

AD-A122 336

RPSL1D (A ONE-DIMENSIONAL VERSION OF REPSIL)(U) ARMY
ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERDEEN
PROV. J D WORTMAN NOV 82 ARBRL-MR-83221

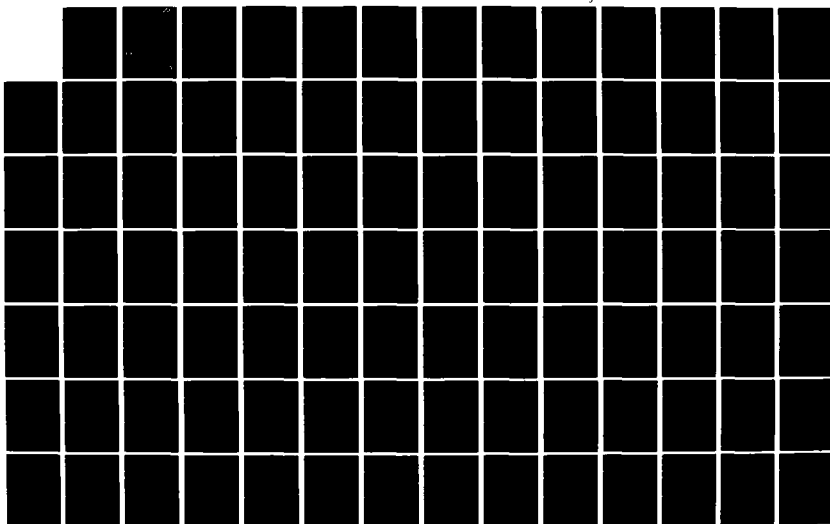
172

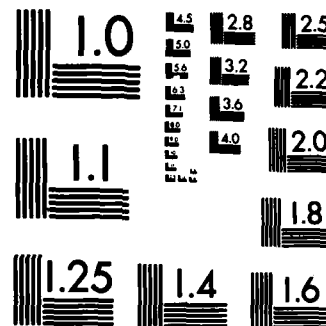
UNCLASSIFIED

SBI-AD-F300 132

F/G 19/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A122336

12

AD-E300136

AD

Memorandum Report ARBRL-MR-03221

RPSL1D

(A ONE-DIMENSIONAL VERSION OF REPSIL)

John D. Wortman

November 1982



US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

DTIC FILE COPY

DTIC
ELECTE
DEC 10 1982
S **D**
D

82 1 19 07

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report is prohibited.

Additional copies of this report may be obtained
from the National Technical Information Service,
U. S. Department of Commerce, Springfield, Virginia
22161.

The findings in this report are not to be construed as
an official Department of the Army position, unless
so designated by other authorized documents.

*The use of trade names or manufacturers' names in this report
does not constitute endorsement of any commercial product.*

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Memorandum Report ARBRL-MR- 03221	2. GOVT ACCESSION NO. AD-A122336	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RPSL1D (A ONE-DIMENSIONAL VERSION OF REPSIL)		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John D. Wortman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Ballistic Research Laboratory ATTN: DRDAR-BLT Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N/A
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Research & Development Command US Army Ballistic Research Laboratory (DRDAR-BL) Aberdeen Proving Ground, MD 21005		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 147
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Director US Army Materiel Systems Analysis Activity Aberdeen Proving Ground, MD 21005		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Structural Response Code Elastic Plastic Behavior Finite Difference Method Impact Pressure Loading Computer Program Axial Symmetric Shells Large Deformation Slab Symmetric Shells Thin Kirchhoff Shells Laterally Symmetric Beams		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) RPSL1D is a one-dimensional version of REPSIL, a computer program which calculates the large plastic deformation of thin Kirchhoff shells. The REPSL1D program is more efficient than REPSIL for axially symmetric shells (shells of revolution) and slab symmetric shells. It has been used most frequently for deformation calculations in laterally symmetric beams. The numerical algorithm differs from that of REPSIL as recorded in Reference 2 in that external, artificial, points are used to define boundary conditions,		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

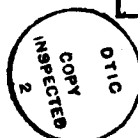
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

and that forces are computed between mesh points as well as at mesh points. This gives a better simulation of clamped ends and a more compact, more stable system of difference equations. The report gives instructions for using RPSL1D. Listings of the program and the companion plotting program are given. Two example problems are discussed with tabulated input and output. A number of options, primarily for beams, are tabulated and discussed.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
1. INTRODUCTION	5
2. MODELS	6
2.1. Axially Symmetric Shells	7
2.2. Slab Symmetric Shells	9
2.3. Laterally Symmetric Beams	10
3. CHANGES FROM REPSIL	11
3.1. Outline of the Development of RPSL1D	12
3.2. New Difference Operators	12
3.3. External Points and End Conditions	15
3.3.1. Hinged Ends	16
3.3.2. Clamped Ends	16
3.3.3. Symmetric Ends	16
3.3.4. Free Ends	17
3.4. Optional Gaussian Integration	17
3.5. Convergence and Stability.	19
4. USER SUBROUTINES	21
4.1. INGEOM	21
4.1.1. INGEOM for a Flat Plate or Beam	23
4.1.2. INGEOM for a Cylindrical Shell	24
4.2. PRESS	24
4.2.1. PRESS for Constant Pressure	25
4.2.2. PRESS for Pressure as a Function of Time	25
4.2.3. PRESS for Two Phase Pressure Decay in a Cylinder	25
4.3. INVEL	27
4.3.1. INVEL for Normal Velocity at Specified Points	27
4.3.2. INVEL for Constant Lateral Velocity	28
4.3.3. INVEL for Beam Vibration in the Fundamental Transverse Mode	28
4.4. INNORM	29
4.5. ENDFRC	30
4.5.1. ENDFRC for One or Both Ends Free	31
4.5.2. ENDFRC for Tabular Forces	31
4.5.3. ENDFRC for Tabular Forces (Modified)	31

TABLE OF CONTENTS (Continued)

	Page
5. OPTIONAL CHANGES	32
5.1. PLOTP (Plot Functions of Time)	32
5.2. SHR3/1 (Compute Moment, Shear and Axial Forces)	32
5.3. MSQSVS (Plot Moment, Shear and Axial Forces)	33
5.4. APLFRC (Applied Forces at Beam Ends)	33
5.5. EAPFRC (Energy for APLFRC)	35
5.6. EROD (Erosion at a Beam End)	36
5.7. BSTRS (Stress-Strain Option for Beams)	37
6. INPUT	39
APPENDICES	
A. LIST OF EQUATIONS	45
B. DEFINITIONS OF FORTRAN VARIABLES	53
C. LISTING OF RPSL1D	71
D. PLOTTING PROGRAM FOR RPSL1D	97
E. EXAMPLES	105
F. SOME ALTERNATE USER SUBROUTINES	141
LIST OF SYMBOLS	145
DISTRIBUTION LIST	147

1. INTRODUCTION

RPSL1D is a spin off of the FORTRAN IV computer program REPSIL^{1,2} developed at the Ballistic Research Laboratory to treat large, transient, elastoplastic deformations in thin shells. Thin shell programs, such as REPSIL, replace the essentially three-dimensional geometry of the deforming shell with a two-dimensional reference surface and some assumptions about deformation through the thickness. The REPSIL family of programs is based on theory that restricts them to thin Kirchhoff shells with negligible rotary inertia. The coding restricts the programs to shells of uniform thickness with clamped or hinged edges and a plane of symmetry. The RPSL1D version further restricts the program to geometries where the reference surface of the shell is a function of one variable and a symmetry assumption.

Three geometries have been incorporated: axially symmetric shells, slab symmetric shells, and laterally symmetric beams. The standard REPSIL (with some modifications) could be used for these problems, but RPSL1D is much faster and uses about one-fifth as much memory. Further, RPSL1D results should be more accurate; the numerical derivatives with respect to the symmetric variable are replaced by exact derivatives, and spacing for the other variable can be made smaller without unreasonable demands on machine storage or time.

The simpler one-dimensional model makes it easier to experiment with alternate methods of computing. Many changes and additions have been tried. Some of these have been incorporated into RPSL1D, as tabulated in Appendix C, and others discussed in Section 5 and Appendices E and F may be useful in the future.

The purpose of this report is to record RPSL1D, as cataloged in the CDC CYBER system at BRL, to record some user subroutines and useful options, to serve as a user's guide, and to give the basic information for further modifications. This is not a complete guide; the user should also obtain a copy of Reference 2, the user's manual for REPSIL. An understanding of the theory given in Reference 1 is also very helpful. The symbols in the present report, which are sometimes used without sufficient explanation, are those of Reference 1.

Sections 2 and 3 and the list of equations in Appendix A should bridge the gap between REPSIL and RPSL1D. In an effort to keep this report within reasonable bounds, we will not repeat theory given in Reference 1 nor will we repeat much of the information given in the REPSIL user's manual, Reference 2.

¹J. M. Santiago, "Formulation of the Large Deflection Shell Equations for Use in Finite Difference Structural Response Computer Codes," U.S. Army Ballistic Research Laboratories, Report No. 1571, Feb. 72.

²J. M. Santiago, H. L. Wisniewski, N. J. Huffington, Jr., "A User's Manual for the REPSIL Code," U.S. Army Ballistic Research Laboratories, Report No. 1744, Oct. 74.

The RPSL1D user must prepare input as described in Section 6 and must supply proper "user" subroutines as described in Section 4. He may retain the user subroutines with the cataloged program listed in Appendix C, replace them with copies of user subroutines listed in Appendices E and F, or replace them with his own subroutines.

For some particular applications it seems inevitable that some modifications are needed or desirable. A number of possibly useful options are described in Section 5 and listed in Appendices E and F.

The program, as stored in UPDATE form in the CDC 7600, is listed in Appendix C. A complete list of FORTRAN names used in RPSL1D is given in Appendix B along with their definition and some explanation. Two example problems are discussed in Appendix E.

The accessory program for plotting output from RPSL1D is discussed and listed in Appendix D.

The main purpose of this report is to serve as a guide to the use and possible modification of the RPSL1D program. The author's interest in this program is as a programmer. Hence, there is a bias in the presentation toward the frequent use of FORTRAN names and formulation, and the trivial details of programming, at the expense of physical meaning.

2. MODELS

The first one-dimensional version of REPSIL treated axially symmetric shells. This was modified to handle slab symmetric shells, and this in turn changed to treat symmetric beams.

The position, $\underline{x}^*$, of any point in the shell may be expressed in terms of a point, $\underline{x}$, on the reference surface, the normal, $\underline{n}$, to the reference surface at $\underline{x}$, and the distance from the reference surface, ζ :

$$\underline{x}^* = \underline{x} + \zeta \underline{n} . \quad (1)$$

Points on the reference surface are expressed in Cartesian coordinates as

$$\underline{x} = Y^j \underline{i}_j \equiv Y^1 \underline{i}_1 + Y^2 \underline{i}_2 + Y^3 \underline{i}_3 , \quad (2)$$

where the $\underline{i}_j$ are the orthonormal basis for vectors in 3-space. The Y^j , and hence other dependent parameters, are functions of the material (Lagrangian) parameters, ξ^1 and ξ^2 , and of time, t .

$$\underline{x} = \underline{x}(\xi^1, \xi^2, t) = Y^j(\xi^1, \xi^2, t) \underline{i}_j . \quad (3)$$

*The symbols of Reference 1 will usually be followed in this report. The tilde below a letter denotes a vector (e.g. $\underline{x}$).

Our characterization of axial and slab symmetry is through specialization of the vector $\underline{x}$ and the material coordinates ξ^1 and ξ^2 . The basic specializations are given in this section. A summary of equations is given in Appendix A.

The type of model is prescribed for RPSL1D by control parameters assigned in the user's subroutine INGEOM.

2.1 Axially Symmetric Shells

An axially symmetric shell is a shell of revolution (see Figure 2.1) such as a circular cylinder or the surface of a truncated right circular cone. A shell of revolution is the surface generated by revolving a plane curve (actually two parallel curves so the shell has thickness) about a line, called the axis of revolution, in its plane. Every point of the revolving curve describes a circle whose center is on the axis. For our shell, the curve must be smooth and must not intersect the axis in the region of interest.

We study the motion of the shell by following the motion of the curve formed by the intersection of the shell's reference surface and a plane through the axis. We do not allow twisting in our model, so the initial motion, and any loading, must be in this plane. Because of the axial symmetry, the internal forces are all in this plane.

Let $\xi^1 = \theta$, $\xi^2 = \xi$, and introduce the orthonormal base vectors $(\hat{\underline{x}}, \hat{\underline{\theta}}, \hat{\underline{k}})$ which vary with angle θ . These are related to $(\underline{i}_1, \underline{i}_2, \underline{i}_3)$ by the relations

$$\hat{\underline{x}} \equiv \hat{\underline{x}}(\theta) = \cos\theta \underline{i}_3 + \sin\theta \underline{i}_1, \quad (4)$$

$$\hat{\underline{\theta}} \equiv \hat{\underline{\theta}}(\theta) = -\sin\theta \underline{i}_3 + \cos\theta \underline{i}_1, \quad (5)$$

$$\hat{\underline{k}} \equiv \underline{i}_2. \quad (6)$$

Then,

$$\underline{x} \equiv \underline{x}(\theta, \xi, t) = R \hat{\underline{x}} + Z \hat{\underline{k}} \quad (7)$$

where

$$R \equiv R(\xi, t) = \{(Y^3)^2 + (Y^1)^2\}^{1/2}, \quad (8)$$

$$Z \equiv Z(\xi, t) = Y^2. \quad (9)$$

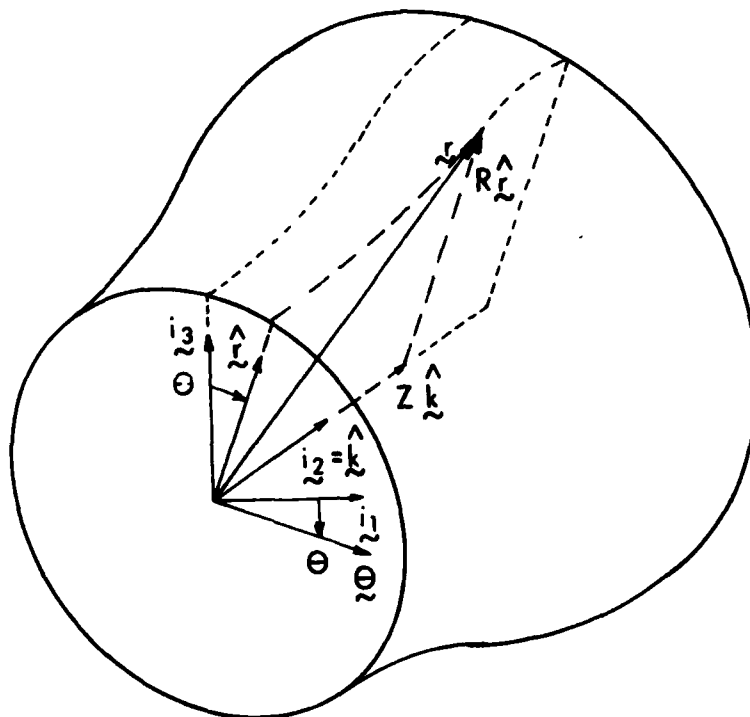


Figure 2.1 - Model for axial symmetry showing the two orthonormal bases $(\hat{i}_1, \hat{i}_2, \hat{i}_3)$ and $(\hat{x}, \hat{\theta}, \hat{k})$ with origin at the center of end 1 of the shell, and a vector $\underline{x}$ to the surface of the axisymmetric shell.

The same formulation holds for every angle θ . Since evaluation is needed at only one angle, we naturally choose $\theta = 0$. Then,

$$\begin{aligned} \underline{x}(0, \xi, t) &= R(\xi, t) \hat{x}(0) + Z(\xi, t) \hat{k} \\ &\equiv Y^3(0, \xi, t) \hat{i}_3 + Y^2(0, \xi, t) \hat{i}_2 . \end{aligned} \quad (10)$$

To test this new program for axial symmetry, the same problem (an initially cylindrical shell subjected to a uniform impulsive load) was simulated with both programs. Both programs were run undamped for 200 time cycles and then damped. Both self terminated on time cycle 262. The results were very close. The maximum difference noted in displacement was less than $2 \cdot 10^{-8}$. Since the displacement was about unity, $2 \cdot 10^{-8}$ was also the relative difference in displacement. Strains and displacement increments are very sensitive to program differences. The largest relative difference noticed in either was 10^{-5} . (There were relative differences of about 0.003 in all the printed energy values. This was because the circumference of the right circular cylinder was a circle in the axial symmetric program but was a 48-sided regular polygon in the standard REPSIL.)

2.2 Slab Symmetric Shells

Slab symmetric shells may be characterized by the displacement of any point being a function of its initial distance, ξ^2 , from end 1 (see Figure 2.2).

Let $Y^1 = \xi^1$, $Y^2 = Y^2(\xi^2, t)$, and $Y^3 = Y^3(\xi^2, t)$. Then,

$$r(\xi^1, \xi^2, t) = \xi^1 \hat{i}_1 + Y^2(\xi^2, t) \hat{i}_2 + Y^3(\xi^2, t) \hat{i}_3. \quad (11)$$

To utilize the formulation and programming for axial symmetry, we renamed variables

$$\xi^2 = \xi, \quad Y^3 = R, \quad \text{and} \quad Y^2 = Z, \quad (12)$$

and confined evaluation to the plane $Y^1 = 0$. Then,

$$\begin{aligned} \mathbf{r} &\equiv \mathbf{r}(0, \xi, t) = R(\xi, t) \hat{i}_3 + Z(\xi, t) \hat{i}_2 \\ &\equiv Y^3(\xi, t) \hat{i}_3 + Y^2(\xi, t) \hat{i}_2. \end{aligned} \quad (13)$$

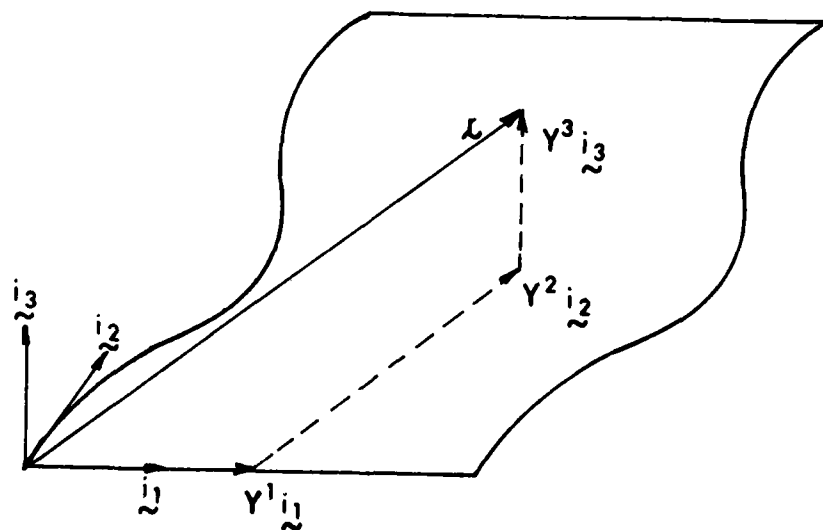


Figure 2.2 - Model for Slab Symmetry.

To maintain slab symmetry, the loading must not contain any component in the i_1 direction.

For both axial and slab symmetry, Z is a distance along the i_2 axis. For axial symmetry, R is the radial distance from the axis of rotation. For slab symmetry, R is the perpendicular distance from the $\gamma^3 = 0$ plane.

The initial slab symmetric coding was also tested by comparing with results from a standard REPSIL run. (A minor coding change was needed to insert cyclic boundary conditions in REPSIL.) The results were nearly identical. The maximum relative errors in any variables were of the order of 10^{-13} .

2.3 Laterally Symmetric Beams

One principal difference between slab symmetric shells and beams is in the change of concept from area to unit length (e.g. pressure, i.e. force/unit area, for shells and force/unit length for beams.) The original beam considered was a narrow rectangular beam. It was treated as a special case of slab symmetry with a uniaxial stress-strain relation replacing the biaxial one. The uniaxial stress-strain was imposed by using a new subroutine BMSTRS, instead of subroutine STRESS, for beams. By setting the width, $\Delta\xi^1$, to unity, the concept of surface area became length without any further changes.

Treating beams that were not rectangular was more involved. It required changes in fact as well as in concept. The main change was for integrals through the thickness. The simplest form of numerical integration through the thickness is used in the standard REPSIL:

$$\int_{-h/2}^{+h/2} f(\zeta) d\zeta \sim \Delta\zeta \sum_{k=1}^L f(\zeta_k) . \quad (14)$$

The thickness, h , is divided into L layers of equal thickness, $\Delta\zeta = h/L$, and the function to be integrated is evaluated at each ζ_k , where ζ_k is in the center of the k 'th layer. The parameter ζ is measured from the reference surface in the center of the shell.

For a beam, an integral through the thickness is replaced by an integral over the cross-sectional area. When multiplied by the proper metric, this becomes the quantity per unit surface area. For a beam, the desired result is the quantity per unit length. For a rectangular beam we could simply multiply the previous result by the width of the beam, $\Delta\xi^1$:

$$\int_A f(\zeta) dA = \Delta\xi^1 \int_{-h/2}^{+h/2} f(\zeta) d\zeta \sim \Delta\xi^1 \Delta\zeta \sum_{k=1}^L f(\zeta_k) . \quad (15)$$

This has been replaced by a more general form for laterally symmetric but not necessarily rectangular beams:

$$\int_A f(\zeta) dA \sim \sum_{k=1}^L w_k f(\zeta_k) \quad (16)$$

where the ζ_k are not necessarily evenly spaced and w_k is the area associated with ζ_k . The ζ_k and w_k should be assigned so that $\sum w_k$ is the cross-sectional area of the beam, $\sum \zeta_k w_k = 0$ (so the reference surface is the neutral surface for bending), and $\sum (\zeta_k)^2 w_k$ is the area moment of inertia of the cross section. (This form of integral approximation is also suitable for Gaussian integration, which is preferred for shells; see Section 3.4.)

See the discussion of INGEOM for a beam (Section 4.1.1) for a particular example. The user subroutine INGEOM for beams must also supply the upper limit of the beam, $ZU \equiv \zeta_u$. Subroutine START assigns $\zeta_l \equiv ZL = ZU - h$. For beams, ZU and ZL merely locate positions, upper and lower respectively, at which surface strains are computed for output. The shell versions set $ZU = h/2$, $ZL = -h/2$, and ZU is used in START to compute a number of program constants.

Results from RPSL1D for small elastic displacements for vibrations initiated in the fundamental transverse modes for clamped, hinged, and cantilever beams are in good agreement with simple beam theory. An example of buckling from lateral forces on the ends also agreed with theory.

3. CHANGES FROM REPSIL

The principal reason for programming a one-dimensional version of REPSIL was to have a simple vehicle for testing possible changes before implementing them in REPSIL. However, the program proved very useful for a variety of problems. Many changes have been coded. Some of these were purely experimental, but most of them were for particular problems. Most of the changes have been superseded or were not considered general enough to transfer to the CDC computer. Some of the changes have been incorporated as part of RPSL1D, as listed in Appendix C, and some are retained as options.

In this section we will outline the changes made to REPSIL to create the RPSL1D program stored in the CDC 7600 as of September 1978, and describe the new boundary condition implementation, the new differencing scheme, Gaussian integration, and the new stability criteria.

3.1 Outline of the Development of RPSLID

Initially, the REPSIL program, listed in Reference 2, was changed to a one-dimensional code for axially symmetric shells (see Section 2.1). A change of notation was introduced to eliminate errors.

This program was modified to include slab symmetry (Section 2.2).

This was then modified to include beams with a vertical axis of symmetry (Section 2.3).

The method of imposing boundary conditions was changed by inserting artificial, external points as described in Section 3.3, and the divided difference approximations for partial derivatives were changed (see Section 3.2) to a form using only central differences. The system was made more compact by evaluating at mid-mesh points.

The code was changed to permit symmetry at either end and permit a free end at end 2. This change was intended for beams only. (An option called APLFRC, see Section 5.4, which permits forces to be applied at either or both ends of beams is available.)

Coding was added to monitor maximum deflection and extreme surface strains, and to print these extremes along with the surface strain prints.

The computation of surface strains was revised. The original surface strain computation was a copy of that described on pages 50 through 53 of Reference 2 except that a_{12} and b_{12} were zero and ζ_u and ζ_l replace $h/2$ and $-h/2$. This was changed so that the interpolation in the non-zero covariant components of strain, ϵ_{11} and ϵ_{22} ($\epsilon_{\alpha\beta} = \frac{1}{2}(\Sigma\Delta a_{\alpha\beta} \pm h \Sigma\Delta b_{\alpha\beta})$) was replaced by interpolation in the changes in the covariant components of the middle surface metric, $\Sigma\Delta a_{11}$ and $\Sigma\Delta a_{22}$, and of the second fundamental tensor, $\Sigma\Delta b_{11}$ and $\Sigma\Delta b_{22}$. Component $\Sigma\Delta a_{22}$ is computed between mesh points.

