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
Technical Report 32-1498

High-Impact Dynamic Response Analysis of Nonlinear Structures

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

<table>
<thead>
<tr>
<th>Source program listing</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-1. Source program listing of main program IMAN</td>
<td>34</td>
</tr>
<tr>
<td>B-2. Source program listing of subroutine ACC</td>
<td>36</td>
</tr>
<tr>
<td>B-3. Source program listing of subroutine ADM</td>
<td>36</td>
</tr>
<tr>
<td>B-4. Source program listing of subroutine ASSEMB</td>
<td>36</td>
</tr>
<tr>
<td>B-5. Source program listing of subroutine CONOJ</td>
<td>36</td>
</tr>
<tr>
<td>B-6. Source program listing of subroutine COORD</td>
<td>37</td>
</tr>
<tr>
<td>B-7. Source program listing of subroutine CORT</td>
<td>37</td>
</tr>
<tr>
<td>B-8. Source program listing of subroutine DCS</td>
<td>37</td>
</tr>
<tr>
<td>B-9. Source program listing of subroutine DFORCE</td>
<td>37</td>
</tr>
<tr>
<td>B-10. Source program listing of subroutine DIEI</td>
<td>37</td>
</tr>
<tr>
<td>B-11. Source program listing of subroutine DIRCOS</td>
<td>37</td>
</tr>
<tr>
<td>B-12. Source program listing of subroutine ELEMT</td>
<td>38</td>
</tr>
<tr>
<td>B-13. Source program listing of subroutine ELIM</td>
<td>38</td>
</tr>
<tr>
<td>B-14. Source program listing of subroutine INV</td>
<td>38</td>
</tr>
<tr>
<td>B-15. Source program listing of subroutine MELCON</td>
<td>39</td>
</tr>
<tr>
<td>B-16. Source program listing of subroutine MMULT</td>
<td>39</td>
</tr>
<tr>
<td>B-17. Source program listing of subroutine NIEL</td>
<td>39</td>
</tr>
<tr>
<td>B-18. Source program listing of subroutine NQII</td>
<td>39</td>
</tr>
<tr>
<td>B-19. Source program listing of subroutine PREST</td>
<td>39</td>
</tr>
<tr>
<td>B-20. Source program listing of subroutine PROMAT</td>
<td>40</td>
</tr>
<tr>
<td>B-21. Source program listing of subroutine QCA</td>
<td>40</td>
</tr>
<tr>
<td>B-22. Source program listing of subroutine QDR</td>
<td>40</td>
</tr>
<tr>
<td>B-23. Source program listing of subroutine SCAN</td>
<td>41</td>
</tr>
<tr>
<td>B-24. Source program listing of subroutine S05</td>
<td>41</td>
</tr>
<tr>
<td>B-25. Source program listing of subroutine S07</td>
<td>41</td>
</tr>
<tr>
<td>B-26. Source program listing of subroutine S11</td>
<td>41</td>
</tr>
<tr>
<td>B-27. Source program listing of subroutine STIMAT</td>
<td>42</td>
</tr>
<tr>
<td>B-28. Source program listing of subroutine STRA</td>
<td>42</td>
</tr>
<tr>
<td>B-29. Source program listing of subroutine STRESS</td>
<td>43</td>
</tr>
<tr>
<td>B-30. Source program listing of subroutine TRANSP</td>
<td>44</td>
</tr>
<tr>
<td>B-31. Source program listing of subroutine TRAN</td>
<td>44</td>
</tr>
<tr>
<td>B-32. Source program listing of subroutine TRIM</td>
<td>44</td>
</tr>
</tbody>
</table>
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-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

1. 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 righthanded 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:

$$[M][\ddot{q}] + [C][\dot{q}] + [K][q] = (P(t)) \quad (1)$$
mass matrix (assumed to be lumped at the nodes)

\[ [M] \]

viscous damping matrix

\[ [C] \]

stiffness matrix

\[ [K] \]

deformation vector

\[ \{q\} \]

time-dependent forcing function vector

\[ \{P(t)\} \]

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

\[
[M]\{\delta q\}_{n,n+1} + [C]\{\delta q\}_{n,n+1} + [K]\{\delta q\}_{n,n+1} = \{\delta P\}_{n,n+1}
\]

