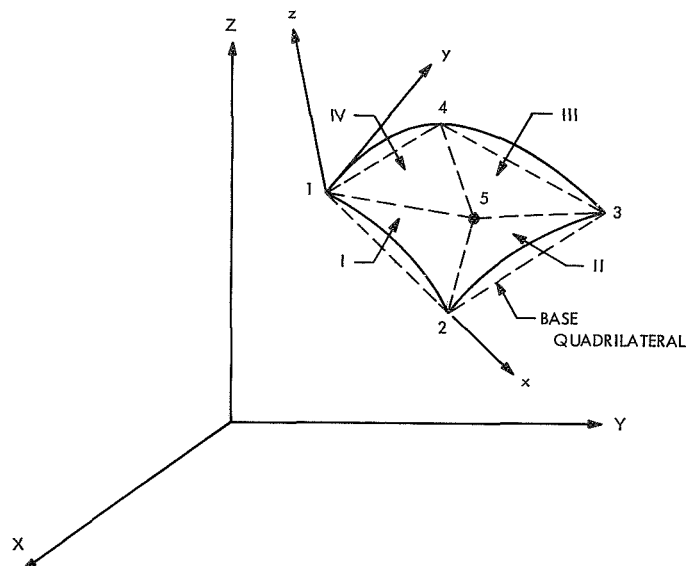


Fig. 1. Quadrilateral shell finite element with four subdivisions of triangular elements

9. Subroutine DIEL. Subroutine DIEL is called by IMAN and ACC to generate the list of connecting nodes and also damping coefficients related to each node under consideration. A maximum of 14 nodes and eight elements are assumed to be connected to any node. The primary steps for each node are as follows:

- (1) Subroutine MELCON is called to form file *M*, listing the element numbers connecting the node. Subroutine NIEL is called to form file *IEL*, listing connecting element descriptions pertaining to file *M* and denoted by their nodal numbers.
- (2) With information provided by files *M* and *IEL*, *INODE*, a list of nodes belonging to the connecting elements, is formed.
- (3) The damping coefficients file *DAMP* is computed using files *PM*, *IEL*, and *INODE*.

10. Subroutine DIRCOS. Subroutine DIRCOS is called by STIMAT to obtain coordinate difference of joints 1-2 and 4-1 of a quadrilateral element, in the global coordinate system, while forming its direction cosines.

11. Subroutine ELEMENT. Subroutine ELEMENT is called by IMAN to read information regarding the finite elements and to store such data in *NNEL*, *TEL*, *IMT*, *INLT*, *NLT*, and *TNLT*.

12. Subroutine ELIM. Subroutine ELIM is called by QDR. It reduces the *A*(25,25) matrix to the final quadri-

lateral stiffness matrix $B(20,20)$ in the local coordinate system by eliminating the fictitious central point corresponding to the center of area of the quadrilateral element.

13. Subroutine INV. Subroutine INV is called by ELIM to invert a matrix of order 5.

14. Subroutine MELCON. Subroutine MELCON is called mostly by DIEL and ACC. It simply converts the element connection information from IELCON into M file for each node under consideration.

15. Subroutine MMULT. Subroutine MMULT is called mostly by PREST, STIMAT, and STRESS to multiply any two matrices.

16. Subroutine NIEL. Subroutine NIEL is called mostly by DIEL and ACC and forms the list of nodal numbers of the elements connecting a node.

17. Subroutine NQII. Subroutine NQII is called by ACC to list the incremental deformations at the particular node under consideration as well as at the connecting nodes.

18. Subroutine PREST. Subroutine PREST is called by IMAN at the end of a Runge-Kutta time step to compute the updated nodal forces at the joints to include the "prestress" effect, the elastic nodal force increments for the time step having been already computed by subroutine ACC and stored in S file. The subroutine also updates the element nodal forces file FEL at the end of the time step. The primary computational steps for the two distinct phases of calculations are given below.

a. First phase. The newly oriented nodal force file is computed. The present S file contains elastic nodal force increments for the time step, whereas the FEL file stores the element nodal forces in the local coordinate system with reference to the geometry of the structure at the beginning of the current time step. The major steps for each node are as follows:

- (1) Subroutine MELCON is called to generate nodal element connection file M . Elements of FEL pertaining to the node under consideration are copied into file SEL .
- (2) For each of the quadrilateral elements of M file, the following computations are next carried out:
 - (a) Elements of FEL pertaining to the element under consideration are copied into file SEL .

- (b) Subroutine DCS is called to form the direction cosine matrix DIR , based on the updated nodal deformation file at the end of the step; e.g., $Q^{n+1} = Q^n + QQI^{n+1}$ configuration. Utilizing SEL and DIR files, element nodal forces are obtained in the global coordinate system pertaining to the element geometry at the end of the time step, thereby automatically incorporating the prestress effect and storing it in an appropriate file SS .

- (3) Elements of SS for each of the four nodes are added to the appropriate elements of the S file.

Repeated computations involving steps 1 to 3 finally yield the updated nodal forces corresponding to the end of the Runge-Kutta time step under consideration.

b. Second phase. The element nodal forces in the local coordinate system, stored presently in FEL , are updated to correspond to the geometry of the structure at the end of the time step. The following major steps have been adopted for each element:

- (1) It is necessary to convert the elements of the QQI^{n+1} file, pertaining to the element, in its local coordinate system. Subroutine DCS is called to form file DIR based on the structural configuration Q^n at the beginning of the time step. Matrix multiplication of DIR and QQI files is done to obtain the accumulating deformation increments file A in the local coordinate system.
- (2) Element stiffness matrix $AK1$ is copied from $B1$ file. The increment in element nodal forces is obtained by multiplying file A by $AK1$, the results being stored in the $SEL1$ (24) file.
- (3) The incremental element nodal forces file of step 2 is added to file FEL , which contains element nodal forces in the local coordinate system. The resulting file FEL is the updated element nodal forces file at the end of the time step.

19. Subroutine PROMAT. Subroutine PROMAT is called by IMAN to read element material properties (E , μ , and c) and to store them in PM .

20. Subroutine QCA. Subroutine QCA is called by subroutine QDR. Given the coordinates of the four corners of a quadrilateral element in the global coordinate system, the subroutine computes the coordinates of

the center of area of the element as well as the area of its surface.

21. Subroutine QDR. Subroutine QDR is called by IMAN to generate the stiffness matrix of a quadrilateral shell element in the local coordinate system and to store the same in matrix $B(20,20)$. The primary steps are as follows:

- (1) Subroutine QCA is called to generate the coordinates of the center of area of the quadrilateral element, which is next divided into four triangular elements with apex at the center of area (Fig. 1).
- (2) For each of the triangular elements the following computations are next carried out:
 - (a) Direction cosines information is computed.
 - (b) Subroutine CORT is called to obtain coordinates of triangular shell element nodes in the local coordinate system.
 - (c) Subroutine S11 is called to generate the stiffness matrix of the element and store it in $C(15,15)$.
 - (d) Subroutine ASSEMB is called to place elements of C in the overall quadrilateral element stiffness matrix $A(25,25)$.
- (3) Subroutine ELIM is called to eliminate the fictitious nodal point at the center of area of the quadrilateral element and to obtain the overall stiffness matrix of the latter in the local coordinate system, which is subsequently stored in $B(20,20)$ in the order $u_1, \dots, u_4, v_1, \dots, v_4, w_1, \dots, w_4, \theta_{x_1}, \dots, \theta_{x_4}, \theta_{y_1}, \dots, \theta_{y_4}$.

22. Subroutine SCAN. Subroutine SCAN is called by IMAN to determine the element numbers connecting each node. Such information is then stored in IELCON.

23. Subroutine SO5. Subroutine SO5 is called by S11 to generate the stiffness matrix of triangular elements under plane stress conditions.

24. Subroutine SO7. Subroutine SO7 is called by S11 to form the stiffness matrix of triangular elements in bending.

25. Subroutine S11. Subroutine S11 is called by QDR. It has been adopted from ELAS (Ref. 6) to generate stiffness matrices for triangular shell elements. Subroutines ADM, STRA, SO5, SO7, TRAN, TRIM, TRM, and INV used in this connection are the modified ver-

sions of their respective counterparts in ELAS and are explained in brief detail only.

26. Subroutine STIMAT. Subroutine STIMAT is called by ACC to form six rows of stiffness matrix $(6,90)$ pertaining to a node. A maximum number of eight elements are assumed to connect a node; the procedure consists in determining the stiffness matrices of individual elements in the global coordinate system and assembling them simultaneously to form the overall stiffness matrix for the node. The essential steps for each element are as follows:

- (1) Stiffness matrix $AK1$ is copied for the quadrilateral element from array $B1$, which stores in-core such stiffness matrices in the local coordinate system for all different types of elements.
- (2) Direction cosine matrix DIR pertaining to the element is next formed. $AK1$ is converted into the global coordinate system, the related formulation being $AK1_G = DIR^T AK1_L DIR$, and the original matrix being replaced by the transformed one.
- (3) The elements in $AK1$ are placed in appropriate positions to form the overall stiffness matrix $AK(6,90)$, utilization being made of nodal and element connection information provided by arrays M and IEL .

27. Subroutine STRA. Subroutine STRA is not being used but has been retained for future use and may be utilized to transform the element matrices description from a local to an overall coordinate system.

28. Subroutine STRESS. Subroutine STRESS is called twice by IMAN during a Runge-Kutta time interval to compute stresses at the center of area of each quadrilateral element. Such calculations are performed at the end of the revised half step and also at the end of the full time step. The related major computational steps for each element are as follows:

- (1) The direction cosine matrix DIR is obtained for the element, based on Q^n , i.e., on the structural configuration at the start of the present time step. QCA is called to obtain surface areas of the two triangles into which the quadrilateral element has been divided by its diagonal.
- (2) The material matrix $D33$ is formed for the plane stress case.
- (3) Nodal coordinates are computed based on Q^n and incremental deformations at the nodes based on

QI^{n+1} . Such values are transformed in the local coordinate system of the element to obtain matrix $A(3,12)$.

- (4) For each of the two triangles the following procedures are next adopted:
 - (a) The coefficient matrix $C8(3,6)$ is formed from A . The deformation file DF is computed incorporating effects of changes in normal curvatures and surface twists, such fields being assumed constant throughout.
 - (b) Triangular element stress increments are then obtained from $D33$, $C8$, and DF files.
- (5) Average stress increments for the quadrilateral element are next calculated from such stresses in the two triangular elements obtained in step 4.

29. Subroutine TRANSP. Subroutine TRANSP is called by subroutine ELIM; it transposes a given matrix A , storing the transposed matrix into a B matrix.

30. Subroutine TRAN. Subroutine TRAN is called by STRA to transform the description of a vector from a local to an overall coordinate system.

31. Subroutine TRIM. Subroutine TRIM is called by S05 to generate certain matrices in connection with the derivation of the plane stress triangular element stiffness matrix.

32. Subroutine TRM. Subroutine TRM is called by S05 and S07 to perform a triple matrix product of order 3.

33. Subroutine UNIT. Subroutine UNIT is called mostly by DCS, STIMAT, and STRESS and is utilized to normalize a vector.

34. Subroutine VECT. Subroutine VECT is called mostly by DCS, STIMAT, and STRESS to perform vector product operations.

V. Sample Problems

An earlier work (Ref. 7) presented details of high-impact dynamic response analysis of a PPO (polyphenylene oxide) plate, the results of which were further compared with the equivalent linear analysis. Present work has mostly been in connection with the solution of heat-sterilizable, impact-resistant battery cells (Ref. 8), the

main components of the cells being a PPO container with silver-zinc negative and zirconium-silver positive plates.

The program IMAN was utilized to analyze the PPO container of a 5-Ah cell, simulating its impact on a rigid surface at a velocity of 120 ft/s. Further analysis was performed on a negative plate subjected to an equivalent impact velocity of 10 ft/s, and such results are presented herein. The damping constants have been assumed to be zero for the rather short impact duration.

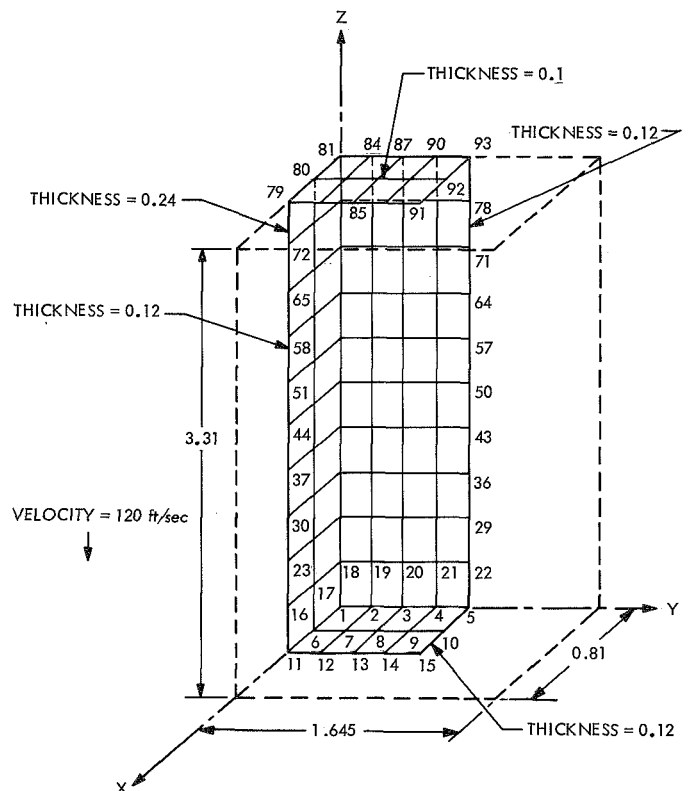


Fig. 2. Battery cell container with finite element idealization (all dimensions in inches)

A. Impact Analysis of a PPO Container

Owing to its symmetric configuration, only one quarter of the PPO cell container structure is considered in its finite element idealization (Fig. 2). The masses are considered to be lumped at the nodes and are based on the total mass of the cell. The primary relevant data used for the analysis are as follows:

Impact velocity, ft/s	120
Uniform time increment, s	0.138×10^{-5}

Number of such increments	34
Total number of nodes	93
Total number of finite elements	76
Young's modulus, lb/in. ²	3.75×10^5
Unit weight, lb/in. ³	0.384×10^{-2}
Poisson's ratio	0.33

Analyses of results depicting propagation of typical support reactions, nodal deformations, and element stresses are presented in Figs. 3-5, respectively. Maximum compressive element stresses are found to be about 10,000 psi, the corresponding yield stress being 17,600 psi. The problem was solved on a UNIVAC 1108 computer, the run time being about 37 min.

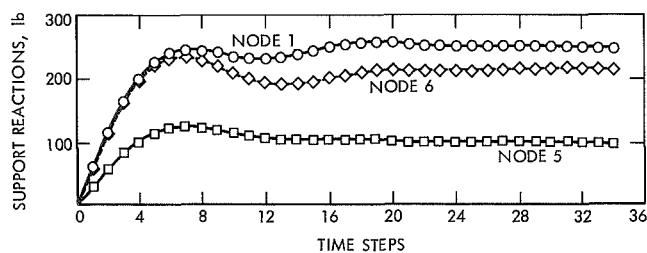


Fig. 3. Support reactions in cell container

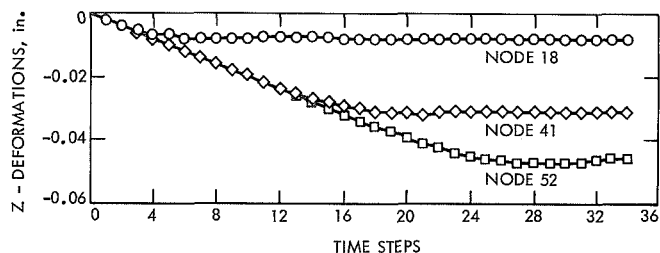


Fig. 4. Nodal deformations in cell container

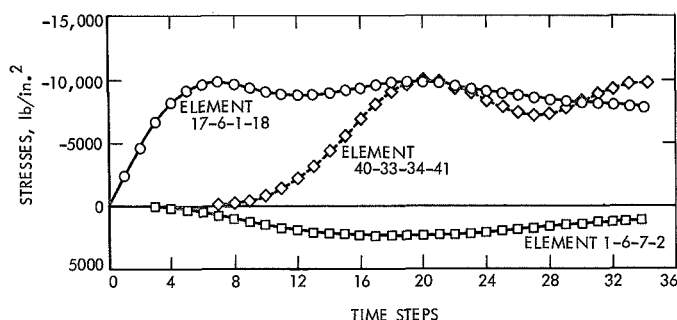


Fig. 5. Stress propagation in typical finite elements of cell container (stresses refer to element top layer)

B. Impact Analysis of a Silver–Zinc Negative Plate

A negative plate of a 5-Ah cell was next considered for impact analysis, and the related finite element model is presented in Fig. 6. The wider upper part of the model is idealized as a uniform silver plate of an equivalent thickness of 0.013 in., thereby replacing the usual 0.037-in.-thick silver grid. The masses are assumed to be lumped at the nodes and consist of the mass of the silver plates and the spongy zinc filling material. Associated relevant data used for the analysis are as follows:

Impact velocity, ft/s	10
Uniform time increment, s	0.19×10^{-5}
Number of such increments	35
Total number of nodes	100
Total number of finite elements	78
Young's modulus, lb/in. ²	10.3×10^6
Unit weight, lb/in. ³	0.381
Poisson's ratio	0.367

Results are presented in graphical form in Figs. 7-10.

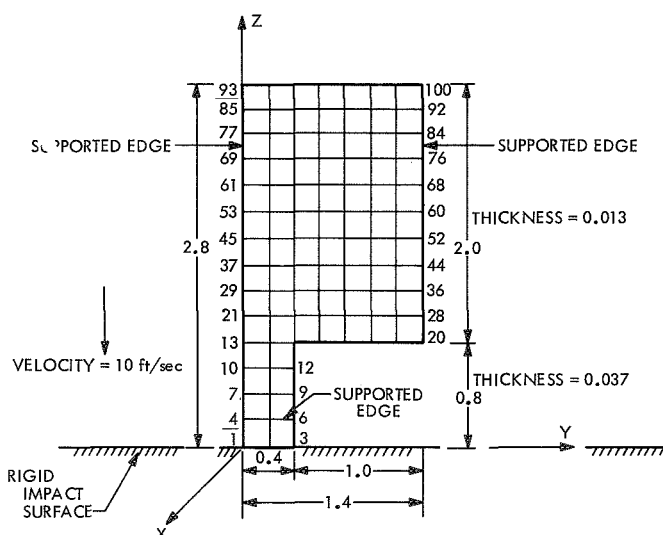


Fig. 6. Finite element idealization of cell negative plate
(all dimensions in inches)

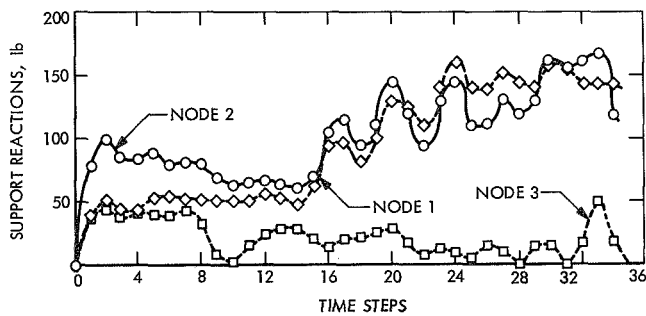


Fig. 7. Support reactions in a negative plate

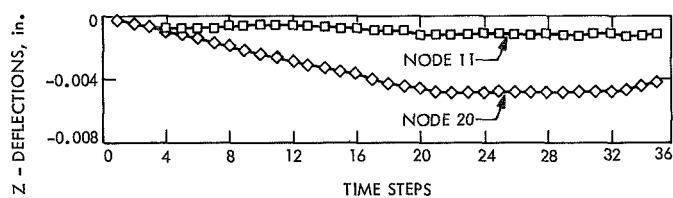


Fig. 8. Nodal deformations in a negative plate

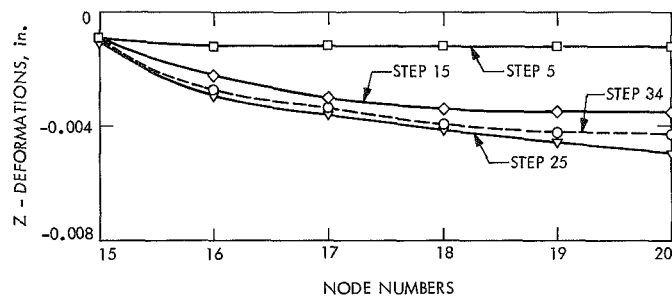


Fig. 9. Deformed configuration of unsupported lower edge of a negative plate at various time steps (edge connecting nodes 15 and 20)

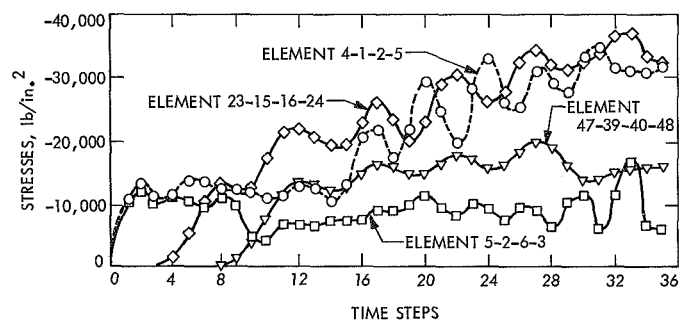


Fig. 10. Stress propagation in typical finite elements of a negative plate

Appendix A

Flowcharts

The flowcharts of the main program IMAN and the associated subroutines, arranged alphabetically, are presented in this appendix as Figs. A-1 through A-25. Each such program flowchart should be considered in conjunction with its respective description in Section IV and with the source program listing in Appendix B. The number associated with any block appearing in the flowchart refers to the first statement number in the corresponding program listing. An asterisk before the number indicates that the first statement in the block appears before the statement corresponding to the number, whereas an asterisk after the number indicates that the first statement in the block appears after the statement corresponding to the number.