Optional Gaussian integration through the thickness of shells was introduced.

Several minor, mostly cosmetic, changes were made.

The coding was transferred from the BRLESC computer to the CDC 7600 (with more minor changes) and stored in the UPDATE form.

3.2 New Difference Operators

The primary purpose of the REPSIL family of programs is to solve the equations of motion. This is done by a sequence of equations (see Reference 1, pp. 99-106) that carry the solution forward one time step.

Difference approximations of derivatives with respect to the material coordinates, ξ^1 and ξ^2 , are required at two points in this sequence of equations. First, we need approximations to the first and second derivatives of position, $\underline{x}$, and the incremental displacement, $\Delta \underline{x}$, to define the local geometry. Finally, we need approximations for first partial derivatives of the components of the stress resultant tensor, $\hat{N}^{*\alpha}$, and second partial derivatives of the normal vector components of the bending resultant tensor, $\hat{M}^{*\alpha\beta}$, for the equations of motion.

The REPSIL equations of motion for a mesh point, as given in Reference 1, are

$$\Delta \underline{u}^+ = \Delta \underline{u} + \frac{(\Delta t)^2}{\gamma^*_0} \left[\frac{\partial^2 \hat{M}^{*\alpha\beta}}{\partial \xi^\alpha \partial \xi^\beta} + \frac{\partial \hat{N}^{*\alpha}}{\partial \xi^\alpha} + \underline{P}^* \right]. \quad (17)$$

The change in the displacement of the point from time t to $t + \Delta t$ is $\Delta \underline{u}^+$. The position at time $t + \Delta t$ is $\underline{x}^+ = \underline{x} + \Delta \underline{u}^+$, where $\underline{x}$ is the position at time t . The mass per unit initial middle surface area, γ^*_0 , is constant for a given point. (The term in brackets has units of force per initial unit area.) $\underline{P}^* = -P a^{\frac{1}{2}} \underline{n}$ is the force per initial unit area along the normal, $\underline{n}$, due to a pressure P . (a is the determinant of the covariant metric tensor.) The one-dimensional assumptions remove a number of the terms in the equations of motion. There are no terms in the $\underline{i}_1$ direction, $\hat{M}^{*12} = \hat{M}^{*21} = 0$, and the derivatives with respect to ξ^1 are simple functions which are computed explicitly. The equations of motion become

$$\Delta \underline{x}^+ = \Delta \underline{x} + \frac{(\Delta t)^2}{\gamma^*_0} \left[\frac{\partial^2 \hat{M}^{*22}}{(\partial \xi^2)^2} + \frac{\partial \hat{N}^{*2}}{\partial \xi^2} + \underline{P}^* - (\hat{M}^{*11} \cdot \underline{i}_2 + \hat{N}^{*1} \cdot \underline{i}_1) \underline{i}_2 \right], \quad (18)$$

$$\underline{x}^+ = \underline{x} + \Delta \underline{x}^+. \quad (19)$$

(The transformation to this equation is outlined in Appendix A.) The vectors $\hat{M}^{*22}$, $\hat{N}^{*2}$, and their derivatives and $\underline{P}^*$ and $\hat{M}^{*11}$ may all have components in both the $\underline{i}_3$ and $\underline{i}_2$ directions. $\hat{N}^{*1}$ has a component in the $\underline{i}_1$ direction only.

For slab symmetry and beams, the term $(\hat{M}^{*11} \cdot \underline{i}_2 + \hat{N}^{*1} \cdot \underline{i}_1)$ is deleted. For beams, γ^*_0 is mass/unit length and the term in brackets is force/unit length.

The difference scheme used in the standard REPSIL has two defects which have occasionally been troublesome: the use of forward, or backward, differences at fixed edges may introduce a slowly developing instability which cannot be eliminated by a reduction in the time step. (The presence of this instability could be detected by an eigenvalue analysis of the linearized equations of motion. We have carried out this analysis for a few cases with very small grids. Unfortunately, for most useful meshes, the eigenvalue analysis would take longer than the REPSIL program.) The other defect is that the differences for first derivatives are not compact enough. (As an experiment with the initial RPSLID code, alternate points in a slab symmetric run were at rest or given an initial axial velocity. The points initially at rest remained at rest. The motion of the moving points produced a reasonable solution.) These two defects have been removed from RPSLID (and from a modified version of REPSIL) by modeling fixed boundaries with artificial external points and by introducing new difference operators which compute and use forces between mesh points as well as at mesh points.

Three sets of central difference operators are used in RPSLID to compute first and second differences with respect to the Lagrangian coordinate $\xi = \xi^2$. To simplify the notation, replace ξ^2 by y and use the notation $f_y(n)$ and $f_{yy}(n)$ to represent the approximations to $\partial f / \partial \xi^2$ and $\partial^2 f / (\partial \xi^2)^2$, respectively, evaluated at the n 'th mesh point.

The first set of difference operators is used in subroutine GRAD to approximate derivatives of position components, and derivatives of components of displacement increments, at mesh points:

$$f_y(n) = \{f(n+1) - f(n-1)\} / (2 \Delta y) , \quad (20)$$

$$f_{yy}(n) = \{f(n+1) - 2 f(n) + f(n-1)\} / (\Delta y)^2 . \quad (21)$$

The second set of difference operators is used in subroutine GRAD to approximate the same derivatives at mesh midpoints:

$$f_y(n+\frac{1}{2}) = \{f(n+1) - f(n)\} / \Delta y , \quad (22)$$

$$f_{yy}(n+\frac{1}{2}) = \frac{1}{2} \{f(n+2) - f(n+1) - f(n) + f(n-1)\} / (\Delta y)^2 . \quad (23)$$

The third set of central difference operators is used in subroutine MOTION to compute terms for the equations of motion:

$$f_y(n) = \{f(n+\frac{1}{2}) - f(n-\frac{1}{2})\} / \Delta y , \quad (24)$$

$$f_{yy}(n) = \{f(n+1) - 2 f(n) + f(n-1)\} / (\Delta y)^2 . \quad (25)$$

These difference operators, which are used at fixed edges as well as at internal points, are all central difference operators. The resulting system of difference equations is stable with the proper choice of time step, Δt . Unfortunately, when the program is run for a beam with a free end at end 2, say $n = N$, backward difference operators have to be used at the free end. In GRAD, for a free end, we use:

$$f_y(N) = \{3 f(N) - 4 f(N-1) + f(N-2)\} / (2 \Delta y) , \quad (26)$$

$$f_{yy}(N) = \{2 f(N) - 5 f(N-1) + 4 f(N-2) - f(N-3)\} / (\Delta y)^2 , \quad (27)$$

$$f_y(N-\frac{1}{2}) = \{f(N) - f(N-1)\} / \Delta y , \quad (\text{as before}) \quad (28)$$

$$f_{yy}(N-\frac{1}{2}) = \frac{1}{2} \{3 f(N) - 7 f(N-1) + 5 f(N-2) - f(N-3)\} / (\Delta y)^2 . \quad (29)$$

In MOTION, for a free end, we have:

$$f_y(N) = \{f(N) - f(N-\frac{1}{2})\} / (\Delta y / 2) , \quad (30)$$

$$f_{yy}(N) = \{2 f(N) - 5 f(N-1) + 4 f(N-2) - f(N-3)\} / (\Delta y)^2 . \quad (31)$$

It is also necessary to approximate $\hat{N}^{*2} = -\partial \hat{M}^{*22} / \partial \xi$ at the free end. This is done in subroutine RESULT with

$$\hat{N}^{*2}(N) = \frac{1}{2} \{4 \hat{M}^{*22}(N-1) - \hat{M}^{*22}(N-2)\} / \Delta \xi , \quad (32)$$

because

$$\hat{M}^{*22}(N) = 0.0 . \quad (33)$$

3.3 External Points and End Conditions

The end conditions for a fixed or symmetric end is imposed in RPSL1D by the initial positioning and the incrementing of a fictional external point. The following notation is used to explain this:

$\underline{x}$ A vector to a point on the reference surface at time t .
($\underline{x} = R \underline{i}_3 + Z \underline{i}_2$)

$\underline{x}^+$ A vector to a point on the reference surface at time $t + \Delta t$.

$\Delta \underline{x}$ $\underline{x}^+ - \underline{x}$

$\underline{x}_B$ A vector to the end point.

$\underline{x}_I$ A vector to the internal mesh point adjacent to $\underline{x}_B$.

$\underline{x}_E$ A vector to the external point adjacent to $\underline{x}_B$.

$\underline{n}_B$ The unit normal to the reference surface at $\underline{x}_B$.
($\underline{n}_B = n_r \underline{i}_3 + n_k \underline{i}_2$)

$\underline{t}$ The unit tangent to the reference surface at $\underline{x}_B$.
($\underline{t} = -n_k \underline{i}_3 + n_r \underline{i}_2$)

The initial position of external points is established in subroutine BOUNDR and the corresponding increments are inserted in BOUNDU. The initial normal at clamped end points is supplied in a new user subroutine INNORM.

3.3.1 Hinged Ends. The initial position of an artificial external point at a hinged end is

$$\underline{x}_E = 2 \underline{x}_B - \underline{x}_I . \quad (34)$$

The increment is

$$\Delta \underline{x}_E = -\Delta \underline{x}_I . \quad (35)$$

We also set

$$\Delta \underline{x}_B = 0 . \quad (36)$$

These inserts impose $\underline{x}_B$ is constant and $\hat{M}^{*22} = 0$ at the boundary. These are the conditions for a hinged end as given in Reference 1.

3.3.2 Clamped Ends. The initial position of an artificial external point adjacent to a clamped end is

$$\underline{x}_E = \underline{x}_I - 2\{(\underline{x}_I - \underline{x}_B) \cdot \underline{i}\} \underline{i} . \quad (37)$$

The corresponding increment is

$$\Delta \underline{x}_E = \Delta \underline{x}_I - 2\{\Delta \underline{x}_I \cdot \underline{i}\} \underline{i} . \quad (38)$$

We also impose

$$\Delta \underline{x}_B = 0 . \quad (39)$$

These inserts force $\underline{x}_B$ and $\underline{n}_B$ to remain constant. These are the conditions for a clamped end given in Reference 1. (The components of $\underline{n}_B$ are set in a user subroutine INNORM and never changed for a clamped end.)

3.3.3 Symmetric Ends. The treatment of symmetric ends is simply a reduction of that for symmetric edges in REPSIL. A symmetry plane is assumed perpendicular to the basis vector $\underline{i}_2$. The initial position of an external point adjacent to a symmetry end is

$$\underline{x}_E = \underline{x}_I - 2\{(\underline{x}_I - \underline{x}_B) \cdot \underline{i}_2\} \underline{i}_2 . \quad (40)$$

The change in $\underline{x}_E$ at each time step is

$$\Delta \underline{x}_E = \Delta \underline{x}_I - 2\{\Delta \underline{x}_I \cdot \underline{i}_2\} \underline{i}_2 . \quad (41)$$

These are the same equations used for clamped ends with $\underline{i}_2 = \underline{i}$.

The external points at symmetry ends represent real points. They are not artificial like those outside the fixed ends. Just as in the standard REPSIL, we need to know membrane and bending resultant components ($\hat{N}^{*2}$ and $\hat{M}^{*22}$, respectively) outside the symmetry end. We have

$$(\hat{M}^{*22})_I \equiv \hat{M}^{*22} n_I = (FM_R^{22})_I i_3 + (FM_K^{22})_I i_2 , \quad (42)$$

defined at x_I , and

$$(\hat{N}^{*1})_I = (FN_\theta^1)_I i_1 , \quad (43)$$

$$(\hat{N}^{*2})_I = (FN_R^2)_I i_3 + (FN_K^2)_I i_2 , \quad (44)$$

defined between x_I and x_B . The external values needed are

$$(FM_R^{22})_E = (FM_R^{22})_I , \quad (45)$$

$$(FM_K^{22})_E = -(FM_K^{22})_I , \quad (46)$$

$$(FN_\theta^1)_E = (FN_\theta^1)_I , \quad (47)$$

$$(FN_R^2)_E = -(FN_R^2)_I , \quad (48)$$

$$(FN_K^2)_E = (FN_K^2)_I . \quad (49)$$

3.3.4 Free Ends. One of the first problems posed for the RPSL1D code was for a cantilever I-beam. To model this beam a free end was needed. The conditions for a free end (see Ref. 2, Eq. 6.52) are imposed by

$$\hat{M}^{*22} = 0.0 , \quad \hat{N}^{*2} = -\partial \hat{M}^{*22} / \partial \xi . \quad (50)$$

Regrettably, it is necessary to use non-central differences to approximate this derivative, and other derivatives, at free ends (see Section 3.2).

The cataloged program permits end 2 to be free but not end 1. (The optional coding called APLFRC, see Section 5.4, allows either end to be free with, or without, applied forces. The applied forces, or absence of forces, are supplied by a user's subroutine called ENDFRC.)

3.4 Optional Gaussian Integration

The original RPSL1D programming approximated integration through the thickness of shells by Riemann sums assuming equal layers about evenly spaced points. This is replaced with Gaussian integration only if IGAUSS is 1 and IB is not 1. (These control parameters are set in the user subroutine INGEOM, see Section 4.1.) IB = 1 specifies a beam. We do not use Gaussian integration with beams. We use Riemann sums with unevenly spaced points and "weights" equal to their associated cross-sectional area (as explained in Section 2.3). For the Riemann sums with L equal layers we choose

$$\zeta_k = \frac{1}{2} h (1 - (2k - 1) / L) , \quad (k = 1, 2, \dots, L) , \quad (51)$$

$$\Delta\zeta = h/L , \quad (52)$$

$$\zeta_u = h/2 , \quad (53)$$

$$\zeta_\ell = - h/2 , \quad (54)$$

and approximate integration through the thickness by

$$\int_{\zeta_\ell}^{\zeta_u} f(\zeta) d\zeta \sim \sum_{k=1}^L f(\zeta_k) \Delta\zeta , \quad (55)$$

where h is the shell thickness, ζ_u and ζ_ℓ are the distances from the reference surface (the middle surface) to the upper and lower surfaces, respectively. L is the number of equally spaced layers, and ζ_k is the center of the k 'th layer.

For Gaussian integration, we replace the ζ_k by

$$\zeta_k = \frac{1}{2} h x_{kL} , \quad (56)$$

set

$$w_k = w_{kL} , \quad (57)$$

and use the approximation

$$\int_{\zeta_\ell}^{\zeta_u} f(\zeta) d\zeta \sim (h/2) \sum_{k=1}^L f(\zeta_k) w_k . \quad (58)$$

where the x_{kL} and w_{kL} are from tables for Gaussian integration (e.g. Table 25.4, Handbook of Mathematical Functions, National Bureau of Standards). The x_{kL} are zeros of the Legendre Polynomial

$$P_L(x) = \frac{1}{2^L L!} \frac{d^L (x^2 - 1)^L}{(dx)^L} \quad (59)$$

Two sets of integrals are affected: the integral for strain energy in subroutine STRESS, and the integrals for force and moment resultants in subroutine RESULT.

The Riemann sums for the force and moment resultants at a point N are of the form

$$I_N = TA \sum_k f_N(\zeta_k) , \quad (60)$$

where

$$TA = \Delta\zeta = h/L . \quad (61)$$

The $f_N(\zeta_k)$ signifies the value of the appropriate function at mesh point N and integration station ζ_k , where the ζ_k are in the center of evenly spaced layers.

All that is needed to change these sums to Gaussian integration is to select the proper ζ_k and W_k , replace $f_N(\zeta_k)$ by $W_k f_N(\zeta_k)$, and set TA to $h/2$. For example,

$$I_N = (h/2) \sum_{k=1}^K W_k f_N(\zeta_k) . \quad (62)$$

(The form is the same for approximations of integrals over cross-sectional area for beams. This is done with Riemann sums but with unevenly spaced ζ_k . The ζ_k , and W_k equal to the area associated with ζ_k , are selected in the user subroutine INGEOM, and TA is set to 1.0 in START.)

3.5 Convergence and Stability

Some examination of convergence and stability is essential for any numerical solution of differential equations. The usual method of examining convergence is to find solutions for a particular problem using an increasing number of meshes until these solutions converge. Such tests with the REPSIL family of programs have shown remarkable insensitivity to the number of meshes for problems with large plastic deformation. In those cases tested it seems that the maximum deflection, usually at the center, decreases slightly with additional meshes, and the strain and deflection increase slightly near the edges. For elastic vibration problems, more meshes may be needed to permit high vibration modes. The choice of mesh, and hence the accuracy, must ultimately be decided by the user.

Numerical solutions of differential equations by explicit methods such as that used with the REPSIL programs will be unstable, because any introduced error grows, unless the increment in the independent variable, Δt in our case, is small enough. Considerable time and effort have been expended in studying the stability for the various REPSIL codes. Even a cursory discussion of the stability analysis is beyond the scope of this report. The following bounds, based on a linear analysis of the non-linear difference equations of motion for RPSL1D assuming the central difference operators, are used in RPSL1D:

$$\Delta t_M = \Delta\xi/C , \quad (63)$$

$$\Delta t_B = \frac{1}{2}(\Delta\xi)^2/(CC_2) . \quad (64)$$

Here, Δt_M and Δt_B are the stable time step for the linearized membrane and bending equations of motion, respectively. C is the longitudinal wave speed.

$$C = (E/\rho)^{1/2}, \text{ for beams.} \quad (65)$$

$$C = \{E/\rho(1 - \nu^2)\}^{1/2}, \text{ for shells.} \quad (66)$$

Here, E is Young's modulus of Elasticity, ρ is mass density, and ν is Poisson's ratio. C_2 is the approximation of $\{\int_A \zeta^2 dA / \int_A dA\}^{1/2}$.

For beams,

$$C_2 = \{\sum w_k (\zeta_k)^2 / \sum w_k\}^{1/2}. \quad (67)$$

For shells h units thick, using Riemann sums with L equal layers,

$$C_2 = \{(1 - 1/L^2) h^2/12\}^{1/2}. \quad (68)$$

For shells with Gaussian integration the approximation is exact:

$$C_2 = (h^2/12)^{1/2}. \quad (69)$$

The program chooses a two digit truncated value 0.95 times the minimum of Δt_M and Δt_B . This is used instead of the input Δt on input card 3 if it is smaller than the input value given, or if the input value is negative or zero.

The term $\Delta \xi$ used by RPSL1D in these equations is the DETA2 supplied by the user through subroutine INGEOM. The equations produce critical values for Δt_M and Δt_B if $\Delta \xi$ is the minimum distance between mesh points. Both Δt_M and Δt_B have been shown to be close to the limits for stability for runs with fixed ends and a linearly elastic stress-strain relation. The user should examine all results for numerical instability. This shows up first as rapid vibrations in strain plots. The inclusion of plasticity tends to damp these vibrations, but they can still be detected. The user should be particularly critical if there is a free or forced end (non-central differences are used), or if the structure is compressed. The stability formulae are based on the assumption that $\Delta \xi$ is the minimum distance between mesh points. The factor 0.95 permits some compression, but only up to 5% compression if Δt_M is critical, or 2.5% if Δt_B is critical. We have not been able to prove the stability of the RPSL1D solutions when non-central difference equations are used, but we have not detected instability when a time step that satisfies the stability criteria is used.

4. USER SUBROUTINES

There are four, or possibly five, subroutines in RPSL1D which the user must either write, or select from existing versions. These "user" subroutines describe the geometry and loading. Three of them, INGEOM, PRESS, and INVEL, are one-dimensional versions of subroutines described in Reference 2. Subroutine INNORM was added to supply the unit normal to the reference curve at clamped ends. Subroutine ENDFRC must be included, to supply the forces and moments on the ends of a beam, if the optional coding APLFRC, which assumes this type of loading (see Section 5.4), is added to the RPSL1D program. Some of these subroutines are always called and some are called under control of input parameters. Any of the user supplied subprograms may require input (this is under the control of the user). The input for them follows all the other input (input is discussed in Section 6). The user subroutines are listed below in the order they are called, with the conditions for calling.

INGEOM is called from START shortly after the start of each run.

INNORM is called immediately after INGEOM from START.

PRESS is called from the initiation part of the main program, RPSL1D, if $LOAD \neq 0$. It will be called each time cycle thereafter as long as $LPRESS \geq NCYCLE$.

INVEL is called in the initiation sequence in RPSL1D if $LOAD \leq 0$.

ENDFRC is called from RESULT each cycle, except cycle 0, if the APLFRC optional coding is included in the program.

Information is brought to each of the user subroutines, and carried back to the rest of the program, through COMMON. The RPSL1D program is intended to be stored in the UPDATE format. All the COMMON is contained in a COMDECK called MAIN which is listed at the beginning of Appendix C. This COMDECK is inserted into a subroutine with the directive *CALL MAIN. Of course, information can also be passed from one user subroutine to another through labeled COMMON. The parameters in the following list are commonly needed in the user subroutines.

NMESH The number of mesh intervals in the length.

LAYER The number of layers in a cross section.

N1B = 2 The index of the mesh point at end 1.

N2B = N1B + NMESH The index of the mesh point at end 2.

N1V and N2V The range of indices of mesh points that move.

4.1 INGEOM

Subroutine INGEOM is called from START. It supplies the initial geometry and some control parameters. The parameters supplied are listed below.

IB This is the dominant control parameter. It should be set to 1 or 0. if $IB > 0$, a beam is assumed. If $IB \leq 0$, it is not.

RADIUS This parameter separates radial symmetry from slab symmetry. If $RADIUS > 0$, radial symmetry is assumed and the parameter IS is set to 1 in START. Otherwise, IS is set to 2 in START. (RADIUS is superseded by IB. If $IB > 0$, RADIUS is set to 0 in START.)

DETA1 ($\Delta\xi^1$) This parameter is a multiplier for energy computations. It should be the width of the reference surface. (If $IB = 1$, DETA1 is set to 1.0 in START. This makes the concept of surface area and beam length interchangeable.)

DETA2 ($\Delta\xi$) This is the distance between mesh points in the material coordinate ξ (ξ^2 of Ref. 1, and η^2 of Ref. 2). (The stability criteria assume that DETA2 is the minimum distance between mesh points, and for some other coding it is tacitly assumed that ξ is arc length from end 1.)

R(N), Z(N) ($N1B \leq N \leq N2B$) The position vector of the N'th mesh point on the reference curve is $\underline{r}_N = R(N) \underline{i}_3 + Z(N) \underline{i}_2$. The point $\underline{r}_N$ corresponds to the material coordinate point $\xi = (N - 1) DETA2$.

ETAD2 and ETAG2(I), $I=1, NSTRN$ The previously read parameters ETAD2 and ETAG2(I) may need to be converted to the same units as DETA2.

IGAUSS Set IGAUSS = 1 for Gaussian integration (see Section 3.4).

Extra Parameters for Beams

If $IB = 1$, the cross section of the beam must be described. It is defined as follows:

ZU (z_u) The distance from the reference surface to the upper surface of the beam.

ZETA(K), $K=1, LAYER$ (z_k) These are the positions of integration stations through the thickness relative to the reference surface, the neutral axis for bending of the beam. ($ZETA(1) < 0$, $ZETA(LAYER) > 0$)

W(K), $K=1, LAYER$ (w_k) The cross-sectional area associated with $ZETA(K)$.

It is assumed that

$$\sum_k w_k = \text{Area of the beam's cross section.}$$

$$\sum_k (z_k)^2 w_k = \text{Moment of inertia.}$$

$$\sum_k \zeta_k w_k = 0 \quad (\text{i.e. } \zeta = \text{zero on the neutral surface}).$$

4.1.1 INGEOM for a Flat Plate or Beam. This is the INGEOM stored with RPSL1D (see Appendix C).

If the run is for slab symmetric motion of an initially flat plate, there is only one input card. The input is simply length, SLABL, and one-half the width, SLABW, as it is for a similar standard REPSIL subroutine. First, the subroutine reads SLABL, SLABW, and IB with FORMAT (2E10.4,I5). The program sets

```

RADIUS = 0.0 ,
DETA1 = 2.0 * SLABW ,
DETA2 = SLABL/FLOAT(NMESH) ,
R(N) = 0.0
Z(N) = (N - N1B) DETA2   } N1B ≤ N ≤ N2B ,
IGAUSS = 1.

```

If $IB \leq 0$, the responding surface is an initially flat plate. The subroutine is finished.

If $IB > 0$, data describing the cross section of the beam is needed. First, the subroutine reads ZU with FORMAT (3E10.2) and prints SLABL and ZU. It then reads WIDTHK, DZETAK, and ZETA(K) with FORMAT (3E10.2), sets $W(K) = \text{WIDTHK} * \text{DZETAK}$, and writes K, WIDTHK, DZETAK, ZETA(K), and W(K) for $K = 1, 2, \dots, \text{LAYER}$.