(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):

\[
\{\delta q\}^P,n = \frac{1}{(M)^P,n} [(P)^P,n - (S)^P,n - \{e^P\}^P\{\delta q\}^P,n]
\]

(3)

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

\[
\{K1\}^n = h\{\delta q\}^n
\]

(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

\[
\{\delta q\}^P,n+\frac{1}{2} = \frac{1}{(M)^P,n+\frac{1}{2}} [(P)^P,n+\frac{1}{2} - (S)^P,n+\frac{1}{2} - \{K^P\}^P,n\{\delta q\}^P,n+\frac{1}{2}]
\]

(5)

in which the extrapolated values are

\[
\{\delta q\}^P,n+\frac{1}{2} = h\{\delta q\}^P,n + \frac{1}{2}(K1)^P,n
\]

(6)

\[
\{\delta q\}^P,n+\frac{1}{2} = (\delta q)^P,n + \frac{1}{2}(\delta K1)^P,n
\]

(7)

and

\[
(K2)^n+\frac{1}{2} = h(\delta q)^n+\frac{1}{2}
\]

(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):

\[
\{\delta q\}^P,R,n+\frac{1}{2} = \frac{1}{(M)^P,n+\frac{1}{2}} [(P)^P,n+\frac{1}{2} - (S)^P,n+\frac{1}{2} - \{K^P\}^P,n\{\delta q\}^P,n+\frac{1}{2} - \{C^P\}^P\{\delta q\}^P,n+\frac{1}{2}]\]

(9)
where

\[ (\delta q)^{n+1/2}_{(Q)} = \frac{h}{2} \{\dot{q}\}^n + \frac{h}{8} (K2)^{P,n+1/2} \]  

(10)

\[ (\delta \dot{q})^{P,n+1/2}_{(Q)} = \{\dot{q}\}^n + \frac{1}{2} (\delta K2)^{P,n+1/2} \]  

(11)

Further, \n
\[ (K3)^{n+1/2}_{(Q)} = h (\dot{q})^{n+1/2} \]

in which

\[ (\delta q)^{n+1}_{P} = \frac{1}{(M)^{P,n+1}} [(P)^{P,n+1} - (S)^{P,n+1} - (K3)^{P,n+1/2}_{(Q)}} \]  

(14)

and

\[ (\delta \dot{q})^{P,n+1}_{(Q)} = (\delta \dot{q})^{P,n} + (\delta K3)^{P,n+1}_{(Q)} \]  

(15)

Also,

\[ (K4)^{n+1} = h (\dot{q})^{n+1} \]  

(16)

is stored.

**Step 4.** The revised stiffness matrix \((K)^{n+1/2}_{(Q)}\) based on the \((q)^{n+1}_{(Q)}\) 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:

\[ \{q\}^{P,n+1}_{(Q)} = \{q\}^{n} + [\{K1\} + \{(K2) + (K3)\} \]  

\[ \{\dot{q}\}^{n} + \{qI\}^{n+1} \]  

(17)

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

(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:

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

(19)

and

\[ \{\delta q\}^{n+1}_{(Q)} \]  

are stored.

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

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

(17a)

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

\[ (RF)^{P,n+r}_{(Q)} = - \frac{(\dot{q})^{P,n+r}_{(Q)}}{(M)^{P,n+r}} \]  

(20)

the corresponding acceleration \((\ddot{q})^{P,n+r}_{(Q)}\) 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 \(z\) 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 characteristics modes, are the associated input data. The characteristic zero-deformation modes numbered 1 to 6 are associated respectively with the degrees of freedom \(\delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z\).