Flowcharts for subroutines adopted from Ref. 6 have not been reproduced here.



Fig. A-1. Flowchart of main program IMAN

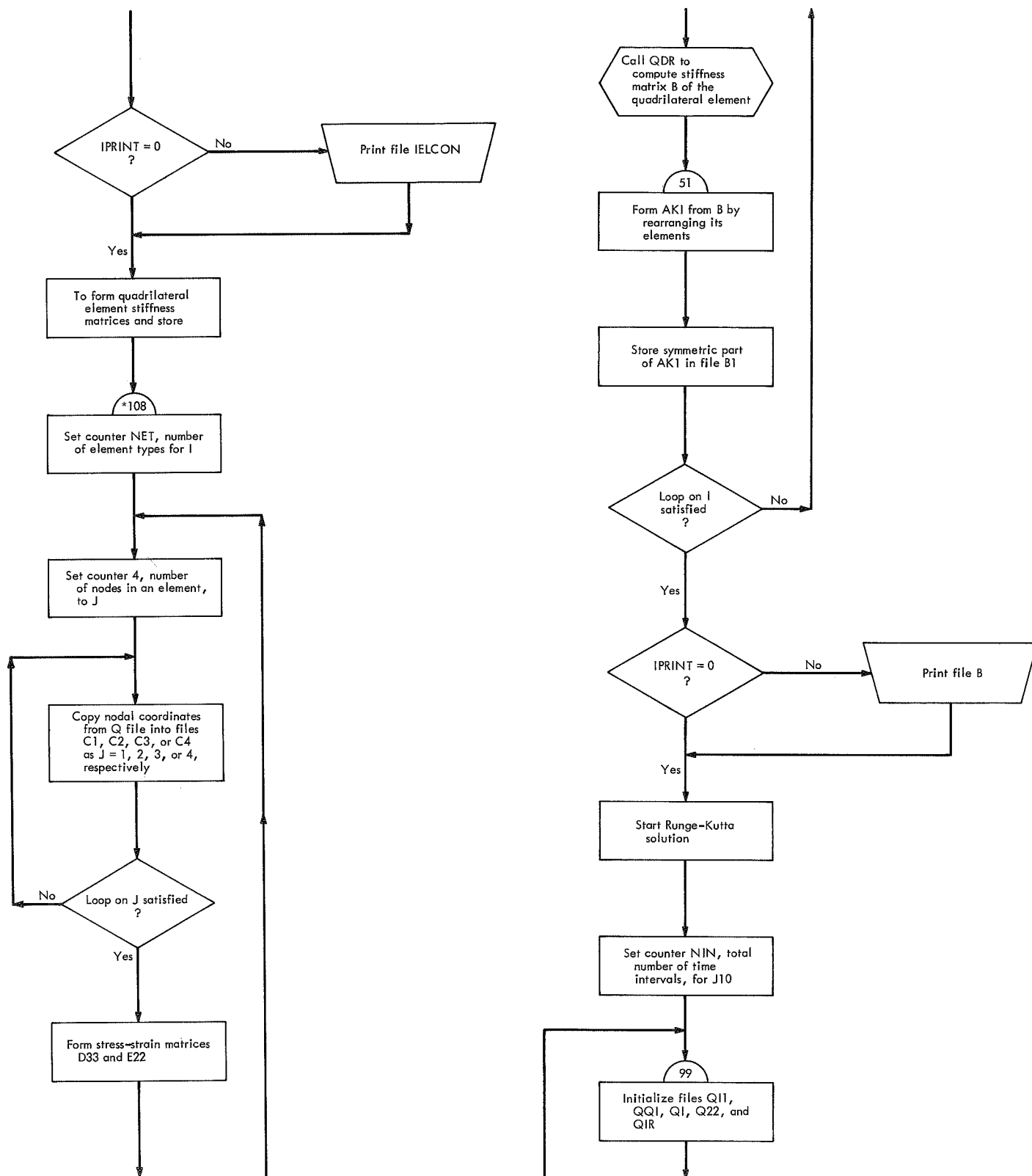


Fig. A-1 (contd)

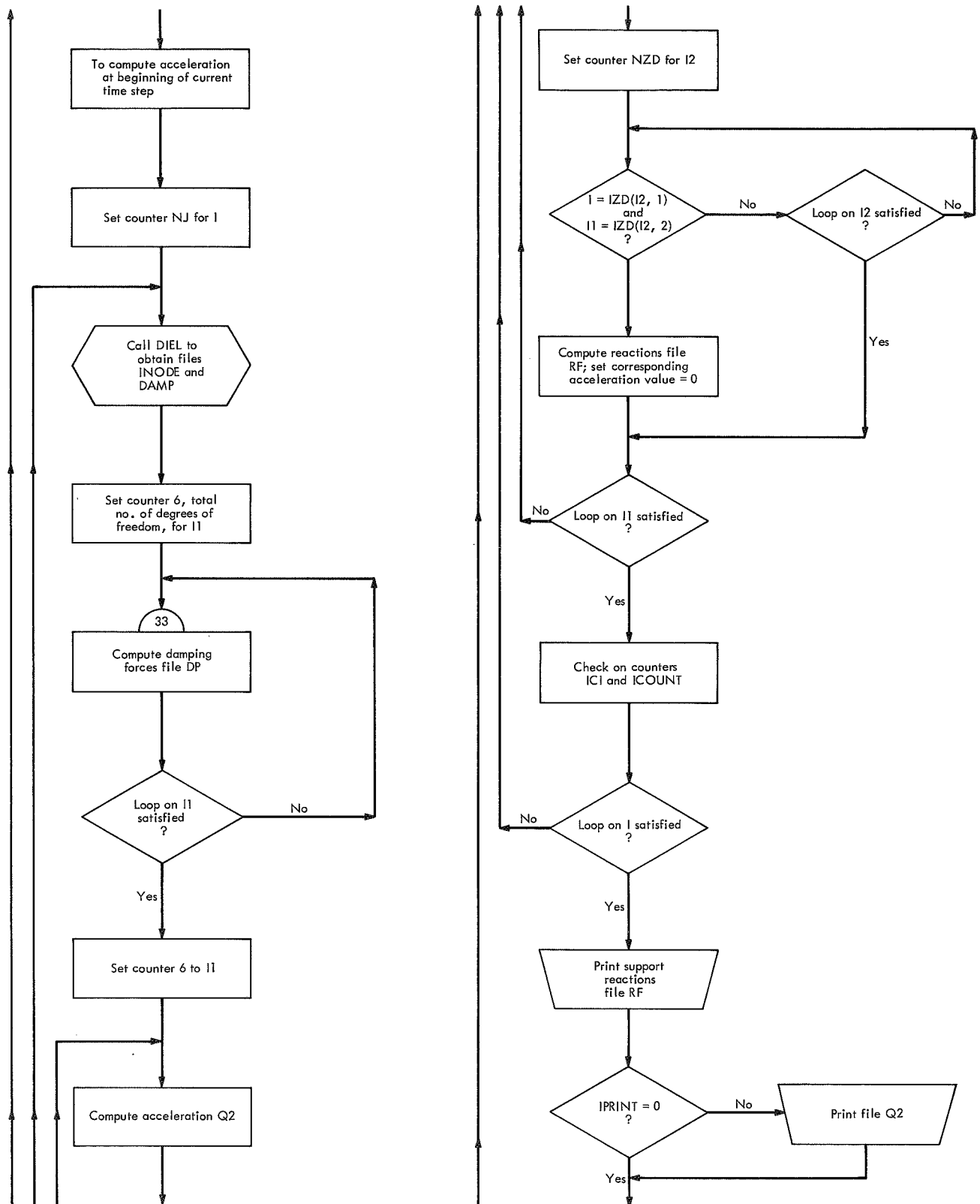


Fig. A-1 (contd)

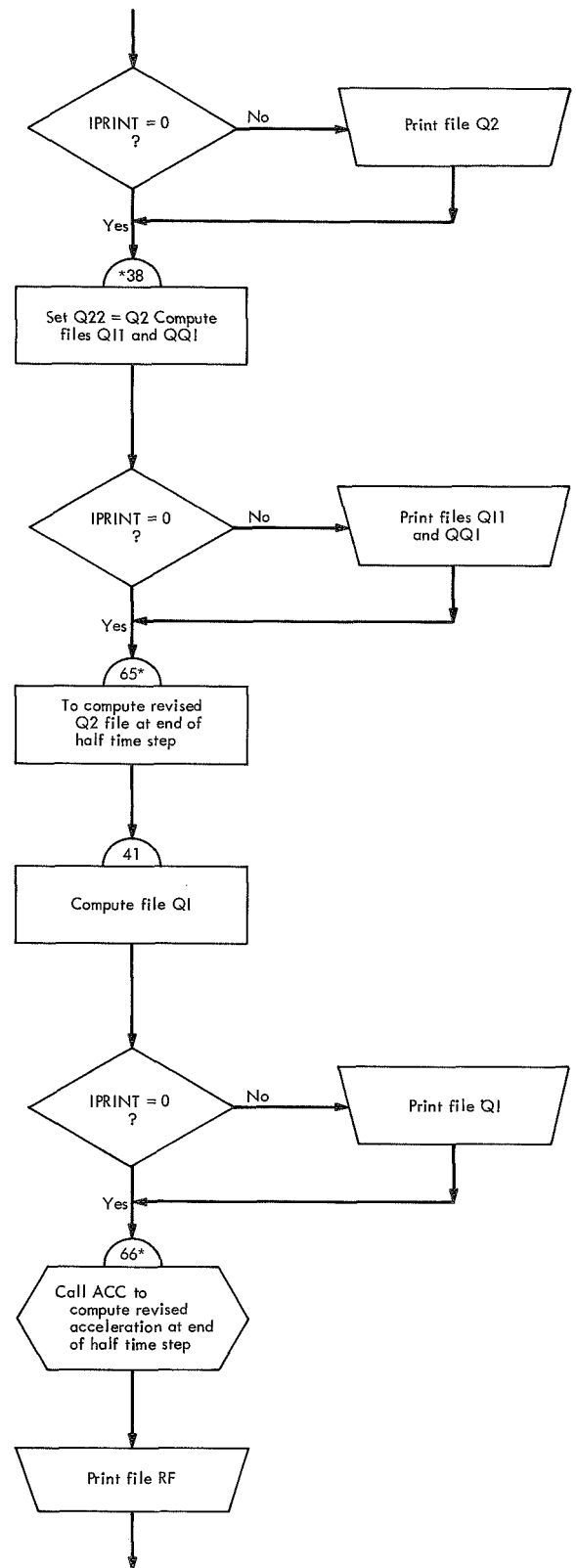
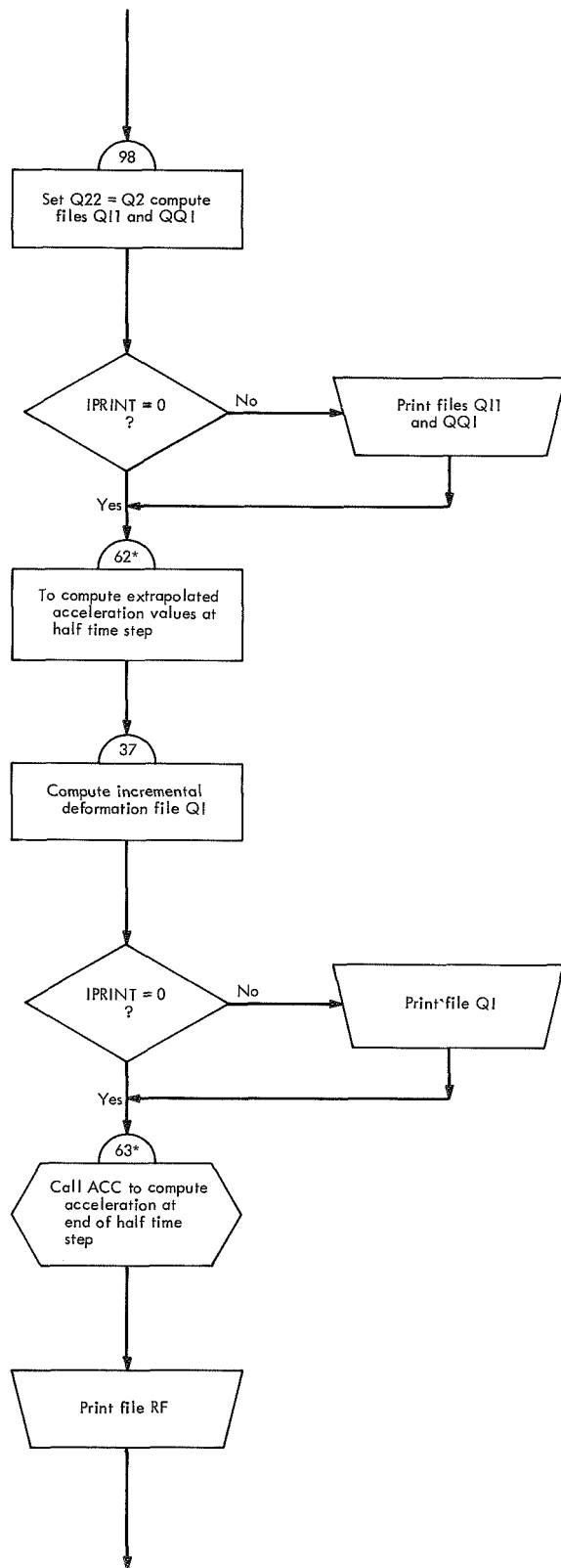


Fig. A-1 (contd)

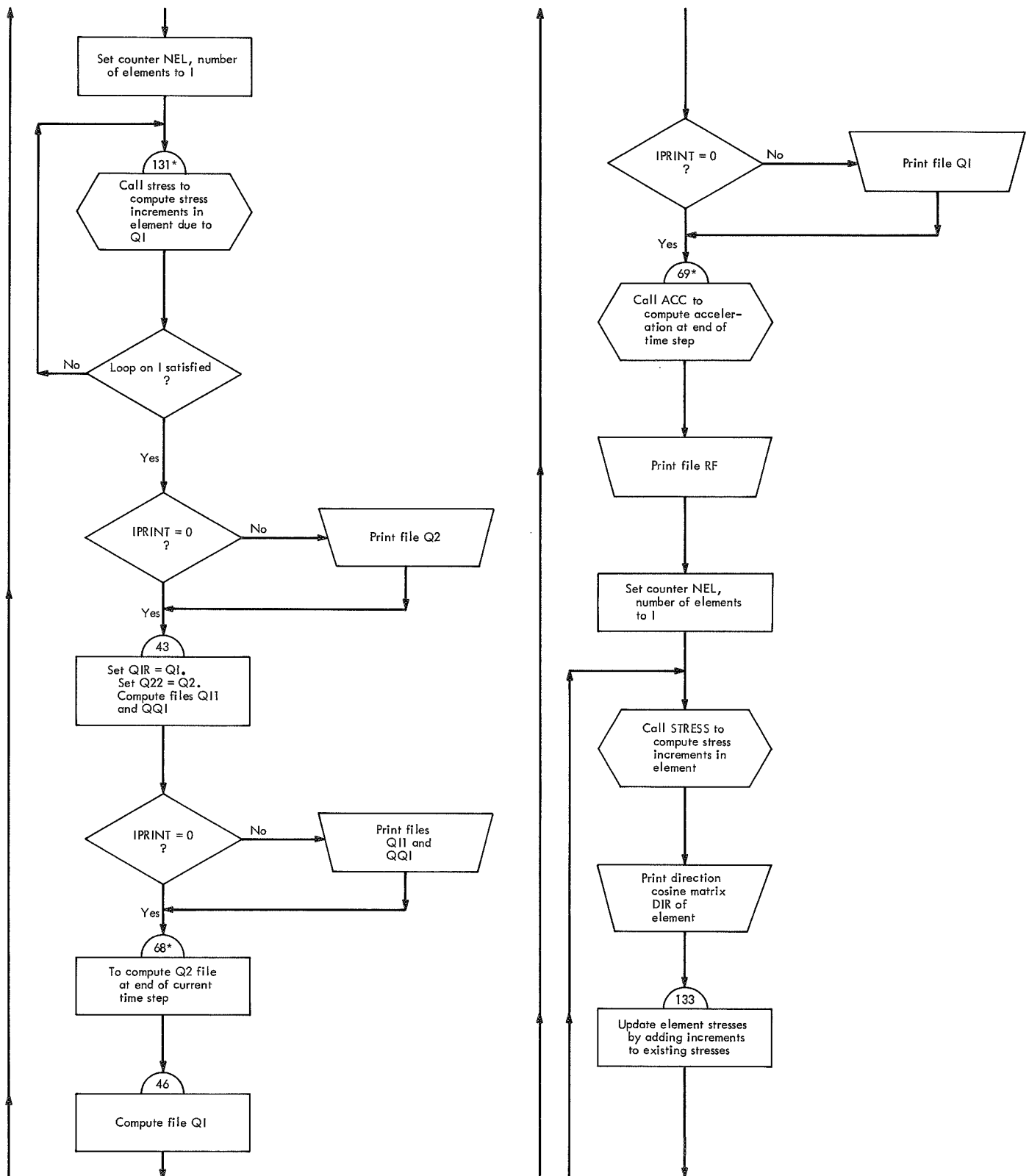


Fig. A-1 (contd)

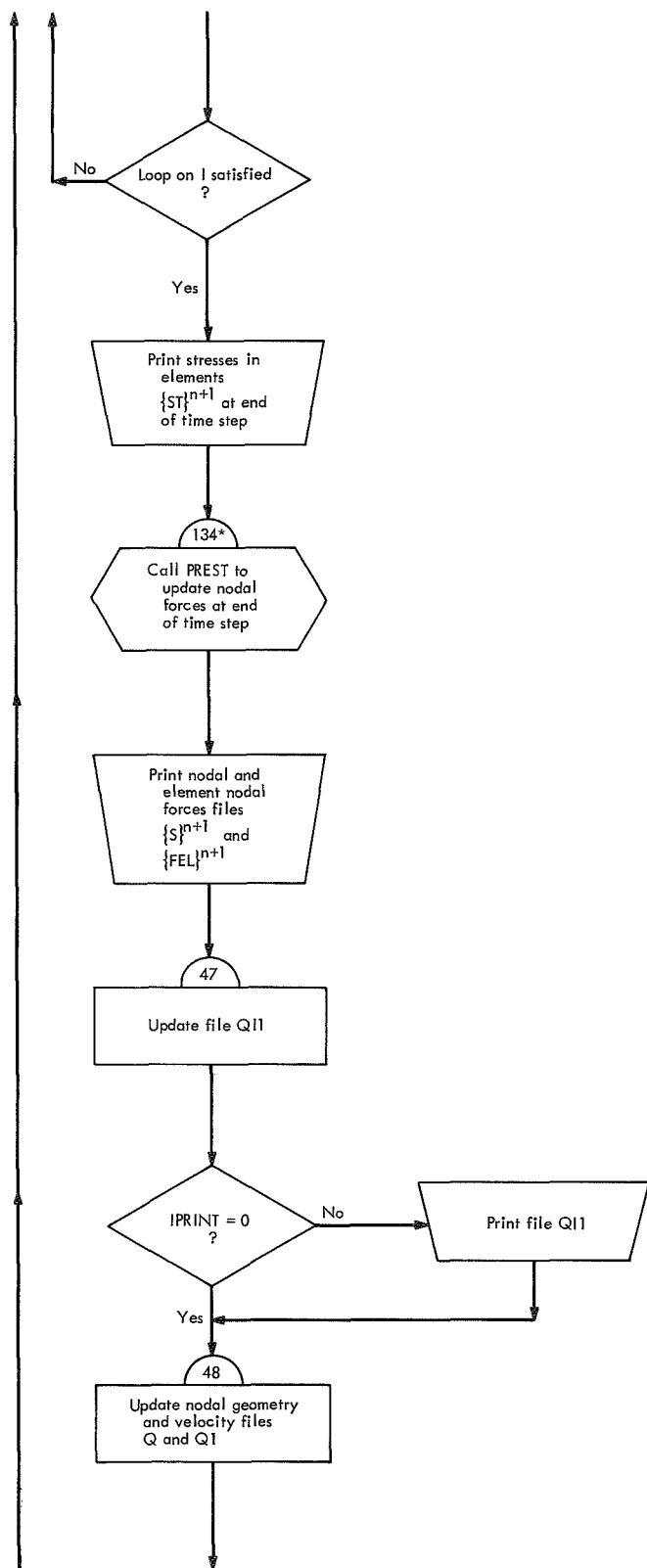
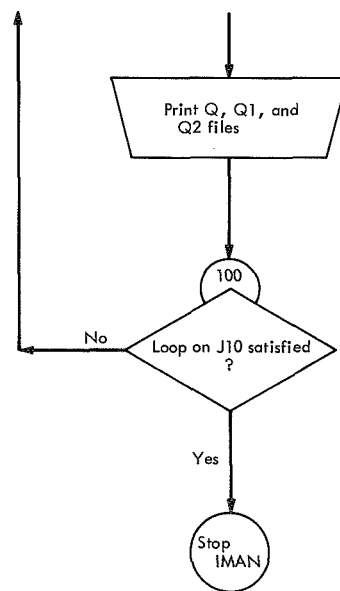