The following table is an example for a 6-layer subdivision of the cross section of a solid, right-circular, cylindrical rod of unit radius:

K	WIDTHK	DZETAK	ZETA(K)
1	1.032494026	1/3	-0.8253794582
2	1.717575183	1/3	-0.4952276749
3	1.962319769	1/3	-0.1650758916
4	1.962319769	1/3	0.1650758916
5	1.717575183	1/3	0.4952276749
6	1.032494026	1/3	0.8253794582

To produce this table we first arbitrarily chose layers of equal thickness, $\text{DZETAK} = 1/3$. Then, we found WIDTHK so that $W(K) = \text{DZETAK} * \text{WIDTHK}$ would equal the area of the circular cross section in that layer. Then we set $Z_K = K/3 - 7/6$ as a tentative location in the center of each layer. Finally we found $\text{ZETA}(K) = c Z_K$ where $c^2 \sum (Z_K)^2 W(K) = \pi/4$, the area moment of inertia for a circle of unit radius.

Similarly, for a rectangular beam with unit thickness and width w:

K	WIDTHK	DZETAK	ZETA(K)	Z_K
1	w	1/6	-0.4225771	-5/12
2	w	1/6	-0.2535462	-3/12
3	w	1/6	-0.0845154	-1/12
4	w	1/6	0.0845154	1/12
5	w	1/6	0.2535462	3/12
6	w	1/6	0.4225771	5/12

(Choosing integration stations and weights for Gaussian integration, see Section 3.4, would probably be better for a rectangular beam.)

The choice of layers of equal thickness is convenient and reasonable for these two examples, but not for all cases. With an I-beam, for example, it would be more reasonable, and convenient, to approximate each flange with a layer and put four layers in the web. If the beam is not symmetric vertically it may be difficult to assign values that produce the correct area moment of inertia without altering the area, or the location of the centroid.

4.1.2 INGEOM for a Cylindrical Shell. The subroutine INGEOM for a cylinder is very simple (see the tabulation in Appendix F). The input is CYLL, RADIUS with FORMAT (2E12.6), where CYLL is the cylinder length and RADIUS is the radius to the midsurface. The subroutine sets

$$DETA1 = 2 \pi \text{ RADIUS}$$

$$DETA2 = \text{CYLL}/\text{FLOAT}(\text{NMESH})$$

$$IB = 0$$

$$R(N) = \text{RADIUS}$$

$$Z(N) = (N - 2) * DETA2$$

If Gaussian integration is desired, IGAUSS = 1.

4.2 PRESS

Subroutine PRESS supplies the pressure, $P(N)$, at each mesh point that moves ($N1V \leq N \leq N2V$). (this is actually the pressure difference for shells and the normal force per unit length for beams, but we will continue to call it pressure.) The sign of $P(N)$ was chosen so that a positive $P(N)$ tends to crush a cylinder or push a flat plate down.

Most of the PRESS subroutines for RPSL1D have been in two sections. The first section reads input and makes other preliminary calculations the first time the subroutine is called. The other part supplies the $P(N)$ each time the subroutine is entered.

PRESS is called in the initial portion of RPSL1D if LOAD = 0. It is called at the start of each succeeding time cycle in RPSL1D if LPRESS $\geq$ NCYCLE. Also, as long as LPRESS $\geq$ NCYCLE, the $P(N)$ are multiplied by $a^{1/2}$ in subroutine DGEOM. This multiplication changes the $P(N)$ from pressure to force per unit initial area. If the program is continued beyond cycle NCYCLE = LPRESS, the values of $P(N)$ do not change. This means the force per unit initial area is fixed, the pressure varies with the change in area. (At the initiation of

damping, when NCYCLE = MDAMP, subroutine DAMP sets LPRESS and all P(N) to zero. This must be changed if the program is to continue with both pressure and damping.) NCYCLE is the count of time cycles. LOAD, LPRESS, and MDAMP are input parameters. If LOAD = 0 and LPRESS = 0, PRESS will never be called and all P(N) will be zero.

One of the weak points of using deformation codes to simulate reality is in accurately describing dynamic pressure loading. It is difficult to obtain accurate, repeatable measurements for enough points even with well-controlled experiments. Such measurements tend to be very oscillatory due to both reality and instrumentation. Even assuming a completely accurate pressure record at some point, the effective pressure on a thin shell would be different because of its rapid reaction. The best we can hope for is an approximate simulation that will lead to useful results. Three examples are discussed.

4.2.1 PRESS for Constant Pressure. This is the PRESS subroutine with the program in Appendix C. In the initial entrance for each run, the constant pressure, PO, is read with FORMAT (E12.6). During each entrance it sets $P(N) = PO$ for $N1V \leq N \leq N2V$.

4.2.2 PRESS for Pressure as a Function of Time. This subroutine, called LINPRS, is tabulated with Example 1 in Appendix E. Pressure is a tabular function of time. On the first entrance, this subroutine reads pairs of time and pressure data into TPR(I) and PPR(I) with FORMAT (2E10.3) until $TPR(I) < 0$. Then IPRESS, the number of data points in the table, is set to I-1. (It is assumed that $IPRESS \leq 50$, $TPR(1) = 0.0$, $TPR(I+1) > TPR(I)$, and $TPR(IPRESS)$ is larger than any time to be reached.

Linear interpolation is used at every entry to compute the pressure, PO, corresponding to time, TIME. The program then sets $P(N) = PO$ for $N1V \leq N \leq N2V$.

4.2.3 PRESS for Two Phase Pressure Decay in a Cylinder. This subroutine, labeled PRESS of 3/10/76, is listed in Appendix F. It simulates the pressure in a cylinder from an explosion at its center. This rather involved subroutine is included because it demonstrates a number of the common features of pressure subroutines that simulate the loading from explosions. The unbalanced pressure P_N at point N is zero until the shock front arrives at time TA_N with peak pressure PO_N . The pressure then decays rapidly. In this subroutine it decays rapidly, exponentially until it reaches a pressure $\bar{P}$ which we call the quasi-static pressure. It then continues to decay exponentially at a much slower rate.

$$\begin{aligned} P_N &= 0.0 && \text{if } t < TA_N, \\ P_N &= PO_N e^{-\alpha(t - TA_N)} && \text{if } TA_N \leq t \leq TB_N, \\ P_N &= \bar{P} e^{-\beta(t - TB_N)} && \text{if } TB_N \leq t, \end{aligned} \quad (70)$$

where α and β are the decay factors and $\bar{P}$ is the quasi-static pressure which is reached at point N at time TB_N found from

$$\bar{P} = PO_N e^{-\alpha(TB_N - TA_N)} .$$

An outline of the subroutine is given below. First, the following input cards are read on the initial entry.

<u>Content</u>	<u>FORMAT</u>
$\bar{P}$, β , (PID(I), I=1,6)	(2E10.3,6A10)
CZ, CT, CP, TD	(4E10.3)
α , ---, ISM	(E10.3,10X,I10)
(ZSI(I), TSI(I), PSI(I), I=1,ISM)	(3E10.3)

Here,

$\bar{P}$ is the quasi-static pressure.

β is the quasi-static decay factor.

PID(I) is an alphanumeric title.

CZ, CT, CP, and TD are conversion factors to transform data to the desired units (see below).

α is the shock pressure decay factor.

ISM is the number of entries in the following table (ISM $\leq$ 50).

ZSI(I) is the distance from the middle, end 1, of the cylinder.

TSI(I) is the shock arrival time at ZSI(I).

PSI(I) is the peak shock pressure at ZSI(I).

These data cards are read, and their images printed, during the initial entrance. The tabular data is then converted to the desired units with the replacement formulas

$$ZSI(I) = CZ * ZSI(I) , \quad (71)$$

$$TSI(I) = CT * (TSI(I) - TD) , \quad (72)$$

$$PSI(I) = CP * PSI(I) . \quad (73)$$

This converted table is also printed. For $N1V \leq N \leq N2V$, $TS(N) \equiv TA_N$ and $PS(N) \equiv PO_N$ are found by linear interpolation at distance $Z(N) = (N - 2) * DETA2$.

(Here is an example where the material parameter ξ is assumed to be length from end 1. Symmetry is assumed at end 1 where $N = N1B = N1V = 2$.)
 $TSB(N) = TB_N$ is found from

$$TSB(N) = TS(N) + \{ \ln(PS(N)/\bar{P}) \} / \alpha . \quad (74)$$

The program then forms $PSB(N)$ and transforms $PS(N)$ to a more convenient form, as follows:

$$PS(N) = PS(N) e^{\alpha TS(N)} \quad (75)$$

$$PSB(N) = \bar{P} e^{\beta TSB(N)} \quad (76)$$

This completes the preparation. For every entry, the subroutine computes $e^{-\alpha t}$ and $e^{-\beta t}$ and then, for $N1V \leq N \leq N2V$,

$$\begin{aligned} P(N) &= 0.0 && \text{if } t < TS(N) , \\ P(N) &= -e^{-\alpha t} PS(N) \equiv -PO_N e^{-\alpha(t - TA_N)} && \text{if } TS(N) \leq t \leq TSB(N) , \\ P(N) &= -e^{-\beta t} PSB(N) \equiv -\bar{P} e^{-\beta(t - TB_N)} && \text{if } TSB(N) < t . \end{aligned} \quad (77)$$

4.3 INVEL

Subroutine INVEL supplies the initial velocity of moving mesh points. The velocity of all mesh points is preset to zero before INVEL is called. This subroutine sets

$$DR(N) = \dot{R} , \quad (78)$$

$$DZ(N) = \dot{Z} , \quad (79)$$

where the velocity at point N is

$$\underline{V} = \dot{R} \underline{i}_3 + \dot{Z} \underline{i}_2 . \quad (80)$$

INVEL is called during the initiation sequence from the main program, RPSL1D, if parameter LOAD is less than or equal to zero. The components of initial velocity are changed to displacement increments by multiplication with Δt in RPSL1D.

4.3.1 INVEL for Normal Velocity at Specified Points. The INVEL subroutine catalogued with RPSL1D (see Appendix C) is a one-dimensional version of the INVEL tabulated in the User's Manual (Reference 2). This subroutine assumes all initial velocities are normal to the surface. The input cards are:

<u>Cards</u>	<u>FORMAT</u>
--,--,NI,NF,VR,NV	(10X,2I5,E12.6,I5)
--, N, V	(5X,I5,E12.6)

NI and NF are the minimum and maximum mesh numbers for an array of points to be given initial velocity VR. NV is the number of mesh points to receive initial velocity different from zero or VR. N is the mesh number for a point receiving velocity V.

Note! This subroutine was written to use input cards for the standard REPSIL. The mesh numbers NI, NF, and N are all relative to the mesh point at end 1 being number 1. In RPSL1D, the mesh point at

end 1 is numbered N1B. (Subroutine START sets N1B = 2.) Therefore, this INVEL subroutine adds N1B-1 (i.e. one) to all input mesh numbers.

The first input card is read and the program sets

$$DR(N) = -SNR(N) * VR , \quad (81)$$

$$DZ(N) = -SNK(N) * VR , \quad (82)$$

for $NI + N1B - 1 \leq N \leq NF + N1B - 1$, where the unit normal at mesh point N is

$$\underline{n} = SNR(N) \underline{i}_3 + SNK(N) \underline{i}_2 .$$

(Notice the reversal in sign. VR is assumed to be an inwardly directed velocity, possibly from impulsive loading, on the outside surface.)

The program then reads NV cards of the second type (NV may be zero), replaces N by N + N1B-1, and sets

$$DR(N) = -SNR(N) * V , \quad (83)$$

$$DZ(N) = -SNK(N) * V . \quad (84)$$

The subroutine prints the initial velocities assigned. These initial velocities are later converted to displacement increments by multiplying them by Δt .

4.3.2 INVEL for Constant Lateral Velocity. This subroutine has been used to simulate a rod striking a wall. The changes in the catalogued INVEL to produce this are tabulated with Example 2 in Appendix E.

The constant velocity, VR, is read with effective FORMAT (20X,E12.6) and the subroutine sets

$$DZ(N) = -VR \quad (85)$$

for $N1V \leq N \leq N2V$.

4.3.3 INVEL for Beam Vibration in the Fundamental Transverse Mode. A number of INVEL subroutines have been coded for beams. They were either very simple or very specialized. As examples of simple codes for initially flat beams, consider the following for the fundamental mode of free transverse vibrations. The transverse velocity of any point x at time t may be written

$$\dot{R}(x,t) = A X(x) \cos(\omega t) \quad (86)$$

for these examples. The velocity at time $t = 0$ is

$$\dot{R}(x,0) = A X(x) . \quad (87)$$

Here ω is the fundamental frequency, $X(x)$ is a function of the end conditions, and the parameter A is arbitrary. We assume evenly spaced points to find x . (These formulas are similar to those given by Enrico Volterra, E. C. Zachmanoglou, Dynamics of Vibration, Charles E. Merrill Books, Inc., Columbus, Ohio.)

Cantilever Beam:

$$x = 1.875104069 (N - N1B)/(N2B - N1B) , \quad (88)$$

$$DR(N) = A \{ \cosh(x) - \cos(x) - 0.7340955137 [\sinh(x) - \sin(x)] \} . \quad (89)$$

Simply Supported Beam:

$$x = \pi(N - N1B)/(N2B - N1B) , \quad (90)$$

$$DR(N) = A \sin(x) . \quad (91)$$

Clamped Beam:

$$x = 4.730040744862 (N - N1B)/(N2B - N1B) , \quad (92)$$

$$DR(N) = A \{ \cosh(x) - \cos(x) - 0.9825022145 [\sinh(x) - \sin(x)] \} . \quad (93)$$

4.4 INNORM

Subroutine INNORM is a new user-supplied subroutine that assigns the unit normal at clamped ends. It is called in START immediately after the call of INGEOM. The version of INNORM catalogued with the program in Appendix C is the only one used up to now. It produces the correct normal if the shape of the reference curve at the end can be written as a quadratic equation in ξ through the last three end points.

The components of the unit normal, $\underline{n}$, are defined in terms of partial derivatives of the position vector $\underline{r}$.

$$\underline{r} = R \underline{i}_3 + Z \underline{i}_2 , \quad (94)$$

$$R2 = \partial R / \partial \xi , \quad (95)$$

$$Z2 = \partial Z / \partial \xi , \quad (96)$$

$$D = \{ (R2)^2 + (Z2)^2 \}^{1/2} , \quad (97)$$

$$SNR(N) = Z2/D , \quad (98)$$

$$SNK(N) = -R2/D , \quad (99)$$

$$\underline{n} = SNR(N) \underline{i}_3 + SNK(N) \underline{i}_2 . \quad (100)$$

The partial derivatives, $R2$ and $Z2$, at the clamped ends are approximated by non-central differences through three end points. N is $N1B$ or $N2B$, depending on which end is clamped.

$$\left(\text{e.g. } \frac{\partial R}{\partial \xi} \right)_{N=N1B} = \{ 3R(N2B) - 4R(N2B - 1) + R(N2B - 2) \} / (2\Delta\xi) . \quad (101)$$

4.5 ENDFRC

Subroutine ENDFRC is a user subroutine that supplies the forces and moments imposed at the ends of a beam. It is really a part of the option called APLFRC (Section 5.4) which was programmed to utilize this type of loading. The subroutine is not needed unless the APLFRC option is included in the program and either IBCE1 = 4 or IBCE2 = 4. With the APLFRC option, IBCE1 = 4 signals an applied force and moment at end 1 and IBCE2 = 4 signals an applied force and moment at end 2. A free end is a special case for which the applied force and moment are zero.

Subroutine ENDFRC is called from subroutine RESULT when $N = N1B$ if IBCE1 = 4, or when $N = N2B$ if IBCE2 = 4 and IBCE1 $\neq$ 4. Since RESULT is not called in the initial sequence, there are no initial forces on the ends.

When ENDFRC is called, the force components and moments are supplied. The force at end 1 is

$$EFR1 \hat{i}_3 + EFZ1 \hat{i}_2 . \quad (102)$$

The moment is EM1. Similarly at end 2, the force components and moment are EFR2, EFZ2, and EM2, respectively. Figure 4.5 and the following list both explain the sign conventions.

- EFR1 > 0 decreases $R(N1B)$,
- EFZ1 > 0 decreases $Z(N1B)$,
- EM1 > 0 decreases $R(N1B)$ and increases $R(N1B + 1)$,
- EFR2 > 0 increases $R(N2B)$,
- EFZ2 > 0 increases $Z(N2B)$,
- EM2 > 0 increases $R(N2B - 1)$ and decreases $R(N2B)$.

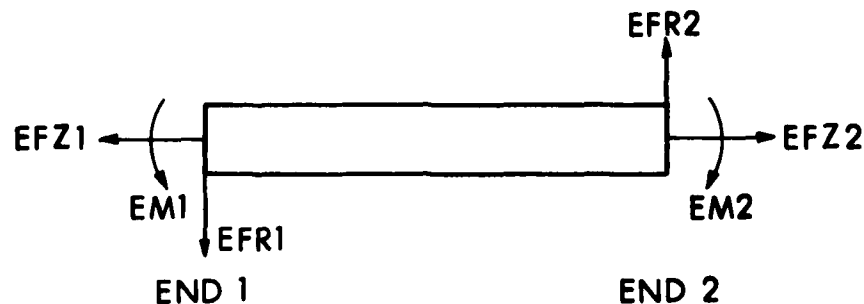


Figure 4.5 Sign Convention for Applied Forces on the End of a Beam.

The program outside ENDFRC does not change these terms. Constant values may be inserted in the initial entry and need not be reset. It does no harm to insert values at an end that is not forced ($IBCE1 \neq 4$ or $IBCE2 \neq 4$); the program ignores them.

4.5.1 ENDFRC for One or Both Ends Free. The initial entry into this ENDFRC sets $EFR1$, $EFZ1$, $EM1$, $EFR2$, $EFZ2$, and $EM2$ to zero. These are never changed. The APLFRC option then imposes a free end at end 1 if $IBCE1 = 4$, and/or a free end at end 2 if $IBCE2 = 4$. This subroutine is tabulated in Appendix F.

4.5.2 ENDFRC for Tabular Forces. This ENDFRC subroutine assumes that end 2 is free and the forces on end 1 are supplied by tables. On the initial entry this subroutine reads three tables. Each has a heading giving the number of entries and an alphanumeric title.

<u>Cards</u>	<u>FORMAT</u>
NFZPTS, (LABELV(I), I=1,7)	(I10,7A10)
(EFZT(I), TMFZ(I), I=1,NFZPTS)	(2E15.7)
NFTHS, (LABELV(I), I=1,7)	(I10,7A10)
(EFHT(I), TMTH(I), I=1,NFTHS)	(2E15.7)
NFMUS, (LABELV(I), I=1,7)	(I10,7A10)
(EFMUT(I), TMMU(I), I=1,NFMUS)	(2E15.7)

The first table is the horizontal force component, $EFZ1$, as a function of time. The other two tables are of parameters θ and μ , respectively, as functions of time.

The initial entrance into the subroutine reads and prints the input and sets $EFZ2 = EFR2 = AM2 = 0$. On each entry, the subroutine finds $EFZ1$, θ , and μ as functions of time by linear interpolation. It then sets

$$EFR1 = \mu EFZ1, \quad (103)$$

$$EM1 = -ZU \{ \sin(\theta) EFZ1 + [1 - \cos(\theta)] EFR1 \}. \quad (104)$$

4.5.3 ENDFRC for Tabular Forces (Modified). This subroutine is like the previous one except that the force on end 1 is assumed relative to the tangent, $\underline{i}_1$, at end 1,

$$\underline{i} = SNK(N1B) \underline{i}_3 - SNR(N1B) \underline{i}_2, \quad (105)$$

rather than the horizontal direction, $\underline{i}_2$. The force components for that subroutine are replaced by

$$EFR1 = EFR1 * SNR(N1B) - EFZ1 * SNK(N1B), \quad (106)$$

$$EFZ1 = EFR1 * SNK(N1B) + EFZ1 * SNR(N1B). \quad (107)$$

A slightly modified version of this is tabulated with example 2 in Appendix E.

5. OPTIONAL CHANGES

As has been stated before, it seems that any particular project for RPSL1D requires some changes in the program. Ideally, these changes are restricted to the user subroutines, discussed in the previous section, which describe the geometry and loading. In this section we will briefly discuss the seven sets of optional coding that are used with the two examples in Appendix E. They are of varying utility. The plotting option PLOTP is usually wanted. It was not added to the catalogued program because it is very short and usually requires some additional coding to select the functions of time to be plotted. The APLFRC option which permits loading the ends of beams by force and moment has been used a number of times. If it is used, the option SHR3/1 must be included and the energy computation EAPFRC should be added.

It is hoped that these options will be useful and serve as a guide for introducing others in the future.

5.1 PLOTP (Plot Functions of Time)

This coding was originally inserted to plot the force/unit initial area (force/unit initial length, for beams) at prescribed mesh points as functions of time. However, any function of time may be plotted by inserting coding that stores that function's value into P(N) ($N_{2B} < N < 104$) each cycle, and requesting a plot for that P(N) on the new input card, card 11a. If the PLOTP option is in the deck, this input card must be included (see Section 6). It selects NNPE ($0 \leq NNPE \leq 9$) values of N for which plots of P(N) are to be made.

The changes to RPSL1D are rather trivial: some new terms are inserted into COMMON and MAIN, the new input card is read and its image is printed in START, and the requested P(N) are stored in the plot data file in PDATA. More extensive changes were required in the plotting program. These changes are included in the plotting program stored in file RPSL1DPLOT except for two orders (the optional coding) which read from the plot data file and activate P(N) plotting. This option is used with both examples in Appendix E. (In example 1 the total large core memory requested for plotting is greater than 131,071 words. Hence, the LCM = I parameter is specified on the FTN control card.)

5.2 SHR3/1 (Compute Moment, Shear and Axial Forces)

This option is for beams only. It computes the moment, M, at each mesh point and computes the shear force, V, and the axial force, Q, between mesh points. The following equations were extracted from Reference 1 by Dr. Santiago.

$$M = a \hat{M}^{22} \quad (108)$$

$$Q = a (\hat{Q}^{22} - b_2^2 \hat{M}^{22}) \quad (109)$$

$$V = (1/a^{1/2}) \partial(a \hat{M}^{22}/\partial \xi^2) \quad (110)$$

The RPSL1D analog of these equations is

$$AMS(N) = A22 * F22 \quad (111)$$

$$QS(N) = A22 * (Q22 - BM22 * F22) \quad (112)$$

$$VS(N) = \{AMS(N+1) - AMS(N)\}/(DETA2 * SRA) . \quad (113)$$

For convenience, the three new arrays are put in COMMON in COMDECK MAIN. The rest of this option is in subroutine RESULT. A table of N, AMS(N), VS(N), QS(N), A22, B22, Q22, and F22 may be printed at cycles prescribed by coding inserted with the option.

5.3 MSQSVS (Plot Moment, Shear and Axial Forces)

This option was inserted to plot the moment, shear force, and axial force computed with option SHR3/1 on the same cycles that cross-sectional plots are made (i.e. whenever NCYCLE = NC3DP(I)). The only change in RPSL1D is one statement in PDATA that stores the three arrays AMS(N), QS(N), and VS(N) on the plot data file. The bulk of this option is inserted into the plotting program catalogued in file RPSL1DPLOT.

5.4 APLFRC (Applied Forces at Beam Ends)

This option was inserted to permit an applied vector force and moment at either or both ends of a beam. We will refer to this as a forced end. A free end is a special case with zero applied force and moment. It is assumed that the option SHR3/1 is included in the program.

The equation of motion, given in Reference 1, reduced to that of a beam with no external forces is

$$\Delta \underline{x}^+ = \Delta \underline{x} + \left\{ \hat{M}^{*22},_{22} + \hat{N}^{*2},_2 \right\} (\Delta t)^2 / (A^{1/2} \Gamma_0) , \quad (114)$$

where

$$\hat{M}^{*22} = a^{1/2} \hat{M}^{22} \underline{n} \quad (115)$$

and

$$\hat{N}^{*2} = a^{1/2} (\hat{Q}^{22} \underline{a}_2 + \Gamma_{22}^2 \hat{M}^{22} \underline{n}) \quad (116)$$

are vectors defining the bending and stress resultant terms. The notation $X_{,2}$ and $X_{,22}$ denotes the partial derivatives $\partial X / \partial \xi$ and $\partial^2 X / (\partial \xi)^2$. The mass per initial unit length for a beam is given by $A^{1/2} \Gamma_0 = A^{1/2} \rho \sum w_k$.