C. List of Variables

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

- \(E\) Young's modulus
- \(\mu\) Poisson's ratio
- \(c\) damping coefficient for the material
- \(\text{IPRINT}\) 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
NEL total number of elements
NMAT number of material types, the latter being dependent on the three variable $E$, $\mu$, and $c$
NZD number of zero deformations, including such deformations at nodes along planes of symmetry
NET number of element types, which is determined by element thickness and dimensions in the local coordinate system
NDF total number of degrees of freedom ($= 6 \times NJ$)
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
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
TI incremental time step
NIN total number of such increments
VIN initial nodal velocity prior to impact
IDFV particular degree of freedom associated with the initial velocity

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

$AK(6,90)$ stiffness matrix in the global coordinate system pertaining to a node

$AK1(24,24)$ final stiffness matrix of a quadrilateral element in the global coordinate system

$AK12(20,5)$ matrices which are portions of the rearranged $A$ matrix utilized to eliminate the fifth central node of the quadrilateral

$AK21(5,20)$ quadrilateral element stiffness matrix in the local coordinate system derived from a matrix by eliminating the fifth central node

$B(20,20)$

$B1(300*NET)$ 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

$C(15,15)$ triangular shell element stiffness matrix in the local coordinate system

$C1,C2,C3,C4,C5$ $C1$ to $C7$ are of order $(3,3)$, whereas $C8$ is of order $(3,4)$; they are used for general computational work

$CMASS(NDF)$ file storing lumped nodal masses

$CNJ(15,3)$ file storing coordinates of nodes connected to any particular node being considered

$DAMP(15)$ damping coefficients file for a node connected to a maximum of 14 other nodes

$D33(3,3)$ elastic stress–strain matrix for plate bending case

$DIR(3,3)$ direction cosine matrix of an element

$DM(8)$ damping coefficients of a maximum number of 8 elements connecting a node

$DP(6)$ damping force at a node corresponding to its 6 degrees of freedom

$E22(3,3)$ 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
IELCON(NJ,A) file storing element nodal numbers (a maximum of 8 elements stored in 4 words in pairs as IC(1)*1000+IC(2), etc.) connecting each node

IMT(NEL) element material type indicator file

INLT(NEL) element type indicator file

INODE(15) file containing nodal numbers connecting a node

IZD(NZD,2) zero-deformation file storing the nodal number and the mode of the associated zero deformation

M(8) file storing element numbers connecting a node

NLT(NET,5) file storing element nodal numbers and its material type

NNEL(NEL,A) file storing 4 nodal numbers for each element

PM(NMAT,3) material properties file containing Young's modulus E, Poisson's ratio μ, and damping coefficient c values for each material type

P(NDF) external loading file

Q(NDF) nodal deformation file, the first three columns of which represent current nodal coordinates

QII(NDF) incremental nodal deformation file pertaining to each incremental sub-time step of the Runge-Kutta procedure

QII(90) incremental nodal deformations pertaining to the nodes connecting the particular node under consideration

QQI(NDF) accumulating nodal incremental deformations during a time interval of the Runge-Kutta step

QIR(NDF) incremental revised nodal deformation QI at half-time interval

Q1(NDF) nodal velocity file

QII(NDF) accumulating nodal incremental velocity file during a time interval of the Runge-Kutta step

Q2(NDF) nodal acceleration file

Q22(NDF) file storing Q2 for use in the Runge-Kutta operations at subtime steps

RF(NZD) file storing reactions at contact points as well as at points of symmetry, i.e., reactions at zero-deformation nodes

S(NDF) nodal forces file

ST(6*NEL) file storing element stresses δx, δy, and τxy 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

TEL(NEL) element thickness file

TINT(NIN) file storing incremental time information

TNLT(NET) file storing thickness of various element types

V1,V2,V3,V4,V10 files of order (3), excepting for V10, which is of order (4), used for storage purposes during various computations

X,Y,Z files of order (3) storing element nodal coordinates and used throughout the computations

XD,YD,ZD files of order (2) storing element side lengths during various computations

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:

(1) 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=
```

(2) 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
@FOR,S 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.
@RPT, 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 $Q1^*$ files.
   c. Acceleration file $Q2^n$ 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. $Q2^n$ 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 $Q1^{n+\frac{1}{2}}$ is computed using files $Q1^n$ and $Q2^n$.
   b. ACC is called to compute $Q2^{n+\frac{1}{2}}$ and $RF$ files.
   c. $Q2^{n+\frac{1}{2}}$ 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 $Q1^R_{\text{new}}$ is computed from $Q1^n$ and $Q2^{n+\frac{1}{2}}$ files.
   b. ACC is called to compute $Q2^R_{\text{new}}$ and $RF$ files.
   c. STRESS is called to compute stress increments in the elements due to incremental deformation $Q1^{n+\frac{1}{2}}$.

4. To compute acceleration values at the end of the time step, the primary steps are as follows:
   a. $QI_{n+1}$ is formed from files $Q1^n$ and $Q2^{n+\frac{1}{2}}$.
   b. ACC is called, IUPDAT being 1 to compute $Q2^{n+1}$, $RF$, and nodal forces incremental file $S$ (denoted by $S$ in ACC) = $K^{n+\frac{1}{2}}Q1I^{n+1}$.
   c. STRESS is called to compute incremental stresses in the elements due to $QI_{n+1}$.
   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 $QII_{n+1}$.
   g. Updated $Q^{n+1} = Q^n + QQI^{n+1}$ and $Q1^{n+1} = Q1^n + QII_{n+1}$ 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 sub-matrices to form the stiffness matrix of a triangular shell element.

3. Subroutine ASSEMB. Subroutine ASSEMB is called by QDR to assemble the stiffness matrices $C$ for each of the four triangular shell elements into the quadrilateral element stiffness matrix $A$.

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, $Q_1$, and $Q_2$ are used.

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 DIFCOS. Subroutine DIFCOS 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 ELEMT. Subroutine ELEMT 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:
   - Elements of $FEL$ pertaining to the element under consideration are copied into file $SEL$.
   - 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 $SS$.

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 ($E$, $\mu$, 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
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 \( u_1, \cdots, u_6, v_1, \cdots, v_6, w_1, \cdots, w_6, \theta_x, \cdots, \theta_y \).

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 S05. Subroutine S05 is called by S11 to generate the stiffness matrix of triangular elements under plane stress conditions.

24. Subroutine S07. Subroutine S07 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, S05, S07, TRAN, TRIM, TRM, and INV used in this connection are the modified versions 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 \(AKL = DIR^T AKL_0 DIR\), 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 \(Q^*\), 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 \(Q^*\) and incremental deformations at the nodes based on
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** $0.138 \times 10^{-3}$

---

**JPL TECHNICAL REPORT 32-1498**
Number of such increments 34
Total number of nodes 93
Total number of finite elements 76
Young's modulus, lb/in.² $3.75 \times 10^5$
Unit weight, lb/in.³ $0.384 \times 10^{-3}$
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.

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 $0.19 \times 10^{-6}$
- Number of such increments 35
- Total number of nodes 100
- Total number of finite elements 78
- Young's modulus, lb/in.² $10.3 \times 10^6$
- Unit weight, lb/in.³ 0.381
- Poisson's ratio 0.367

Results are presented in graphical form in Figs. 7-10.
Fig. 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
matrices and store

Start Runge-Kutta solution

Form symmetric part of $A_{kl}$ in file $B_1$

Loop on $i$ satisfied?

Yes

Set counter $N_{IN}$, total number of time intervals, for J10

Initialize files $Q_{II}$, $Q_{QI}$, $Q_I$, $Q_{22}$, and $Q_{1K}$

Print file 8

Fig. A-1 (contd)
To compute acceleration at beginning of current time step

Set counter NJ for I

Call DIEL to obtain files INOISE and DAMP

Set counter 6, total no. of degrees of freedom, for I1

Compute damping forces file DP

Loop on I1 satisfied?

Yes

Set counter 6 to I1

Compute acceleration Q2

No

I = IZD(I2, 1) and I1 = IZD(I2, 2)?

Loop on I2 satisfied?

Yes

No

No

Compute reactions file RF; set corresponding acceleration value = 0

Check on counters IC1 and ICOUNT

Yes

No

Loop on I satisfied?

Yes

No

Print support reactions file RF

IPRINT = 0?

Yes

No

Print file Q2

Fig. A-1 (contd)
Set Q22 = Q2 compute files Q11 and QQ1

If IPRINT = 0
    Print files Q11 and QQ1
Else
    To compute extrapolated acceleration values at half time step
        Compute incremental deformation file Q1
        If IPRINT = 0
            Print file Q1
        Else
            Call ACC to compute acceleration at end of half time step
            Print file RF

Set Q22 = Q2 compute files Q11 and QQ1

If IPRINT = 0
    Print files Q11 and QQ1
Else
    To compute revised Q2 file at end of half time step
        Compute file Q1
        If IPRINT = 0
            Print file Q1
        Else
            Call ACC to compute revised acceleration at end of half time step