Fig. A-1 (contd)



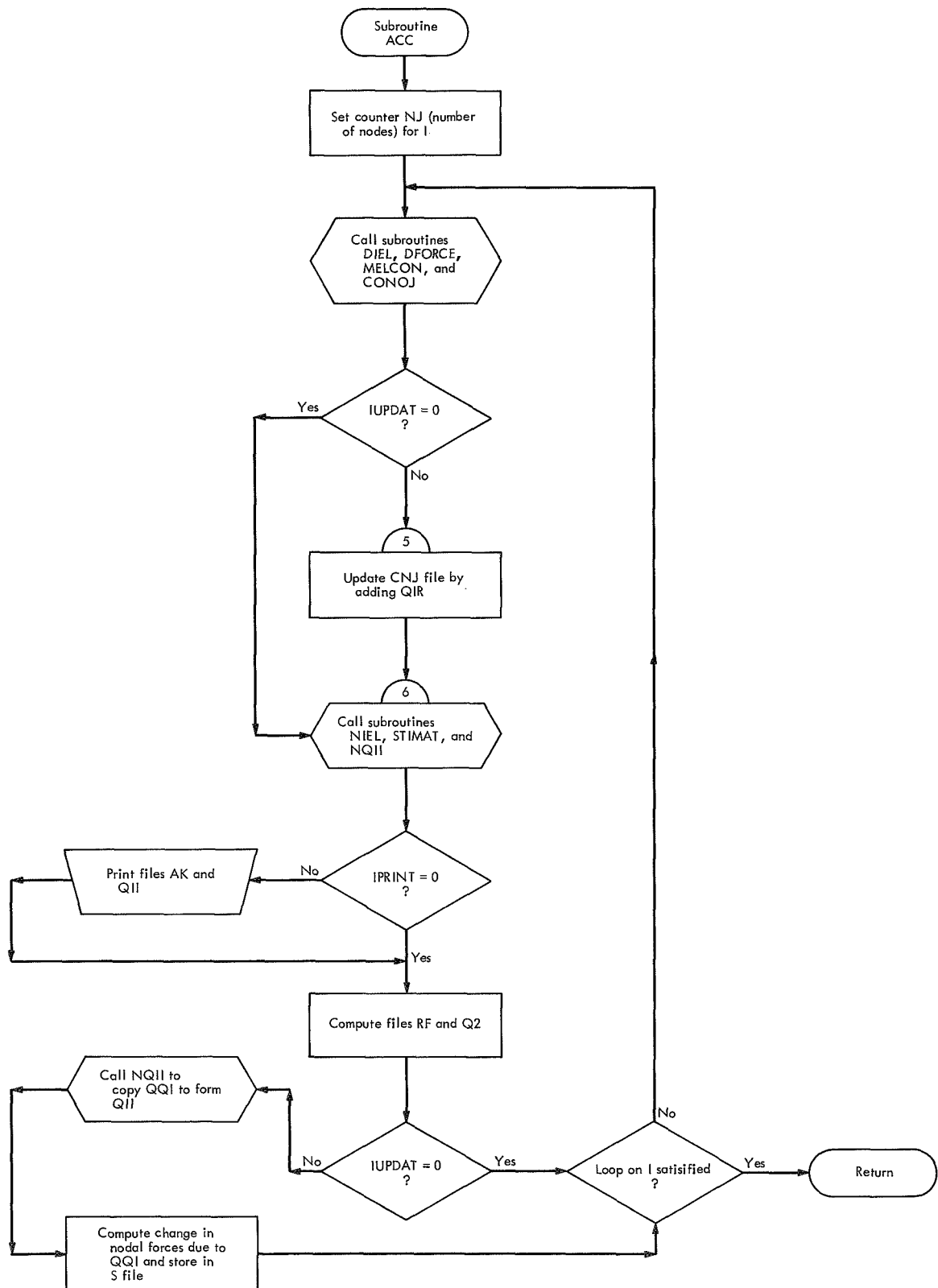


Fig. A-2. Flowchart of subroutine ACC

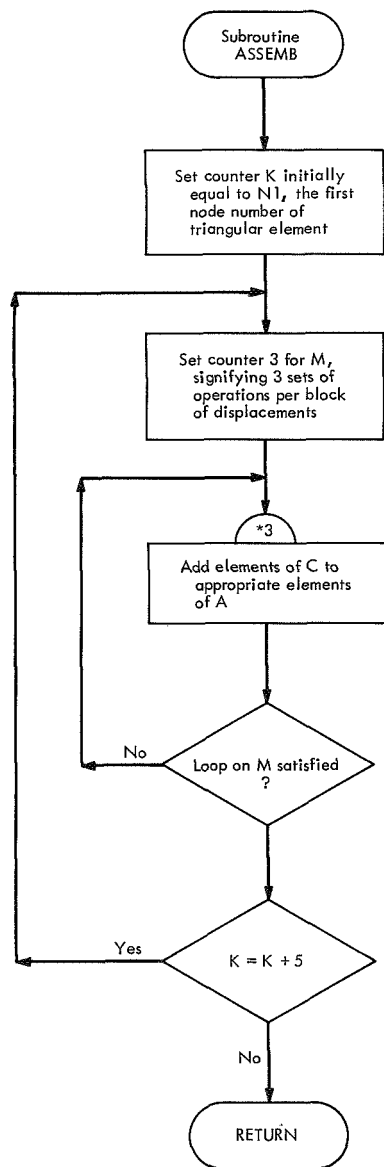


Fig. A-3. Flowchart of subroutine ASSEMB

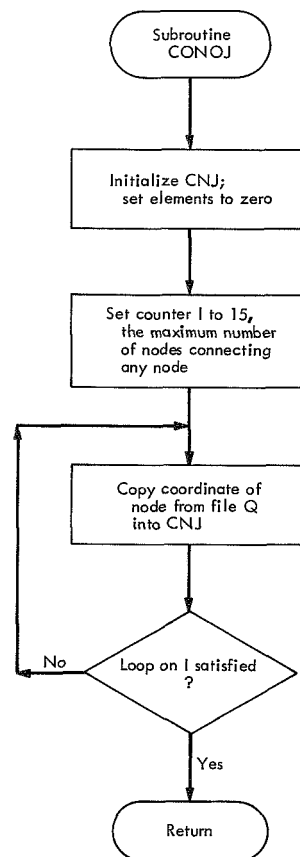


Fig. A-4. Flowchart of subroutine CONOJ

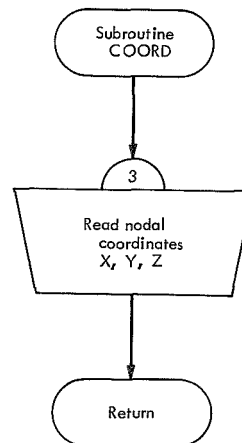


Fig. A-5. Flowchart of subroutine COORD

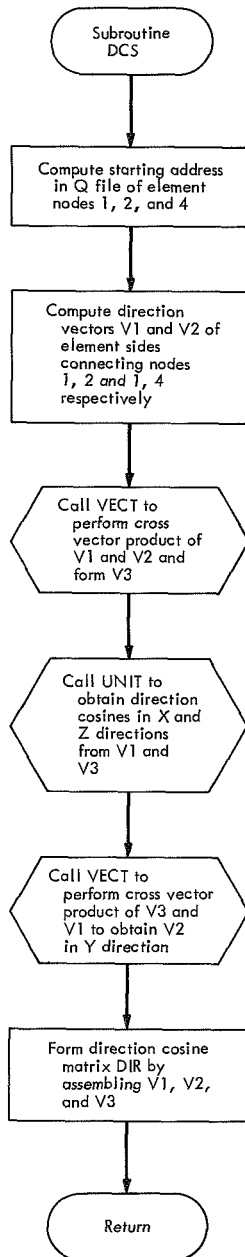


Fig. A-6. Flowchart of subroutine DCS

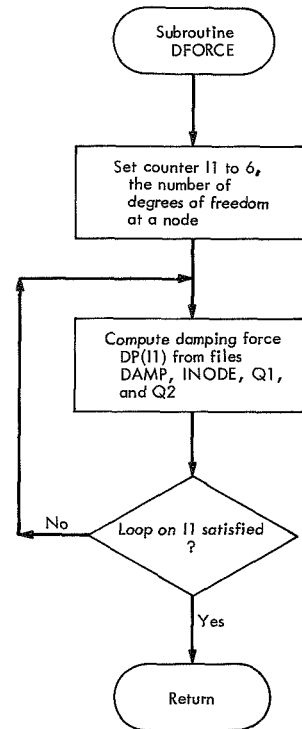


Fig. A-7. Flowchart of subroutine DFORCE

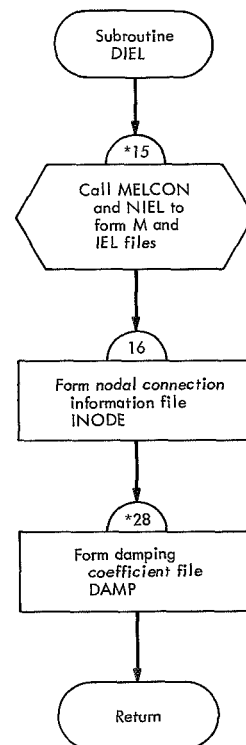


Fig. A-8. Flowchart of subroutine DIEL

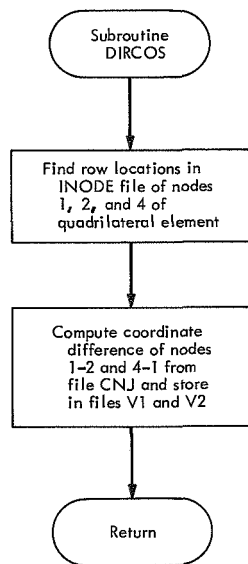


Fig. A-9. Flowchart of subroutine DIRCOS

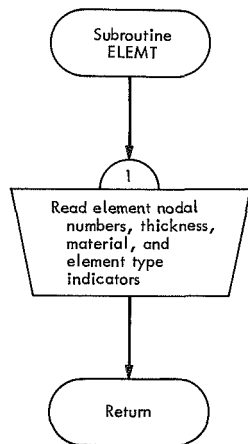


Fig. A-10. Flowchart of subroutine ELEMt

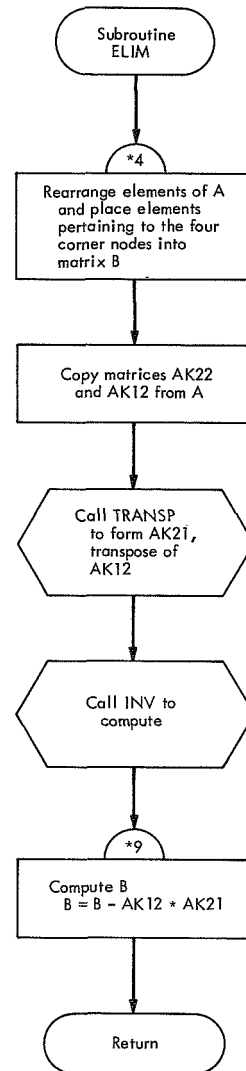


Fig. A-11. Flowchart of subroutine ELIM

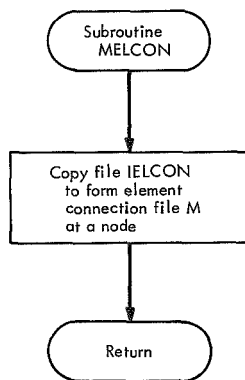


Fig. A-12. Flowchart of subroutine MELCON

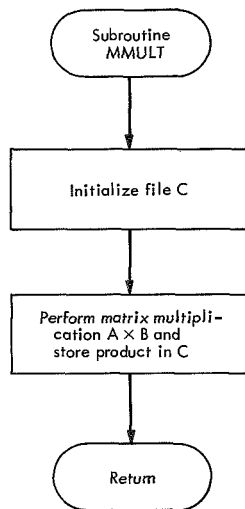


Fig. A-13. Flowchart of subroutine MMULT

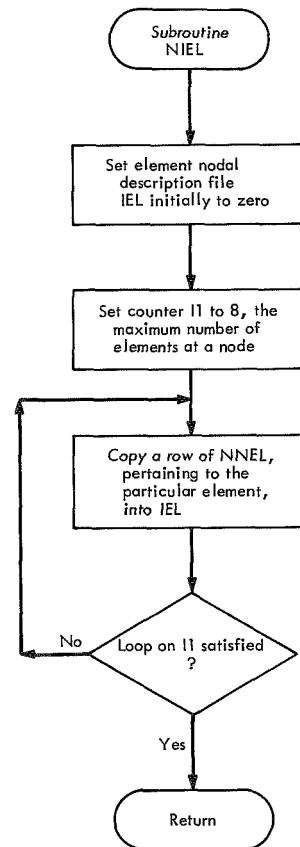


Fig. A-14. Flowchart of subroutine NIEL

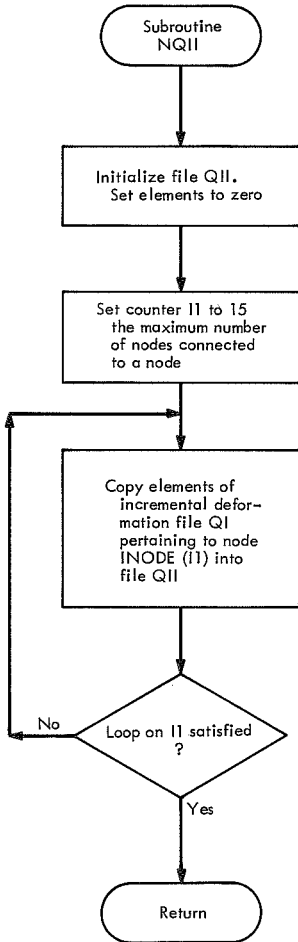


Fig. A-15. Flowchart of subroutine NQII

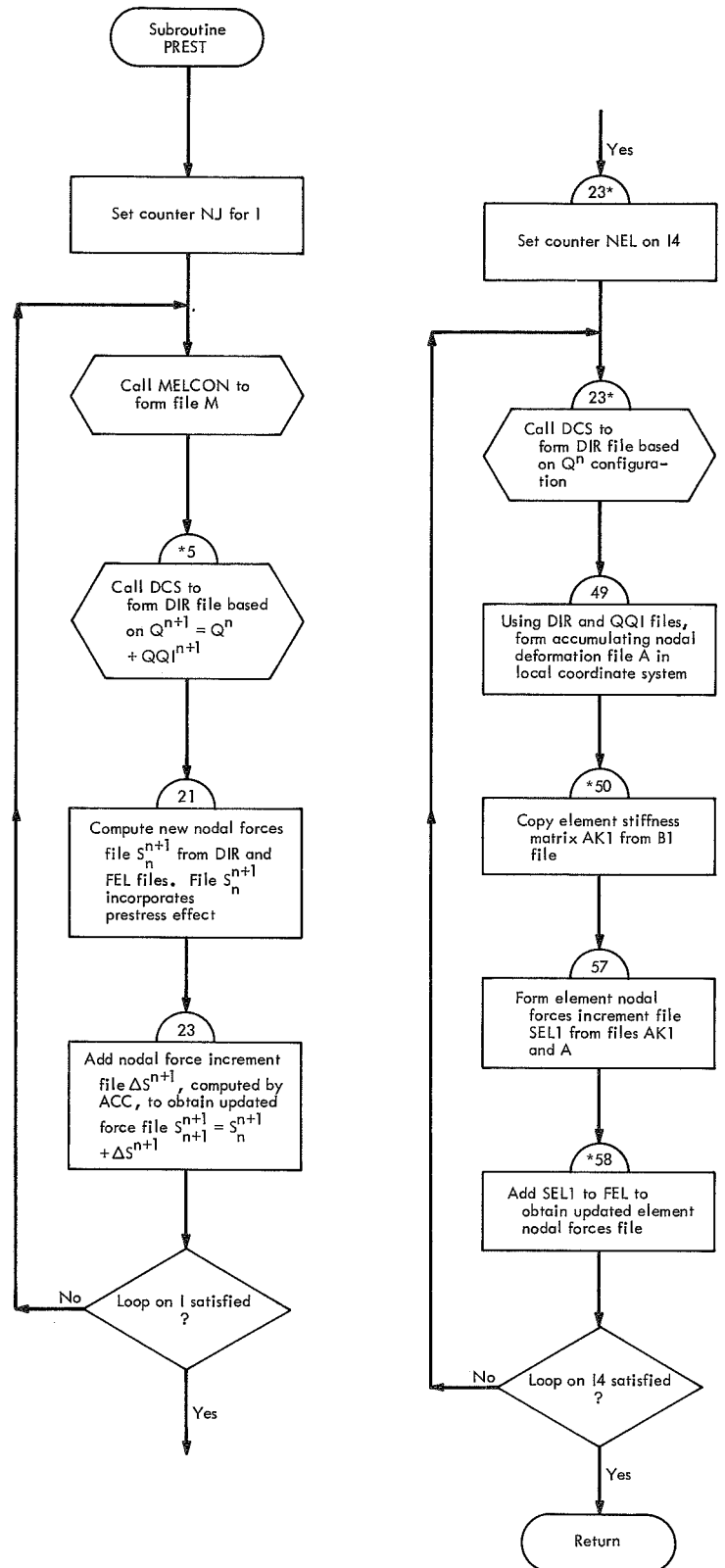


Fig. A-16. Flowchart of subroutine PREST

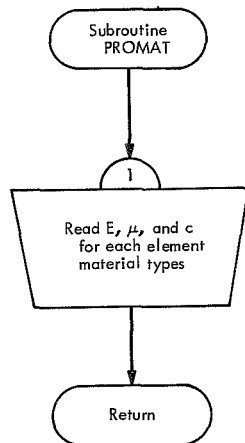


Fig. A-17. Flowchart of subroutine PROMAT

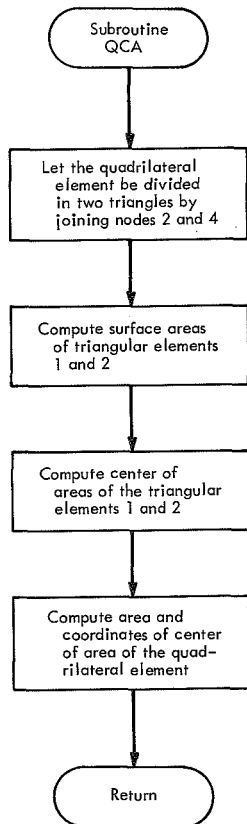


Fig. A-18. Flowchart of subroutine QCA

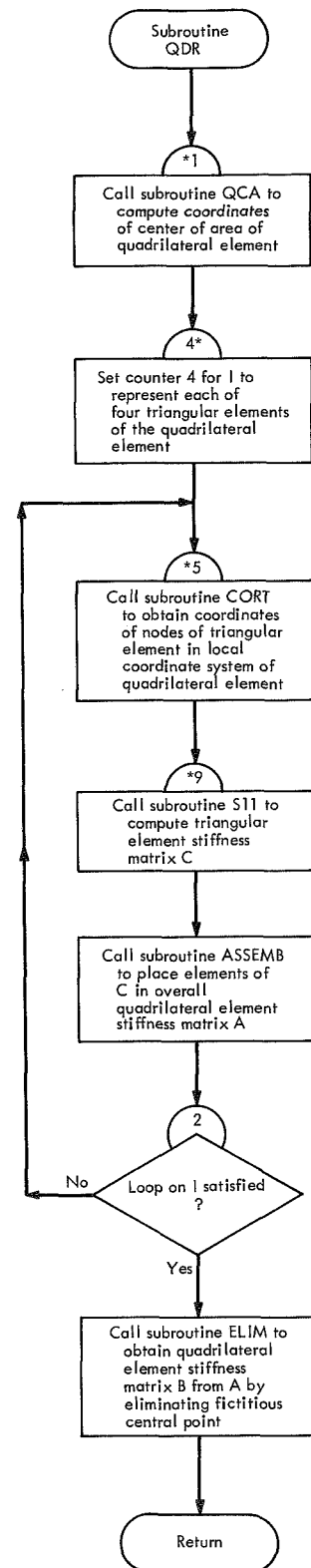


Fig. A-19. Flowchart of subroutine QDR

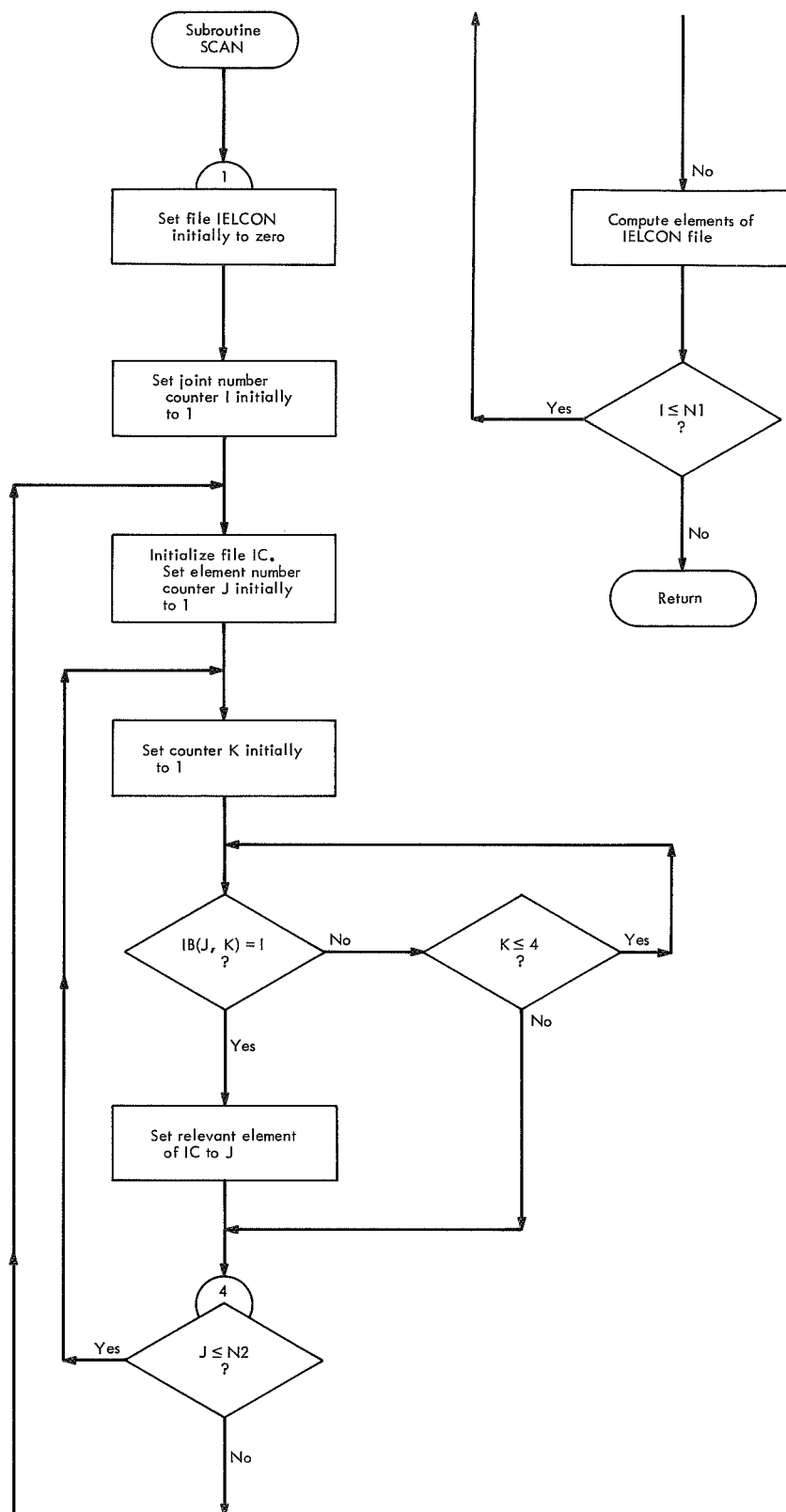


Fig. A-20. Flowchart of subroutine SCAN

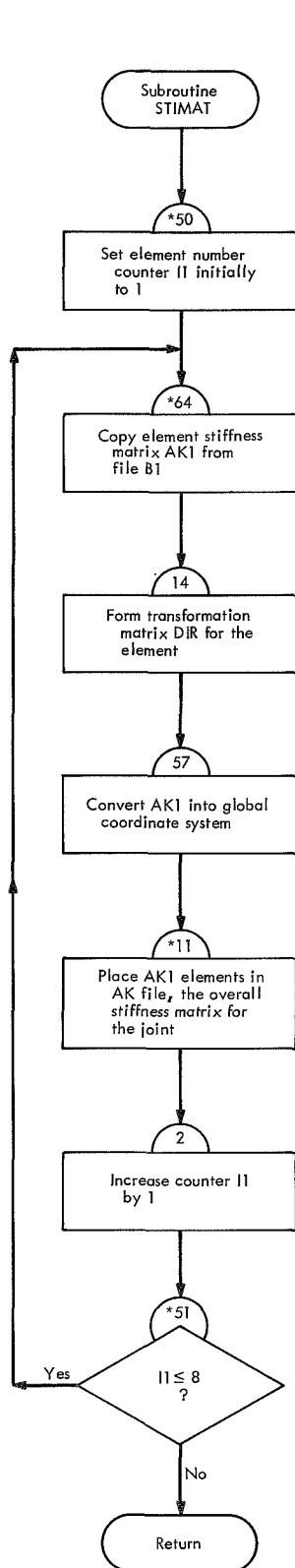


Fig. A-21. Flowchart of subroutine STIMAT

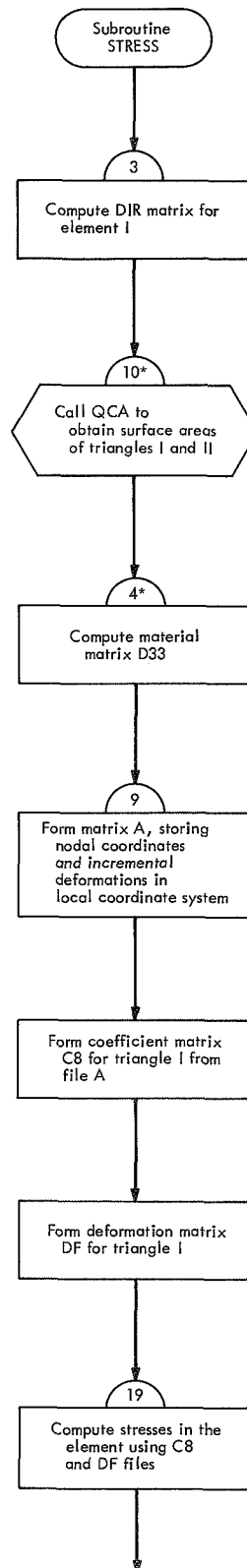


Fig. A-22. Flowchart of subroutine STRESS

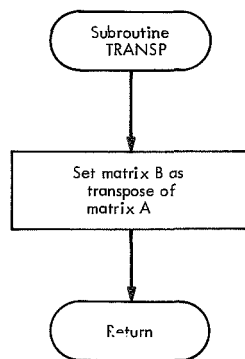


Fig. A-23. Flowchart of subroutine TRANSP

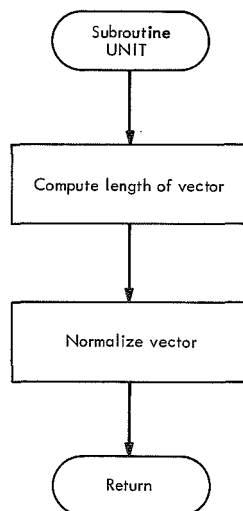


Fig. A-24. Flowchart of subroutine UNIT

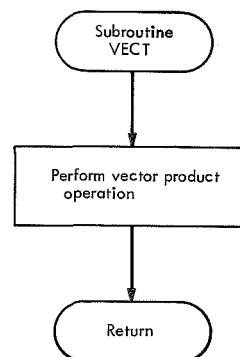


Fig. A-25. Flowchart of subroutine VECT

Appendix B

Source Program Listings

Tables B-1 through B-35 present the program listings of the main program IMAN and the associated subroutines, arranged alphabetically. Detailed explanations of the associated variables and arrays may be obtained from Section III.

Table B-1. Source program listing of main program IMAN

```

C MAIN DRIVER PROGRAM IMAN IMAN0001
C GENERALIZED HIGH IMPACT DYNAMIC RESPONSE ANALYSIS OF GEOMETRICALLY IMAN0002
C NON LINEAR STRUCTURES BY K.K. GUPTA. IMAN0003
C STRUCTURAL IDEALISATION IS BASED ON FINITE ELEMENT METHOD USING IMAN0004
C QUADRILATERAL SHELL ELEMENTS. IMAN0005
C MATRIX DISPLACEMENT METHOD EMPLOYING A NODEWISE PREDICTOR-CORRECT IMAN0006
C -OR RUNGE-KUTTA APPROACH IS ADOPTED FOR THE SOLUTION IMAN0007
C THE PROGRAM CATERES TO GEOMETRICAL NONLINEARITY INVOLVING LARGE IMAN0008
C DEFORMATIONS INCLUDING LARGE ROTATIONS OF THE FINITE ELEMENTS. IMAN0009
C PARAMETER NJ=3,NEL=76,NZD=115,NMAT=1,NEL=6,NIN=35 IMAN0010
C PARAMETER ITIME=0,IPRINT=0 IMAN0011
C PARAMETER NDF=6,NJ,NEL=300,NEL,NST=6,NEL,NEL=29,NEL IMAN0012
C FIRST TWO PARAMETER CARDS MUST BE CHANGED FOR EACH PROBLEM. IMAN0013
C DIMENSION NNEL(NEL,4),TEL(NEL),INT(NEL),PMINMAT(3),CMASS(NDF), IMAN0014
C IZD(NZD,2),TJOB(13),IELCON(NJ,4),C1(3),C2(3),C3(3),C4(3),C7(3), IMAN0015
C ZIC(8),MIR(3),V1(3),V2(3),V3(3),V4(3),DP(6),R1(20,20),Q1(NDF),Q2(NDF), IMAN0016
C 3Q1(90),Q3(NDF),Q4(NDF),Q1(NDF),Q1(NDF),Q2(NDF),Q2(NDF),Q2(NDF), IMAN0017
C NS(NDF),PMINMAT(1),AK1(5),AK1(29,29),INODE(15),TEL(8,4),DAMP(15), IMAN0018
C 5DM(8),CN1(15,3),BL(NEL),RF(NZD),D3(13,3),E22(3,3),DIR(3,3), IMAN0019
C 6MLT(NEL,5),TNLT(NEL,1),INT(NEL),ST(NST),FEL(NEL,2),TJNT(MTN), IMAN0020
C WRITE(6,1) IMAN0021
C 1 FORMAT(1H1) IMAN0022
C READ(5,2) TJOB IMAN0023
C 2 FORMAT(13A6) IMAN0024
C N3=4 IMAN0025
C INDS=0 IMAN0026
C INDP=0 IMAN0027
C IF (ITIME.EQ.0) GO TO 222 IMAN0028
C READ(5,199) (TINT(I),I=1,NIN) IMAN0029
C 199 FORMAT(3F11.3) IMAN0030
C GO TO 225 IMAN0031
C 222 READ(5,223) TI IMAN0032
C 223 FORMAT(E11.3) IMAN0033
C DO 224 I=1,NIN IMAN0034
C 224 TINT(I)=TI IMAN0035
C 225 CONTINUE IMAN0036
C SETTING INITIAL VELOCITY FILE IMAN0037
C READ(5,4) VIN,IOFY IMAN0038
C 4 FORMAT(E11.3,I5) IMAN0039
C DO 5 I=1,NDF IMAN0040
C 5 Q1(I)=0.0 IMAN0041
C DO 6 I=1,IOFY,NDF,6 IMAN0042
C 6 Q1(I)=VIN IMAN0043
C READ(5,7) ((IZD(I,J),J=1,2),I=1,NZD) IMAN0044
C 7 FORMAT(5(2I5)) IMAN0045
C DO 8 I=1,NZD IMAN0046
C 8 J=(IZD(I,1)-1)*6+IZD(I,2) IMAN0047
C Q1(J)=0.0 IMAN0048
C WRITE(6,1) IMAN0049
C WRITE(6,3) TJOB IMAN0050
C 9 FORMAT(1H 13A6) IMAN0051
C WRITE(6,161) IMAN0052
C 161 FORMAT(22H ZERO DEFORMATION FILE) IMAN0053
C WRITE(6,160) IMAN0054
C 160 FORMAT (1H ///) IMAN0055
C WRITE(6,162) ((I7D(K10,10),K10=1,2),K10=1,NZD) IMAN0056
C 162 FORMAT(1H 215) IMAN0057
C WRITE(6,1) IMAN0058
C WRITE(6,163) IMAN0059
C 163 FORMAT(27H INITIAL VELOCITY (Q1) FILE) IMAN0060
C WRITE(6,160) IMAN0061
C WRITE(6,172) (Q1(K10),K10=1,NDF) IMAN0062
C SETTING INITIAL COORDINATE AND DEFORMATION FILE IMAN0063