From the equations for shear force, axial force, and moment, respectively,

$$V = a^{\frac{1}{2}} (\hat{M}^{22},_2 + 2\Gamma_{22}^2 \hat{M}^{22} = (a \hat{M}^{22}),_2 / a^{\frac{1}{2}}, \quad (117)$$

$$Q = a (\hat{Q}^{22} - b_2^2 \hat{M}^{22}), \quad (118)$$

$$M = a \hat{M}^{22}, \quad (119)$$

and the relations

$$a^{\frac{1}{2}},_2 = \Gamma_{22}^2 a^{\frac{1}{2}}, \quad (120)$$

$$\underline{n},_2 = -b_2^2 \underline{a}_2, \quad (121)$$

$$\underline{a}_2 = a^{\frac{1}{2}} \underline{\tau}, \quad (122)$$

which are also taken from Reference 1, we can transform the vector

$$(\hat{M}^{*22},_{22} + \hat{N}^{*2},_2) \quad (123)$$

into the form

$$(V \underline{n} + Q \underline{\tau}),_2 \quad (124)$$

(The shear force and axial force, V and Q , are computed at all midmesh points, and the moment, M , is computed at all mesh points, except at forced ends, by option SHR3/1.) Vector $\underline{n}$ is the unit normal to the reference surface and $\underline{\tau}$ is the unit tangent.

$$\underline{n} = n_k \underline{i}_2 + n_r \underline{i}_3. \quad (125)$$

$$\underline{\tau} = n_r \underline{i}_2 - n_k \underline{i}_3. \quad (126)$$

The force vector

$$\underline{F} = (V \underline{n} + Q \underline{\tau}) \quad (127)$$

can be computed at all midmesh points if the moment $M = a \hat{M}^{22}$ is known at the mesh points.

The APLFRC option assumes a new user subroutine, ENDFRC (see Section 4.5), which supplies M and $\underline{F}$ at forced ends. In the following, we use the subscript N to denote quantities at the N 'th mesh point, $N = N1B$ at end 1 and $N2B$ at end 2, and subscript $N+\frac{1}{2}$ to denote midmesh quantities. If end 1 is forced,

$$\underline{F}_{N1B} = EFZ1 \underline{i}_2 + EFR1 \underline{i}_3, \quad M_{N1B} = EM1. \quad (128)$$

If end 2 is forced,

$$\underline{F}_{N2B} = EFZ2 \underline{i}_2 + EFR2 \underline{i}_3, \quad M_{N2B} = EM2. \quad (129)$$

For $N1B \leq N < N2B$,

$$\underline{F}_{N+1/2} = (V_{N+1/2} \underline{n}_{N+1/2} + Q_{N+1/2} \underline{t}_{N+1/2}) \quad (130)$$

Then, the force per unit length at $N = N1B$ and $N1B+1$ if end 1 is forced, and at $N = N2B$ and $N2B-1$ if end 2 is forced, is found by the difference formulas:

$$(\underline{F}, 2)_{N1B} = (\underline{F}_{N1B+1/2} - \underline{F}_{N1B})/(\Delta\xi/2), \quad (131)$$

$$(\underline{F}, 2)_{N1B+1} = (\underline{F}_{N1B+3/2} - \underline{F}_{N1B+1/2})/(\Delta\xi), \quad (132)$$

$$(\underline{F}, 2)_{N2B-1} = (\underline{F}_{N2B-1/2} - \underline{F}_{N2B-3/2})/(\Delta\xi), \quad (133)$$

$$(\underline{F}, 2)_{N2B} = (\underline{F}_{N2B} - \underline{F}_{N2B-1/2})/(\Delta\xi/2). \quad (134)$$

The first and last of these are not central differences. The form

$$(\underline{F}, 2)_N = (\underline{M}^{*22}, 22 + \underline{N}^{*2}, 2), \quad (135)$$

where only central differences are employed, is used at all other moving mesh points (see Section 3.2).

This option includes minor changes in MAIN, START, BOUNDR, and BOUNDU, and more extensive changes in GRAD, RESULT, and MOTION, plus the user supplied subroutine ENDFRC. Option SHR3/1 must also be used, and option EAPFRC should be.

5.5 EAPFRC (Energy for APLFRC)

In the section for APLFRC we described the option which allows forces to be applied at either end of a beam. The optional coding labeled EAPFRC, which computes the work on the beam by the applied forces and moments, is properly a part of the APLFRC option.

The work by a directed force, $\underline{F}$, applied at a point that moves a vector distance $\Delta \underline{r}$ is

$$\Delta W_F = \underline{F} \cdot \Delta \underline{r}. \quad (136)$$

The work by a moment of magnitude M applied while the normal vector changes by an angle $\Delta\theta$ is

$$\Delta W_M = \pm M \Delta\theta. \quad (137)$$

(The sign in this equation depends on the sign convention used.)

This work must be compatible with work from pressure loading (see Reference 2, pages 45 and 46), so an analogous method is used in RPSL1D. That is, the external work from time $t-\Delta t$ to time t , $\Delta W_F(t-\frac{1}{2}\Delta t)$ and $\Delta W_M(t-\frac{1}{2}\Delta t)$, will be the average of "work increments" computed at times $t-\Delta t$ and t . We compute:

$$\Delta W_F(t) = F(t) \cdot \{\Delta \underline{r}(t-\frac{1}{2}\Delta t) + \Delta \underline{r}(t+\frac{1}{2}\Delta t)\}/2, \quad (138)$$

$$\Delta W_F(t-\frac{1}{2}\Delta t) = \{\Delta W_F(t-\Delta t) + \Delta W_F(t)\}/2, \quad (139)$$

$$\Delta W_M(t) = -M(t) \{\Delta \theta(t-\frac{1}{2}\Delta t) + \Delta \theta(t+\frac{1}{2}\Delta t)\}/2, \quad (140)$$

$$\Delta W_M(t-\frac{1}{2}\Delta t) = \{\Delta W_M(t-\Delta t) + \Delta W_M(t)\}/2, \quad (141)$$

at the forced ends.

Except for some preliminary set up, all the computation is carried out in MOTION. The values of $\Delta W_F(t-\Delta t)$, $\Delta W_M(t-\Delta t)$, $\Delta \theta(t-\frac{1}{2}\Delta t)$, and the components of $\Delta \underline{r}(t-\frac{1}{2}\Delta t)$ at any forced end, is saved from the previous cycle. The components of $\Delta \underline{r}(t+\frac{1}{2}\Delta t)$ are already stored in the arrays DR(N) and DZ(N) by MOTION. We need to compute $\Delta \theta(t+\frac{1}{2}\Delta t)$. We define $\Delta \theta(t+\frac{1}{2}\Delta t)$ by

$$\sin \{\Delta \theta(t+\frac{1}{2}\Delta t)\} \underline{i}_1 = \underline{n}(t) \times \underline{n}(t+\Delta t). \quad (142)$$

The unit normal at time t is available,

$$\underline{n}(t) = \text{SNK}(N) \underline{i}_2 + \text{SNR}(N) \underline{i}_3, \quad (143)$$

but $\underline{n}(t+\Delta t)$ must be computed. This is computed from

$$\underline{a}_1 = \underline{i}_1, \quad (144)$$

$$\underline{a}_2 = \partial \underline{r}^+ / \partial \xi, \quad (145)$$

$$\underline{n}(t+\Delta t) = (\underline{a}_1 \times \underline{a}_2) / |\underline{a}_1 \times \underline{a}_2|, \quad (146)$$

where

$$\underline{r}^+ = \underline{r}(t) + \Delta \underline{r}(t+\frac{1}{2}\Delta t) \quad (147)$$

and $\underline{a}_2$ is approximated by a forward or backward difference.

5.6 EROD (Erosion at a Beam End)

This option depends on the applied force option, APLFRG, being in RPSL1D. The basis for our model is to assume an end mesh point, denoted by $N = \text{N1B}$, which moves according to forces on the true, nearby, eroding end. Whenever enough erosion has occurred, the end mesh point is shifted

to the next mesh point. (The eroding end cannot approach the midpoint of the first mesh too closely, or the difference equations at that end would be unreliable.)

We inserted a new subroutine, ERODE, which supplies the distance, DE, from the mesh point N1B to the eroding end and advances the endpoint indices when necessary. DE is zero initially. Whenever DE is greater than $\Delta\xi/4$, we subtract $\Delta\xi$ from DE and add 1 to N1B, N1V, and N1A (these are all indices for the end point). The difference equations for the shear in RESULT and force/unit length on the end point in MOTION were changed to reflect the position of the eroding end. The ENDFRC subroutine was altered to use quadratic interpolation to find the normal at the eroding end. (This last change was not really tested since the rod remained straight in our application.) Several minor inserts were made to permit reasonable plots.

This option is not as well tested as other features of RPSL1D. This brief discussion is included because the option is used with Example 2 (Appendix E). The motivation for Example 2 was to demonstrate that the RPSL1D program could reproduce experimental strain records at two points along a long rod penetrator, assuming a stress-strain curve, and a table of force on the end and position of the end as functions of time, all derived from the Karmann-Taylor theory³. From the two strain records, the density, and the cross-sectional area of the rod, one can derive the stress and force for any strain, and the velocity at which the strain is propagated. From this velocity, one has a linear relation between initial time and position of the generating force for each strain. Two quite different tables of erosion and force as functions of time were generated, one for a slow erosion rate and the other for a rapid erosion rate. Both were used as input for RPSL1D, and both produced strain records in fair agreement with the test data.

5.7 BSTRS (Stress-Strain Option for Beams)

The optional coding labelled BSTRS is a stress-strain routine for beams, which models the uniaxial curve more smoothly than the mechanical sublayer model by using more points. The mechanical sublayer model in the catalogued routine is generally superior, but the number of segments in the polygonal stress-strain curve must be restricted.

In the erosion problem discussed in Section 5.6, we wanted to compare the experimental strain-time response in a bar with output from RPSL1D. The strain response from RPSL1D at points along a bar agree closely with those predicted by the Karmann-Taylor theory. That is, the velocity at which a particular strain level is propagated down the bar is proportional to the square root of the slope of the stress-strain curve at that strain. Therefore, the strain-time records from RPSL1D at points away from the end of the rod, in response to an increasing force at the end, have near discontinuities corresponding to the corners of the polygonal stress-strain curve where there are discontinuities in the slope. To remove these discontinuities, one needs

³A.D. Gupta and J.D. Wortman, "An Eroding Long Rod Penetrator Model for Hard Target Penetration," *Proceedings of the Third ASCE/EMD Specialty Conference*, September 17-19, 1979, University of Texas, Austin, Texas, pp 714-717.

a smoother stress-strain curve. This was impractical with the mechanical sublayer model, so the BSTRS option was coded. If enough points are used, the uniaxial loading curve will be smooth enough. This option is not programmed for a strain rate effect nor for biaxial loading and there is no obvious simple way to include either of them.

The positive portion of the uniaxial stress-strain curve is entered as a table of points, (ϵ_i, σ_i) , with FORMAT (2E15.7), which terminates with $\epsilon_{\text{NEST}} < 0.0$ ($\text{NEST} \leq 100$). Then ϵ_{NEST} is replaced by 1000.0 (infinity) and σ_{NEST} is replaced by $\sigma_{\text{NEST}-1}$. As before, for the mechanical sublayer model, the point (ϵ_1, σ_1) is replaced by (SIGZ/E, SIGZ), the elastic limit. Also, as before, these are not engineering stress nor strain. If $(\epsilon_{ei}, \sigma_{ei})$ are engineering values,

$$\epsilon_i = \epsilon_{ei} (1 + \frac{1}{2} \epsilon_{ei}) , \quad (148)$$

$$\sigma_i = \sigma_{ei} (1 + \sigma_{ei}) . \quad (149)$$

These correspond to the mixed components ϵ_2^2, σ_2^2 of the strain and stress tensors used in RPSL1D. The negative portion of the curve is defined by $\sigma(-\epsilon) = -\sigma(\epsilon)$.

For each integration station at each mesh point and at each midmesh, we keep track of the strain, $\epsilon = \epsilon_2^2$, and the mean value of possible elastic strain variation, ϵ_m , in addition to $\sigma = \sigma_2^2$.

Initially we set $\epsilon = \epsilon_m = \sigma = 0.0$.

Assume we have ϵ, σ , and ϵ_m from time $t-\Delta t$ (call these ϵ^-, σ^- , and ϵ_m^-), and $\Delta\epsilon$, the change in ϵ from time $t-\Delta t$ to time t . We recognize five cases for $\epsilon_m^- \geq 0$. (For $\epsilon_m^- < 0$, we change the sign on $\epsilon^-, \sigma^-, \epsilon_m^-$, and $\Delta\epsilon$, and proceed as for $\epsilon_m^- \geq 0$. At the end, we change the signs on ϵ, σ , and ϵ_m .)

In all five cases we set

$$\epsilon = \epsilon^- + \Delta\epsilon . \quad (150)$$

Case	Conditions	Results
(1)	$\Delta\epsilon \geq 0$	$\sigma = \sigma^- + E \Delta\epsilon$
	$\epsilon \leq \epsilon_m^- + \epsilon_1$	$\epsilon_m = \epsilon_m^-$
(2)	$\Delta\epsilon < 0$	$\sigma = \sigma^- + E \Delta\epsilon$
	$\epsilon > \epsilon_m^- - \epsilon_1$	$\epsilon_m = \epsilon_m^-$

<u>Case</u>	<u>Conditions</u>	<u>Results</u>
(3)	$\Delta\epsilon > 0$ $\epsilon > \epsilon_m + \epsilon_1$	$\sigma = \sigma(\epsilon)$ $\epsilon_m = \epsilon - \epsilon_1$
(4)	$\Delta\epsilon < 0$ $\epsilon < \epsilon_m - \epsilon_1$ $\epsilon \geq -\epsilon_1$	$\sigma = \sigma(\epsilon + 2\epsilon_1) - 2\sigma_1$ $\epsilon_m = \epsilon + \epsilon_1$
(5)	$\Delta\epsilon < 0$ $\epsilon < \epsilon_m - \epsilon_1$ $\epsilon < -\epsilon_1$	$\sigma = -\sigma(-\epsilon)$ $\epsilon_m = \epsilon + \epsilon_1$

The functional value $\sigma(\epsilon)$ is found from the table (ϵ_i, σ_i) by linear interpolation.

(We should mention that the computation of strain energy is as if unloading to $\sigma = 0$ can take place entirely elastically. That is, the strain energy is assumed proportional to $(\sigma)^2$ where $\sigma = E \epsilon$. This assumption, also used in REPSIL and the catalogued RPSL1D, is only true if $|\sigma| < 2\sigma_1$.)

The UPDATE cards for this option include some changes and additions to COMMON in COMDECK MAIN, a few orders to initiate some new arrays to zero in the main program, RPSL1D, a change to read and store the new form for the stress-strain table (input cards 7) in subroutine START, and a complete replacement of BMSTRS, the stress subroutine for beams.

6. INPUT

RPSL1D originally used the same input as the standard REPSIL (see Reference 2, pp 54-71). This is still true if the user-coded subroutines are compatible. Table 6.1 lists the input for RPSL1D. The first part of this table is like table 3.1 of Reference 2. Variables that are no longer used have a line drawn through them. A brief description for each variable is given after the table. More complete descriptions for some variables may be found in Reference 2. These descriptions are still valid and will not be repeated here. Units are indicated in brackets: T for time, L for length, and F for force. Limits are those for the catalogued RPSL1D (Appendix C); most of these can be easily increased.

The input for user subroutines is listed in the order the subroutines will be called. Examples of input are given, but the descriptions of the input variables for the user subroutines are purposely sketchy. The user must check these subroutines and their input. INGEOM is always called first and usually requires input from cards. INNORM is called next; no input has been needed so far. PRESS is called next if LOAD $\neq$ 0. INVEL is called

next if $LOAD \leq 0$. (PRESS will be called next if $LPRESS > 0$; so far no version of PRESS has been coded for RPSLID that required input after the initial entrance. The initial entrance should occur with $LOAD \neq 0$, but it could occur here.) ENDFRC will be called next if option APLFRC is used. Subroutine ERODE, part of option EROD, would be called last.

CARD	VARIABLES	FORMAT
1	TITLE	8A10
2	~~MESH~~ , NMESH, LAYER, YLDFAC	3I5,E12.6
3	MAXC, NCONT, ~~WRITE~~ , DELTAT	3I5,E12.6
4	~~IBCE1~~ , IBCE2, ~~IBCE3~~ , IBCE4	16I5
5	LOAD, LPRESS, MDAMP, DAMPF, DFACT	3I5,2E12.6
6	E, FNU, SIGZ, RHO, THICKN, NSFL, ISR	5E12.6,2I5
7	(SSIG(J), SEPS(J), DSR(J), PSR(J), J=1,NSFL)	4E15.7
8	NPRINT, (JCHK(I), I=1,3)	16I5
9	NUMCY, (NCYCH(I), I=1,NUMCY)	16I5
10	NLPRIN, (JCYNLP(I), I=1,NLPRIN)	16I5
11	N3D, (NC3DP(I), I=1,N3D)	16I5
11a	NNPE, (NPE(I), I=1,NNPE)	16I5
12	~~ETAD1~~ , ETAD2, NSTRN	2E10.4,I5
13	(~~ETAG1(I)~~ , ETAG2(I), ANGLE(I), ANDLB(I), NETAG(I), I=1,NSTRN)	4E10.4,I5
14	Input for subroutine INGEOM	
15	Input for subroutine INNORM	
16	Input for subroutine PRESS	
17	Input for subroutine INVEL	
18	Input for subroutine ENDFRC	
19	Input for subroutine ERODE	

Table 6.1 List of Input Cards

- IBCE4 from card 4 is stored in position IBCE1 and used as end 1 condition control.
- Omit card 7 if $NSFL = 0$ (or if $NSFL = 1$ and $ISR = 0$), unless the BSTRS option is used. Card 7 is read with a different form and FORMAT with option BSTRS.
- Card 11a is included if, and only if, the PLOTP option is used.

The input variables in table 6.1 are described below. Additional descriptions are given in Chapter 3 of Reference 2.

- Card 1 TITLE Alphanumeric statement for output identification.
- Card 2 ~~MESH~~ Not used.
- NMESH Number of meshes. (NMESH < 102)
- LAYER Number of integration stations through the thickness.
 This may be the number of layers as in REPSIL.

	YLDFAC	Parameter controlling substep size for plastic yielding in shells. (2.0 is suggested)
Card 3	MAXC	Final time step. (Total number of time steps to final time.)
	NCONT	Initial time step for run. This must be zero. Restart capability has been deleted from the CDC version.
	WRITE	Not used.
	DELTAT	Suggested time step. This will be replaced if it is larger than the computed stable Δt or if entered as 0.0. [T]
Card 4	~~IBCE1~~ IBCE2 ~~IBCE3~~ IBCE4	Numbers for end conditions. IBCE1 and IBCE3 are unused. IBCE2 is for end 2 (ξ a maximum). IBCE4 is for end 1 (ξ a minimum). IBCE4 is stored in FORTRAN position IBCE1). 1 - clamped end, 2 - symmetry end, 3 - hinged end, 4 - free end, end 2 only with catalogued RPSL1D, forced end, either end with APLFRC option.
Card 5	LOAD	Parameter which controls calls of INVEL and PRESS in the initiation sequence. (If LPRESS > 0, LOAD $\neq$ 0) 1 - call PRESS (supplies initial pressure) 0 - call INVEL (supplies initial velocity) -1 - call PRESS and then call INVEL
	LPRESS	Subroutine PRESS is called each time cycle to supply pressures in P(N), and then P(N) is converted to force/initial unit surface area, until NCYCLE > LPRESS. The force/initial unit area is constant after NCYCLE > LPRESS. (For beams, unit area is replaced by unit length.)
	MDAMP	If NCYCLE $\geq$ MDAMP, the damping procedures are carried out. If NCYCLE = MDAMP, the program sets LOAD = 0, LPRESS = 0, and all P(N) = 0.
	DAMPF	Viscous damping coefficient. [FT/L ³]
	DFACT	Parameter controlling termination of program during damping. (Suggest .001)
Card 6	E	Young's modulus. [F/L ²]
	FNU	Poisson's ratio, ν .
	SIGZ	Yield stress, σ_0 . [F/L ²] (See card 7.)
	RHO	Initial mass density per unit volume, ρ . [FT ² /L ⁴ , (i.e. mass/L ³)]
	THICKN	Thickness of shell. [L]
	NSFL	Number of changes of slope in the polygonal approximation of the uniaxial loading curve. (NSFL < 5) 0 - elastic behavior, no plasticity. 1 - elastic perfectly plastic response, >1 - elastoplastic, strain hardening response.

ISR Strain rate sensitivity control.
 0 - no strain rate effect,
 1 - plasticity is strain rate dependent.

Card 7 SSIG(J), Stress, σ [F/L²], and strain, ϵ [L/L], at the corners
 SEPS(J) of the polynomial, uniaxial loading curve. (SSIG(1) is
 replaced by SIGZ and SEPS(1) by SIGZ/E.) (If (e_s, σ_s)
 are 'engineering' strain and stress, $\sigma = \sigma_s(1 + e_s)$,
 $\epsilon = e_s(1 + e_s/2)$.)

 DSR(J), Constants for strain rate behavior. These must be
 PSR(J) included if ISR = 1.

With the BSTRS option, cards 7 become:

 (EEB(I), SSB(I), I = 1,2,...) FORMAT(2E15.7)

 EEB(I), Table of strain and stress (reverse order from above)
 SSB(I) for up to 100 points. Table is terminated by
 EEB(NEST) < 0. SSB(1) is replaced by SIGZ, and EEB(1)
 by SIGZ/E, EEB(NEST) by 1000, and SSB(NEST) by
 SSB(NEST-1).

Card 8 NPRINT Number of cycles between surface strain prints at points
 defined on cards 13. Maximum deflection and extreme
 strains are also printed.

 JCHK(I) Numbers controlling printing at cycles on cards 9.
 JCHK(1), or JCHK(2), > 0 requests prints of coordinates,
 displacement increments, and pressure at mesh points.
 JCHK(3) > 0 requests prints of surface normals, and
 a print of surface strains on both surfaces, at all
 mesh points.

Card 9 NUMCY Number of cycles for JCHK(I) controlled prints (card 8)
 and energy balance summary. (<51)

 NCYCH(I) Cycles for prints.

Card 10 NLPRIN Number of JCNLP(I) to follow. (<51)

 JCNLP(I) Cycles to print IMAT(N,K) and IMATM(N,K) arrays.

Card 11 N3D Number of cycles at which displacement plots are to be
 made. (<51)

 NC3DP(I) Cycles for plots.

Card 11a (Include if PLOTP option is used.)

 NNPE Number of plots to be made. ($0 \leq NNPE \leq 9$)

 NPE(I) Index, N, of P(N) to plot as a function of time. P(N)
 ($N1V \leq N \leq N2V$) is force per initial unit area (initial
 length for beams). Other functions of time may be
 plotted by storing them in P(N) ($N2B < N < 104$) and
 listing N on this card.

[Time dependent pressure, LINPRS]

((TPR(I), PPR(I), I=1,2,...) (2E10.3))
((-1.0) (2E10.3))

[PRESS with tabular data, exponential decay. 3/10/76]

(PBAR, BETA, (PID(I), I=1,6) (2E10.3,6A10))
(CONCZ, CONCT, CONCP, TDIF (4E10.3))
(ALPHAS, NOUSE, ISM (2E10.3,I10))
((ZSI(I), TSI(I), PSI(I), I=1,ISM) (3E10.3))

Card 17

[Subroutine INVEL]

[Catalogued INVEL. Like INVEL in Reference 2.]

(--, NI, NF, VR, NV (10X,2I5,E12.6,I5))
(--, N, V (5X,I5,E12.6))

NI, NF Mesh points NI+1 through NF+1 are given initial normal
velocity -VR. Mesh point at end 1 is numbered N1B = 2.

VR Velocity. [L/T]

NV Number of cards to follow.

N Mesh point N+1 given initial normal velocity -V.

V Velocity. [L/T]

Card 18

[Subroutine ENDFRC. Needed with APLFRC option.]

[End 2 free. Tabular forces on end 1.]

(NFZPTS, (LABELV(I), I=1,7) (I10,7A10))
((EFZT(I), TMFZ(I), I=1,NFZPTS) (2E15.7))
(NFTHS, (LABELV(I), I=1,7) (I10,7A10))
((EFTHT(I), TMTHT(I), I=1,NFTHS) (2E15.7))
(NFMUS, (LABELV(I), I=1,7) (I10,7A10))
((EFMUT(I), TMMU(I), I=1,NFMUS) (2E15.7))

Card 19

[Subroutine ERODE. Part of EROD option.]