            Print file RF

Fig. A-1 (contd)
Set counter NEL, number of elements to 1

Call stress to compute stress increments in element due to Q1

Loop on I satisfied?

No

Yes

IPRINT = 0?

No

Print file Q2

Yes

Set QIR = QI, Set Q22 = Q2, Compute files Q11 and QQI

IPRINT = 0?

No

Print files Q11 and QQI

Yes

To compute Q2 file at end of current time step

Compute file Q1

Fig. A-1 (contd)
Loop on I satisfied?

Print stresses in elements \( \{ S \}^{n+1} \) at end of time step

Call PREST to update nodal forces at end of time step

Print nodal and element nodal forces files \( \{ E \}^{n+1} \) and \( \{ F \}^{n+1} \)

Update file Q11

JPRINT = 0 ?

Print file Q11

Update nodal geometry and velocity files Q and Q1

Loop on J10 satisfied?

Print Q, Q1, and Q2 files

Stop IMAN

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
Set counter I1 to 6, the number of degrees of freedom at a node

Compute damping force DF(I1) from files DAMP, INODE, Q1, and Q2

No

Loop on IT satisfied?

Yes

Return

Subroutine DFORCE

Fig. A-7. Flowchart of subroutine DFORCE

Subroutine DIEL

Subroutine DCS

Compute starting address in Q file of element nodes 1, 2, and 4

Compute direction vectors V1 and V2 of element sides connecting nodes 1, 2 and 1, 4 respectively

Call VECT to perform cross vector product of V1 and V2 and form V3

Call UNIT to obtain direction cosines in X and Z directions from V1 and V3

Call VECT to perform cross vector product of V3 and V1 to obtain V2 in Y direction

Form direction cosine matrix DI8 by assembling V1, V2, and V3

Return

Fig. A-6. Flowchart of subroutine DCS

Fig. A-8. Flowchart of subroutine DIEL
Find row locations in INODE file of nodes 1, 2, and 4 of quadrilateral element

Compute coordinate difference of nodes 1-2 and 4-1 from file CNJ and store in files V1 and V2

Return

Subroutine DIRCOS

Subroutine ELEM

Read element nodal numbers, thickness, material, and element type indicators

Return

Subroutine ELIM

Rearrange elements of A and place elements pertaining to the four corner nodes into matrix B

Copy matrices AK22 and AK12 from A

Call TRANS to form AK21, transpose of AK12

Call INV to compute

Compute B

B = B - AK12 * AK21

Return

Fig. A-9. Flowchart of subroutine DIRCOS

Fig. A-10. Flowchart of subroutine ELEM

Fig. A-11. Flowchart of subroutine ELIM
Copy file IELCON to form element connection file M at a node

Return

Subroutine MELCON

Perform matrix multiplication $A \times B$ and store product in C

Return

Subroutine MMULT

Set element nodal description file IEL initially to zero

Set counter I1 to $B$, the maximum number of elements at a node

Copy a row of NIEL, pertaining to the particular element, into IEL

Loop on I1 satisfied?

Yes

Return

Subroutine NIEL

Fig. A-12. Flowchart of subroutine MELCON

Fig. A-13. Flowchart of subroutine MMULT

Fig. A-14. Flowchart of subroutine NIEL
Subroutine PREST

Initialize file QII, set elements to zero

Set counter I1 to 15, the maximum number of nodes connected to a node

Copy elements of incremental deformation file QI pertaining to node INODE (I1) into file QII

Loop on I1 satisfied?

Yes

Return

No

Subroutine PREST

Set counter NJ for I

Call MELCON to form file M

Call DCS to form DIR file based on $Q_n^{n+1} = Q_n^n + QQ^n_{n+1}$

Compute new nodal forces file $S_n^{n+1}$ from DIR and FEL files, file $S_n^n$ incorporates prestress effect

Add nodal force increment file $\Delta S_n^{n+1}$ computed by ACC, to obtain updated force file $S_n^{n+1} = S_n^n + \Delta S_n^{n+1}$

Loop on I1 satisfied?

Yes

Return

No

Set counter NEL on I

Call DCS to form DIR file based on $Q^n$ configuration

Using DIR and QQI files, form accumulating nodal deformation file A in local coordinate system

Copy element stiffness matrix AK1 from BL file

Form element nodal forces increment file SEL1 from files AK1 and A

Add SEL1 to FEL to obtain updated element nodal forces file

Loop on I4 satisfied?

Yes

Return

No
PROMAT

obtain quadrilateral element stiffness matrix B from A by eliminating fictitious central point

Read E, μ, and c for each element material type

Return

Fig. A-17. Flowchart of subroutine PROMAT

Subroutine QCA

Let the quadrilateral element be divided into two triangles by joining nodes 2 and 4

Compute surface areas of triangular elements 1 and 2

Compute center of areas of the triangular elements 1 and 2

Compute area and coordinates of center of area of the quadrilateral element

Return

Fig. A-18. Flowchart of subroutine QCA

Subroutine QDR

Call subroutine QCA to compute coordinates of center of area of quadrilateral element

Set counter 4 for 1 to represent each of four triangular elements of the quadrilateral element

Call subroutine CORT to obtain coordinates of nodes of triangular element in local coordinate system of quadrilateral element

Call subroutine S21 to compute triangular element stiffness matrix C

Call subroutine ASSEMB to place elements of C in overall quadrilateral element stiffness matrix A

Loop on i satisfied

No

Yes

Call subroutine ELIM to obtain quadrilateral element stiffness matrix B from A by eliminating fictitious central point

Return

Fig. A-19. Flowchart of subroutine QDR
Set file IELCON initially to zero

Set joint number counter I initially to 1

Initialize file IC, set element number counter J initially to 1

Set counter K initially to 1