C CALL COORDIN(NDF) IMAN0064
C WRITE(6,1) IMAN0065
C WRITE(6,165) IMAN0066
C 165 FORMAT(38H INITIAL DEFORMATION/NODAL COORD. FILE) IMAN0067
C WRITE(6,160) IMAN0068
C WRITE(6,172) (Q(K10),K10=1,NDF) IMAN0069
C DO 10 I=1,NDF IMAN0070
C Q2(I)=0.0 IMAN0071
C S(I)=0.0 IMAN0072
C Q1(I)=0.0 IMAN0073
C Q3(I)=0.0 IMAN0074
C Q4(I)=0.0 IMAN0075
C DO 14 I=1,NEL24 IMAN0076
C 14 FEL(I)=0.0 IMAN0077
C DO 32 I=1,NST IMAN0078
C 32 ST(I)=0.0 IMAN0079
C TO READ ELEMENT NODAL NUMBERS,THICKNESS AND MATERIAL TYPE IMAN0080
C CALL ELMENT(NNEL,TEL,INT,INLT,NLT,NEL,NEL,4) IMAN0081
C WRITE(6,1) IMAN0082
C WRITE(6,167) IMAN0083
C 167 FORMAT(31H ELEMENT NODAL DESCRIPTION FILE) IMAN0084
C WRITE(6,160) IMAN0085
C WRITE(6,168) ((NNEL(K10,L10),L10=1,4),TEL(K10),INT(K10),INLT(K10), IMAN0086
C IK10=1,NEL) IMAN0087
C 168 FORMAT(1H WTS,F8.4*2I4) IMAN0088
C WRITE(6,1) IMAN0089
C WRITE(6,113) IMAN0090
C 113 FORMAT(18H ELEMENT TYPE FILE) IMAN0091
C WRITE(6,160) IMAN0092
C WRITE(6,113) ((INLT(K10,L10),L10=1,5),TNLT(K10),K10=1,NEL) IMAN0093
C 114 FORMAT(1H S15,F8.4) IMAN0094
C TO READ MATERIAL PROPERTY MATRIX IMAN0095
C CALL PRONAT(PH,NMAT,3) IMAN0096
C WRITE(6,1) IMAN0097
C WRITE(6,169) IMAN0098
C 169 FORMAT(25H MATERIAL PROPERTIES FILE) IMAN0099
C WRITE(6,160) IMAN0100
C WRITE(6,170) ((PM(K10,L10),L10=1,3),K10=1,NMAT) IMAN0101
C 170 FORMAT(1H E10.2,F6.3,F8.3) IMAN0102
C READ(5,11) (CMASS(I),I=1,NDF) IMAN0103
C 11 FORMAT(E11.3) IMAN0104
C WRITE(6,1) IMAN0105
C WRITE(6,171) IMAN0106
C 171 FORMAT(16H NODAL MASS FILE) IMAN0107
C WRITE(6,160) IMAN0108
C WRITE(6,172) (CMASS(K10),K10=1,NDF) IMAN0109
C 172 FORMAT(1H 6E15.6) IMAN0110
C WRITE(6,1) IMAN0111
C WRITE(6,173) IMAN0112
C 173 FORMAT(17H NODAL FORCE FILE) IMAN0113
C WRITE(6,160) IMAN0114
C WRITE(6,172) (P(K10),K10=1,NDF) IMAN0115
C TO FORM ELEMENT NUMBER FILE CONNECTING INDIVIDUAL NODES IMAN0116
C CALL SCAN(TELCON,NNEL,IC,NJ,NEL,N3) IMAN0117
C WRITE(6,1) IMAN0118
C WRITE(6,175) IMAN0119
C 175 FORMAT(37H ELEMENT NODE CONNECTION FILE(TELCON)) IMAN0120
C WRITE(6,160) IMAN0121
C WRITE(6,176) ((TELCON(K10,L10),L10=1,4),K10=1,NJ) IMAN0122
C 176 FORMAT(1H 4I10) IMAN0123
C TO FORM QUADRILATERAL ELEMENT STIFFNESS MATRICES AND STORE IMAN0124
C DO 108 I=1,NEL1 IMAN0125
C 108 B1(I)=0.0 IMAN0126
C DO 12 J=1,NEL IMAN0127
C DO 80 J=1,4 IMAN0128
C J1=INT(I/J) IMAN0129
C J1=5*(J1-1) IMAN0130
C DO 85 J2=1,3 IMAN0131
C J11=J1+J2 IMAN0132
C I20=J IMAN0133
C GO TO (81,82,83,84),I20 IMAN0134
C 81 C1(J2)=Q(J11) IMAN0135
C GO TO 85 IMAN0136
C 82 C2(J2)=Q(J11) IMAN0137
C GO TO 85 IMAN0138
C 83 C3(J2)=Q(J11) IMAN0139
C GO TO 85 IMAN0140
C 84 C4(J2)=Q(J11) IMAN0141
C 85 CONTINUE IMAN0142
C 80 CONTINUE IMAN0143
C IM=NLTI(1,5) IMAN0144
C EEPK(IM,1) IMAN0145
C VMM=PM IM,2) IMAN0146
C BEE/(2.0*(1.0+VMM)) IMAN0147
C FM=EE/(1.0+VMM**2) IMAN0148
C DO 151 I=1,3 IMAN0149
C DO 151 J1=1,3 IMAN0150
C D33(I1,J1)=0.0 IMAN0151
C 151 E22(I1,J1)=0.0 IMAN0152
C D33(I1,J1)=FM IMAN0153
C D33(I1,2)=FM*VMM IMAN0154
C D33(2,1)=D33(I1,2) IMAN0155
C D33(2,2)=D33(1,1) IMAN0156
C D33(3,3)=FM*(1.0+VMM)/2.0 IMAN0157
C E22(1,1)=G IMAN0158
C E22(2,2)=S IMAN0159
C TE=TNLT(I) IMAN0160
C J0=15 IMAN0161
C CALL QDR(B,NLT(I,1),NLT(I,2),NLT(I,3),NLT(I,4),C1,C2,C3,C4,C7, IMAN0162
C ITC,D33,E22) IMAN0163
C DO 49 I3=1,24 IMAN0164
C DO 49 J3=1,24 IMAN0165
C 49 AK1(I3,J3)=0.0 IMAN0166
C DO 50 I3=1,4 IMAN0167
C DO 50 J3=1,5 IMAN0168
C I4=6*(I3-1)+J3 IMAN0169
C J4=4*(J3-1)+I3 IMAN0170
C DO 51 I5=1,5 IMAN0171
C DO 51 J5=1,4 IMAN0172
C I6=I5+6*(J5-1) IMAN0173
C J6=J5+4*(I5-1) IMAN0174
C 51 AK1(I4,I6)=B(J4,J6) IMAN0175
C 50 CONTINUE IMAN0176
C I3=300*(I-1)+1 IMAN0177
C DO 52 I4=1,24 IMAN0178
C DO 52 J4=1,24 IMAN0179
C B1(I3)=AK1(I4,J4) IMAN0180
C 52 I3=I+1 IMAN0181
C 12 CONTINUE IMAN0182
C IF (IPRINT.EQ.0) GO TO 96 IMAN0183
C WRITE(6,115) IMAN0184
C 115 FORMAT(10H B1 MATRIX) IMAN0185
C WRITE(6,160) IMAN0186
C WRITE(6,116) (B1(K10),K10=1,NEL1) IMAN0187
C 116 FORMAT(1H 10E11.3) IMAN0188
C 96 CONTINUE IMAN0189
C DO 121 I=1,NZD IMAN0190
C 121 RE(I)=0.0 IMAN0191
C RUNGE KUTTA SOLUTION IMAN0192
C DO 100 J10C=1,NIN IMAN0193
C J100=J10 IMAN0194
C TI=TINT(J10) IMAN0195
C IF (TI) 100,100,200 IMAN0196
C 200 CONTINUE IMAN0197
C DO 99 J9=1,NDF IMAN0198
C Q11(J9)=0.0 IMAN0199
C Q1(J9)=0.0 IMAN0200
C Q1(J9)=0.0 IMAN0201
C Q22(J9)=0.0 IMAN0202
C 99 Q1R(J9)=0.0 IMAN0203
C TO FIND ACCELERATION AT BEGINNING OF STEP IMAN0204
C DO 31 I=1,NJ IMAN0205
C CALL DIEL(I,H,IELCON,NJ,IEL,NNEL,NEL,INODE,DAMP,DM,PM,INT,NMAT) IMAN0206
C DO 31 I1=1,6 IMAN0207
C C0=0.0 IMAN0208
C J5=6*(I1-1)+1 IMAN0209
C DO 34 I2=1,15 IMAN0210
C J1=6*(INODE(I2)-1)+I1 IMAN0211
C 34 CD=CD+DAMP(I2)*(Q1(J)-Q1(J1)) IMAN0212
C 33 DP(I1)=C0 IMAN0213
C DO 72 I1=1,6 IMAN0214
C J5=6*(I1-1)+1 IMAN0215
C Q2(J1)=(DP(J1)-S(J1))-DP(I1)/CMASS(J) IMAN0216
C DO 122 I2=1,NZD IMAN0217
C IF (I.EQ.IZD(I2,1).AND.I1.EQ.IZD(I2,2)) GO TO 123 IMAN0218
C 122 CONTINUE IMAN0219
C GO TO 72 IMAN0220
C 123 RE(I2)=Q2(J1)/CMASS(J) IMAN0221
C IF (RF(I2)) 72,226,226 IMAN0222
C IMAN0223
C IMAN0224
C IMAN0225
C IMAN0226