(NTEPTS, (LABEL(I), I=1,7) (I10,7A10))
((TEROD(I), XEND(I), I=1,NTEPTS) (2E15.7))

APPENDIX A

LIST OF EQUATIONS

In this section, we list the principal equations used in the standard REPSIL and the corresponding equations used for axial symmetry, slab symmetry, and beams, respectively. The notation in column 1 is that of Reference 1, except we have assigned an orthonormal cartesian vector basis ($\hat{i}_1, \hat{i}_2, \hat{i}_3$) and defined corresponding Cartesian components of position, Y^1, Y^2 , and Y^3 .

The notation 'same' in any column means that the entry is the same as that in the column to the left.

Standard REFSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\underline{x} = \underline{x}(r^1, r^2, r, t)$ $= \underline{x}(r^1, r^2, t) + \underline{x}_0(r^1, r^2, t)$	$\underline{x} = \underline{x}(0, \xi, t) = \underline{x} + \underline{x}_0$ Evaluate at $\theta = 0$.	$\underline{x} = \underline{x}(V^1, \xi, t) = \underline{x} + \underline{x}_0$	$\underline{x} = \underline{x}(0, \xi, t) = \underline{x} + \underline{x}_0$
$\underline{x} = V^k \underline{\hat{x}}_k = V^1 \underline{\hat{x}}_1 + V^2 \underline{\hat{x}}_2 + V^3 \underline{\hat{x}}_3$ $V^k = V^k(r^1, r^2, t)$	$\underline{x} = R(\xi, t) \underline{\hat{x}}(0) + Z(\xi, t) \underline{\hat{k}}$ $\underline{\hat{x}} = \cos \theta \underline{\hat{x}}_3 + \sin \theta \underline{\hat{x}}_2 (= -d\hat{Q}/d\theta)$ $\underline{\hat{Q}} = -\sin \theta \underline{\hat{x}}_3 + \cos \theta \underline{\hat{x}}_2 (= d\hat{r}/d\theta)$ $R = (V^3(0, \xi, t))^2 + (V^1(0, \xi, t))^2)^{1/2}$ $= V^3(0, \xi, t)$ $Z = V^2(0, \xi, t) = V^2(0, \xi, t)$	$\underline{x} = R(\xi, t) \underline{\hat{x}}_3 + Z(\xi, t) \underline{\hat{x}}_2 + V^1 \underline{\hat{x}}_1$ Evaluation at $V^1 = 0$. $R = V^3(V^1, \xi, t) = V^3(0, \xi, t)$ $Z = V^2(V^1, \xi, t) = V^2(0, \xi, t)$	$\underline{x} = R(\xi, t) \underline{\hat{x}}_3 + Z(\xi, t) \underline{\hat{x}}_2$ $R = V^3(0, \xi, t)$ $Z = V^2(0, \xi, t)$
$\underline{\hat{x}}_1 = \underline{\hat{x}}_{1,1} = \partial \underline{x} / \partial r^1$	$\underline{\hat{x}}_1 = R \underline{\hat{Q}} + \underline{\hat{x}}_2 = R, \underline{\hat{x}} + Z, \underline{\hat{k}}$	$\underline{\hat{x}}_1 = \underline{\hat{x}}_1, \underline{\hat{x}}_2 = R, \underline{\hat{x}}_3 + Z, \underline{\hat{x}}_2 \underline{\hat{x}}_2$	same
$\underline{\hat{x}}_{1,R} = \partial^2 \underline{x} / \partial r^1 \partial r^1$	$\underline{\hat{x}}_{1,11} = -R \underline{\hat{x}}_{1,12} = R, \underline{\hat{x}} \text{ unused}$ $\underline{\hat{x}}_{1,22} = R, \underline{\hat{x}} + Z, \underline{\hat{k}}$	$\underline{\hat{x}}_{1,11} = \underline{\hat{x}}_{1,12} = \underline{\hat{x}}_{1,21} = 0$ $\underline{\hat{x}}_{1,22} = R, \underline{\hat{x}}_2 \underline{\hat{x}}_3 + Z, \underline{\hat{x}}_2 \underline{\hat{x}}_2$	same same
$a_{\alpha\beta} = \underline{\hat{x}}_{\alpha} \cdot \underline{\hat{x}}_{\beta}$	$a_{11} = (R)^2 (a_{12} = a_{21} = 0)$ $a_{22} = (R, \underline{\hat{x}})^2 + (Z, \underline{\hat{x}})^2 (= (D)^2)$	$a_{11} = 1$ same	same same
$a = a_{11} a_{22} - (a_{12})^2$	$a = a_{11} a_{22} (= (RD)^2) \{a^k = RD\}$	$a = a_{11} a_{22} (= (D)^2)$	same
$a^{11} = a_{22}/a, a^{22} = a_{11}/a$ $a^{12} = a^{21} = -a_{12}/a$	same $\{a^{11} = 1/(R)^2, a^{22} = 1/(D)^2\}$ $\{a^{12} = a^{21} = 0\}$	$a^{11} = 1, a^{22} = 1/(D)^2$ same	same same

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\vec{u} = (a_1 \times a_2) / a^3 (a^3 = a_1 \times a_2)$	$\vec{n} = n_r \hat{r} + n_k \hat{k}$ $n_r = z_2/D, n_k = -R_z/D$	$\vec{n} = n_r \hat{r}_3 + n_k \hat{k}_2$ same	same same
$b_{\alpha\beta} = \vec{n} \cdot \hat{r}_{\alpha\beta}$	$b_{11} = -n_r R$ $b_{22} = n_r R_{,22} + n_k^2 z_2$	$b_{11} = 0$ same	same same
$b_{\alpha}^{\beta} = a^{\beta\gamma} b_{\gamma\alpha}$	$b_1^1 = a^{11} b_{11}, b_2^2 = a^{22} b_{22}$	same $b_1^1 = 0$	same
$a_{\alpha}^{\alpha} = a^{\alpha\beta} a_{\beta\alpha}$	$a_1^1 = a^{11} R \hat{r}$ $a_2^2 = a^{22} R_{,2} \hat{r} + a^{22} z_2 \hat{k}$	$a_1^1 = a^{11} \hat{k}_1$ $a_2^2 = a^{22} R_{,2} \hat{k}_3 + a^{22} z_2 \hat{k}_2$	same same
$\Gamma_{\alpha\beta}^{\gamma} = a^{\gamma\delta} \cdot \hat{r}_{\delta\alpha\beta}$	$\Gamma_{11}^2 = -a^{22} R_{,2} R$ $\Gamma_{22}^2 = a^{22} (R_{,2} R_{,22} + z_2 z_2^2,22)$ { $\Gamma_{12}^1 = \Gamma_{21}^1 = a^{11} R_{,2} R$ unused}	$\Gamma_{11}^2 = 0$ same { $\Gamma_{12}^1 = \Gamma_{21}^1 = 0$ }	same same same
$\Delta \vec{u} = (\vec{r} - \vec{r}') \cdot (\Delta u^k \hat{k}_k = \Delta \vec{r})$	$\Delta \vec{r} = \Delta R \hat{r} + \Delta z \hat{k}$	$\Delta \vec{r} = \Delta R \hat{k}_3 + \Delta z \hat{k}_2$	same
$\Delta n_{\alpha} = -\vec{r} \cdot \Delta \vec{u}_{\alpha}$	$\Delta n_2 = -(n_r \Delta R_{,2} + n_k \Delta z_{,2})$	same	same
$\Delta n = (a_{\alpha} + \frac{\Delta n_{\alpha}}{1 + \vec{r} \cdot \vec{r}} \cdot \vec{r}) a^{\alpha\beta} \Delta n_{\beta}$	$\Delta n = \Delta n_r \hat{r} + \Delta n_k \hat{k}$ $\Delta n_r = (R_{,2} + n_r \Delta n) a^{22} \Delta n_2$ $\Delta n_k = (z_2 + n_k \Delta n) a^{22} \Delta n_2$ $\Delta n = \Delta n_2 / (1 + n_r \vec{r} + n_k \vec{r})$	$\Delta n = \Delta n_r \hat{r} + \Delta n_k \hat{k}$ same same same	same same same same

Standard REPSIL

Axial Symmetry

Slab Symmetry

Laterally Symmetric Beam

$$g_{\alpha\beta} = \frac{1}{2}(\bar{a}_{\alpha\beta} - 2\bar{c}_{\alpha\beta})$$

$$g = g_{11}g_{22} - (g_{12})^2$$

$$\text{same } (g_{12} = g_{21} = 0)$$

$$g = g_{11}g_{22}$$

$$\text{same } (g_{11} = 1)$$

$$\text{same } (g = g_{22})$$

$$\text{same } (g_{11} \text{ not used})$$

$$\text{same}$$

$$g_{11} = g_{22}/g, \quad g_{22} = R_{11}/g$$

$$g_{12} = g_{21} = -g_{12}/g$$

$$\text{same } (g_{11} = 1/g_{11}, \quad g_{22} = 1/g_{22})$$

$$g_{12} = g_{21} = 0$$

$$\text{same } (g_{11} \text{ not used})$$

$$\text{same}$$

$$\Delta c_{\alpha\beta} = \frac{1}{2}(\bar{a}_{\alpha\beta} \cdot \Delta \bar{u}_{\alpha\beta} + \bar{a}_{\beta\alpha} \cdot \Delta \bar{u}_{\alpha\beta})$$

$$- \Delta \bar{u}_{\alpha\beta} \cdot \Delta \bar{u}_{\alpha\beta}$$

$$- \bar{c}_{\alpha\beta}(\bar{u}_{\alpha\beta} \cdot \Delta \bar{u}_{\alpha\beta} + \Delta \bar{u}_{\alpha\beta} \cdot \bar{u}_{\alpha\beta})$$

$$\Delta c_{\alpha\beta} = \frac{1}{2}(\Delta \bar{a}_{\alpha\beta} - \Delta \bar{c}_{\alpha\beta}) \quad (\Delta c_{12} = \Delta c_{21} = 0)$$

$$\Delta a_{11} = 0$$

$$\text{same}$$

$$\text{same}$$

$$\text{same}$$

$$\Delta b_{11} = -(\Delta R_{11} + R \Delta n_{11})$$

$$\Delta b_{22} = (n_{11}^2 \Delta R_{11} + n_{22}^2 \Delta R_{22})$$

$$+ (\Delta n_{11} R_{11} + \Delta n_{22} R_{22})$$

$$\Delta b_{11} = 0$$

$$\text{same}$$

$$\text{same}$$

$$\text{same}$$

$$\Delta c_{\alpha\beta} = g_{\alpha\beta} \Delta c_{\alpha\beta}$$

$$\Delta c_{11} = g_{11} \Delta c_{11}, \quad \Delta c_{22} = g_{22} \Delta c_{22}$$

$$\text{same}$$

$$\sigma_{\alpha\beta} = g_{\alpha\beta} \sigma_{\alpha\beta}$$

$$\sigma_{11} = g_{11} \sigma_{11}, \quad \sigma_{22} = g_{22} \sigma_{22}$$

$$\text{same } (\sigma_{11}^{-1} \text{ not used})$$

$$\Delta \sigma_{\alpha\beta} = \frac{E}{1+\nu} (\Delta c_{\alpha\beta} + \frac{\nu}{1-\nu} \Delta c_{\gamma\gamma} \delta_{\alpha\beta})$$

$$\Delta \sigma_{11} = (\Delta c_{11} + \nu \Delta c_{22}) E / (1-\nu^2)$$

$$\text{same } (\Delta \sigma_{11}^{-1} \text{ not used})$$

$$\Delta \sigma_{22} = (\nu \Delta c_{11} + \Delta c_{22}) E / (1-\nu^2)$$

$$\text{same}$$

$$\text{same}$$

$$T_{\alpha\beta} = \sigma_{\alpha\beta} + \Delta \sigma_{\alpha\beta}$$

$$T_{11} = \sigma_{11}^{-1} + \Delta \sigma_{11}^{-1}, \quad T_{22} = \sigma_{22}^{-2} = \sigma_{22}^{-2} + \Delta \sigma_{22}^{-2}$$

$$\text{same } (T_{11}^{-1} \text{ not used})$$

Standard REPSIL

$$\psi_T = \frac{1}{2}(\sigma_1^2 - \sigma_2^2) - (\sigma_0)^2$$

$$\text{If } \psi_1 < 0, \sigma_1^a = \sigma_2^a$$

If $\psi_T > 0$, the stress increment

is partly plastic. Correction is

$$\sigma_1^a = \sigma_1^a - \frac{1-2\nu}{3(1-\nu)} \sigma_1^a \sigma_2^a$$

$$\sigma_1^a = \sigma_1^a - \lambda \sigma_1^a$$

where λ is the smaller root of

$$3\sigma_1^a \sigma_2^a - (\sigma_1^a)^2 = 2(\sigma_0)^2$$

$$\sigma_1^a = g \sigma_1^a$$

This stress calculation is for an elastic perfectly plastic material with no complications. The mechanical sublayer model is used for strain hardening. Strain rate dependency may be used with the equations:

$$\sigma_0 = \sigma_0(\text{static}) [1 + (\dot{\epsilon}/\dot{\epsilon}_0)^{1/p}]$$

$$\dot{\epsilon} = \frac{1}{\Lambda t} \left[\frac{3}{2} \Lambda \sigma_1^a \sigma_2^a - \frac{1}{2} (\Lambda \sigma_1^a)^2 \right]^{1/2}$$

If ψ_T is large, the above procedure is repeated L times with $\Delta \sigma_0^a/L$ in place of $\Delta \sigma_0^a$. If λ is negative or complex, L is increased. If L is too large, or if $g < 0$, the run is aborted.

Axial Symmetry

$$\psi_T = \sigma_1^2 (\sigma_1^2 - \sigma_2^2) + (\sigma_2^2)^2 - (\sigma_0)^2$$

Same for σ_1^2 and σ_2^2

same

$$\sigma_1^{-1} = (2-\nu)\sigma_1^{-1} - (1-2\nu)\sigma_2^{-1}$$

$$\sigma_2^{-1} = (2-\nu)\sigma_2^{-1} - (1-2\nu)\sigma_1^{-1}$$

same for σ_1^2 and σ_2^2 , where

$$\Lambda \lambda^2 - B \lambda + C = 0$$

$$A = \sigma_1^2 (\sigma_1^2 - \sigma_2^2) + (\sigma_2^2)^2, C = \psi_T$$

$$B = \sigma_1^2 (2\sigma_1^2 - \sigma_2^2) + \sigma_2^2 (2\sigma_2^2 - \sigma_1^2)$$

$$\sigma_1^{11} = g \sigma_1^{11}, \sigma_2^{22} = g \sigma_2^{22}$$

same

same

$$\dot{\epsilon} = [\Delta \epsilon_1^2 (\Delta \epsilon_1^2 - \Delta \epsilon_2^2) + (\Delta \epsilon_2^2)^2]^{1/2} / \Delta t$$

same

Slab Symmetry

Laterally Symmetric Beam

$$(\psi_T = (\sigma_2^2)^2 - (\sigma_0)^2)$$

$$\text{If } |\sigma_2^2| < \sigma_0, \sigma_2^2 = \sigma_2^2$$

The uniaxial stress-strain assumption gives the simple corrections:

$$\text{If } \sigma_2^2 > \sigma_0, \sigma_2^2 = \sigma_0$$

$$\text{If } \sigma_2^2 < -\sigma_0, \sigma_2^2 = -\sigma_0$$

$$\sigma_2^2 = g \sigma_2^2$$

same

same

$$\dot{\epsilon} = \Delta \epsilon_2^2 / \Delta t$$

No iteration.

Standard REPSIL

$\sigma^{\alpha\beta}$ for each pseudo-sublayer is stored for future use. A weighted stress, $\sigma_k^{\alpha\beta}$ is found for layer k.

$$\sigma_a^{\alpha\beta} = \sum_k \sigma_k^{\alpha\beta} (\tau_k)^a \Lambda^a, \quad (a=0,1,2)$$

$$\hat{Q}^{\alpha\beta} = a^{\frac{1}{2}} (\sigma_0^{\alpha\beta} - b_{\gamma}^{\gamma} \sigma_1^{\alpha\beta})$$

$$\hat{M}^{\alpha\beta} = a^{\frac{1}{2}} \{ \sigma_1^{\alpha\beta} - b_{\gamma}^{\gamma} \sigma_2^{\alpha\beta} - b_{\gamma}^{(\alpha} \sigma_2^{\gamma\beta)} \}$$

$$\hat{N}^{\alpha\beta} = \hat{Q}^{\alpha\beta} z_{\beta} + \Gamma_{\beta\gamma}^{\alpha} M^{\beta\gamma} n_{\beta}$$

$$\hat{M}^{\alpha\beta} = \hat{M}^{\alpha\beta} n_{\beta}$$

Axial Symmetry

same for σ^{11} and σ^{22}

same for σ^{11} and σ^{22}

$$\hat{Q}^{*11} = a^{\frac{1}{2}} \{ \sigma_0^{11} - (b_1^1 + b_2^2) \sigma_1^{11} \}$$

$$\hat{Q}^{*22} = a^{\frac{1}{2}} \{ \sigma_0^{22} - (b_1^1 + b_2^2) \sigma_1^{22} \}$$

$$\hat{M}^{*11} = a^{\frac{1}{2}} \{ \sigma_1^{11} - (2b_1^1 + b_2^2) \sigma_2^{11} \}$$

$$\hat{M}^{*22} = a^{\frac{1}{2}} \{ \sigma_1^{22} - (b_1^1 + 2b_2^2) \sigma_2^{22} \}$$

$$\hat{N}^{*1} = FN_0^{11} \hat{\sigma} = Q^{*11} R \hat{\sigma}$$

$$\hat{N}^{*2} = FN_r^{11} \hat{\sigma} + FN_k^{11} \hat{\sigma}$$

$$FN_r^{11} = \hat{Q}^{*22} R_{r,2} + n_{r,CSM}^2$$

$$FN_k^{11} = \hat{Q}^{*22} Z_{r,2} + n_{k,CSM}^2$$

$$CSM^2 = \Gamma_{11}^2 \hat{M}^{*11} + \Gamma_{22}^2 \hat{M}^{*22}$$

$$\hat{M}^{*aa} = FN_r^{aa} \hat{\sigma} + FN_k^{aa} \hat{\sigma} \quad (\text{no sum})$$

$$FM_r^{11} = \hat{M}^{*11} n_r, \quad FM_r^{22} = \hat{M}^{*22} n_r$$

$$FM_k^{22} = \hat{M}^{*22} n_k \quad (FM_k^{11} \text{ not used})$$

Slab Symmetry

same

same

same ($b_1^1 = 0$)

same

same

same

$\hat{N}^{*1} = \hat{Q}^{*11} i_1$ not used

$\hat{N}^{*2} = FN_r^{11} i_3 + FN_k^{11} i_2$

same

same

same ($\Gamma_{11}^2 = 0$)

$\hat{M}^{*aa} = FN_r^{aa} i_3 + FN_k^{aa} i_2$ (no sum)

same

same

Laterally Symmetric Beam

same for σ^{22}

$$\sigma_a^{22} = \sum_k M_k \sigma_k^{22} (\tau_k)^a \quad (M_k = \text{Area}_k)$$

$Q^{*11} = 0$ not used

same

$M^{*11} = 0$ not used

same

same ($N^{*1} = 0$)

same

same

same

same

same ($M^{*11} = 0$)

same

same

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\underline{E}^+ = -a^2 \Delta P \underline{R}$	$\underline{E}^+ = F_I^+ \underline{E} + F_K^+ \underline{E}$	$\underline{E}^+ = F_I^+ \underline{E}_3 = F_K^+ \underline{E}_2$	same
	$F_I^+ = -a^2 \Delta P n_I, F_K^+ = -a^2 \Delta P n_K$	same	same
$\Delta \underline{U}^+ = \Delta \underline{U} + \Delta t^2 (\dot{M}^{ab} \underline{e}_{ab} + \dot{N}^{ab} \underline{e}_a + \underline{E}^+) / \gamma_0^2$	$\Delta R^+ = \Delta R + \Delta t^2 (F_I^{22} - F_I^{11} + F_I^2 - F_I^1 + F_I^+) / \gamma_0^2$	$\Delta R^+ = \Delta R + \Delta t^2 (F_I^{22} + F_I^2 - F_I^1 + F_I^+) / \gamma_0^2$	same
$= \Delta \gamma^{k+} \underline{E}_k$	$\Delta Z^+ = \Delta Z + \Delta t^2 (F_K^{22} + F_K^2 - F_K^1 + F_K^+) / \gamma_0^2$	same	same
	$+ F_K^+ / \gamma_0^2$		
$\gamma_0^+ = \rho h \Lambda^{1/2} (\Lambda = a \text{ at time } t=0)$	same	same	$\gamma_0^+ = \rho \Lambda^{1/2} \Delta \underline{U}_k$
$\underline{E}^+ = \underline{E} + \Delta \underline{U}^+$	same	same	same
$\gamma^{k+} = \gamma^k + \gamma^{k+}$	$R^+ = R + \Delta R^+, Z^+ = Z + \Delta Z^+$	same	same

APPENDIX B

DEFINITION OF FORTRAN VARIABLES

In this appendix we list the FORTRAN variables used in RPSL1D as listed in Appendix C, plus additional variables used in the options PLOTP, SHR3/1, MSQSVS, APLFRC, EAPFRC, EROD, or BSTRS. Extra columns and a lot of symbology were used in an attempt to shorten the appendix but still give enough information.

Column 1 is the FORTRAN name. Closely related names may be grouped. Arrays are indicated by enclosing a symbolic integer in parentheses (e.g. AMS(N)).

Column 2 lists the symbol used in this text or one of the references. If the symbol is in braces, { }, it was first used in this report; if in brackets, [], it was first used in Reference 2; if in neither braces or brackets, the symbol originated in Reference 1. Examples:

{M} Symbol for moment in this report.

[A] Coefficient used in Reference 2.

a_{11} Symbol of a covariant metric component first used in Reference 1.

Column 3 partially describes the status of the variable:

C denotes a name in COMMON.

I denotes input.

S_0 denotes a variable that is rarely changed throughout a computer run, as opposed to

S denoting a term that is fixed for one time step, or

T denoting a transient value.

D denotes a dummy argument for a subroutine.

Column 4 names the subroutine which stores the variable. (If other subroutines change or store these variables, they will usually be listed in Column 6.)

Column 5 gives the stage of development when the variable originated:

U denotes a variable described in Appendix C of the User's Manual (Reference 2),

R denotes a newer name included in the RPSL1D file,

any name is the option that first used the variable.