$IB(J, K) = 1$ ?

$K \leq 4$ ?

Set relevant element of IC to J

$J \leq N2$ ?

Compute elements of IELCON file

$I \leq N1$ ?

Return

Fig. A-20. Flowchart of subroutine SCAN
Subroutine STIMAT

Set element number counter II initially to 1

Copy element stiffness matrix AK1 from file A

Form transformation matrix DR for the element

Convert AK1 into global coordinate system

Place AK1 elements in AK file, the overall stiffness matrix for the joint

Increase counter II by 1

Yes

11 ≤ 0

No

Return

Subroutine STRESS

Compute DIR matrix for element 1

Call QCA to obtain surface areas of triangles 1 and II

Compute material matrix D33

Form matrix A, storing nodal coordinates and incremental deformations in local coordinate system

Form coefficient matrix CB for triangle I from file A

Form deformation matrix DF for triangle I

Compute stresses in the element using CB and DF files

Form file CB for triangle II from file A

Form DF file for triangle II

Using CB and DF files, compute stresses in the element

Combine stresses in elements I and II to obtain average stress in the quadrilateral element

Return

Fig. A-21. Flowchart of subroutine STIMAT

Fig. A-22. Flowchart of subroutine STRESS
Fig. A-23. Flowchart of subroutine TRANSP

Subroutine TRANSP

Set matrix B as transpose of matrix A

Return

Fig. A-24. Flowchart of subroutine UNIT

Subroutine UNIT

Compute length of vector

Normalize vector

Return

Fig. A-25. Flowchart of subroutine VECT

Subroutine VECT

Perform vector product operation

Return
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.

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

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>34</td>
<td>JPL TECHNICAL REPORT 32-1498</td>
</tr>
</tbody>
</table>
Table B.1 (contd)
Table B-1 (contd.)

| FORMATION UI FILE AT END OF STEP NO.111 | IMAN023 |
| WRITE(6,110) | IMAN024 |
| WRITE(6,111) 16,1601 | IMAN025 |

Table B-2. Source program listing of subroutine ACC

| SUBROUTINE ACC | SAD0001 |
| WRITE(6,110) | SAD0002 |
| WRITE(6,111) 16,1601 | SAD0003 |

Table B-3. Source program listing of subroutine ADM

| SUBROUTINE ADM (15,10,16,10,16) | SAD0004 |
| WRITE(6,110) | SAD0005 |
| WRITE(6,111) 16,1601 | SAD0006 |

Table B-4. Source program listing of subroutine ASSEMB

| SUBROUTINE ASSEMB | SAD0007 |
| WRITE(6,110) | SAD0008 |
| WRITE(6,111) 16,1601 | SAD0009 |

Table B-5. Source program listing of subroutine CONOJ

| SUBROUTINE CONOJ (16,10,16,10,16) | SCD0001 |
| WRITE(6,110) | SCD0002 |
| WRITE(6,111) 16,1601 | SCD0003 |

36  JPL TECHNICAL REPORT 32-1498
Table B-12. Source program listing of subroutine ELEM

Table B-13. Source program listing of subroutine ELM

Table B-14. Source program listing of subroutine INV
Table B-20. Source program listing of subroutine PROMAT

SUBROUTINE PROMAT(N,N11)
C
C SUBROUTINE TO GENERATE PROPERTIES MATRIX
C
DIMENSION C(143), C(143, 143), C(143, 143, 143)
C
RETURN
END

Table B-21. Source program listing of subroutine QCA

SUBROUTINE QCA(N,EN,KK,B)
C
C SUBROUTINE TO GENERATE QUADRILATERAL ELEMENT STIFFNESS MATRIX
C
DIMENSION C(143 ,143), C(143, 143, 143)
C
RETURN
END

Table B-22. Source program listing of subroutine QDR

SUBROUTINE QDR(N,EN,KK,B)
C
C SUBROUTINE TO GENERATE QUADRILATERAL ELEMENT STIFFNESS MATRIX
C
DIMENSION C(143 ,143), C(143, 143, 143)
C
RETURN
END
### Table B-23. Source program listing of subroutine SCAN

<table>
<thead>
<tr>
<th>SUBROUTINE SCAN (A=0.0+EC-N1+N2+N3)</th>
<th>CDCNS001</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMENSION N1(N2,N3,E3),E1(E2,E3)</td>
<td>CDCNS002</td>
</tr>
<tr>
<td>DO J J(J-1)+2</td>
<td>CDCNS003</td>
</tr>
<tr>
<td>I (J-1)+1</td>
<td>CDCNS004</td>
</tr>
<tr>
<td>A I(J-1)+1</td>
<td>CDCNS005</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS006</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS007</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS008</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS009</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS010</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS011</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS012</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS013</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS014</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS015</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS016</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS017</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS018</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS019</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS020</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS021</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS022</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS023</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS024</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS025</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS026</td>
</tr>
</tbody>
</table>

### Table B-24. Source program listing of subroutine S07

<table>
<thead>
<tr>
<th>SUBROUTINE S07 (A=0.0+EC-N1+N2+N3)</th>
<th>CDCNS001</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMENSION N1(N2,N3,E3),E1(E2,E3)</td>
<td>CDCNS002</td>
</tr>
<tr>
<td>DO J J(J-1)+2</td>
<td>CDCNS003</td>
</tr>
<tr>
<td>I (J-1)+1</td>
<td>CDCNS004</td>
</tr>
<tr>
<td>A I(J-1)+1</td>