```

Table B-1 (contd)

226 Q2(J)=0.0	IMAN0227	WRITE(6,160)	IMAN0340
72 CONTINUE	IMAN0228	DO 135 I=1,NEL	IMAN0341
ICI=ICI+1	IMAN0229	CALL STRESS(I,NMAT,NDF,NEL,PM,D33,DIR,INT,INT,I,T,QT,Q,N,NEL)	IMAN0342
IF(ICI.NE.15) GO TO 31	IMAN0230	ICI=CI3	IMAN0343
ICOUNT=ICOUNT+ICI	IMAN0231	WRITE(6,132) I,CI(I10),I10=1,3),(CI(I10),I10=1,3)	IMAN0344
ICI=0	IMAN0232	132 FORMAT(1H 16,E17.4,5E14.4)	IMAN0345
31 CONTINUE	IMAN0233	135 CONTINUE	IMAN0346
WRITE(6,204) J10	IMAN0234	IF(IPRINT.EQ.0) GO TO 67	IMAN0347
204 FORMAT(30H RF FILE AT START OF STEP NO.=I3)	IMAN0235	WRITE(6,11)	IMAN0348
WRITE(6,160)	IMAN0236	WRITE(6,219) J10	IMAN0349
WRITE(6,227)	IMAN0237	219 FORMAT(30H Q2 FILE FOR REVISED STEP NO.=I3)	IMAN0350
227 FORMAT(27H NODE DIRECTION REACTION)	IMAN0238	WRITE(6,160)	IMAN0351
WRITE(6,160)	IMAN0239	WRITE(6,172) (Q2(K10),K10=1,NDF)	IMAN0352
WRITE(6,206) (IZD(K10,1),IZD(K10,2),RF(K10),K10=1,NZD)	IMAN0240	67 CONTINUE	IMAN0353
206 FORMAT(1H 19,E16.5)	IMAN0241	DO 42 I=1,NDF	IMAN0354
IF(IPRINT.EQ.0) GO TO 97	IMAN0242	42 QIR(I)=QI(I)	IMAN0355
WRITE(6,11)	IMAN0243	DO 43 I=1,NDF	IMAN0356
WRITE(6,205) J10	IMAN0244	Q2(I)=Q2(I)	IMAN0357
205 FORMAT(31H Q2 FILE AT START FOR STEP NO.=I3)	IMAN0245	QI(I)=QI(I)+TI*Q2(I)/3.0	IMAN0358
WRITE(6,172) (Q2(K10),K10=1,NDF)	IMAN0246	QOI(I)=QOI(I)+TI**2*(Q2(I)/6.0)	IMAN0359
WRITE(6,160)	IMAN0247	43 QOI(I)=QOI(I)+TI*QI(I)	IMAN0360
97 CONTINUE	IMAN0248	IF(IPRINT.EQ.0) GO TO 68	IMAN0361
DO 98 I=1,NDF	IMAN0249	WRITE(6,11)	IMAN0362
Q2(I)=Q2(I)	IMAN0250	WRITE(6,211) J10	IMAN0363
QI(I)=QI(I)+TI*Q2(I)/6.0	IMAN0251	211 FORMAT(31H Q1 FILE REVISED FOR STEP NO.=I3)	IMAN0364
98 QOI(I)=QOI(I)+TI**2*(Q2(I)/6.0)	IMAN0252	WRITE(6,160)	IMAN0365
IF(IPRINT.EQ.0) GO TO 62	IMAN0253	WRITE(6,172) (Q1(K10),K10=1,NDF)	IMAN0366
WRITE(6,11)	IMAN0254	WRITE(6,11)	IMAN0367
WRITE(6,207) J10	IMAN0255	WRITE(6,212) J10	IMAN0368
207 FORMAT(32H Q1 FILE AT START FOR STEP NO.=I3)	IMAN0256	212 FORMAT(31H QOI FILE REVISED FOR STEP NO.=I3)	IMAN0369
WRITE(6,160)	IMAN0257	WRITE(6,160)	IMAN0370
WRITE(6,172) (Q1(K10),K10=1,NDF)	IMAN0258	WRITE(6,172) (QOI(K10),K10=1,NDF)	IMAN0371
WRITE(6,11)	IMAN0259	68 CONTINUE	IMAN0372
WRITE(6,208) J10	IMAN0260	C TO FIND ACCELERATION VALUES AT END OF TIME STEP	IMAN0373
208 FORMAT(32H QOI FILE AT START FOR STEP NO.=I3)	IMAN0261	IF(INDS.EQ.0) GO TO 44	IMAN0374
WRITE(6,160)	IMAN0262	44 IF(INDP.EQ.0) GO TO 45	IMAN0375
WRITE(6,172) (QOI(K10),K10=1,NDF)	IMAN0263	45 DO 46 I=1,NDF	IMAN0376
62 CONTINUE	IMAN0264	46 QI(I)=TI*(QI(I)+TI*Q2(I)/2.0)	IMAN0377
TO FIND ACCELERATION AT HALF STEP	IMAN0265	IF(IPRINT.EQ.0) GO TO 69	IMAN0378
IF(INDS.EQ.0) GO TO 35	IMAN0266	WRITE(6,11)	IMAN0379
35 IF(INDP.EQ.0) GO TO 36	IMAN0267	WRITE(6,220) J10	IMAN0380
36 DO 37 I=1,NDF	IMAN0268	220 FORMAT(28H Q1 FILE AT END OF STEP NO.=I3)	IMAN0381
37 QI(I)=TI*(QI(I)+TI**2*(Q2(I)/8.0)	IMAN0269	WRITE(6,160)	IMAN0382
IF(IPRINT.EQ.0) GO TO 63	IMAN0270	WRITE(6,172) (QI(K10),K10=1,NDF)	IMAN0383
WRITE(6,11)	IMAN0271	69 CONTINUE	IMAN0384
WRITE(6,214) J10	IMAN0272	CALL ACCDAMP,INODE,TI,QI,Q2,DP,NDF,1.0,M,IELCON,CNJ,Q,IEL,NNEL,	IMAN0385
214 FORMAT(44H Q1 FILE AT START OF HALF STEP FOR STEP NO.=I3)	IMAN0273	INEL,AK,AK1,V1,V2,V3,V4,QI,QI,QIR,P,S,CHASS,NJ,QOI,Q,RF,NZD,IZD,B,	IMAN0386
WRITE(6,160)	IMAN0274	2022,NET,NET1,B1,INLT,DM,PM,INT,NMAT,IPRINT)	IMAN0387
WRITE(6,172) (QI(K10),K10=1,NDF)	IMAN0275	WRITE(6,150)	IMAN0388
63 CONTINUE	IMAN0276	WRITE(6,221) J10	IMAN0389
CALL ACCDAMP,INODE,TI,QI,Q2,DP,NDF,2.0,M,IELCON,CNJ,Q,IEL,NNEL,	IMAN0277	221 FORMAT(28H RF FILE AT END OF STEP NO.=I3)	IMAN0390
INEL,AK,AK1,V1,V2,V3,V4,QI,QI,QIR,P,S,CHASS,NJ,QOI,Q,RF,NZD,IZD,B,	IMAN0278	WRITE(6,160)	IMAN0391
2022,NET,NET1,B1,INLT,DM,PM,INT,NMAT,IPRINT)	IMAN0279	WRITE(6,227)	IMAN0392
WRITE(6,160)	IMAN0280	WRITE(6,160)	IMAN0393
WRITE(6,215) J10	IMAN0281	WRITE(6,206) (IZD(K10,1),IZD(K10,2),RF(K10),K10=1,NZD)	IMAN0394
215 FORMAT(35H RF FILE AT HALF STEP FOR STEP NO.=I3)	IMAN0282	WRITE(6,11)	IMAN0395
WRITE(6,160)	IMAN0283	WRITE(6,228) J10	IMAN0396
WRITE(6,227)	IMAN0284	124 FORMAT(58H DIR FILE FOR CONFIGURATION Q(N) AT END OF (N+1) STEP NO=I3)	IMAN0397
WRITE(6,160)	IMAN0285	WRITE(6,150)	IMAN0398
WRITE(6,206) (IZD(K10,1),IZD(K10,2),RF(K10),K10=1,NZD)	IMAN0286	WRITE(6,125)	IMAN0399
IF(IPRINT.EQ.0) GO TO 64	IMAN0287	125 FORMAT(116H FLEM. NO. LXX LYZ LXZ LZY LZZ)	IMAN0400
WRITE(6,11)	IMAN0288	LX LYY LYZ LXZ LZY LZZ	IMAN0401
WRITE(6,216) J10	IMAN0289	WRITE(6,160)	IMAN0402
216 FORMAT(35H Q2 FILE AT HALF STEP FOR STEP NO.=I3)	IMAN0290	DO 138 I=1,NEL	IMAN0403
WRITE(6,160)	IMAN0291	CALL STRESS(I,NMAT,NDF,NEL,PM,D33,DIR,INT,INT,I,T,QT,Q,N,NEL)	IMAN0404
WRITE(6,172) (Q2(K10),K10=1,NDF)	IMAN0292	ICI=CI3	IMAN0405
64 CONTINUE	IMAN0293	WRITE(6,126) I,CI(DIR(I2,J2),J2=1,3),I2=1,3)	IMAN0406
DO 39 I=1,NDF	IMAN0294	126 FORMAT(1H 16,F15.5,8F12.5)	IMAN0407
Q2(I)=Q2(I)	IMAN0295	I1=CI(I1,1)	IMAN0408
QI(I)=QI(I)+TI*Q2(I)/3.0	IMAN0296	DO 133 I2=1,3	IMAN0409
38 QOI(I)=QOI(I)+TI**2*(Q2(I)/6.0)	IMAN0297	ST(I1,I2)=ST(I1,I2)+CI(I2)	IMAN0410
IF(IPRINT.EQ.0) GO TO 65	IMAN0298	ST(I1,I2)=ST(I1,I2)+CI(I2)	IMAN0411
WRITE(6,11)	IMAN0299	133 I1=I1+1	IMAN0412
WRITE(6,209) J10	IMAN0300	138 CONTINUE	IMAN0413
209 FORMAT(43H Q1 FILE AT END OF HALF STEP FOR STEP NO.=I3)	IMAN0301	WRITE(6,11)	IMAN0414
WRITE(6,160)	IMAN0302	WRITE(6,137) J10	IMAN0415
WRITE(6,172) (Q1(K10),K10=1,NDF)	IMAN0303	137 FORMAT(23H STRESSES AT END OF STEP NO.=I3)	IMAN0416
WRITE(6,11)	IMAN0304	WRITE(6,160)	IMAN0417
WRITE(6,210) J10	IMAN0305	WRITE(6,131)	IMAN0418
210 FORMAT(43H QOI FILE AT END OF HALF STEP FOR STEP NO.=I3)	IMAN0306	WRITE(6,160)	IMAN0419
WRITE(6,160)	IMAN0307	DO 134 I=1,NEL	IMAN0420
WRITE(6,172) (QOI(K10),K10=1,NDF)	IMAN0308	I1=CI(I1,1)+1	IMAN0421
65 CONTINUE	IMAN0309	I2=I1+5	IMAN0422
TO FIND REVISED ACCELERATION VALUES AT HALF STEP	IMAN0310	WRITE(6,132) I,ST(K10),K10=1,12)	IMAN0423
IF(INDS.EQ.0) GO TO 39	IMAN0311	134 CONTINUE	IMAN0424
39 IF(INDP.EQ.0) GO TO 40	IMAN0312	CALL PREST(J10,S,NDF,FEL,NEL24,B1,NETI,Q,QOI,NNEL,NEL,IELCON,NJ,M,	IMAN0425
40 DO 41 I=1,NDF	IMAN0313	LAKI,INT)	IMAN0426
41 QI(I)=TI*(QI(I)+2.0*TI*Q2(I)/8.0)	IMAN0314	WRITE(6,11)	IMAN0427
IF(IPRINT.EQ.0) GO TO 66	IMAN0315	WRITE(6,141) J10	IMAN0428
WRITE(6,11)	IMAN0316	141 FORMAT(42H NODAL FORCES (S FILE) AT END OF STEP NO.=I3)	IMAN0429
WRITE(6,217) J10	IMAN0317	WRITE(6,160)	IMAN0430
217 FORMAT(30H Q1 FILE REVISED FOR STEP NO.=I3)	IMAN0318	WRITE(6,228)	IMAN0431
WRITE(6,160)	IMAN0319	228 FORMAT(95H NODE ALONG X ALONG Y ALONG Z	IMAN0432
WRITE(6,172) (Q1(K10),K10=1,NDF)	IMAN0320	1 ABOUT X ABOUT Y ABOUT Z)	IMAN0433
66 CONTINUE	IMAN0321	WRITE(6,160)	IMAN0434
CALL ACCDAMP,INODE,TI,QI,Q2,DP,NDF,2.0,M,IELCON,CNJ,Q,IEL,NNEL,	IMAN0322	DO 230 I=1,NJ	IMAN0435
INEL,AK,AK1,V1,V2,V3,V4,QI,QI,QIR,P,S,CHASS,NJ,QOI,Q,RF,NZD,IZD,B,	IMAN0323	I1=I1+5	IMAN0436
2022,NET,NET1,B1,INLT,DM,PM,INT,NMAT,IPRINT)	IMAN0324	I2=I1+5	IMAN0437
WRITE(6,160)	IMAN0325	230 WRITE(6,229) I,(S(K10),K10=I1,I2)	IMAN0438
WRITE(6,218) J10	IMAN0326	229 FORMAT(1H 14,E15.6)	IMAN0439
218 FORMAT(30H RF FILE REVISED FOR STEP NO.=I3)	IMAN0327	IF(IPRINT.EQ.0) GO TO 53	IMAN0440
WRITE(6,160)	IMAN0328	WRITE(6,11)	IMAN0441
WRITE(6,227)	IMAN0329	WRITE(6,142) J10	IMAN0442
WRITE(6,160)	IMAN0330	142 FORMAT(46H ELEMENT NODAL FORCES FILE AT END OF STEP NO.=I3)	IMAN0443
WRITE(6,206) (IZD(K10,1),IZD(K10,2),RF(K10),K10=1,NZD)	IMAN0331	WRITE(6,160)	IMAN0444
WRITE(6,11)	IMAN0332	WRITE(6,172) (FEL(K10),K10=1,NEL24)	IMAN0445
WRITE(6,130) J10	IMAN0333	53 CONTINUE	IMAN0446
130 FORMAT(50H STRESSES FOR REVISED HALF STEP FOR TIME STEP NO.=I3)	IMAN0334	DO 47 I=1,NDF	IMAN0447
WRITE(6,160)	IMAN0335	47 QI(I)=QI(I)+TI*Q2(I)/6.0	IMAN0448
WRITE(6,131)	IMAN0336	IF(IPRINT.EQ.0) GO TO 70	IMAN0449
131 FORMAT(92H ELEMENT NO. SXT SYT SXYT	IMAN0337	WRITE(6,11)	IMAN0450
J SXR SYR SXYR)	IMAN0338	WRITE(6,213) J10	IMAN0451
	IMAN0339		IMAN0452

Table B-1 (contd)

213	FORMAT(29H Q11 FILE AT END OF STEP NO.=I3)	IMAND453	180	FORMAT(49H NODAL VELOCITY (Q1 FILE) AT END OF STEP NO.=I3)	IMAND474
	WRITE(6,160)	IMAND454		WRITE(6,160)	IMAND475
	WRITE(6,172) (Q11(K10),K10=1,NDF)	IMAND455		WRITE(6,228)	IMAND476
70	CONTINUE	IMAND456		WRITE(6,160)	IMAND477
C	TO PREPARE BASIC DATAS TO FIND REVISED ACCELERATION VALUES	IMAND457		DO 232 I=1,NJ	IMAND478
C	S VALUES ALREADY UPDATED	IMAND458		I1=I+6-5	IMAND479
	DO 48 J=1,NDF	IMAND459		I2=I1+5	IMAND480
	Q(J)=Q(J)+Q01(J)	IMAND460	232	WRITE(6,229) I,(Q1(K10),K10=I1,I2)	IMAND481
48	Q1(J)=Q1(J)+Q11(J)	IMAND461		WRITE(6,1)	IMAND482
	WRITE(6,1)	IMAND462		WRITE(6,181) J10	IMAND483
	WRITE(6,179) J10	IMAND463	181	FORMAT(49H NODAL ACCELERATION (Q2 FILE) AT END OF STEP NO.=I3)	IMAND484
179	FORMAT(47H NODAL DEFLECTIONS (Q FILE) AT END OF STEP NO.=I3)	IMAND464		WRITE(6,160)	IMAND485
	WRITE(6,160)	IMAND465		WRITE(6,228)	IMAND486
	WRITE(6,172)	IMAND466		WRITE(6,160)	IMAND487
	WRITE(6,160)	IMAND467		DO 233 I=1,NJ	IMAND488
	DO 231 I=1,NJ	IMAND468		I1=I+6-5	IMAND489
	I1=I+6-5	IMAND469		I2=I1+5	IMAND490
	I2=I1+5	IMAND470	233	WRITE(6,229) I,(Q2(K10),K10=I1,I2)	IMAND491
231	WRITE(6,229) I,(Q(K10),K10=I1,I2)	IMAND471	100	CONTINUE	IMAND492
	WRITE(6,1)	IMAND472		STOP	IMAND493
	WRITE(6,180) J10	IMAND473		END	IMAND494

Table B-2. Source program listing of subroutine ACC

[illegible]