Column 6 defines the variable and may give additional information such as a reference, a limiting value, a formula, the input card that it is entered on, or any other item of interest.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
AA	[A]	T	STRESS	U	Coefficient for computing TANBDA.
AM2(I)	[M]	C,S	RESULT	SURS/I	Moment at mesh point N. AMS = A22*F22.
ALFN		T	STRAIN	U	Alphanumeric for output: "INNER" or "OUTER".
ANGEL	[θ]	T	STRAIN	U	ANGLE(I) or ANGLB(I) in radians.
ANGLR(I), ANGLB(I)	[φ]	I,C,S ₀	START	U	Angles to locate surface strains. (Input 13 in degrees) (Ref. 2, p. 52)
AR11, AR22	a_{11}, a_{22}	T	DGEOM	U	Contravariant components, reference surface metric.
ASA(I), ASB(I)	[2α ²]	C,S	STRAIN	U	(α = sinθ. Ref. 2, p. 52)
AY3UN	$\bar{u} \cdot \bar{u}$	T	STRAIN	R	Square of displacement.
A11,A22	a_{11}, a_{22}	C,T	DGEOM	U	Covariant components, reference surface metric.
A11(I),A22(I)	A_{11}, A_{22}	C,S ₀	ABINIT	U	Initial a_{11}, a_{22} at NI1(I).
A112(I),A222(I)	A_{11}, A_{22}	C,S ₀	ABINIT	U	Initial a_{11}, a_{22} at NI2(I).
B	[-2B]	T	STRESS	U	Coefficient for computing TANBDA.
BEPI, BEPI24, BEPS		T	BMSTRS	BSTRS	Temporary.
BM11, BM22	b_1^1, b_2^2	C,T	DGEOM	U	Mixed components of 2nd fundamental tensor.
BSA(I),BSB(I)	[2B ²]	C,S ₀	STRAIN	U	(B = cosθ. Ref. 2, p. 52)
BT	b_1^1, b_2^2	C,T	DGEOM	U	Trace of 2nd fundamental tensor.
B11,B22	b_{11}, b_{22}	C,T	DGEOM	U	Covariant components, 2nd fundamental tensor.
B11(I),B22(I)	B_{11}, B_{22}	C,S ₀	ABINIT	U	Initial b_{11}, b_{22} at NI1(I).
B112(I),B222(I)	B_{11}, B_{22}	C,S ₀	ABINIT	U	Initial b_{11}, b_{22} at NI2(I).
C	[c] (C)	T	START	U	' Sound speed. $C = \sqrt{E/(\rho(1-\nu^2))}$. If IB>0, $\dot{c} = \sqrt{E/\rho}$.
CA	$\Delta \xi^2$	C,S ₀	START	U	Surface area. Used for kinetic energy and work.
CB		C,S ₀	START	U R R	$CB = \frac{1}{2} \Delta \xi^2 / E$. If IB>0, $CB = \frac{1}{2} \Delta \xi^2 / E$. If IGAUSS = 1, $CB = \frac{1}{2} \Delta \xi^2 / E$.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
CINER	[T*]	C,S	DAMP	U	Kinetic energy, KE, removed by KEA. (Ref. 2, p.46)
CINER	[T*]	C,S	KINET	U	KE at time $t-\Delta t/2$.
CINER	[T*,T*,]	C,S	KINET	U	KE at time $t+\Delta t/2$ or $t+3\Delta t/2$. (REPSLIP,DAMP,DESTEP)
CINER	[T*]	C,T	DAMP	U	Previous CINES in DAMP and DESTEP.
CINES2	[T]	C,T	DAMP	R	Previous CINES in DAMP. (Unused)
CINET	[T]	C,S	KINET	U	Kinetic energy at current time, t. (DESTEP)
CA		T	KINET	U	Portion of CA used with mesh point. (PWORK)
CSM2		T	RESULT	U	Normal component of $\dot{N}^2/TB$.
CS211,CS222	$\Gamma_{11}^2, \Gamma_{22}^2$	C,T	DGEOM	U	Christoffel symbols. Used in RESULT.
CTWO	$\{C_2\}$	T	START	R	Constant for computing DEFLM. (See 3.5)
C1	$[C2/(C2+4)]$	C,S ₀	START	U	Constant for equation of motion. (DESTEP)
C101D		T	DESTEP	U	Old value of C1 in DESTEP.
C2		C,S ₀	START	U	$(C2 = 2 * DELTAT * DAMPF / CAMZ.)$ (DESTEP)
D	$(a_{22})^{1/2}$	T	DGEOM	R	Temporary storage in DGEOM, MOTION, and INNORM.
D	$[B^2-AC]$	T	STRESS	U	Discriminant of quadratic equation for TAMBD.
DA	a	T	DGEOM	U	Determinant of surface metric. $DA = A11*A22$.
DAMPF	[D]	C,I,S ₀	START	U	Viscous damping coefficient. (Input 5)
DAT(1)		T	PDATA	U	Array for storage on plotting file.
DA1(N)	$1/2(a_{11}-A_{11})$	C,S	DGEOM	R	Σ DA11 at mesh point N. (A11 at time 0.)
DA2(N)	$1/2(a_{22}-A_{22})$	C,S	DGEOM	R	Σ DA22 at mesh point N. (A22 at time 0.)
DA11,DA22	$1/2a_{11}, 1/2a_{22}$	T	DGEOM	U	Half the increment in a_{11}, a_{22} from $t-\Delta t$ to t.
DA22P	$1/2a_{22}$	T	DGEOM	R	$1/2a_{22}$ at previous midmesh.
DB1(N)	$b_{11}-B_{11}$	C,S	DGEOM	R	Σ DB11 at mesh point N.
DB2(N)	$b_{22}-B_{22}$	C,S	DGEOM	R	Σ DB22 at mesh point N.
DB11,DB22	$Ab_{11}, \Delta b_{22}$	T	DGEOM	U	Increment in b_{11}, b_{22} from $t-\Delta t$ to t.
DE		C,S	ERODE	EROD	Erosion from end mesh point. May be negative.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
DECOM	$(\Delta \epsilon / 4)$	S_0	EROIDE	EROD	Maximum permitted DE before changing end point.
DELY		C, S	STRAIN	R	Maximum displacement.
DEFXT		T	STRAIN	R	Maximum displacement in current cycle.
DELFB	$[\Delta \epsilon_g]$	T	START	U	Critical bending time increment.
				R	$\Delta t_g = \frac{1}{2}(\Delta \epsilon_{A2})^2 / (C/C_{TWO})$
				R	$\Delta t_g = \sqrt{3}(\Delta \epsilon_{A2})^2 / (Ch)$, if IGAUSS = 1.
DELJDE		T	ERODE	EROD	Erosion increment from t- Δt to t.
DELJAN	$(\Delta t)^2 / \gamma_0$ $(\Delta t)^2 / GMEZ$	C, S_0	RPSLID	U	Constant used in DGFOM for TEMP(N).
DELJIN	Δt	T, I	START	U	Temporary storage for input Δt .
DELJIN	$[\Delta t_M]$	T	START	U	Critical membrane time increment. $\Delta t_M = \Delta \epsilon / C$.
DELJIN		T	START	U	$0.95 * \text{MIN}[\text{DELFB}, \text{DEJIN}]$ rounded.
DELR, DELS		T	DESTEP	U	Factors used in DESTEP to change DELTAT.
DELSQ	$(\Delta t)^2$	C, S_0	START	U	Square of time increment. (DESTEP)
DELTAT	Δt	C, I, S_0	START	U	Time increment. (Input 13) May change to minimum of DELJIN and DELJIN. (DESTEP)
DELTR, DELTZ		T	BOUND	R	Components used to locate external point.
DEPS1(K), DEPS2(K)	$\Delta \epsilon_{11}, \Delta \epsilon_{22}$	C	DGEOM	U	Covariant components of strain increment at t_k .
		C	DGEOM	U	$\Delta \epsilon_{\alpha\beta} = \frac{1}{2} \Delta a_{\alpha\beta} - \zeta_k \Delta b_{\alpha\beta}$
DEPS11, DEPS22	$\Delta \epsilon_1^1, \Delta \epsilon_2^2$	T	STRESS	U	Mixed components of strain increment. (BNSTRS)
DETA1	$\Delta \epsilon^1 \{ \Delta n^1 \}$	C, S_0	INGEOM	U	Increment in the ξ^1, ξ^2 Lagrange coordinates.
DETA2	$\Delta \epsilon^2 \{ \Delta n^2 \} \{ \Delta \epsilon \}$	C, S_0	INGEOM	U	(If IB=0, $\Delta \epsilon^1=1.0$ in START.)
DEFACT		C, I, S_0	START	U	Factor for terminating damped run. (Input 5)
DEFG		S_0	PRESS		In PRESS of 10/12/76, DFTG = -DAMPF/GANZ.
DG	g	T	STRESS	U	Determinant of metric. DG = G11G22.
DN	Δn	T	DGEOM	U	Normal component of Δg . $\Delta g = \Delta n_2 g^2 + \Delta n_1 g$.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
DN1(I),DN2(I), DN3(I)		C,S ₀ C,S ₀	START START	U R	Interpolation coefficients for surface strains. DN3(I) added with revision.
DNK,DNR	$\{\Delta n_k, \Delta n_r\}$	T	DGEOM	R	Cartesian components of Δn . $\Delta n = DNK \hat{i}_2 + DNR \hat{i}_3$.
DNL2,DNR2	$\Delta n_2, \Delta n_2^2$	T	DGEOM	U	Covariant, contravariant tangential component of Δn .
DR(N),DZ(N)	$\{\vec{R}_N, \vec{z}\}$ $\{\Delta R, \Delta Z\}$	C,S	INVEL	R	[Replaces $U_j(N,N)$] Components of velocity (INVEL). Displacement increments RPSLID, MOTION, and other.
DRI,DZI	$\{\Delta R^-, \Delta Z^-\}$	T	MOTION	R	Components of displacement increment, $t-\Delta t$ to t .
DR11,DZ11	$\{\Delta R^-, \Delta Z^-\}$	T	MOTION	EAPERC	DRI and DZI at end 1.
DRSN	$\{\Delta R\}$	T	DGEOM	R	Radial displacement increment.
DRS,DZS	$\{\Delta R, \Delta Z\}$	T	MOTION	R	Components of displacement increment, t to $t+\Delta t$.
DR2, DZ2	$\{\Delta R, \Delta Z\}$ $\{\Delta Z, \Delta Z\}$	C,T C,T	DGEOM DGEOM	R R	[Replace U2i.] First finite differences of displacement increment components.
DR22, DZ22	$\{\Delta R, \Delta Z\}$ $\{\Delta Z, \Delta Z\}$	C,T C,T	DGEOM DGEOM	R R	[Replace U22i.] Second finite differences of displacement increment components.
DSG11,DSG22L	$\frac{E_a}{\Delta \sigma_B/L}$	T	STRESS	U	Subincrement division of $\Delta \sigma_B$.
DSIG11,DSIG22	$\Delta \sigma_1^E, \Delta \sigma_1^E$	T	STRESS	U	Mixed components of elastic stress increment.
DSQDID		T	DESTEP	U	Old $(\Delta t)^2$.
DSR(I)	$\{d_j\}$	C,I,S ₀	START	U	Constant for strain rate (Input 7).
DTHEF	$\{\Delta \theta\}$	T	MOTION	EAPERC	Angle increment $[t, t+\Delta t]$ current end.
DTHTIP,DTHT2P	$\{\Delta \theta_j^-\}$	C,S ₀	MOTION	EAPERC	Angle increment $[t-\Delta t, t]$ (end 1, end 2).
DWE1P,DWE2P	$\{\Delta W_j^-\}$	C,S ₀	MOTION	EAPERC	Work increment $[t-\Delta t, t]$ (end 1, end 2).
DWF	$\{\Delta W_F^-\}$	T	MOTION	EAPERC	Work increment $[t, t+\Delta t]$, from force on the end.
DWT	$\{\Delta W_M^-\}$	T	MOTION	EAPERC	Work increment $[t, t+\Delta t]$, from moment on the end.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
DM	$\Delta \cdot \Delta \Sigma$	T	PHORK	U	Normal component of $\Delta \Sigma$.
DN (N)	$\{\Delta Z\}$	C,S	MOTION	R	See DR(N).
DZETAK	$\{\Delta \Sigma_k\}$	I,T	INGEOM	R	Thickness of k'th layer of a beam. (Input 14)
DZ1, DZ11, DZS		T	MOTION	R	See DRI, DRII, DRS.
DZ2, DZ22		C,T	INGEOM	R	See DR2 and DR22.
D1, D2	$\{u^3, u^2\}$	C,S	STRAIN	U	Components of displacement at ETAD2. $\{\Delta R, \Delta Z\}$
E	E	C,I,S ₀	START	U	Young's modulus. (Input 6)
EFB(1),SSB(1)	$\{\epsilon_1, \sigma_1\}$	C,I,S ₀	STRAI	BSTRS	Uniaxial stress-strain curve. $1 < I < 100$. (Input 7) (See comments at SEPS(J).)
EFM1	$\{u(t)\}$	T	ENDFRC	APLFR	μ at time t. $EFM1 = EFMU \neq EFZ1$.
EFMT(1),TMRM(1)	$\{u_1, t_1\}$	I,S ₀	ENDFRC	APLFR	Table of μ vs. time. (Input 18)
EFRI		C,S	ENDFRC	APLFR	i_3 force, end 1. $EFRI > 0$ decreases $\Delta R(N1B)$.
EFR2		C,S	ENDFRC	APLFR	i_3 force, end 2. $EFR2 > 0$ increases $\Delta R(N2B)$.
EFTH	$\{\theta(t)\}$	T	ENDFRC	APLFR	θ at time t.
EFTH(1),TMRH(1)	$\{\theta_1, t_1\}$	I,S ₀	ENDFRC	APLFR	Table of θ vs. time. (Input 18)
EFZT(1),TMRZ(1)		I,S ₀	ENDFRC	APLFR	Table of force, EFZ1, vs. time, t. (Input 18)
EFZ1		C,S	ENDFRC	APLFR	i_2 force, end 1. $EFZ1 > 0$ decreases $\Delta Z(N1B)$.
EFZ2		C,S	ENDFRC	APLFR	i_2 force, end 1. $EFZ2 > 0$ increases $\Delta Z(N2B)$.
EMI		C,S	ENDFRC	APLFR	Applied moment at end 1. $EMI > 0$ decreases $\Delta R(N1B)$, increases $\Delta R(N1B+1)$.
EM2		C,S	ENDFRC	APLFR	Applied moment at end 2. $EM2 > 0$ increases $\Delta R(N2B)$, decreases $\Delta R(N2B-1)$.
EN	$\{\Delta W(t-\frac{1}{2}\Delta t)\}$	T	MOTION	U	Work increment, $\{t-\Delta t, t\}$. $(EN = \frac{1}{2}CA*(EMR+ENS))$
ENR	$\{\Delta W(t-\Delta t)/CA\}$	C,S	MOTION	U	Work increment/CA, $\{t-\frac{3}{2}\Delta t, t-\frac{1}{2}\Delta t\}$. (RPSL10)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
ENS	$\{ \Delta M(t) / \Delta A \}$	C,S	PNORK	U	Work increment/CA, $[t - \frac{1}{2}\Delta t, t + \frac{1}{2}\Delta t]$. (RPSLID)
EPIC(I)		T	STRAIN	R	Temporary for computing extreme surface strains.
EPCL1, EPCL2	$[E_1, E_2]$	T	DGEOM	R	Elongational components of strain on the lower surface in ξ^1 and ξ^2 directions. (Output)
EPCL1, EPCL2	$[E_1, E_2]$	T	DGEOM	R	Elongational components of strain on the upper surface in ξ^1 and ξ^2 directions. (Output)
EPLEB(N,K)	$\{\epsilon_m\}$	C,S	BNSTRS	BSTRS	Mean of possible elastic variation of EPSB(N,K).
EPLEB(N,K)	$\{\epsilon_m\}$	C,S	BNSTRS	BSTRS	Mean of possible elastic variation of EPSB(N,K).
EPSANR(I)	$[E_\theta]$	T	STRAIN	U	Elongational surface strains in θ directions at location I. (Output)
EPSANG(I)	$[E_\theta]$	T	STRAIN	U	
EPSB(N,K)	ϵ_2^2	C,S	BNSTRS	BSTRS	Uniaxial strain at mesh point N, layer K.
EPSBN(N,K)	ϵ_2^2	C,S	BNSTRS	BSTRS	Uniaxial strain at midmesh N, layer K.
EPSDOT	$\dot{\epsilon}$	T	STRESS	U	Deviator strain rate in strain rate equation.
EPSLI(N), EPSL2(N)	$\epsilon_{11}, \epsilon_{22}$	C	DGEOM	U	Covariant strain components on the lower surface at mesh point N.
EPSR1, EPSR2	$\{\epsilon_1, \epsilon_2\}$	T	STRAIN	U	Intermediate values in surface strain calculations.
EPSSI(I), EPSS2(I)	$[F_1, E_2]$	C	STRAIN	U	Elongational components of strain in the ξ^1 and ξ^2 directions at position I.
EPSU1(N), EPSU2(N)	$\epsilon_{11}, \epsilon_{22}$	C	DGEOM	U	Covariant strain components on the upper surface at mesh point N.
EPSX(I)		C,S	STRAIN	R	Extreme strains. (I = 1,8) (Output)
EPSXT(I)		T	STRAIN	R	Extreme strains at current time. (Output)
EPSZ	σ_0/E	C,S ₀	START	BSTRS	Strain at yield stress.
EROD		T	ERODE	EROD	Erosion from initial end 1 up to time t.
ERODP		S	ERODE	EROD	Erosion from initial end 1 up to time t- Δt .