<td>CDCNS005</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS006</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS007</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS008</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS009</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS010</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS011</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS012</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS013</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS014</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS015</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS016</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS017</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS018</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS019</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS020</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS021</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS022</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS023</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS024</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS025</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS026</td>
</tr>
</tbody>
</table>

### Table B-25. Source program listing of subroutine S05

<table>
<thead>
<tr>
<th>SUBROUTINE S05 (A=0.0+EC-N1+N2+N3)</th>
<th>CDCNS001</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMENSION N1(N2,N3,E3),E1(E2,E3)</td>
<td>CDCNS002</td>
</tr>
<tr>
<td>DO J J(J-1)+2</td>
<td>CDCNS003</td>
</tr>
<tr>
<td>I (J-1)+1</td>
<td>CDCNS004</td>
</tr>
<tr>
<td>A I(J-1)+1</td>
<td>CDCNS005</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS006</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS007</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS008</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS009</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS010</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS011</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS012</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS013</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS014</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS015</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS016</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS017</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS018</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS019</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS020</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS021</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS022</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS023</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS024</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS025</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS026</td>
</tr>
</tbody>
</table>

### Table B-26. Source program listing of subroutine S11

<table>
<thead>
<tr>
<th>SUBROUTINE S11 (A=0.0+EC-N1+N2+N3)</th>
<th>CDCNS001</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMENSION N1(N2,N3,E3),E1(E2,E3)</td>
<td>CDCNS002</td>
</tr>
<tr>
<td>DO J J(J-1)+2</td>
<td>CDCNS003</td>
</tr>
<tr>
<td>I (J-1)+1</td>
<td>CDCNS004</td>
</tr>
<tr>
<td>A I(J-1)+1</td>
<td>CDCNS005</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS006</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS007</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS008</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS009</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS010</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS011</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS012</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS013</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS014</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS015</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS016</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS017</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS018</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS019</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS020</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS021</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS022</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS023</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS024</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS025</td>
</tr>
<tr>
<td>J(J-1)+1</td>
<td>CDCNS026</td>
</tr>
</tbody>
</table>
### Table B-27. Source program listing of subroutine STIMAT

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>2</td>
<td>CALL UNITIV</td>
</tr>
<tr>
<td>3</td>
<td>CALL MVECTVL</td>
</tr>
<tr>
<td>4</td>
<td>DO 1 J=1,JL</td>
</tr>
<tr>
<td>5</td>
<td>IF J**2 EQ IE NO GO TO 6</td>
</tr>
<tr>
<td>6</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>7</td>
<td>CALL MVECTVL</td>
</tr>
<tr>
<td>8</td>
<td>CALL SUBROUTINE</td>
</tr>
</tbody>
</table>

### Table B-28. Source program listing of subroutine STRA

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION</td>
</tr>
<tr>
<td>3</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>4</td>
<td>CALL TRANS</td>
</tr>
<tr>
<td>5</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

---

**Table B-27. Source program listing of subroutine STIMAT**