Table B-3. Source program listing of subroutine ADM

	SUBROUTINE ADD(S,I0S,A*,M,JB,C)	SADM0001
C	ADD SUBMATRICES TO FORM STIFFNESS MATRIX	SADM0002
	DIMENSION A(4*4),S(11)	SADM0003
	DO 10 I=1,M	SADM0004
	IS=IB+I-1	SADM0005
	IF (IB-JB) 12,11,12	SADM0006
11	J1=I	SADM0007
	GO TO 13	SADM0008
12	J1=1	SADM0009
13	DO 90 J=J1,M	SADM0010
	JS=JB+J-1	SADM0011
	ISS=(JS-1)*I0S+IS	SADM0012
	S(ISS)=S(ISS)+A(I,J)*C	SADM0013
	ISR=(IS-1)*I0S+JS	SADM0014
	S(ISR)=S(ISR)	SADM0015
90	CONTINUE	SADM0016
10	CONTINUE	SADM0017
	RETURN	SADM0018
	END	SADM0019

Table B-4. Source program listing of subroutine ASSEMB

	SUBROUTINE ASSEMB (A,C,N1,N2,N3,N4,N5)	\$ASS0001
C	TO ASSEMBLE TRIANGULAR ELEMENT STIFFNESS MATRIX	\$ASS0002
C	TO FORM QUADRILATERAL STIFFNESS MATRIX IN LOCAL COORDINATE SYSTEM	\$ASS0003
	DIMENSION A(25,25),C(15,15)	\$ASS0004
	J1=1	\$ASS0005
	DO 1 K=N1,25,5	\$ASS0006
	IND=N4	\$ASS0007
	L=K	\$ASS0008
	DO 2 M=1,3	\$ASS0009
	J=N1	\$ASS0010
	DO 3 I=1,15,3	\$ASS0011
	A(I,J)=A(I,J)+C(J,I,I)	\$ASS0012
	I11=I+1	\$ASS0013
	J12=J+N2	\$ASS0014
	A(I,J12)=A(I,J12)+C(J1,I11)	\$ASS0015
	I12=I+2	\$ASS0016
	J13=J+N3	\$ASS0017
	A(I,J13)=A(I,J13)+C(J1,I12)	\$ASS0018
3	J=J+5	\$ASS0019
	J1=J1+1	\$ASS0020
	L=(L+1)+IND	\$ASS0021
2	IND=IND+N5	\$ASS0022
1	CONTINUE	\$ASS0023
	RETURN	\$ASS0024
	END	\$ASS0025

Table B-5. Source program listing of subroutine CONOJ

```
C SUBROUTINE CONOJ (CNJ,Q,INODE,NOF) SC0NN001
   TO COPY Q TO FORM COORDINATES OF NODES CONNECTING A JOINT SC0NN002
   DIMENSION CNJ(15,3),Q(INOF),INODE(15) SC0NN003
   DO 3 I=1,15 SC0NN004
     DO 3 J=1,3 SC0NN005
       1 CNJ(I,J)=Q SC0NN006
       DO 1 I=1,15 SC0NN007
         J=INODE(I) SC0NN008
         IF(J.EQ.Q) GO TO 1 SC0NN009
         J=6-(J-1) SC0NN010
         DO 2 II=1,3 SC0NN011
           JII=J-II SC0NN012
           2 CNJ(II,I)=Q(JII) SC0NN013
       1 CONTINUE SC0NN014
     RETURN SC0NN015
   END SC0NN016
```

Table B-6. Source program listing of subroutine COORD

```

SUBROUTINE COORD(A,N)                                SC000001
C TO GENERATE NODAL COORDINATES IN INCREASING ORDER OF JOINT NUMBERS SC000002
DIMENSION A(N)                                       SC000003
N1=N/6                                              SC000004
DO 1 I=1,N                                          SC000005
1 A(I)=0.0                                          SC000006
DO 2 I=1,N1                                         SC000007
J1=(I-1)*6+1                                       SC000008
J2=J1+2                                             SC000009
READ(5,3) (A(K),K=J1,J2)                          SC000010
3 FORMAT(3F9.4)                                     SC000011
2 CONTINUE                                         SC000012
RETURN                                             SC000013
END                                                 SC000014

```

Table B-9. Source program listing of subroutine DFORCE

```

SUBROUTINE DFORCE(DAMP,INODE,JI,Q1,Q2,DP,XDIV,NDF,II) SDF00001
C TO FORM DAMPING FORCES                          SDF00002
DIMENSION DAMP(15),INODE(15),Q1(NDF),Q2(NDF),DP(6) SDF00003
DO 1 I=1,6                                          SDF00004
CD=0.0                                              SDF00005
J2=JI-1                                           SDF00006
DO 2 I2=1,15                                       SDF00007
IF(INODE(I2),EQ,0) GO TO 2                        SDF00008
J1=6*(INODE(I2)-1)+1                               SDF00009
CD=CD+DAMP(I2)*(Q1(J1)-Q1(J11))+JI*(Q2(J1)-Q2(J11))/XDIV SDF00010
2 CONTINUE                                         SDF00011
1 DP(I1)=CD                                         SDF00012
RETURN                                             SDF00013
END                                                 SDF00014

```

Table B-7. Source program listing of subroutine CORT

```

SUBROUTINE CORT(IK,XD,YD,ZD,X,Y,Z)                  SCORT001
C OBTAINS COORDINATES OF TRIANGULAR SHELL ELEMENT IN LOCAL COORDINATES SCORT002
DIMENSION XD(2),YD(2),ZD(2),X(3),Y(3),Z(3),DUMX(3),DUNX(3),DIR(3,3) SCORT003
1 IF(IK-1) 15,10,15                                SCORT004
10 EL=SGR(XD(1),XD(1)+YD(1)+YD(1),ZD(1)+ZD(1)) SCORT005
DIR(1,1)=XD(1)/EL                                  SCORT007
DIR(1,2)=YD(1)/EL                                  SCORT008
DIR(1,3)=ZD(1)/EL                                  SCORT009
DIR(3,1)=DIR(1,2)+ZD(2)-DIR(1,3)+YD(2)            SCORT010
DIR(3,2)=DIR(1,3)+XD(2)-DIR(1,1)+ZD(2)            SCORT011
DIR(3,3)=DIR(1,1)+YD(2)-DIR(1,2)+XD(2)            SCORT012
EL=SGR(DIR(3,1)+DIR(3,1)+DIR(3,1)+DIR(3,2)+DIR(3,2)+DIR(3,3)) SCORT013
DIR(3,1)=DIR(3,1)/EL                               SCORT015
DIR(3,2)=DIR(3,2)/EL                               SCORT016
DIR(3,3)=DIR(3,3)/EL                               SCORT017
DIR(2,1)=DIR(3,2)+DIR(1,3)-DIR(3,3)+DIR(1,2)     SCORT018
DIR(2,2)=DIR(3,3)+DIR(1,1)-DIR(3,1)+DIR(1,3)     SCORT019
DIR(2,3)=DIR(3,1)+DIR(1,2)-DIR(3,2)+DIR(1,1)     SCORT020
15 DO 1 I=1,3                                       SCORT021
DUMX(I)=X(I)                                       SCORT022
DUNX(I)=Y(I)                                       SCORT023
DUMX(3)=Z(I)                                       SCORT024
DO 2 J=1,3                                         SCORT025
DUNX(J)=0.                                         SCORT026
DO 3 K=1,3                                         SCORT027
DUNX(K)=DUMX(J)+DIR(J,K)+DUMX(K)                 SCORT028
3 CONTINUE                                         SCORT029
X(I)=DUMX(I)                                       SCORT030
Y(I)=DUNX(I)                                       SCORT031
Z(I)=DUNX(I)                                       SCORT032
DO 4 I=1,3                                         SCORT033
XD(I-1)=X(I)-X(I)                                 SCORT034
YD(I-1)=Y(I)-Y(I)                                 SCORT035
ZD(I-1)=Z(I)-Z(I)                                 SCORT036
4 CONTINUE                                         SCORT037
END

```

Table B-10. Source program listing of subroutine DIEL

```

SUBROUTINE DIEL(I,M,IELCON,NJ,IEL,NNEL,NEL,INODE,DAMP,DH,PM,INT, S0IE0001
INMAT)                                              S0IE0002
DIMENSION M(8),IELCON(NJ+4),IEL(8+4),NNEL(NEL+4),INODE(15),DAMP(15) S0IE0003
1 IDMH(8),PM(INMAT),INT(INMAT)                   S0IE0004
C TO FORM INODE AND DAMP ARRAYS FOR A JOINT       S0IE0005
CALL MELCON(M,IELCON,NJ)                          S0IE0006
CALL NIEL(IEL,M,NNEL,NEL)                          S0IE0007
DO 15 I1=1,15                                       S0IE0008
15 INODE(I1)=0                                       S0IE0009
DO 16 I2=1,4                                         S0IE0010
16 INODE(I2)=IEL(I1,I2)                             S0IE0011
16=16+1                                             S0IE0012
DO 17 I2=2,8                                         S0IE0013
IF(M(I2),EQ,0) GO TO 17                            S0IE0014
DO 18 I3=1,4                                         S0IE0015
DO 19 I4=1,15                                       S0IE0016
IF(IEL(I2,I3),EQ,INODE(I4)) GO TO 18              S0IE0017
19 CONTINUE                                         S0IE0018
INODE(I6)=IEL(I2,I3)                               S0IE0019
16=16+1                                             S0IE0020
18 CONTINUE                                         S0IE0021
17 CONTINUE                                         S0IE0022
C TO FORM DAMPED VELOCITY TERM                    S0IE0023
DO 24 I1=1,8                                         S0IE0024
24 DM(I1)=0.0                                       S0IE0025
DO 25 I1=1,8                                         S0IE0026
KI=M(I1)                                           S0IE0027
IF(KI,EQ,0) GO TO 25                               S0IE0028
K2=INT(KI)                                          S0IE0029
DM(I1)=PM(K2,3)                                    S0IE0030
25 CONTINUE                                         S0IE0031
DO 26 I1=1,15                                       S0IE0032
26 DAMP(I1)=0.0                                       S0IE0033
ICOUNT=0                                           S0IE0034
DO 27 I1=1,15                                       S0IE0035
IF(INODE(I1),EQ,0,OR,INODE(I1),EQ,1) GO TO 27     S0IE0036
DO 28 I2=1,8                                         S0IE0037
DO 29 I3=1,4                                         S0IE0038
IF(INODE(I1),EQ,IEL(I2,I3)) GO TO 30              S0IE0039
29 CONTINUE                                         S0IE0040
GO TO 28                                           S0IE0041
30 ICOUNT=ICOUNT+1                                   S0IE0042
DAMP(I1)=DAMP(I1)+DM(I2)                           S0IE0043
28 CONTINUE                                         S0IE0044
CCOUNT=ICOUNT                                       S0IE0045
27 ICOUNT=0                                           S0IE0046
RETURN                                             S0IE0047
END                                                 S0IE0048

```

Table B-8. Source program listing of subroutine DCS

```

SUBROUTINE DCS(I,NNEL,NEL,Q,QOI,NDF,DIR,INDEX)      DCS00001
C TO COMPUTE DIRECTION COSINE OF ELEMENT NUMBER I DCS00002
C FOR INDEX=1, DIR IS BASED ON Q(N+1)=Q(N)+QOI(QIN) AND QOI DCS00003
C BEING RETAINED IN ORIGINAL FORM                 DCS00004
C FOR INDEX=0, DIR IS BASED ON Q(N) ONLY           DCS00005
DIMENSION NNEL(NEL+4),Q(NDF),QOI(NDF),DIR(3,3), DCS00006
1 V1(3),V2(3),V3(3)                                DCS00007
J1=6*(NNEL(I1)-1)+1                                DCS00008
J2=6*(NNEL(I2)-1)+1                                DCS00009
J3=6*(NNEL(I3)-1)+1                                DCS00010
DO 1 I1=1,3                                         DCS00011
V1(I1)=Q(J1)-Q(J11)                                DCS00012
V2(I1)=Q(J2)-Q(J11)                                DCS00013
IF(INDEX,EQ,0) GO TO 2                              DCS00014
V1(I1)=V1(I1)+QOI(J2)-QOI(J1)                      DCS00015
V2(I1)=V2(I1)+QOI(J2)-QOI(J1)                      DCS00016
2 J1=J1+1                                           DCS00017
J2=J2+1                                           DCS00018
J3=J3+1                                           DCS00019
CALL VECT(V1,V2,V3)                                DCS00020
CALL UNIT(V3)                                       DCS00021
CALL UNIT(V1)                                       DCS00022
CALL VECT(V3,V1,V2)                                DCS00023
DO 3 I2=1,3                                         DCS00024
DIR(1,I2)=V1(I2)                                   DCS00025
DIR(2,I2)=V2(I2)                                   DCS00026
3 DIR(3,I2)=V3(I2)                                  DCS00027
RETURN                                             DCS00028
END                                                 DCS00029

```

Table B-11. Source program listing of subroutine DIRCOS

```

SUBROUTINE DIRCOS(V1,V2,INODE,CNJ)                 SDIR0001
C TO FIND DIRECTION COSINES OF AN ELEMENT          SDIR0002
DIMENSION V1(3),V2(3),INODE(15),CNJ(15,3)         SDIR0003
DO 1 I1=1,3                                         SDIR0004
DO 2 I2=1,15                                       SDIR0005
J1=V1(I1)                                           SDIR0006
IF(INODE(I2),EQ,INODE(I2)) GO TO 3                SDIR0007
3 V2(I1)=I2                                         SDIR0008
1 CONTINUE                                         SDIR0009
J1=V2(I1)                                           SDIR0010
J2=V2(I2)                                           SDIR0011
J3=V2(I3)                                           SDIR0012
DO 4 I1=1,3                                         SDIR0013
V1(I1)=CNJ(J2,I1)+CNJ(J1,I1)                      SDIR0014
4 V2(I1)=CNJ(J3,I1)+CNJ(J1,I1)                    SDIR0015
RETURN                                             SDIR0016
END                                                 SDIR0017

```

Table B-12. Source program listing of subroutine ELEM T

```

SUBROUTINE ELEM T(I A,B,IC,INLT,NLT,TNLT,J,NET,J1)
C TO GENERATE ELEMENT NODAL NUMBER ARRAY AND ELEMENT THICKNESS
C ALSO TO STORE ELEMENT MATERIAL TYPE INFORMATIONS
C DIMENSION I A(1,2,3,4,5),B(1,2,3,4,5),IC(1,2,3,4,5),INLT(1,2,3,4,5),TNLT(1,2,3,4,5)
READ(5,1) ((I A(1,2,3,4,5),B(1,2,3,4,5),IC(1,2,3,4,5),INLT(1,2,3,4,5),TNLT(1,2,3,4,5))
1 FORMAT(4I5,F8.3,2I4)
READ(5,2) ((INLT(1,2,3,4,5),B(1,2,3,4,5),TNLT(1,2,3,4,5),NET)
2 FORMAT(5I5,F8.4)
RETURN
END

```

Table B-13. Source program listing of subroutine ELIM

```

SUBROUTINE ELIM(I A,B,I11)
C TO FORM EQUIVALENT QUADRILATERAL ELEMENT STIFFNESS MATRIX BY
C ELIMINATING CENTRAL POINT
C DIMENSION A(25,25),B(20,20),AK12(20,5),AK21(5,20),AK22(5,5)
I1=1
DO 1 I=1,5
DO 2 J=1,5
J1=1
N2=5*(I-1)+J
IF(J,5,5) GO TO 6
DO 3 L=1,5
N1=5*(J-1)+L
DO 4 K=1,5
L2=L-1
N12=N1+L2
B(I1,J1)=A(N2,N12)
4 J1=J+1
AK12(I1,J1)=A(N2,N1+9)
3 CONTINUE
IF(J,NE,5) GO TO 7
DO 5 L3=1,5
5 AK22(I1,L3)=A(N2,5+L3)
GO TO 2
7 I1=I+1
2 CONTINUE
1 CONTINUE
CALL TRANSP(AK12,AK21,20,5)
CALL INV(AK22,5,AK21,20,DETERM)
DO 8 I=1,20
DO 9 J=1,20
V=0.0
DO 10 K=1,5
V=V+AK12(I,K)*AK21(K,J)
B(I1,J)=B(I1,J)+V
9 CONTINUE
8 CONTINUE
RETURN
END

```

Table B-14. Source program listing of subroutine INV

```

SUBROUTINE INV(I A,N,B,M,DETERM)
C INVERTS MATRICES UP TO ORDER 8 BY GAUSS ELIMINATION
C DIMENSION IPIVOT(5),A(5,5),B(5,20),INDEX(5,2),PIVOT(5)
C EQUIVALENCE (IROW,JROW), (ICOLU M,JCOLU M), (AMAX,T,SWAP)
C INITIALIZATION
C 10 DETERM=1.0
15 DO 20 J=1,N
20 IPIVOT(J)=0
88=1.F-30
GO TO 9003
9002 DETERM=0.
GO TO 740
9003 CONTINUE
30 DO 550 I=1,N
C SEARCH FOR PIVOT ELEMENT
C 40 AMAX=0.0
45 DO 105 J=1,N
50 IF (IPIVOT(J)-1) 60, 105, 60
50 DO 100 K=1,N
60 IF (IPIVOT(K)-1) 80, 100, 9002
80 IF (ABS (AMAX)-ABS (A(J,K))) 85, 100, 100
85 IROW=J
90 ICOLU M=K
95 AMAX=A(J,K)
100 CONTINUE
105 CONTINUE
IF (ABS (AMAX)-88) 9002,9002,110
110 IPIVOT(ICOLU M)=IPIVOT(ICOLU M)+1
C INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C 130 IF (IROW-ICOLU M) 140, 260, 140
140 DETERM=-DETERM
150 DO 200 L=1,N
160 SWAP=A(IROW,L)
170 A(IROW,L)=A(ICOLU M,L)
200 A(ICOLU M,L)=SWAP
205 IF(M) 260, 260, 210
210 DO 250 L=1,M
220 SWAP=B(IROW,L)
230 B(IROW,L)=B(ICOLU M,L)
250 B(ICOLU M,L)=SWAP
260 INDEX(I,1)=IROW
270 INDEX(I,2)=ICOLU M
310 PIVOT(I)=A(ICOLU M,ICOLU M)
320 DETERM=DETERM*PIVOT(I)
C DIVIDE PIVOT ROW BY PIVOT ELEMENT
C 330 A(ICOLU M,ICOLU M)=1.0
340 DO 350 I=1,N
350 A(ICOLU M,L)=A(ICOLU M,L)/PIVOT(I)
355 IF(M) 360, 360, 360
360 DO 370 L=1,M
370 B(ICOLU M,L)=B(ICOLU M,L)/PIVOT(I)
C REDUCE NON-PIVOT ROWS
C 380 DO 550 L=1,N
390 IF(L1-ICOLU M) 400, 550, 400
400 T=A(L1,ICOLU M)
420 A(L1,ICOLU M)=0.0
430 DO 450 L=1,N
450 A(L1,L)=A(L1,L)-A(ICOLU M,L)*T
455 IF(M) 550, 550, 460
460 DO 500 L=1,M
500 B(L1,L)=B(L1,L)-B(ICOLU M,L)*T
550 CONTINUE
C INTERCHANGE COLUMNS
C 600 DO 710 I=1,N
610 L=N+1-I
620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
630 JROW=INDEX(L,1)
640 JCOLUM=INDEX(L,2)
650 DO 705 K=1,N
660 SWAP=A(IROW,K)
670 A(IROW,K)=A(JROW,K)
700 A(K,JCOLUM)=SWAP
705 CONTINUE
710 CONTINUE
740 RETURN
END

```

Table B-15. Source program listing of subroutine MELCON

```

SUBROUTINE MELCON(M,IELCON,I,NJ)
C TO FORM ELEMENT NUMBERS CONNECTING A JOINT
DIMENSION M(8),IELCON(4)
I1=1
DO 2 I2=1,4
M(I1)=IELCON(I+I2)/1000
M(I1+1)=IELCON(I+I2)-M(I1)/1000
2 I1=I1+2
RETURN
END
SMEL0001
SMEL0002
SMEL0003
SMEL0004
SMEL0005
SMEL0006
SMEL0007
SMEL0008
SMEL0009
SMEL0010