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
ETADZ	$\xi^2 (\xi) [n^2]$	C, I, S ₀	START	U	Material coordinate of the point at which displacement is plotted. (Input 12) (INCEOM?)
ETADZ(1)	$\xi^2 (\xi) [n^2]$	C, I, S ₀	START	U	Material coordinate for the 1'th surface strain. (Input 13) (INCEOM?)
FLOATL	[1/L]	T	STRESS	U	Proportion of stress increment per step.
FN11R(N)	$\{ \dot{M}^{*11} n_r \}$				Components of bending resultant tensor times the normal at mesh point N. (These replace FN _{αβ} (M,N) of Reference 2.)
FN22K(N)	$\{ \dot{M}^{*22} n_k \}$	C, T	RESULT	R	
FN22R(N)	$\{ \dot{M}^{*22} n_r \}$				
FN11		C, I, S ₀	START	U	Poisson's ratio. (Input 6)
FN11R(N), FN22K(N)		C, T	RESULT	R	Components of stress resultant tensor at midmesh N. [These replace FN _{αj} of Reference 2.]
FN11, FN22	$\{ \dot{M}^{*11} \dot{M}^{*22} / (1B + 1B) \}$	T	RESULT	U	Proportional to bending resultant components.
G	$\dot{M}^2 / (1 + \nu)$	C, S ₀	START	U	Shear modulus. Used in STRESS.
GAME	$\gamma_0^* [\Gamma_0]$	C, S ₀	START	U	If IB=0, $\Gamma_0 = \rho h = \text{mass/unit surface area}$. If IB=1, $\Gamma_0 = \rho * \text{SUMAR} = \text{mass/unit length}$.
GI11(I), GI22(I)	G^{11}, G^{22}	C, S ₀	STRAIN	U	Contravariant components of initial metric at I.
GL11(N), GL22(N)	G_{11}, G_{22}	C, S ₀	RPSLID	R	Initial metric, lower surface, mesh point N.
GR11, GR22	g^{11}, g^{22}	T	STRESS	U	Contravariant metric components. (BMSTRS)
GU11(N), GU22(N)	G_{11}, G_{22}	C, S ₀	RPSLID	R	Initial metric, upper surface, mesh point N.
G11, G22	g_{11}, g_{22}	T	STRESS	U	Metric at ξ_k in STRESS and BMSTRS.
I	G_{11}, G_{22}	T	STRAIN	U	Initial metric on surfaces in STRAIN.
IB	[1,1]	T		U	General index. Multiple use.
		C, S ₀	INCEOM	R	Signal from INCEOM: 0 or 1. IB=1 denotes a beam.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
IBCE1		C,I,S ₀	START	R	Boundary condition end 1. Edge 4 of REPSIL.
IBCE2		C,I,S ₀	START	U	Boundary condition end 2. Edge 2 of REPSIL.
IBCE4		(I)	(NONE)	U	REPSIL name. Read into IBCE1. (See input 4)
IENDFR		S ₀	ENDFR	APLFR	Signal for initial entry into ENDFRC.
IFRODE		S ₀	ERODE	EROD	Signal for initial entry into ERODE.
IFLAG		T	PDATA	U	Control number in subroutine PDATA.
IGAUSS		C,S ₀	INGEOM	R	Signal from INGEOM. IGAUSS=1 denotes Gaussian integration through thickness. (If IB=1, IGAUSS=0 in START.)
II		S ₀	PDATA	U	Dimension of DAT(I) array. II = 2*NSTRN+8.
IP		C	DGEOM	R	IP=1 denotes mesh point, IP=2 denotes midmesh.
IPRESS		S ₀	PRESS	R	Signal for initial entry into PRESS.
IS		C,S ₀	START	R	(IS=1 or 2) 1-radial symmetry, 2-slab symmetry.
ISR		C,I,S ₀	START	U	Strain rate sensitivity control. (Input 6)
J	[J]	T		U	General index. Occasionally mechanical sublayer.
JCHK(I)		C,I,S ₀	START	U	Print controls. (Input 8)
JCYNLP(I)		C,I,S ₀	START	U	Time cycles to print LMAT and LMATN. (Input 10)
J1,J2		T	STRAIN	U	Temporary indices equal to NI1(I) and NI2(I).
J3		T	STRAIN	R	Temporary index equal to NI3(I).
K	[k]	T		U	General index. Frequently integration stations.
KD	[k]	D	DGEOM	U	Dummy argument for station K in STRESS and BMSTRS.
KEY		D	(none)	U	Dummy argument in WRTAPE. WRTAPE is not called.
KJ		T	STRESS	U	Index. J'th mechanical sublayer at K'th station.
KJMAX		C,S ₀	START	U	Maximum KJ. KJMAX = NSFL*SLAYER.
KN		T	STRESS	U	KN = NSFL*(K-1). So, KJ = KN + J. (BMSTRS)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
L	[1]	T	STRESS	U	Number of subincrements for stress increment.
L		T	BMSTRS	R	L=0, or L=1 + INT(4.0*(SIG22 /SIGYZ - 1.))
L		T		R	General index.
LABELV(1)		I,S ₀	ENDFRC	APLFR	Label for force table. (Input 18)
LABLET(1)		I,S ₀	ERODE	EROD	Label for XEND vs. TEROD table. (Input 19)
LAYER	{L} [K]	C,I,S ₀	START	U	Number of integration stations through thickness. (Input 2)
LC		T	STRESS	U	Counter in STRESS. 1 ≤ LC ≤ L+1.
LEPSX(1)		S ₀	STRAIN	R	Label for EPSX(I) in output. (Output)
LINK		T,D	STRAIN	U	Control switch in STRAIN and PDATA.
LMAT(N,K)		C	STRESS	U	Matrix of LMK at mesh points. (BMSTRS) (Output)
LMATM(N,K)		C	STRESS	R	Matrix of LMK at midmeshes. (BMSTRS) (Output)
LMK		T	STRESS	U	Maximum l, station K, mesh or midmesh N. (BMSTRS)
LOAD		C,I,S ₀	START	U	Mode of loading. (Input 5) (DAMP)
LPRESS		C,I,S ₀	START	U	Last time step to call PRESS in RPSLID and modify P(N) in DGEOM. (Input 5) (DAMP)
M		T	MOTION	APLFR	Temporary index. [Not M of REPSIL in Ref. 2.]
MAXC		C,I,S ₀	START	U	Final cycle. (Input 3) (DAMP)
MDAMP		C,I,S ₀	START	U	Time cycle at which damping begins. (Input 5)
WRITE		C,S ₀	START	U	Time cycle to store restart data. (unused)
N	[n]	T		U	Mesh number in ξ ² direction. 1 ≤ N ≤ NN < 105.
NCONT		C,I,S ₀	START	U	Initial time step. (Input 3) (Always 0)
NCYC		T	POSITN	U	NCYC = NCYCLE-1.
NCYCH(1)		C,I,S ₀	START	U	Cycles for JCHK and energy summary print. (Input 9)
NCYCLF	[2]	C,S	RPSLID	U	Time cycle. (DAMP)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
NC3DP(1)		C,I,S ₀	START	U	Time steps for "3D" plots. (Input 11) (DAMP)
ND		D	DGEOM	U	Dummy for N in STRESS, BMSTRS, GRAD, and RESULT.
NDEFX		C,S	STTAIN	R	N corresponding to DEFEX.
NDEFXT		T	STRAIN	R	N corresponding to DEFXT.
NDELP		C,S ₀	START	U	Time steps between surface strain prints.
NEPSX(1)		C,S	STRAIN	R	Mesh number of EPSX(1).
NEPSXT(1)		T	STRAIN	R	Mesh number of EPSXT(1).
NEST		C,S ₀	START	BSTRS	Number of EEB,SSB pairs in table. (NEST<100)
NETAG(1)		C,I,S ₀	START	U	Surface for I'th surface strain. (Input 13)
NF		T,I	INVEL	U	Maximum mesh point for uniform velocity. (Input 15)
NFMUS		I,S ₀	ENDFRC	APLFR	Number of EFMUT entries in table. (Input 18)
NFTIS		I,S ₀	ENDFRC	APLFR	Number of EFTHT entries in table. (Input 18)
NTZPTS		I,S ₀	ENDFRC	APLFR	Number of EFZT entries in table. (Input 18)
NI		T,I	INVEL	U	Minimum mesh point for uniform velocity. (Input 17)
NI1(1),NI2(1)		C,S ₀	START	U	Mesh points that bracket surface strain points.
NI3(1)		C,S ₀	START	R	Midmesh just less than surface strain points.
NLP		C,S	RPSLID	U	Counter for array JCYNLP(1). (DGEOM)
NLPRIN		T,I	START	U	Number of JCYNLP(1) entries. (Input 10)
NMESII		C,I,S ₀	START	U	Number of meshes. (Input 2)
NN		C,S ₀	START	U	NN = N2B+1. (NN=N2B if IBC2=4) (NN<105)
NNN		C,S	RPSLID	U	Counter indexing NCYCH(1). (MOTION)
NNPE		C,I,S ₀	START	PLOTP	Number of "PLOTP" plots to make. (Input 11a)
NN3D		C,S	PDATA	U	Counter indexing array NC3DP(1).
NNISE		T	START	R	Position for unwanted input.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
NPE(1)		C, I, S ₀	START	PLOTP	N of P(N) at which "PLOTP" plots are desired. If N>NN, P(N) may be any function of time inserted by programming. (Input 11a)
NPLOT		C, S ₀	RPSILD	U	File number for plotting data. (NPLOT = 3)
NPRINT		C, S I	STRAIN START	U U	Next time step for surface strain print. Print interval. (Input 8)
NPTS		T	PDATA	R	Number of points in "3D" plots. NPTS = N2B-N1B+1.
NQ1, NQ2		C, S ₀	START	U	Mesh points bracketing ETAD2 location.
NRITE		C, I, S ₀	START	U	Restart print interval. (Input 3) (unused)
NSFL	[J]	C, I, S ₀	START	U	Number of mechanical stress sublayers. (Input 6)
NSTRN		C, I, S ₀	START	U	Number of surface strain locations. (Input 12)
NTEPTS		I, S ₀	ERODE	EROD	Number of TEROD, XEND pairs in table. (Input 19)
NMNCY		T, I	START	U	Number of NCYCH(I) entries. (Input 9)
NV		T, I	INVEL	U	Number of points with individual velocities. (Input 15)
N1A, N2A		C, S ₀	START (ERODE)	R EROD	Limits on N for loop in DECOM. (Same as N1B, N2B) N1A increases if end point erodes (DE>DECOM.)
N1B, N2B		C, S ₀	START (ERODE)	R EROD	N at end 1, end 2. N1B = 2. N2B = NMESH+2. N1B increases if end point erodes (DE>DECOM.)
N1V, N2V		C, S ₀	START (ERODE)	R EROD	Limits on N for points that move. N1V increases if end point erodes (DE>DECOM.)
N3D		T, I	START	U	Number of time steps for "3D" plots. (Input 11)
P(N)	AP, [P, P*]	C, S	PRESS	U	Pressure or force/unit initial area at mesh point N. (DECOM) (If IB=1, P(N) is force/length.)
PDATA(1)		C, S	PDATA	PLOTP	Array for P(N), N=NPE(1), for plots.
PHIT	φ, [φ]	T	STRESS	U	Yield function.
PI	π	S ₀	STRAIN	U	3.141592653589793. (Too long for CDC)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
PLAST	$\{W, I\}$	C, S	MOTION	U	Plastic work. PLAST = TNRG-CINET-STREN-TDAMP.
PN(I)	$\{n\}$	C, S ₀	START	U	PN(I) = 2.0 + ETAG2(I)/DETA2.
PN1, PN2		T	START	U	PN1 + NI1(I), PN2 = NI2(I).
PPR(I)		I	PRESS	LINPRS	Pressure at time TPR(I). (Input 16)
PRAT	$E/(1-\nu^2)$	C, S ₀	START	U	Material constant used in STRESS.
PSR(J)		C, I, S ₀	START	U	Constant for strain rate. (Input 7)
P0		I, S ₀	PRESS	R	Pressure in some PRESS subroutines. (Input 16)
QN	$\{n\}$	C, S ₀	START	U	QN = 2 + ETAD2/DETA2. NQ1 < QN < NQ2 = NQ1+1.
QN1, QN2		C, S ₀	START	U	Interpolants for displacement at ETAD2.
QS(N)	$\{Q\}$	C, S	RESULT	SIR3/1	Axial force at midmesh N.
Q11, Q22		T	RESULT	U	Proportional to membrane components of stress resultant. (Q11 = SUM11 - BT*SUMB11)
R(N), Z(N)	$\{Y^3, Y^2\}$	C, S	INGEOM	R	Coordinate of mesh point N. $\{X = R \frac{1}{2} + Z \frac{1}{2}\}$ [Replaces $X = Y^j \frac{1}{2}$ of REPSIL] (POSITN)
RADIUS		C, S ₀	INGEOM	R	Signals Type of symmetry. If RADIUS>0, radial symmetry. (If IB>0, RADIUS = 0.0 in START.)
RD22	$\{1/(\Delta E)^2\}$	C, S ₀	START	U	Program constants for differences.
RD22M	$\{1/2/(\Delta E)^2\}$	C, S ₀	START	R	
RG	1/g	T	STRESS	U	RG = 1.0/DG:
RHO	ρ	T, I	START	U	Mass density. (Input 6)
RMSN	$\{R\}$	T	DGEOM	R	R at current mesh point or midmesh.
RNK, RNR	$\{n_k^-, n_r^-\}$	T	DGEOM	R	Components of normal at time t-Δt. [Replace RN _j]
RSUM		T	START	U	Temporary for computing Δt.
RTD2	$\{1/2/\Delta E\}$	C, S ₀	START	U	Constant for differences.
RTD2M	$\{1/\Delta E\}$	C, S ₀	START	R	Constant for differences.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
RZ(N), ZZ(N)		C, S ₀	DGEOM	R	Initial position of mesh point N.
RZ, RZ2	$[Y_2^3, Y_{22}^3]$	C, T	GRAD	R	Difference approximations of $\partial R/\partial \xi$, $\partial^2 R/(\partial \xi)^2$.
SA, SB	$[\alpha, \beta]$	T	STRAIN	U	Temporary constants for surface strains.
SEF(J)		C, S ₀	START	U	Slope of J'th segment of stress-strain curve.
SEPS(J), SSIG(J)	(ϵ_j, σ_j)	C, I, S ₀	START	U	Stress, strain at J'th corner of uniaxial stress-strain curve. (If (ϵ_s, σ_s) is a corresponding engineering stress-strain point, $\epsilon_j = \epsilon_s(1+\epsilon_s/2)$, $\sigma_j = \sigma_s(1+\epsilon_s)$ (Input 7)) $((\epsilon_1, \sigma_1) = (\text{SIGZ}/E, \text{SIGZ}))$
SIGYSQ	$(\sigma_0)^2$	T	STRESS	U	Square of dynamic yield stress.
SIGYZ	σ_0	T	BMSTRS	R	Dynamic yield stress.
SIGZ	σ_0	C, I, S ₀	START	U	Static uniaxial yield stress. (Input 6)
SIGZSQ(J)	$[(\sigma_{0j})^2]$	C, S ₀	START	U	Square of static yield stress, sublayer j.
SIGZZ(J)	$[\sigma_{0j}]$	C, S ₀	START	R	Static yield stress, sublayer j. ($\sigma_{0j} = E \cdot \text{SEPS}(J)$)
SIG1(N, KJ), SIG2(N, KJ)	σ^{11}, σ^{22}	C, S	STRESS	U	Contravariant components of stress, mesh point N, station K, sublayer J. (BMSTRS, RPSLID)
SIG1M(N, KJ), SIG2M(N, KJ)	σ^{11}, σ^{22}	C, S	STRESS	R	Contravariant components of stress, midmesh N, station K, sublayer J. (BMSTRS, RPSLID)
SIG11, SIG22	σ_1^1, σ_2^2	T	STRESS	U	Mixed components of stress. (BMSTRS)
SIG11D, SIG22D	$C_1 C_2$ σ_1^1, σ_2^2	T	STRESS	U	Mixed components of plastic flow corrector stress.
SIG111, SIG221	$\sigma_1^{-1}, \sigma_2^{-2}$	T	STRESS	U	Mixed components of stress at time t-Δt. (BMSTRS)
SIG11L, SIG22L	$T_1 T_2$ σ_1^1, σ_2^2	T	STRESS	U	Mixed components of trial stress.
SLABL, SLARW		T, I	INGEOM	R	Length, width of slab symmetric plate. (Input 14)
SNK(N), SNR(N)	(n_k, n_r)	C, S	DGEOM	R	Components of unit normal, mesh point N. (INNORM) [Replace Sni(M,N)]
SNEM(N), SNRM(N)	(n_k, n_r)	C, S	DGEOM	R	Components of unit normal, midmesh N.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
SNKN, SNRN	(n_k, n_r)	T	DGEOM	R	Components of unit normal at time t.
		T	MOTION	EAPFRC	Components of unit normal at time t+dt.
SPA	a_i^j	C, T	DGEOM	U	SRA = SQRT(DA).
SPG	g_i^j	T	STRESS	U	SRG = SQRT(DG).
SSB(1)	$\{\sigma_i\}$	C, I, S ₀	START	BSTRS	See EEB(1).
SSIG(J)	$\{\sigma_j\}$	C, I, S ₀	START	U	See SEPS(J).
SSIMN(K), SS2NN(K)	σ^{11}, σ^{22}	C	STRESS	U	Contravariant components of stress, station K. (BNSTRS)
SS11, SS22	σ_1^1, σ_2^2	T	STRESS	U	Mixed components of stress for layer.
STREN	$[V]$	C, S	STRESS	U	Elastic strain energy. (BNSTRS, DGEOM)
SUM		T	START	U	Temporary for computing Δt .
SUMAR		T	START	R	IN(K), cross-sectional area for beams.
SUM11, SUM22	$\epsilon_0^{11}, \epsilon_0^{22}$	T	RESULT	U	Approximation of $\int_{ZL}^{ZU} \sigma^{ab} d\zeta$.
SIMR11, SIMR22	$\epsilon_1^{11}, \epsilon_1^{22}$	T	RESULT	U	Approximation of $\int_{ZL}^{ZU} \sigma^{ab} d\zeta$.
SIMC11, SIMC22	$\epsilon_2^{11}, \epsilon_2^{22}$	T	RESULT	U	Approximation of $\int_{ZL}^{ZU} (\epsilon)^2 \sigma^{ab} d\zeta$.
SUMZAR		T	START	R	$\Sigma(ZETA(K))^2 * W(K)$. {Moment of inertia for a beam.}
TA		C, S ₀	START	U, R	If IB=1, TA = 1. If IGAUSS = 1, TA = THICKN/2. Otherwise, TA = $\Delta \zeta$ = THICKN/ZAYER.
TAMBDA	$\Delta \lambda$	T	STRESS	U	Factor for amount of plastic flow. (Ref. 1, App. C)
TB		T	RESULT	U	Program constant. TB = SRA*TA.
TDAMP	$[W_D]$	C, S	DAMP	U	Damping work. (RPSLID)
TDEFX		C, S	STRAIN	R	Time of DEFY, maximum displacement.
TEMP(N)		C, S ₀	DGEOM	U	Program constant for mesh point N. Used in MOTION and KINET. TEMP(N) = DELGAN/A ^{1/2} .

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
TEMPU		T	ROUND	R	Temporary. (ROUNDU)
TEPSX(I)		C,S	STRAIN	R	Time for EPSX(1), the 1'th extreme strain.
TEROD(I), XEND(I)		I,S ₀	ERODE	EROD	Table of erosion vs. time at end 1. (Input 19)
THICKN	h	T,I	START	U	Thickness of shell. (Input 6)
TIME	t	C,S	RPSLID	U	Time. (START--temporary for output) (DAMP)
TITLE(I)		I	START	U	Array for image of identification. (Input 1)
TNFZ(I)		I,S ₀	ENDFRC	APLFR	Time with FFZT(1). (Input 18)
TNSU(I)		I,S ₀	ENDFRC	APLFR	Time with FFMU(1). (Input 18)
TMTU(I)		I,S ₀	ENDFRC	APLFR	Time with EFTU(1). (Input 18)
TNRG	[M]	C,S	MOTION	U	Total energy. Initial kinetic energy plus work by external forces.
TPR(I), PPR(J)		I,S ₀	PRESS	LINPRS	Time, pressure table for PRESS. (Input 16)
V	[v]	T,I	INVEL	U	Velocity. (Input 17)
VR	[v]	T,I	INVEL	U	Velocity. (Input 17)
VR,VZ		T	MOTION	R	Forces, equations of motion. [Replace VM _i +VN _i +VF _i]
VS(N)	(V)	C,S	RESULT	SURS/1	Shearing force at midmesh N.
W(K)	(W _k)	C,S ₀	INGEOM START	R R	If IB = 1, W(K) is weight with K'th layer in beam. If IGAUSS=1, W(K) is weight with K'th Gauss point.
WIDTH(K)		T,I	INGEOM	R	Width of K'th layer of beam. (Input 14)
WT(J)		C	START	U	Weight associated with J'th mechanical sublayer.
WZETA(K)	(W _{kL})	C,S ₀	START	R	Constants for Gaussian integration.
WZTSQ(K)	(W _{kL}) ²	C,S ₀	START	R	Constants for Gaussian integration.
XEND(I)		I,S ₀	ERODE	EROD	See TEROD(I).
YLDFAC		C,I,S ₀	START	U	Parameter for plastic flow computation. (Input 2)
Z(N)		C,S	INGEOM	R	See R(N). (POSITN)

<u>NAME</u>	<u>SYMBOL</u>	<u>STATUS</u>	<u>STORED</u>	<u>ORIGIN</u>	<u>COMMENTS</u>
ZAYER		T	START	U	Floating point representation of LAYER.
ZB		T	STRAIN	R	z on upper, or lower, surface, (ZU or ZL)
ZETA(K)	$z, \{z_k\}$	C, S ₀ I	START	R R U	If IGAUSS=1, Gaussian points through thickness. If IB=1, integration points in beams. (Input 14) Otherwise, midpoints of equal layers.
ZETAK	$[2z]$	T	STRESS	U	$2.0 * ZETA(K)$.
ZETASQ(K)	$(z)^2$	C, S ₀	START	U	Square of ZETA(K).
ZL	$\{z_k\}$	C, S ₀	START	R	z on lower surface. [Replaces -H.]
ZU	$\{z_u\}$	C, S ₀	START	R	z on upper surface. [Replaces H.] (Input 14)
ZZ(N)		C, S ₀	DGEOM	R	See RZ(N).
ZZ, ZZZ	$[V_2^2, V_{22}^2]$	T	GRAD	R	Approximations for $\partial z / \partial \xi$, $\partial^2 z / (\partial \xi)^2$.

APPENDIX C

LISTING OF REPSL1D

The version of RPSL1D discussed in this report is catalogued in file RPSL1D, cycle 2, in UPDATE form at the time of this writing. There is one COMDECK called MAIN. This is listed first. The remainder of the listing is the COMPILE file image of RPSL1D formed through UPDATE with the COMDECK images replaced with the COMMENT statement *CALL MAIN HERE.

This listing gives the correct UPDATE card identifiers. It could be used for simple changes through UPDATE. For involved changes, the user would be wise to use the FTN compiler to create symbolic reference maps.

*COMDECK MAIN

COMMON W(103),Z(103),PH(103),OZ(103),FM11P(103),FM22W(103),	MAIN	2
1 FM22K(103),FN1TH(103),FN2R(103), FN2K(103),SNW(103),SNK(103),	MAIN	3
2 TEMP(103),P(103),FPSL1(103),EPS(2(103),FPSU1(103),FPSU2(103),	MAIN	4
3 SIG1(103,36),SIG2(103,36),LMAT(103,6),	MAIN	5
4 DEPS1(6),DEPS2(6),ZETA(6),ZETASQ(6),SS1M(6),SS2M(6),W(6),	MAIN	6
5 NCYCH(50),AC3DH(50),JCYNLP(50)	MAIN	7
COMMON NN, RD22,WT02,DETA1,DETA2,RADIUS,AN3D,DAMP,	MAIN	8
1 DFACT,TDAMP,TDAMP,LOAD,DEL GAM	MAIN	9
COMMON F,FN1,G,PHAT,SIGZ,GAMZ,ZU,ZI,LAYER,DELTAT,TIME,LPPES,	MAIN	10
1 NN,NCYCLF,NWITF,ACONT,ASTN,CINER,CINFS,CINFP,C1,C2,NPLUT,	MAIN	11
2 DELSQ,TA,MAXC,WHITE,CA,CH,CINET,STREN,PLAST,TANG, EPS,FAP,	MAIN	12
3 NPPINT,ADFLP,NMESH,IMCE1,IRCE2,ISH,NSFL,KJMAX,YLDFAC,NLP	MAIN	13
COMMON N11(6),N12(6),DN1(6),DN2(6),PN(6),FTAG2(6),ANGLE(6),	MAIN	14
1 ANGLP(6),NETAG(6),FPSS1(6),FPSS2(6),JCHK(3)	MAIN	15
COMMON GN,NQ1,NQ2,GN1,GN2,D1,D2,FTA02	MAIN	16
COMMON ASA(6),HSA(6),ASB(6),HSH(6),GI11(6),GI22(6),	MAIN	17
1 A111(6),A221(6),H111(6),H221(6),A112(6),A222(6),H112(6),H222(6)	MAIN	18
COMMON DSH(6),PSH(6),SSIG(6),SEPS(6),SF(6),SIGZSQ(6),WT(6)	MAIN	19
COMMON SIGZ2(6),IH	MAIN	20
COMMON F2,72,OW2,072,F22,722,OR22,DZ22	MAIN	21
COMMON A11,A22,SHA,CS211,CS222,H11,F22,BT,FM11,FM22,CINES1,CINES2	MAIN	22
COMMON IS	MAIN	23
COMMON N1R,N2R,N1V,N2V	MAIN	24
COMMON N1A,N2A,IP,HT02M,PD22M	MAIN	25
COMMON SNM(103),SNM(103),SIG1M(103,36),SIG2M(103,36),LMATM(103,6)	MAIN	26
COMMON DEFX,DEFX,TDEFX,EPSX(H),NFPX(R),TFPSX(H),	MAIN	27
1 RZ(103),ZZ(103),GU11(103),GU22(103),GL11(103),GL22(103)	MAIN	28
COMMON DP3(6),N13(6),DA1(103),DP1(103),DA2(103),DP2(103)	MAIN	29
COMMON IGAUSS,WZFTA(6),WZFTSQ(6)	MAIN	30

PROGRAM REPSIL (INPUT, OUTPUT, PLOT, TAPE3=INPUT, TAPE4=OUTPUT, 1 TAPE3=PLOT)	REPSIL0	2
*CALL MAIN HERE	REPSIL0	3
CHANGES IN SURFACE STRAIN COMPUTATIONS 5/26/76	TAPE	1
OPTIONAL GAUSSIAN INTEGRATION 4/9/76. USED IF IGAUSS=1.	REPSIL0	5
PERMIT UP TO 6 MECHANICAL SURLAYERS. NSFL .LE. 6	REPSIL0	6
(1-D)REPSIL MASTER PROGRAM FOR CLC. 10/13/77	REPSIL0	7
INGROM FOR BEAM OR SLAB SYMMETRY. PRESS FOR CONSTANT PRESSURE.	REPSIL0	8
IRCE1=1.2, OR 3. IRCE2=1.2, 3, OR 4. 8/28/74 1/7/76	REPSIL0	9
CHANGES TO PERMIT SYMMETRY AT END 1 INSERTED ON 1/7/76	REPSIL0	10
PERMIT UP TO 4 MECHANICAL SURLAYERS. NSFL .LE. 4.	REPSIL0	11
END? FREE IF IRCE2=4. 8/28/74. MAY ONLY BE GOOD FOR BEAMS.	REPSIL0	12
DIFFERENCES FOR FM CENTERED AT MESH POINTS. FOR FM IN MESHES	REPSIL0	13
TAPE (NPLOT) PLOTTING DATA	REPSIL0	14
INITIATE PROGRAM	REPSIL0	15
DO 901 L=1,4	REPSIL0	16
EPSX(L)=100000.	REPSIL0	17
901 EPSX(L+4)=100000.	REPSIL0	18
DEFX=0.0	REPSIL0	19
NPLOT=3	REPSIL0	20
REWIND 3	REPSIL0	21
NLP=1	REPSIL0	22
NN=1	REPSIL0	23
CALL START	REPSIL0	24
DELETE RESTART. (I.E. WRTAPE)	REPSIL0	25
IF (NCONT .LE. 0) GOTO 15	REPSIL0	26
CALL WRTAPE (2)	REPSIL0	27
NPRINT=(NOCYCLF+MOD(ACYCLE,NDELPH))+NHELP	REPSIL0	28
GOTO 49	REPSIL0	29
15 NOCYCLF=0	REPSIL0	30
TIME=0.0	REPSIL0	31
CINES=0.0	REPSIL0	32
TDAMP=0.0	REPSIL0	33
CINFP=0.0	REPSIL0	34
EAS=0.0	REPSIL0	35
D1=0.0	REPSIL0	36
D2=0.0	REPSIL0	37
CALL TIME	REPSIL0	38
SFT INITIAL DISPLACEMENT, PRESSURE, STRAIN AND STRESS = 0	REPSIL0	39
17 DO 28 N=1, NN	REPSIL0	40
DP(N)=0.0	REPSIL0	41
IZ(N)=0.0	REPSIL0	42
W(N)=0.0	REPSIL0	43
FPSL1(N)=0.0	REPSIL0	44
FPSL2(N)=0.0	REPSIL0	45
FPSU1(N)=0.0	REPSIL0	46
FPSU2(N)=0.0	REPSIL0	47
DO 29 K=1, KJMAX	REPSIL0	48
SIG1(N,K)=0.0	REPSIL0	49
SIG2(N,K)=0.0	REPSIL0	50
SIG1M(N,K)=0.0	REPSIL0	51
SIG2M(N,K)=0.0	REPSIL0	52
28 CONTINUE	REPSIL0	53
IF (TIME .GT. 0) CALL REPSIL	REPSIL0	54
	REPSIL0	55
	REPSIL0	56
	REPSIL0	57
	REPSIL0	58
	REPSIL0	59
	REPSIL0	60
	REPSIL0	61
	REPSIL0	62
	REPSIL0	63

```

DELGAM=DELSQ/647
CALL DGEUP
CALL STWAIN
DO 33 N=M1H,N2F
A22 = 0.5*(DA2(N)+DA2(N-1))
IF(N.EQ.N1H)A22=DA2(N)
IF(N.EQ.N2F)A22=DA2(N-1)
GU11(N) = 1.0/(DA1(N)-2.0*ZU*DP1(N))
GU22(N) = 1.0/(A22-2.0*ZU*DH2(N))
GL11(N) = 1.0/(DA1(N)-2.0*ZL*DP1(N))
GL22(N) = 1.0/(A22-2.0*ZL*DH2(N))
EA1(N) = 0.0
DA2(N-1) = 0.0
DP1(N) = 0.0
DH2(N) = 0.0
33 CONTINUE
DA2(N2F) = 0.0
EPSU2(N2F) = 0.0
EPL2(N2F) = 0.0
IF(LOAD.GT.0)GOTO 43
C
CALL INVEL
C
DO 35 N=M1V,N2V
D6(N)=DELTAT*D6(N)
D7(N)=DELTAT*D7(N)
35 CONTINUE
CALL ROUNDF
CALL KINFT
CINFS=2.0*CINFT
TMRG=CINFS
C
IF(LOAD) 42,45,43
C
42 CALL PRONK
C
43 DO 44 N=M1V,N2V
D6(N)=D6(N)-F(N)*SNR(N)*TEMP(N)
D7(N)=D7(N)-F(N)*SNK(N)*TEMP(N)
44 CONTINUE
CALL ROUNDF
CALL PRONK
FAR=FNC
CALL KINFT
TMRG=CINFT
45 IF(NCYCH(1).EQ.0)ANN=2
C
WRITE(6,300)
WRITE(6,300)
WRITE(6,410)(N,P(N),Z(N),P(N),N=1,NN)
C
49 CALL PDATA(1)
C
END INITIALIZATION
C
50 NCYCLE=NCYCLE+1
TIME=TIME+DELTAT
CHECK FOR FINAL STEP
IF(NCYCLE.GT.MAXC)GOTO 70
CHECK IF CALL PRESS IS NEEDED
IF(LPRESS.GE.NCYCLE)CALL PRESS
CALL POSITN
CALL STWAIN

```

```

WFSL10 64
WFSL10 65
WFSL10 66
WFSL10 67
WFSL10 68
WFSL10 69
WFSL10 70
WFSL10 71
WFSL10 72
WFSL10 73
WFSL10 74
WFSL10 75
WFSL10 76
WFSL10 77
WFSL10 78
WFSL10 79
WFSL10 80
WFSL10 81
WFSL10 82
WFSL10 83
WFSL10 84
WFSL10 85
WFSL10 86
WFSL10 87
WFSL10 88
WFSL10 89
WFSL10 90
WFSL10 91
WFSL10 92
WFSL10 93
WFSL10 94
WFSL10 95
WFSL10 96
WFSL10 97
WFSL10 98
WFSL10 99
WFSL10 100
WFSL10 101
WFSL10 102
WFSL10 103
WFSL10 104
WFSL10 105
WFSL10 106
WFSL10 107
WFSL10 108
WFSL10 109
WFSL10 110
WFSL10 111
WFSL10 112
WFSL10 113
WFSL10 114
WFSL10 115
WFSL10 116
WFSL10 117
WFSL10 118
WFSL10 119
WFSL10 120
WFSL10 121
WFSL10 122
WFSL10 123
WFSL10 124
WFSL10 125
WFSL10 126