```
SUBROUTINE STIMAT, INCOE, ILAK, IE, CN, B, N, NO, NET, NET, NET, NET, NET

DIMENSION W(13,13), M(13,13), V(13,13), V(13,13), V(13,13)
DIMENSION M(13,13), V(13,13), V(13,13), V(13,13)

CONTINUE
RETURN
END
```

**Table B-28. Source program listing of subroutine STRA**

```
SUBROUTINE STRAIDS(4,5,13,9,5,5)
DIMENSION W(13,13), M(13,13), V(13,13), V(13,13), V(13,13)
CONTINUE
RETURN
END
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Code</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>To compute stresses in a quadrilateral element</td>
<td>SUBROUTINE STRESSI</td>
<td>C (STRESSI)</td>
</tr>
<tr>
<td>27</td>
<td>32-149</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table B-30. Source program listing of subroutine TRANSPIAI

<table>
<thead>
<tr>
<th>Line</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE TRANSPIAI B J1 J2</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION A1(J1) A2(J2)</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

### Table B-31. Source program listing of subroutine TRAN

<table>
<thead>
<tr>
<th>Line</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE TRAN(JF GDIR M3)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION A1(J1) A2(J2)</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

### Table B-32. Source program listing of subroutine TRIM

<table>
<thead>
<tr>
<th>Line</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE TRIM(JF GDIR M3)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION A1(J1) A2(J2)</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

### Table B-33. Source program listing of subroutine TRM

<table>
<thead>
<tr>
<th>Line</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE TRM(JF GDIR M3)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION A1(J1) A2(J2)</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

### Table B-34. Source program listing of subroutine UNIT

<table>
<thead>
<tr>
<th>Line</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE UNIT(JF GDIR M3)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION A1(J1) A2(J2)</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

### Table B-35. Source program listing of subroutine VECT

<table>
<thead>
<tr>
<th>Line</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE VECT(JF GDIR M3)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION A1(J1) A2(J2)</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
</tr>
</tbody>
</table>
References