```

Table B-16. Source program listing of subroutine MMULT

```

SUBROUTINE MMULT(A,B,C,J1,J2,J3)
C TO PERFORM MATRIX MULTIPLICATION
DIMENSION A(J1,J2),B(J2,J3),C(J1,J3)
DO 1 I=1,J1
DO 1 J=1,J3
1 C(I,J)=0.0
DO 2 I1=1,J1
DO 3 J1=1,J3
DO 4 K=1,J2
4 C(I,J)=C(I,J)+A(I,K)*B(K,J)
3 CONTINUE
2 CONTINUE
RETURN
END
SMUL0001
SMUL0002
SMUL0003
SMUL0004
SMUL0005
SMUL0006
SMUL0007
SMUL0008
SMUL0009
SMUL0010
SMUL0011
SMUL0012
SMUL0013
SMUL0014

```

Table B-19. Source program listing of subroutine PREST

```

SUBROUTINE PREST(IJID,S,NDF,FEL,NEL24,B1,NET1,0,
1901,NNEL,NEL,IELCON,NJ,M,AK1,INLT)
C TO INCORPORATE PRESTRESS EFFECT
C TO COMPUTE UPDATED NODAL FORCES FOR JOINTS IN LCS
C TO COMPUTE UPDATED ELEMENT NODAL FORCES IN LCS
DIMENSION M(8),AK1(24,24),S(NDF),Q(NDF),QI(NDF),
IFEL(NEL24),B1(NET1),NNEL(NEL,4),IELCON(NJ,4),
2DIR(3,3),DIR1(3,3),SEL(24),SEL1(24),S1(6),SS(6),
3C1(3),C2(3),C3(3),C4(3),C13(4),A1(3,3),A2(3,3),A3(12),
4INLT(NEL)
IF(IJID.EQ.1) GO TO 23
C TO COMPUTE NEWLY ORIENTED S,DS=K,QQT BEING ALREADY COMPUTED IN ACCPREST012
DO 1 I=1,NJ
CALL MELCON (M,IELCON,I,NJ)
16=6*(I-1)
I1=0
DO 2 I2=1,8
IF(M(I2).EQ.0) GO TO 2
I1=I1+1
2 CONTINUE
DO 3 I2=1,I1
DO 4 I3=1,24
4 SEL(I3)=0.0
J=M(I2)
I4=24*(J-1)+1
DO 5 I5=1,24
SEL(I5)=FEL(I4)
5 I4=I4+1
CALL DCS(J,NNEL,NEL,0,QQI,NDF,DIR,1)
DO 16 I3=1,3
DO 16 J3=1,3
16 DIR1(I3,J3)=DIR(J3,I3)
DO 17 I3=1,4
IF(I1.EQ.0) GO TO 18
17 CONTINUE
18 I3=6*(I3-1)
DO 19 I4=1,6
19 SS(I4)=0.0
DO 20 I4=1,6
I34=I3+I4
20 S1(I4)=SEL(I34)
DO 21 I4=1,3
DO 21 J4=1,3
SS(I4)=SS(I4)+DIR1(I4,J4)*S1(I4)
21 SS(I4+3)=SS(I4+3)+DIR1(I4,J4)*S1(I4+3)
DO 22 I4=1,6
164=16+I4
22 S1(164)=S1(164)+SS(I4)
3 CONTINUE
1 CONTINUE
23 CONTINUE
C TO UPDATE NODAL FORCES IN L.C.S.
DO 16 I1=1,NEI
C TO OBTAIN ELEMENT DISPLACEMENT VECTOR IN L.C.S.
I1=1
CALL DCS(I1,NNEL,NEL,0,QQI,NDF,DIR,0)
DO 30 J1=1,4
J1=NNEL(I1)
J11=6*(J1-1)+1
DO 35 J2=1,3
J15=J
GO TO (31,32,33,34),J15
31 C1(J2)=S(J11)
PREST0001
PREST0002
PREST0003
PREST0004
PREST0005
PREST0006
PREST0007
PREST0008
PREST0009
PREST0010
PREST0011
PREST0012
PREST0013
PREST0014
PREST0015
PREST0016
PREST0017
PREST0018
PREST0019
PREST0020
PREST0021
PREST0022
PREST0023
PREST0024
PREST0025
PREST0026
PREST0027
PREST0028
PREST0029
PREST0030
PREST0031
PREST0032
PREST0033
PREST0034
PREST0035
PREST0036
PREST0037
PREST0038
PREST0039
PREST0040
PREST0041
PREST0042
PREST0043
PREST0044
PREST0045
PREST0046
PREST0047
PREST0048
PREST0049
PREST0050
PREST0051
PREST0052
PREST0053
PREST0054
PREST0055
PREST0056
PREST0057
PREST0058
PREST0059
PREST0060
PREST0061
PREST0062
PREST0063

```

Table B-17. Source program listing of subroutine NIEL

```

SUBROUTINE NIEL(IEL,M,NNEL,NEL)
C TO FORM ELEMENT NODAL DESCRIPTION CONNECTING A JOINT
DIMENSION IEL(8,4),M(8),NNEL(NEL,4)
DO 1 I1=1,8
DO 1 I2=1,4
1 IEL(I1,I2)=0.0
DO 2 I1=1,8
IF(M(I1).EQ.0) GO TO 2
K1=M(I1)
DO 3 I2=1,4
3 IEL(I1,I2)=NNEL(K1,I2)
2 CONTINUE
RETURN
END
SNIE0001
SNIE0002
SNIE0003
SNIE0004
SNIE0005
SNIE0006
SNIE0007
SNIE0008
SNIE0009
SNIE0010
SNIE0011
SNIE0012
SNIE0013
SNIE0014

```

Table B-18. Source program listing of subroutine NQII

```

SUBROUTINE NQII(QII,INODE,01,NDF)
C TO COPY QI AND FORM INCREMENTAL DEFORMATION FILE AT A JOINT
DIMENSION QII(90),INODE(15),QI(NDF)
DO 3 I1=1,90
3 QI(I1)=0.0
DO 1 I1=1,15
J1=6*(I1-1)
J=INODE(I1)
IF(J.EQ.0) GO TO 1
J5=J+1
DO 2 I2=1,6
J2=J+I2
J12=J+I2
2 QI(I12)=QI(J2)
1 CONTINUE
RETURN
END
SNQI0001
SNQI0002
SNQI0003
SNQI0004
SNQI0005
SNQI0006
SNQI0007
SNQI0008
SNQI0009
SNQI0010
SNQI0011
SNQI0012
SNQI0013
SNQI0014
SNQI0015
SNQI0016
SNQI0017

```

Table B-20. Source program listing of subroutine PROMAT

	SUBROUTINE PROMAT(A,N,N1)	SPR00001
C	TO GENERATE MATERIAL PROPERTIES MATRIX	SPR00002
	DIMENSION A(N,N1)	SPR00003
	READ(5,1) ((A(I,J),J=1,N1),I=1,N)	SPR00004
	1 FORMAT(10.2,F6.2,F8.3)	SPR00005
	RETURN	SPR00006
	END	SPR00007

Table B-21. Source program listing of subroutine QCA

	SUBROUTINE QCA(I1,I2,I3,I4,C1,C2,C3,C4,C5,C6,C7,AR1,AR2)	QCA00001
C	TO GENERATE COORDINATE OF CENTRE OF AREA OF QUADRILATERAL ELEMENT	QCA00002
	IN GLOBAL SYSTEM	QCA00003
	DIMENSION C1(3),C2(3),C3(3),C4(3),C5(3),C6(3),C7(3)	QCA00004
	X21=C2(1)-C1(1)	QCA00005
	X41=C4(1)-C1(1)	QCA00006
	Y21=C2(2)-C1(2)	QCA00007
	Y41=C4(2)-C1(2)	QCA00008
	Z21=C2(3)-C1(3)	QCA00009
	Z41=C4(3)-C1(3)	QCA00010
	X32=C3(1)-C2(1)	QCA00011
	X42=C4(1)-C2(1)	QCA00012
	Y32=C3(2)-C2(2)	QCA00013
	Y42=C4(2)-C2(2)	QCA00014
	Z32=C3(3)-C2(3)	QCA00015
	Z42=C4(3)-C2(3)	QCA00016
	AR1=(Y21*Z41-Y41*Z21)**2+(X41*Z21-X21*Z41)**2+(X21*Y41-X41*Y21)**2	QCA00017
	AR1=D.5*SQR(AR1)	QCA00018
	AR2=(Y32*Z42-Y42*Z32)**2+(X42*Z32-X32*Z42)**2+(X32*Y42-X42*Y32)**2	QCA00019
	AR2=D.5*SQR(AR2)	QCA00020
	C5(1)=(C1(1)+C2(1)+C4(1))/3.0	QCA00021
	C5(2)=(C1(2)+C2(2)+C4(2))/3.0	QCA00022
	C5(3)=(C1(3)+C2(3)+C4(3))/3.0	QCA00023
	C6(1)=(C2(1)+C3(1)+C4(1))/3.0	QCA00024
	C6(2)=(C2(2)+C3(2)+C4(2))/3.0	QCA00025
	C6(3)=(C2(3)+C3(3)+C4(3))/3.0	QCA00026
	AR=AR1+AR2	QCA00027
	C7(1)=(AR1*C5(1)+AR2*C6(1))/AR	QCA00028
	C7(2)=(AR1*C5(2)+AR2*C6(2))/AR	QCA00029
	C7(3)=(AR1*C5(3)+AR2*C6(3))/AR	QCA00030
	RETURN	QCA00031
	END	QCA00032

Table B-22. Source program listing of subroutine QDR

	SUBROUTINE QDR(I1,I2,I3,I4,C1,C2,C3,C4,C5,C6,C7,AR1,AR2)	QDR00001
C	TO FORM QUADRILATERAL ELEMENT STIFFNESS MATRIX	QDR00002
	DIMENSION A(25+25),B(20+20),C(15+15),C1(3),C2(3),C3(3),C4(3),C5(3)	QDR00003
	1,C6(3),C7(3),C8(3),X(3),Y(3),Z(3),XD(2),YD(2),ZD(2),D33(3,3)	QDR00004
	2E22(3,3),S1225)	QDR00005
	EQUILVALENCE (C,S1)	QDR00006
	CALL QCA(I1,I2,I3,I4,C1,C2,C3,C4,C5,C6,C7,AR1,AR2)	QDR00007
	DO 1 I=1,3	QDR00008
	C8(I,1)=C1(I)	QDR00009
	C8(I,2)=C2(I)	QDR00010
	C8(I,3)=C3(I)	QDR00011
	1 C8(I,4)=C4(I)	QDR00012
	DO 4 I=1,25	QDR00013
	DO 4 J=1,25	QDR00014
	4 A(I,J)=0.0	QDR00015
	DO 2 I=1,4	QDR00016
	X(3)=C7(1)	QDR00017
	Y(3)=C7(2)	QDR00018
	Z(3)=C7(3)	QDR00019
	DO 3 J=1,2	QDR00020
	K=I-1+J	QDR00021
	IF(I.EQ.4.AND.J.EQ.2) K=1	QDR00022
	X(J)=C8(1,K)	QDR00023
	Y(J)=C8(2,K)	QDR00024
	3 Z(J)=C8(3,K)	QDR00025
	XD(1)=X(2)-X(1)	QDR00026
	XD(2)=X(3)-X(1)	QDR00027
	YD(1)=Y(2)-Y(1)	QDR00028
	YD(2)=Y(3)-Y(1)	QDR00029
	ZD(1)=Z(2)-Z(1)	QDR00030
	ZD(2)=Z(3)-Z(1)	QDR00031
	IS=I	QDR00032
	CALL SORT(15,XD,YD,ZD,X,Y,Z)	QDR00033
	DO 5 J1=1,15	QDR00034
	DO 5 J2=1,15	QDR00035
	5 C(J1,J2)=0.0	QDR00036
	IOS=15	QDR00037
	CALL S11(XD,YD,TE,S,IOS,D33,E22)	QDR00038
	GO TO (5,7,8,9),I	QDR00039
	9 J10=1	QDR00040
	DO 10 J1=1,5	QDR00041
	DO 11 J2=1,15	QDR00042
	D1=C(J10,J2)	QDR00043
	D2=C(J10+1,J2)	QDR00044
	C(J10,J2)=D2	QDR00045
	11 C(J10+1,J2)=D1	QDR00046
	10 J10=J10+3	QDR00047
	J10=1	QDR00048
	DO 12 J1=1,5	QDR00049
	DO 13 J2=1,15	QDR00050
	D1=C(J2,J10)	QDR00051
	D2=C(J2,J10+1)	QDR00052
	C(J2,J10)=D2	QDR00053
	13 C(J2,J10+1)=D1	QDR00054
	12 J10=J10+3	QDR00055
	CALL ASSEMB(A,C,1,3,4,2,-2)	QDR00056
	GO TO 130	QDR00057
	6 CALL ASSEMB(A,C,1,1,4,0,2)	QDR00058
	GO TO 130	QDR00059
	7 CALL ASSEMB(A,C,2,1,3,0,1)	QDR00060
	GO TO 130	QDR00061
	8 CALL ASSEMB(A,C,3,1,2,0,0)	QDR00062
	130 CONTINUE	QDR00063
	2 CONTINUE	QDR00064
	I11=1	QDR00065
	CALL ELIM(A,B,I11)	QDR00066
	RETURN	QDR00067
	END	QDR00068

Table B-23. Source program listing of subroutine SCAN

```

SUBROUTINE SCAN (IA,IB,IC,N1,N2,N3)          SSCN0001
C TO DETERMINE ELEMENT NUMBERS CONNECTING NODES SSCN0002
DIMENSION IA(N1,N3),IB(N2,N3),IC(N3)        SSCN0003
DO 1 I=1,N1                                  SSCN0004
DO 1 J=1,N3                                  SSCN0005
1 IA(I,J)=0                                  SSCN0006
K1=0                                          SSCN0007
DO 3 I=1,N1                                  SSCN0008
DO 2 II=1,8                                  SSCN0009
2 IC(II)=0                                  SSCN0010
DO 4 J=1,N2                                  SSCN0011
DO 5 K=1,8                                  SSCN0012
IF (IB(J,K).EQ.1) GO TO 6                  SSCN0013
5 CONTINUE                                  SSCN0014
GO TO 4                                      SSCN0015
6 K1=K1+1                                    SSCN0016
IC(K1)=J                                    SSCN0017
4 CONTINUE                                  SSCN0018
I2=1                                          SSCN0019
DO 7 I=1,N3                                  SSCN0020
IA(I,I1)=1000*IC(I2)+IC(I2+1)              SSCN0021
7 I2=I2+2                                    SSCN0022
K1=0                                          SSCN0023
3 CONTINUE                                  SSCN0024
RETURN                                       SSCN0025
END                                           SSCN0026

```

Table B-24. Source program listing of subroutine SO5

```

SUBROUTINE SO5(S,TE,D33,XD,YD,IDS)          SS050001
C GENERATES FOR ELEMENT TYPE 5 STIFFNESS AND LOAD MATRICES SS050002
DIMENSION S(225),D33(3,3),XD(2),YD(2),EM(4,4),EN(4,4),EQ(4,4),Q(4) SS050003
147                                          SS050004
C TRIANGULAR PLANE ELEMENT SS050005
A2=XD(1)*YD(2)-XD(2)*YD(1) SS050006
41 CALL TRIM(EM,EN,EQ,XD,YD) SS050007
CALL TRM (D33,EM,Q,3,3) SS050008
F1=TE/(2.*A2) SS050009
CALL ADM (S,IDS,Q,3,1,1,F1) SS050010
CALL TRM (D33,EN,Q,3,3) SS050011
CALL ADM (S,IDS,Q,3,4,4,F1) SS050012
CALL TRM (D33,EN,EM,3,-3) SS050013
CALL ADM (S,IDS,EM,3,1,9,F1) SS050014
9 RETURN SS050015
END SS050016

```

Table B-25. Source program listing of subroutine SO7

```

SUBROUTINE SO7(XD,YD,TE,E22,S,IDS,D33)      SS070001
C GENERATES FOR ELEMENT TYPE 7 STIFFNESS AND LOAD MATRICES SS070002
DIMENSION S(225),XD(2),YD(2),D33(3,3),E22(3,3),X(3),Y(3),EM(4,4), SS070003
EN(4,4),ER(4,4),Q(4,4) SS070004
C TRIANGULAR BENDING ELEMENT, CONSTANT TRACE SCHEME SS070005
41 CALL TRIM(EM,EN,EQ,XD,YD) SS070007
F1=TE/(2.*A2) SS070008
CALL TRM (E22,EQ,Q,2,3) SS070009
IF (Q(2,1)) 431,432,433 SS070010
431 IF (Q(3,1)) 432,432,436 SS070011
432 IF (Q(3,2)) 433,433,437 SS070012
433 Q(3,1)=Q(3,1)+Q(2,1)/2. SS070013
Q(3,2)=Q(3,2)+Q(2,1)/2. SS070014
Q(2,1)=0. SS070015
GO TO 438 SS070016
436 Q(2,1)=Q(2,1)+Q(3,1)/2. SS070017
Q(3,2)=Q(3,2)+Q(3,1)/2. SS070018
Q(3,1)=0. SS070019
GO TO 438 SS070020
437 Q(2,1)=Q(2,1)+Q(3,2)/2. SS070021
Q(3,1)=Q(3,1)+Q(3,2)/2. SS070022
Q(3,2)=0. SS070023
438 Q(1,1)=Q(2,1)-Q(3,1) SS070024
Q(1,2)=Q(2,1) SS070025
Q(1,3)=Q(3,1) SS070026
Q(2,2)=Q(1,2)-Q(3,2) SS070027
Q(2,3)=Q(3,2) SS070028
Q(3,3)=Q(1,3)-Q(2,3) SS070029
433 CALL ADM (S,IDS,Q,3,1,1,F1) SS070030
DO 4022 L=1,3 SS070031
IF (L=2) 4018,4018,4019 SS070032
4018 DO 4021 I1=4,9 SS070033
DO 4021 J1=1,3 SS070034
ISS=(JJ-1)*IDS+I1 SS070035
ISR=I1I-1+IDS+J1 SS070036
4021 S(IISR)=ISS SS070037
4019 DO 4010 I=1,3 SS070038
I1=I-(I/3)*3+1 SS070039
I2=I1-(I1/3)*3+1 SS070040
K=3*(L-1)+I SS070041
ISS=(K-1)*IDS SS070042
ISS1=ISS+1 SS070043
ISS2=ISS+1 SS070044
ISS3=ISS+2 SS070045
X(1)=0.0 SS070046
X(2)=XD(1) SS070047
X(3)=XD(2) SS070048
Y(1)=0.0 SS070049
Y(2)=YD(1) SS070050
Y(3)=YD(2) SS070051
CX=(X(1)-X(I1))*S(ISS2) SS070052
CY=(X(1)-X(I1))*S(ISS3) SS070053
CX=(Y(1)-Y(I1))*S(ISS2) SS070054
CY=(Y(1)-Y(I1))*S(ISS3) SS070055
S(ISS1+3)=(CX+DX)/2. SS070056
S(ISS2+3)=CX/2. SS070057
S(ISS3+3)=CY/2. SS070058
S(ISS1+6)=(CY+DY)/2. SS070059
S(ISS2+6)=CY/2. SS070060
4010 S(ISS3+6)=DY/2. SS070061
4022 CONTINUE SS070062
CALL TRM (D33,EM,Q,3,3) SS070063
F2=F1+TE/TE/12. SS070064
CALL ADM (S,IDS,Q,3,7,7,F2) SS070065
CALL TRM (D33,EN,Q,3,3) SS070066
CALL ADM (S,IDS,Q,3,4,4,F2) SS070067
CALL TRM (D33,EN,EM,3,-3) SS070068
F2=F2 SS070069
CALL ADM (S,IDS,EM,3,7,4,F2) SS070070
452 CONTINUE SS070071
9 RETURN SS070072
END SS070073

```

Table B-26. Source program listing of subroutine S11

```

SUBROUTINE S11(XD,YD,TE,S,IDS,D33,E22)      SS110001
C GENERATES FOR ELEMENT TYPE 11 STIFFNESS AND LOAD MATRICES SS110002
DIMENSION XD(2),YD(2),S(225),D33(3,3),E22(3,3) SS110003
C TRIANGULAR SHELL ELEMENT SS110004
CALL SO7(XD,YD,TE,E22,S,IDS,D33) SS110005
ISS=9+IDS SS110006
DO 1 I=1,9 SS110007
DO 1 J=1,9 SS110008
IESS=IESS-IDS SS110009
IESS=IESS SS110010
DO 2 J=1,9 SS110011
IESS=IESS+1 SS110012
IYE=IESS+ISHF SS110013
S(IYE)=S(IESS) SS110014
2 S(IES)=0. SS110015
1 CONTINUE SS110016
CALL SO5(S,TE,D33,XD,YD,IDS) SS110017
RETURN SS110018
END SS110019

```

Table B-27. Source program listing of
subroutine STIMAT

```

SUBROUTINE STIMAT(M,INODE,I,AK,IEL,CNJ,B,NDF,NET,NET1,B1,INLT,NEL,ISTIM0001
DIMENSION M(8),INODE(15),AK(6,90),IEL(8,9),AK1(29,29),CNJ(15,3),V1(ISTIM0002
(3),V2(3),V3(3),V10(4),B(20,20),DIR(3,3)
2,B1(NET1),INLT(INFL,DIR(3,3),V1(3,3),V2(3,3)
DO 1 I1=1,6
DO 1 I2=1,90
1 AK(I1,I2)=0.0
C TO FORM SIX ROWS OF STIFFNESS MATRIX AT A JOINT
I1=1
50 CONTINUE
IF(IEL(I1,1).EQ.0) GO TO 2
J=M(I1)
IF(J.EQ.0) GO TO 2
J1=INLT(J)
J1=300+(J1-1)+1
DO 53 I4=1,24
DO 54 J4=1,24
AK(I1,J1)=B1(J1)
64 J1=J1+1
63 CONTINUE
DO 13 I4=2,24
I5=I4-1
DO 13 J4=1,16
13 AK(I1,J1)=AK1(J4,I5)
V1(1)=IEL(I1,1)
V1(2)=IEL(I1,2)
V1(3)=IEL(I1,4)
CALL DIRCOS(V1,V2,INODE,CNJ)
CALL VECT(V1,V2,V3)
CALL UNIT(V3)
CALL UNIT(V1)
CALL VECT(V3,V1,V2)
DO 14 I2=1,3
DIR(1,I2)=V1(I2)
DIR(2,I2)=V2(I2)
14 DIR(3,I2)=V3(I2)
DO 53 I3=1,3
DO 53 J3=1,3
53 DIR1(I3,J3)=DIR(J3,I3)
INODEX=0
61 I4=1
60 DO 54 I3=1,4,8
I5=3+(I4-1)
J5=3+(I3-1)
DO 55 I6=1,3
DO 55 J6=1,3
I56=I5+I6
J56=J5+J6
55 V11(I5,J5)=AK1(I56,J56)
IF(INODEX.EQ.1) GO TO 57
CALL MMULT(V11,DIR,V22,3,3,3)
GO TO 59
57 CALL MMULT(DIR1,V11,V22,3,3,3)
59 DO 56 I6=1,3
DO 56 J6=1,3
I56=I5+I6
J56=J5+J6
56 AK1(I56,J56)=V22(I6,J6)
54 CONTINUE
I4=I4+1
IF(I4=8) 60,60,58
58 INODEX=INODEX+1
IF(INODEX=1) 61,61,52
62 CONTINUE
DO 55 I4=2,24
I5=I4-1
DO 55 J4=1,16
65 AK1(I4,J4)=AK1(J4,I4)
DO 5 J4=1,4
IF(I.EQ.IEL(I1,J4)) GO TO 12
5 CONTINUE
12 J=J4
DO 6 I2=1,4
DO 7 I3=1,15
IF(IEL(I1,I2).EQ.INODE(I3)) GO TO 8
7 CONTINUE
8 V10(I2)=I3
6 CONTINUE
J=6*(J-1)
DO 10 I3=1,6
J=J+1
DO 9 I2=1,4
J1=V10(I2)
J1=5*(J1-1)+1
I5=6*(I2-1)+1
DO 11 I4=1,6
AK(I1,J1)=AK(I3,I1)+AK1(J,I5)
J1=J1+1
11 I5=I5+1
9 CONTINUE
10 CONTINUE
2 I1=I1+1
IF(I1=8) 50,50,51
51 CONTINUE
RETURN
END

```

Table B-28. Source program listing of subroutine STRA

```

SUBROUTINE STRA(IMS,S,DIR)
C TRANSFORMS DESCRIPTION OF ELEMENT MATRICES FROM LOCAL TO OVERALL
DIMENSION DIR(3,3),S(576)
DATA IMS/9/
J=-IDS
DO 5 I=1,IDS
J=J+IDS
CALL TRANS(J,DIR,IMS)
5 CONTINUE
IABB=-IDS
IEBB=-IDS
DO 6 I=1,IDS
IABB=IABB+1
IEBB=IEBB+IDS
IEE=IEEB
DO 7 J=1,IDS
IAB=IAB+IDS
IEE=IEE+1
IF(IAB=IEE) 7,7,13
13 TEMP=S(IEE)
S(IEE)=S(IAB)
S(IAB)=TEMP
7 CONTINUE
6 CONTINUE
J=-IDS
DO 8 I=1,IDS
J=J+IDS
CALL TRANS(J,DIR,IMS)
8 CONTINUE
RETURN
END

```

Table B-29. Source program listing of subroutine STRESS

SUBROUTINE STRESS(I,NMAT,NDF,NET,NEL,PH,D33,DIR,INT,INLT,TNLT,QIN,STSS0001	C8(1,3)=A(2,1)-A(2,4)	STSS0086
IG,NNEL,C1,C3)	C8(2,3)=A(1,10)-A(1,9)	STSS0087
C TO COMPUTE STRESSES IN A QUADRILATERAL ELEMENT	C8(2,5)=A(1,11)-A(1,10)	STSS0088
DIMENSION PHINMAT(3),D33(3,3),DIR(3,3),V1(3),V2(3),V3(3),TMT(NEL),	C8(2,6)=A(1,9)-A(1,1)	STSS0089
INLT(NEL),TNLT(NEL),QIN(NDF)+QIN(NDF)+NDEL(NEL,4)+A1(3,3)+A2(3,3),	C8(3,1)=C8(2,4)	STSS0090
2A1(3,12)+C1(3)+C2(3)+C3(3)+C4(3)+C5(3)+C6(3)+C7(3)+C8(3)+C9(3)+C10(3),	C8(3,2)=C8(2,5)	STSS0091
3DF(6),DF1(6)	C8(3,3)=C8(2,6)	STSS0092
C TO COMPUTE DIRECTION COSINES OF THE ELEMENT	C8(3,4)=C8(1,1)	STSS0093
J1=6*(NDEL(I,1)-1)+1	C8(3,5)=C8(1,2)	STSS0094
J2=6*(NDEL(I,2)-1)+1	C8(3,6)=C8(1,3)	STSS0095
J4=6*(NDEL(I,4)-1)+1	DF(1)=A(1,2)+Z*A(2,3)	STSS0096
DO 2 J1=1,3	DF1(1)=A(1,2)-Z*A(2,3)	STSS0097
V1(I1)=0(J2)-0(J1)	DF(2)=A(1,5)+Z*A(2,6)	STSS0098
V2(I1)=0(J4)-0(J1)	DF1(2)=A(1,5)-Z*A(2,6)	STSS0099
J1=J1+1	DF(3)=A(1,11)+Z*A(2,12)	STSS0100
J2=J2+1	DF1(3)=A(1,11)-Z*A(2,12)	STSS0101
2 J4=J4+1	DF(4)=A(2,2)+Z*A(1,3)	STSS0102
CALL VECT(V1,V2,V3)	DF1(4)=A(2,2)-Z*A(1,3)	STSS0103
CALL UNIT(V3)	DF(5)=A(2,5)+Z*A(1,6)	STSS0104
CALL UNIT(V1)	DF1(5)=A(2,5)-Z*A(1,6)	STSS0105
CALL VECT(V3,V1,V2)	DF(6)=A(2,11)+Z*A(1,12)	STSS0106
DO 3 I2=1,3	DF1(6)=A(2,11)-Z*A(1,12)	STSS0107
DIR(1,I2)=V1(I2)	DO 17 I1=1,3	STSS0108
DIR(2,I2)=V2(I2)	C1(I1)=0.0	STSS0109
3 DIR(3,I2)=V3(I2)	C2(I1)=0.0	STSS0110
DO 10 J=1,4	C3(I1)=0.0	STSS0111
J1=NDEL(I,J)	17 C4(I1)=0.0	STSS0112
J1=6*(J1-1)+1	DO 18 I1=1,3	STSS0113
DO 15 J2=1,3		
GO TO (11,12,13,14),J	DO 18 I2=1,6	STSS0114
11 C1(J2)=0(J11)	C2(I1)=C2(I1)+C8(I1,I2)*DF(I2)	STSS0115
C(J2,J1)=C1(J2)	18 C4(I1)=C4(I1)+C8(I1,I2)*DF1(I2)	STSS0116
GO TO 15	AR=1.0/(2.0+AR1)	STSS0117
12 C2(J2)=0(J11)	DO 19 I2=1,3	STSS0118
C(J2,J1)=C2(J2)	DO 19 I2=1,3	STSS0119
GO TO 15	C1(I1)=C1(I1)+AR*D33(I1,I2)+C2(I2)	STSS0120
13 C3(J2)=0(J11)	19 C3(I1)=C3(I1)+AR*D33(I1,I2)+C4(I2)	STSS0121
C(J2,J1)=C3(J2)	C TO COMPUTE STRESSES IN TRIANGULAR ELEMENT 2	STSS0122
GO TO 15	DO 20 I1=1,3	STSS0123
14 C4(J2)=0(J11)	DO 20 I1=1,3	STSS0124
C(J2,J1)=C4(J2)	20 C8(I1,J1)=0.0	STSS0125
15 J1=J1+1	C8(1,1)=A(2,7)-A(2,10)	STSS0126
10 CONTINUE	C8(1,2)=A(2,10)-A(2,4)	STSS0127
CALL QCA(NDEL(I,1),NDEL(I,2),NDEL(I,3),NDEL(I,4),C1,C2,C3,C4,C5,C6,STSS0044	C8(1,3)=A(2,4)-A(2,7)	STSS0128
I,C7,AR1,AR2)	C8(2,4)=A(1,10)-A(1,7)	STSS0129
C TO FORM MATERIAL MATRIX FOR THE ELASTIC CASE	C8(2,5)=A(1,9)-A(1,10)	STSS0130
IM=INT(I)	C8(2,6)=A(1,7)-A(1,4)	STSS0131
FE=PM(IM,1)	C8(3,1)=C8(2,4)	STSS0132
VMU=PM(IM,2)	C8(3,2)=C8(2,5)	STSS0133
FM=FE/(1.0-VMU**2)	C8(3,3)=C8(2,6)	STSS0134
DO 4 I1=1,3	C8(3,4)=C8(1,1)	STSS0135
DO 4 J1=1,3	C8(3,5)=C8(1,2)	STSS0136
4 D33(I1,J1)=0.0	C8(3,6)=C8(1,3)	STSS0137
D33(I1,1)=FM	DF(1)=A(1,5)+Z*A(2,6)	STSS0138
D33(1,2)=FM*VMU	DF1(1)=A(1,5)-Z*A(2,6)	STSS0139
D33(2,1)=D33(1,2)	DF(2)=A(1,8)+Z*A(2,9)	STSS0140
D33(2,2)=D33(1,1)	DF1(2)=A(1,8)-Z*A(2,9)	STSS0141
D33(3,3)=FM*((1.0-VMU)/2.0)	DF(3)=A(1,11)+Z*A(2,12)	STSS0142
IM=INT(I)	DF1(3)=A(1,11)-Z*A(2,12)	STSS0143
TE=TNLT(IM)	DF(4)=A(2,5)+Z*A(1,6)	STSS0144
DO 5 J1=1,4	DF1(4)=A(2,5)-Z*A(1,6)	STSS0145
I1=NDEL(I,J)	DF(5)=A(2,8)+Z*A(1,9)	STSS0146
DO 6 J2=1,3	DF1(5)=A(2,8)-Z*A(1,9)	STSS0147
6 A2(J2,1)=C(J2,J1)-C(J2,1)	DF(6)=A(2,11)+Z*A(1,12)	STSS0148
I2=6*(I1-1)+1	DF1(6)=A(2,11)-Z*A(1,12)	STSS0149
I3=I2+3	DO 21 I1=1,3	STSS0150
DO 7 J2=1,3	C2(I1)=0.0	STSS0151
A2(J2,2)=0IN(I2)	C4(I1)=0.0	STSS0152
A2(J2,3)=0IN(I3)	C5(I1)=0.0	STSS0153
I2=I2+1	21 C6(I1)=0.0	STSS0154
7 I3=I3+1	DO 22 I1=1,3	STSS0155
CALL MMULT(DIR,A2,A1,3,3,3)	DO 22 I2=1,6	STSS0156
J1=3*(J1-1)+1	C2(I1)=C2(I1)+C8(I1,I2)*DF(I2)	STSS0157
DO 8 J2=1,3	22 C4(I1)=C4(I1)+C8(I1,I2)*DF1(I2)	STSS0158
DO 9 I2=1,3	AR=1.0/(2.0+AR2)	STSS0159
9 A(I2,J1)=A(I2,J2)	DO 23 I1=1,3	STSS0160
0 J1=J1+1	DO 23 I1=1,3	STSS0161
5 CONTINUE	C5(I1)=C5(I1)+AR*D33(I1,I2)+C2(I2)	STSS0162
C TO COMPUTE STRESSES IN TRIANGULAR ELEMENT 1	23 C6(I1)=C6(I1)+AR*D33(I1,I2)+C4(I2)	STSS0163
Z=TE/2.0	AR=AR+AR2	STSS0164
DO 16 I1=1,3	DO 24 I1=1,3	STSS0165
DO 16 J1=1,6	C1(I1)=C1(I1)+AR2*C5(I1,1)/AR	STSS0166
16 C8(I1,1)=A(2,4)-A(2,10)	24 C3(I1)=C3(I1)+AR2*C6(I1,1)/AR	STSS0167
C8(1,2)=A(2,10)-A(2,1)	RETURN	STSS0168
C8(1,3)=A(2,1)-A(2,4)	END	STSS0169

Table B-30. Source program listing of subroutine TRANSP

	SUBROUTINE TRANSP(A,B,J1,J2)	STRP0001
C	TO FORM TRANSPOSE OF A MATRIX	STRP0002
	DIMENSION A(J1,J2),B(J2,J1)	STRP0003
	DO 1 I=1,J2	STRP0004
	DO 1 J=1,J1	STRP0005
	B(J,I)=A(I,J)	STRP0006
	RETURN	STRP0007
	END	STRP0008

Table B-31. Source program listing of subroutine TRAN

	SUBROUTINE TRAN(IFS,DIR,IMS)	STRAND01
C	TRANSFORMS THE DESCRIPTION OF A VECTOR FROM LOCAL TO OVERALL	STRAND02
	DIMENSION A(576),DIR(3,3),DUM(24)	STRAND03
	DATA IGEM/1/	STRAND04
10	IGEMP=IGEM+1	STRAND05
	LJ=0	STRAND06
	LK=IFS-4*IMS	STRAND07
20	DO 300 I=1,IGEMP	STRAND08
	LK=LK+3*IMS	STRAND09
30	DO 200 J=1,3	STRAND10
	DO 200 I=1,IMS	STRAND11
	LJI=LJI+1	STRAND12
	DUM(LJI)=0.	STRAND13
	LKI=LK+I	STRAND14
40	DO 100 K=1,3	STRAND15
	LKI=LKI+IMS	STRAND16
	DUM(LJI)=DUM(LJI)+DIR(K,J)*A(LKI)	STRAND17
	200 CONTINUE	STRAND18
	300 CONTINUE	STRAND19
	INI=3*IGEMP+IMS	STRAND20
	DO 400 I=1,INI	STRAND21
	II=IFS+I	STRAND22
400	A(II)=DUM(I)	STRAND23
	RETURN	STRAND24
	END	STRAND25

Table B-32. Source program listing of subroutine TRIM

	SUBROUTINE TRIM(EM,EN,EQ,XD,YD)	STRIM001
C	GENERATES M, N AND L MATRICES OF THIN TRIANGULAR ELEMENT	STRIM002
	DIMENSION EM(4,4),EN(4,4),EQ(4,4),XD(2),YD(2)	STRIM003
	EM(1,2)=YD(2)	STRIM004
	EM(1,3)=-YD(1)	STRIM005
	EM(1,4)=YD(1)-YD(2)	STRIM006
	EM(2,1)=0.	STRIM007
	EM(2,2)=0.	STRIM008
	EM(2,3)=0.	STRIM009
	EM(3,1)=XD(2)-XD(1)	STRIM010
	EM(3,2)=-XD(2)	STRIM011
	EM(3,3)=XD(1)	STRIM012
	DO 42 J=1,3	STRIM013
	EM(1,J)=EM(2,J)	STRIM014
	EM(2,J)=EM(3,J)	STRIM015
	EM(3,J)=EM(1,J)	STRIM016
42	CONTINUE	STRIM017
	DO 430 J=1,3	STRIM018
	EQ(1,J)=EM(1,J)	STRIM019
430	EQ(2,J)=EM(3,J)	STRIM020
	RETURN	STRIM021
	END	STRIM022

Table B-33. Source program listing of subroutine TRM

	SUBROUTINE TRM(A,B,C,M,N1)	STRM0001
C	PERFORMS TRIPLE MATRIX PRODUCT OF ORDER 3	STRM0002
	DIMENSION A(3,3),B(4,4),C(4,4),D(4,4)	STRM0003
	NE=1	STRM0004
	IF (N) 2,1,1	STRM0005
2	N=N-1	STRM0006
	GO TO 8	STRM0007
1	DO 3 I=1,M	STRM0008
	DO 3 J=1,N	STRM0009
3	C(I,J)=B(I,J)	STRM0010
8	DO 4 I=1,N	STRM0011
	DO 4 J=1,M	STRM0012
	D(I,J)=0.	STRM0013
	DO 5 K=1,M	STRM0014
5	D(I,J)=D(I,J)+C(K,I)*A(K,J)	STRM0015
4	CONTINUE	STRM0016
	DO 6 I=1,N	STRM0017
	DO 6 J=1,N	STRM0018
	C(I,J)=0.	STRM0019
	DO 7 K=1,M	STRM0020
7	C(I,J)=C(I,J)+D(I,K)*B(K,J)	STRM0021
6	CONTINUE	STRM0022
	RETURN	STRM0023
	END	STRM0024

Table B-34. Source program listing of subroutine UNIT

	SUBROUTINE UNIT(V1)	SUNIT0001
C	TO NORMALISE A VECTOR	SUNIT0002
	DIMENSION V1(3)	SUNIT0003
	V1=SQRT(V1(1)**2+V1(2)**2+V1(3)**2)	SUNIT0004
	DO 1 I=1,3	SUNIT0005
1	V1(I)=V1(I)/V1	SUNIT0006
	RETURN	SUNIT0007
	END	SUNIT0008

Table B-35. Source program listing of subroutine VECT

	SUBROUTINE VECT(V1,V2,V3)	SVECD0001
C	TO PERFORM VECTOR PRODUCT OPERATION	SVECD0002
	DIMENSION V1(3),V2(3),V3(3)	SVECD0003
	V3(1)=V1(2)*V2(3)-V2(2)*V1(3)	SVECD0004
	V3(2)=V2(1)*V1(3)-V1(1)*V2(3)	SVECD0005
	V3(3)=V1(1)*V2(2)-V2(1)*V1(2)	SVECD0006
	RETURN	SVECD0007
	END	SVECD0008

References

1. *Heat Sterilizable and Impact Resistant Ni-Cd Battery Development*, Modification 6, JPL Contract 951972. Texas Instruments, Inc., Attleboro, Mass., Jan. 1970.
2. Gupta, K. K., "Free Vibrations of Single-Branch Structural Systems," *J. Inst. Math. Appl.* Vol. 5, No. 3, pp. 351-362, Sept. 1969.
3. Gupta, K. K., "Vibration of Frames and Other Structures With Banded Stiffness Matrix," *Int. J. Num. Meth. Eng.*, Vol. 2, No. 2, pp. 221-228, Apr. 1970.
4. Zienkiewicz, O. C., and Cheung, Y. K., *The Finite Element Method in Structural and Continuum Mechanics*, p. 181. McGraw-Hill Book Co., Inc., New York, 1967.
5. Utku, S., "Stiffness Matrices for Thin Triangular Elements of Nonzero Gaussian Curvature," *AIAA J.*, Vol. 5, No. 9, pp. 1650-1667, Sept. 1967.
6. Utku, S., and Akyuz, F. A., *ELAS—A General-Purpose Computer Program for the Equilibrium Problems of Linear Structures: Volume I, User's Manual*, Technical Report 32-1240. Jet Propulsion Laboratory, Pasadena, Calif., Feb. 1, 1968.
7. Gupta, K. K., "High Impact Dynamic Response Analysis of Nonlinear Structures," in *Supporting Research and Advanced Development*, Space Programs Summary 37-60, Vol. III, pp. 142-145. Jet Propulsion Laboratory, Pasadena, Calif., Dec. 1969.
8. *Heat Sterilizable Impact Resistant Cell Development*, JPL Contract 951296, ESB, Inc., Raleigh, N.C., June 1969.