```

CALL MOTION	WPSL10	127
CALL PDATA (2)	WPSL10	128
CALL DAMP	WPSL10	129
C CHECK FOR RESTART LUMP	WPSL10	130
C IF (NCYCLE .NE. NWRITE) GOTO 50	WPSL10	131
C CALL WRTAPE (1)	WPSL10	132
C NWRITE=NWRITE+NWRITE	WPSL10	133
C CALL PDATA (3)	WPSL10	134
GOTO 50	WPSL10	135
70 IF (NCYCLE .LT. 2) STOP MAIN. NCYCLE .LT.	WPSL10	136
CALL PDATA (3)	WPSL10	137
CALL PDATA (4)	WPSL10	138
STOP PROGRAM COMPLETE	WPSL10	139
300 FORMAT (1H1,20X,19HINITIAL COORDINATES,23X,4HPRESSURE//	WPSL10	140
1 3X,1HN,11X,4HP(N),21X,4HZ(N),21X,4HP(N))	WPSL10	141
400 FORMAT (14,3(2X,E23.16)))	WPSL10	142
END	WPSL10	143
SUBROUTINE START	START	2
C *CALL MAIN HERE	TAH	1
DIMENSION TITLE(8)	START	4
READ(5,100) TITLE	START	5
WRITE(6,140) TITLE	START	6
READ(5,105) NOUSE, NMESH, LAYER, YLDFAC	START	7
READ(5,105) MAXC, NCONT, NWRITE, DELTAT	START	8
READ(5,110) NOUSE, IPCE2, NOUSE, IPCE1	START	9
READ(5,115) LOAD, LPRESS, MDAMP, DAMPF, OFACT	START	10
READ(5,120) E, FNU, SIGZ, FHO, THICKN, NSFL, ISF	START	11
WRITE(6,160) E, FNU, SIGZ, FHO, THICKN	START	12
WRITE(6,170) NCONT, MAXC, NPRINT, NWRITE	START	13
WRITE(6,175) LAYER, NSTRN, LOAD, LPRESS	START	14
WRITE(6,180) IPCE1, IPCE2	START	15
IF (NSFL .EQ. 1 .AND. ISF .EQ. 0) GOTO 700	START	16
IF (NSFL .EQ. 0) GOTO 700	START	17
READ(5,125) (SSIG(J), SFPS(J), DSR(J), PSF(J), J=1, NSFL)	START	18
700 IF (NSFL .LT. 1) ISW=-1	START	19
IF (NSFL .LT. 1) NSFL=1	START	20
READ(5,110) NPRINT, (JCHK(J), J=1, 3)	START	21
READ(5,110) NLMCY, (NCYCH(J), J=1, NLMCY)	START	22
READ(5,110) NLMFIN, (JCYNLP(J), J=1, NLMFIN)	START	23
READ(5,110) N3D, (NC3DP(J), J=1, N3D)	START	24
SSIG(1)=SIGZ	START	25
SFPS(1)=SIGZ/F	START	26
KUMAX=LAYER*NSFL	START	27
SF(1)=E	START	28
SIGZSQ(1)=SSIG(1)**2	START	29
SIGZZ(1)=SSIG(1)	START	30
DO 795 J=1, NSFL	START	31
IF (ISF .LT. 1) GOTO 794	START	32
IF (DSF(J) .GT. 0.0 .AND. PSF(J) .GT. 0.0) GOTO 793	START	33
791 WRITE(6,792)	START	34
792 FORMAT (//46H FROM IN STRAIN HARDENING ON STRAIN RATE DATA)	START	35
STOP START. FROM IN STRAIN RATE DATA	START	36
793 PSF(J)=1.0/PSF(J)	START	37
794 IF (J .EQ. 1) GOTO 795	START	38
IF (SFPS(J) .LT. SFPS(J-1)) GOTO 791	START	39
SF(J)=(SSIG(J)-SSIG(J-1))/(SFPS(J)-SFPS(J-1))	START	40
WT(J-1)=(SF(J-1)-SF(J))/E	START	41
SIGZSQ(J)=(E*SFPS(J))**2	START	42
SIGZZ(J)=(E*SFPS(J))	START	43
795 CONTINUE	START	44
WT(NSFL)=SF(NSFL)/E	START	45
C SET UP BOUNDARY CONDITIONS	START	46
C	START	47

	N1=N2	START	44
	N2P=N1+N2	START	45
	N1V=N1+1	START	46
	N2V=N2+1	START	47
	N1A=N1V-1	START	48
	N2A=N2V-1	START	49
	IF (IHCF1 .EQ. 2) N1V=N1H	START	50
	IF (IHCF2 .EQ. 2) N2V=N2H	START	51
	N1=N2A+1	START	52
	IF (IHCF2 .EQ. 4) N1=N2H	START	53
	HEAD(5,130) ACUSE=ETAG2.NSTRN	START	54
	HEAD(5,15) (N1USE.ETAG2(I).ANGLE(I).ANGLE(I).NETAG(I).I=1.NSTRN)	START	55
	CALL INGEOM	START	56
	CALL INNOPM	START	57
C	SIGNALS TO PROGRAM FROM INGEOM.	START	58
C	AXIAL SYMMETRY (RADIUS > 0.0, IH=0).	START	59
C	SLAB SYMMETRY (RADIUS = 0.0, IF=0). REAM (IH>0).	START	60
	IS = 1	START	61
	IF (IH .GT. 0) RADIUS = 0.0	START	62
	IF (RADIUS .LE. 0.0) IS=2	START	63
	IF (IR .GT. 0) DELTA=1.0	START	64
	GN=FLOAT(N1H) + ETAG2/DELTA2	START	65
	AG1=GN	START	66
	AG2=AG1+1	START	67
	GN1=GN-FLOAT(AG1)	START	68
	GN2=FLOAT(AG2)-GN	START	69
	DO 20 I=1.NSTRN	START	70
	PN(I)=FLOAT(N1H) + ETAG2(I)/DELTA2	START	71
	N11(I)=PN(I)	START	72
	N12(I)=N11(I)+1	START	73
	PN1=N11(I)	START	74
	PN2=N12(I)	START	75
	PN1(I)=PN(I)-PN1	START	76
	PN2(I)=PN2-PN(I)	START	77
	IF (DN1(I) .GT. 0.5) GOTO 16	START	78
	DN1(I) LE 1/2.	START	79
C	N13(I) = N11(I)-1	START	80
	DN3(I) = DN1(I) + 0.5	START	81
	GOTO 17	START	82
C	DN1(I) GT 1/2	START	83
	N13(I) = N11(I)	START	84
	DN3(I) = DN1(I) - 0.5	START	85
	IF (N13(I) .GT. 1) GOTO 18	START	86
C	N13(I) LT 2. SHIFT TO 2.	START	87
	N13(I) = 2	START	88
	DN3(I) = 0.0	START	89
	GOTO 20	START	90
	IF (N13(I) .LT. N2H-1) GOTO 20	START	91
C	N13(I) GT N2H-2. SHIFT TO N2H-2.	START	92
	N13(I) = N2H-2	START	93
	DN3(I) = 1.0	START	94
	CONTINUE	START	95
	ZAYFR=LAYFR	START	96
C	PROGRAM CONSTANTS	START	97
	IF (IH .LE. 0) ZU=0.5*THICKN	START	98
	ZL=70*THICKN	START	99
	GAMZ=THICKN*PI	START	100
	ADFLP=NPFINT	START	101
	WHITE=NCNT*WHITE	START	102
	SUMAP=0.0	START	103
	SUM2AP=0.0	START	104
C	IGAUSS=1 IS SIGNAL FOR GAUSSIAN INTEGRATION THIN THICKNESS.	START	105
	IF (IH .EQ. 1) IGAUSS=0	START	106

IF (IGAUSS .NE. 1) GOTO 51	START	111
SET UP FOR GAUSSIAN INTEGRATION WITH LAYER (LAYER) POINTS	START	112
GOTO (41, 42, 43, 44, 45, 46) LAYER	START	113
41 ZETA(1) = 0.0	START	114
W(1) = 2.0	START	115
GOTO 50	START	116
42 ZETA(2) = 0.5773502691-4626*ZU	START	117
ZETA(1) = -ZETA(2)	START	118
W(1) = 1.0	START	119
W(2) = 1.0	START	120
GOTO 50	START	121
43 ZETA(3) = 0.774596669241483*ZU	START	122
ZETA(2) = 0.0	START	123
ZETA(1) = -ZETA(3)	START	124
W(1) = 0.5555555555555555	START	125
W(2) = 0.5555555555555555	START	126
W(3) = W(1)	START	127
GOTO 50	START	128
44 ZETA(4) = 0.861136311594053*ZU	START	129
ZETA(3) = 0.334441043596456*ZU	START	130
ZETA(2) = -ZETA(3)	START	131
ZETA(1) = -ZETA(4)	START	132
W(1) = 0.347854845137454	START	133
W(2) = 0.652145154862546	START	134
W(3) = W(2)	START	135
W(4) = W(1)	START	136
GOTO 50	START	137
45 ZETA(5) = 0.906179845938664*ZU	START	138
ZETA(4) = 0.538469310105683*ZU	START	139
ZETA(3) = 0.0	START	140
ZETA(2) = -ZETA(4)	START	141
ZETA(1) = -ZETA(5)	START	142
W(1) = 0.236926885056189	START	143
W(2) = 0.478628670499378	START	144
W(3) = 0.568888888888889	START	145
W(4) = W(2)	START	146
W(5) = W(1)	START	147
GOTO 50	START	148
46 ZETA(6) = 0.932489514203152*ZU	START	149
ZETA(5) = 0.691209386646256*ZU	START	150
ZETA(4) = 0.238619186083197*ZU	START	151
ZETA(3) = -ZETA(4)	START	152
ZETA(2) = -ZETA(5)	START	153
ZETA(1) = -ZETA(6)	START	154
W(1) = 0.171324492379170	START	155
W(2) = 0.360761573048139	START	156
W(3) = 0.467913934572691	START	157
W(4) = W(3)	START	158
W(5) = W(2)	START	159
W(6) = W(1)	START	160
50 CONTINUE	START	161
IF (3-K).LAYER	START	162
IF (IGAUSS .EQ. 1) GOTO 51	START	163
IF (JL .LE. 0) ZETA(K) = ZU * (1.0 - (2.0 * FLUAT(K) - 1.0) / ZAYFW)	START	164
51 CONTINUE	START	165
SUMAP = SUMAP + W(K)	START	166
ZETASQ(K) = ZETA(K) ** 2	START	167
SUM2AP = SUM2AP + W(K) * ZETASQ(K)	START	168
WZETA(K) = W(K) * ZETA(K)	START	169
WZETSQ(K) = ZETA(K) * WZETA(K)	START	170
3 CONTINUE	START	171
TA = THICKN / ZAYFW	START	172
IF (JL .GT. 0) JAP = 1.0	START	173

IF (ICAUSS .EQ. 1) TA=20	START	174
IF (IH .GT. 0) GAMZ=H*0.5*SLMAN	START	175
C TIME INCREMENT BY VAN NEUMAN	START	176
C=SQRT(5/H*H0)	START	177
IF (IR .LE. 0) C=C/SQRT(1.0-FNU**2)	START	178
IFLN=DELTA2/C	START	179
CTW=(1.0-1.0/7AYFH**2)*THICK**2/12.0	START	180
IF (IH .GT. 0) CTWU=SUMPAH/SUMPH	START	181
IFLR= 0.5*DELTA2**2/(C*SQRT(CTW0))	START	182
IF (ICAUSS .EQ. 1) DELH = SQRT(3.)*DELTA2**2/(C*THICK)	START	183
IFLMIN=AMIN1(DELH,DELM)	START	184
DELMIN= 0.95*AMIN1(DELH,DELM)	START	185
SIM=1.0	START	186
25 RSUM=1.0/SUM	START	187
SUM=SIM*10.0	START	188
IF (RSUM .LT. DELMIN) GOTO 30	START	189
GOTO 25	START	190
30 DELMIN=AMIN1(DELMIN*SUM)/SUM	START	191
DELIN=DELTAT	START	192
IF (DELIN .GT. 0.0) DELMIN=AMIN1(DELIN,DELMIN)	START	193
DELTAT=IFLMIN	START	194
DELSU=DELTAT**2	START	195
G=.5*F/(1.0-FNU)	START	196
PHAT=F/(1.0-FNU**2)	START	197
WTU2=1.0/(2.0*DELTA2)	START	198
WU22=1.0/DELTA2**2	START	199
WTU2M=1.0/DELTA2	START	200
WU22M=0.5/DELTA2**2	START	201
CA=DELTA1*DELTA2	START	202
CF = 0.5*TA*DELTA1*DELTA2/E	START	203
DAMPING CLASTICS	START	204
C2=2.0*DELTAT*DAMP/GAMZ	START	205
C1=C2/(4.0+C2)	START	206
WRITE (6,130)	START	207
WRITE (6,140) TITLE	START	208
WRITE (6,150) DELTA1, NMESH,DELTA2	START	209
WRITE (6,300) DELH,DELM,DELIN,DELTAT	START	210
WRITE (6,160) F,FNU,SIGZ,RHO,THICK	START	211
WRITE (6,170) ACNT,MAXC,NPHINT,WRITE	START	212
WRITE (6,175) LAYER,ASTRA,LOAD,LPHFSS	START	213
WRITE (6,180) INCE1,INCE2	START	214
WRITE (6,185) (JCHK(I),I=1,3)	START	215
WRITE (6,190) (JCYN(I),I=1,NPHCY)	START	216
WRITE (6,500) (JCYN(I),I=1,NPHCY)	START	217
WRITE (6,195) (JCYN(I),I=1,NPHCY)	START	218
IF (ISF .EQ. -1) WRITE (6,400)	START	219
IF (NSFL .EQ. 1 .AND. ISF .EQ. 1) WRITE (6,405)	START	220
IF (NSFL .GT. 1 .AND. ISF .EQ. 0) WRITE (6,410)	START	221
IF (NSFL .EQ. 1 .AND. ISF .EQ. 0) WRITE (6,415)	START	222
IF (NSFL .GT. 1 .AND. ISF .EQ. 1) WRITE (6,420)	START	223
IF (NSFL .GT. 1) WRITE (6,420) NSFL	START	224
WRITE (6,421) (J,SSIG(J),SEPS(J),DSP(J),PSH(J),J=1,NSFL)	START	225
WRITE (6,110)	START	226
TIME=DELTAT*FLOAT(NDAMP)	START	227
WRITE (6,200) NDAMP,TIME,DAMP,DEFACT	START	228
IF (INCE1 .LT. 4 .AND. INCE2 .LT. 5) RETURN	START	229
WRITE (6,96474)	START	230
STOP, START, END END DEFINITION	START	231
96474 FORMAT(20X, 'END CONDITIONS NOT ALLOWED')	START	232
15 FORMAT(4F10.4,15)	START	233
10 FORMAT(10I10)	START	234
105 FORMAT(3I5,5I12,0)	START	235

110	FORMAT(1A15)	START	237
115	FORMAT(3I5,2F12.6)	START	238
120	FORMAT(5E12.6,2I5)	START	239
125	FORMAT(4E14,7)	START	240
130	FORMAT(2E10,4,I5)	START	241
140	FORMAT(1F1,53X,15HREL REFERSIL CODE//24X,4A10//)	START	242
150	FORMAT(43X,74HCOMPUTATIONS FOR ENERGY USE DETA1=E12.6/	START	243
1	3HX,14,25H MESHES IN ETA2 DIRECTION,3X,7H(ETA2=E12.6,1H//)	START	244
160	FORMAT(32X,17HYOUNG'S MODULUS =E12.6	START	245
1	/32X,17HPOISSON'S RATIO =E12.6,10X,17HYIELD STRESS =E12.6	START	246
2	/32X,17HMASS DENSITY =E12.6,10X,17HTHICKNESS =E12.6	START	247
3/		START	248
170	FORMAT(55X,1HSTART AT TIME STEP15/55X,1HFINAL TIME STEP 15/	START	249
1	55X,21HSURFACE STRAINS EVERY15,10H TIME STEP/	START	250
2	55X,21HSTART WHITE EVERY15,10H TIME STEP/)	START	251
175	FORMAT(43X,7HLAYEN =15,18X,4HSTRAN =15/	START	252
1	43X,7HLOAD =15,1HX,4HLPRESS =15/)	START	253
180	FORMAT(54X,1HBOUNDARY CONDITIONS/24X,11/2/3/4/ = CLAMPED//	START	254
1	1HSYMMETRY/HINGED/FREE/	START	255
2	50X,1HEND1 (HCE4) =1,14/50X,1HEND2 (HCE2) =1,14/)	START	256
185	FORMAT(50X,25HPRINT OPTION CONTROL CARD/52X,20HNO/I = NO PRINT/PHIN	START	257
1	17/50X,14,24H DISPLACEMENT INCREMENTS/	START	258
2	50X,14,32H CARTESIAN COORDINATES, HPRESSURE/	START	259
3	50X,14,33H SURFACE NORMAL VECTOR COMPONENTS/)	START	260
190	FORMAT(24X,45HPRINT INFORMATION AT THE FOLLOWING TIME STEPS/(24X,	START	261
1(1A15)))		START	262
195	FORMAT(24X,3H3-D PLOTS FOR THE FOLLOWING TIME STEPS/(24X,(1A15)))	START	263
200	FORMAT(44X,29HSTART DAMPING AFTER TIME STEP15,5X,4HTIME =E10.4/	START	264
140X,7HDAWPF =E10.4,16X,7HDFACT =E10.4)		START	265
300	FORMAT(47X,25HENDING TIME INCREMENT= E12.6/47X,25HMEMBRANE TIME	START	266
1	INCREMENT= E12.6/47X,25HINPUT TIME INCREMENT= E12.6//44X,31HTIM	START	267
2	INCREMENT USED BY REFERSIL= E12.6/)	START	268
400	FORMAT(1/37X,32HCONSTITUTIVE RELATION ELASTIC)	START	269
405	FORMAT(1/37X,7HCONSTITUTIVE RELATION ELASTOPLASTIC-NO WORK HARD	START	270
1	ENING-STRAIN RATE DEPENDENT)	START	271
410	FORMAT(1/37X,77HCONSTITUTIVE RELATION ELASTOPLASTIC-WORK HARDENI	START	272
1	INC-STRAIN RATE INDEPENDENT)	START	273
415	FORMAT(1/37X,80HCONSTITUTIVE RELATION ELASTOPLASTIC-NO WORK HARDI	START	274
1	NING-STRAIN RATE INDEPENDENT)	START	275
420	FORMAT(1/37X,75HCONSTITUTIVE RELATION ELASTOPLASTIC-WORK HARDENI	START	276
1	INC-STRAIN RATE DEPENDENT)	START	277
500	FORMAT(24X,49HPRINT L MATRIX (LMAT) AT THE FOLLOWING TIME STEPS/	START	278
1(24X,1A15)))		START	279
620	FORMAT(35X,31HSTRESS-STRAIN APPROXIMATION HAS,13,10H SUFFICES)	START	280
621	FORMAT(44X,40HSTRESS-STRAIN AND STRAIN RATE PARAMETERS/	START	281
130X,1HJ,4X,7HSSIG(J),4X,7HSEPS(J),14X,4HDSH(J),4X,4H1/PSH(J)/		START	282
2(26X,15,6X,1H2H15,7,5X,2E15,7))		START	283
END		START	284
SUBROUTINE INVEL		INVEL	2
C	CALL MAIN HERE	TAH	1
C	EVALUATE THE INITIAL VELOCITY AT TIME=0 FOR ALL MESH POINTS	INVEL	4
C	INPUT INDICES RELATIVE TO END1 POINT = 1	INVEL	5
	IFAD(5,100)NI,NF,VH,NV	INVEL	6
	N1=N1+N1F=1	INVEL	7
	NF=NF+N1F=1	INVEL	8
	WRITE(6,200) NI,NF,VH	INVEL	9
C		INVEL	10
	DO 30 N=N1,NF	INVEL	11
	DF(N)=VH*SNH(N)	INVEL	12
	DZ(N)=VH*SNK(N)	INVEL	13
30	CONTINUE	INVEL	14
	IF(NV.NE.0)GO TO 50	TAH	15
	WRITE(6,300)	INVEL	16

```

      DO 45 K=1,NV
      READ(5,END=N,V
      N=N+1)F=1
      WRITE(6,400)N,V
      U(N)=V*SMU(N)
      DZ(N)=V*SMK(N)
45 CONTINUE
50 RETURN

C
100 FORMAT(10X,2F5.4,12.4,15)
200 FORMAT(1H1.40X,2H2=.13, 10,13, 15H VELOCITY (VM)=.F12.6/)
300 FORMAT(52X,27HOTHER VELOCITY DISTRIBUTION/63X,1HN,4X,1HV/)
400 FORMAT(59X,15.2X,E23.14)
500 FORMAT(5X,15,F12.6)
      END
      SUBROUTINE POSITN
      *CALL MAIN HERE
C
      DO 50 N=1,NN
      P(N)=P(N)+DN(N)
      Z(N)=Z(N)+DZ(N)
50 CONTINUE
      IF(NCYCLE .NE. NCYCH(MN))GOTO 75
      IF(JCHK(1).LE.0.AND.JCHK(2).LE.0)GOTO 75
C
      WRITE DISPLACEMENT INCREMENTS, COORDINATES, AND PRESSURE
      NCYC=NCYCLE-1
      WRITE(6,101)NCYCLE,TIME
      WRITE(6,102)NCYC,NCYCLE
      WRITE(6,103)
      WRITE(6,100) (N,DH(N),DZ(N),H(N),Z(N),P(N),N=1,NN)
75 RETURN
100 FORMAT((15,5(2X,F23.16)))
101 FORMAT(10H1TIME STEP,15,4X,4HTIME,F16.8)
102 FORMAT(/6X,36HDISPLACEMENT INCREMENTS BETWEEN T.S.,14,4F AND,14,
1 21X,11HCOORDINATES,27X,8HMPRESSURE/)
103 FORMAT(4X,1HN,12X,5HDH(N),20X,5HDZ(N),21X,4FH(N),21X,4FZ(N),21X,4F
1P(N))
      END
      SUBROUTINE DEFORM
      *CALL MAIN HERE
C
      STREN=0.0
      A11=1.0
      H11=0.0
      CS211=0.0
      CS11=0.0
      DH11=0.0
      DO 41 IP=1,2
C
      IP=1 DENOTES MESH POINTS, IP=2 DENOTES MESHES
      DO 90 N=N1A,N2A
      IF(IP .EQ. 2)GOTO 12
      HNSN=H(N)
      FNSN=FN(N)
      HNS=SNH(N)
      FNS=SNK(N)
      GOTO 13
12 HNS=SNH(N)
      FNS=SNK(N)
C
      AVERAGE RADIUS AND CHANGE OF RADIUS, IN MESH
      RMSN=0.5*(H(N)+H(N+1))
      DEFSN=0.5*(D4(N)+D4(N+1))
13 CONTINUE
      CALL GRAP(N)
      IF(ITS .EQ. 1)A11=RMSN**2

```

```

INVEL 17
INVEL 18
INVEL 19
INVEL 20
INVEL 21
INVEL 22
INVEL 23
INVEL 24
INVEL 25
INVEL 26
INVEL 27
INVEL 28
INVEL 29
INVEL 30
INVEL 31
POSITN 2
TAF 1
POSITN 4
POSITN 5
POSITN 6
POSITN 7
POSITN 8
POSITN 9
POSITN 10
POSITN 11
POSITN 12
POSITN 13
POSITN 14
POSITN 15
POSITN 16
POSITN 17
POSITN 18
POSITN 19
POSITN 20
POSITN 21
POSITN 22
POSITN 23
DEFORM 2
TAF 1
DEFORM 4
DEFORM 5
DEFORM 6
DEFORM 7
DEFORM 8
DEFORM 9
DEFORM 10
DEFORM 11
DEFORM 12
DEFORM 13
DEFORM 14
DEFORM 15
DEFORM 16
DEFORM 17
DEFORM 18
DEFORM 19
DEFORM 20
DEFORM 21
DEFORM 22
DEFORM 23
DEFORM 24
DEFORM 25
DEFORM 26

```

I=SGHT(A22)	DEFCM	28
PA=A11*22	DEFCM	29
SA2=SGHT(PA)	DEFCM	30
SA2N=722/1	DEFCM	31
SA2N=-422/1	DEFCM	32
IF(IP,FG,2)GOTO 22	DEFCM	33
IF(I=1)ENOTES WESH POINTS. IF=2 ENOTES PESHES	DEFCM	34
IF(LPRESS,GF,NCYCLE)P(N)=SMA*P(N)	DEFCM	35
IF(N,EG,N1P,AND,INCE1,EG,1)GOTO 20	DEFCM	36
IF(N,EG,N2P,AND,INCE2,EG,1)GOTO 20	DEFCM	37
SAW(N)=SMA*P	DEFCM	38
SNK(N)=SMA*P	DEFCM	39
GOTO 23	DEFCM	40
20 SAWN=SAW(N)	DEFCM	41
SAKN=SNK(N)	DEFCM	42
GOTO 23	DEFCM	43
22 SAWM(N)=SAWN	DEFCM	44
SAKM(N)=SAKN	DEFCM	45
23 CONTINUE	DEFCM	46
IF(IS,FG,1)F11=-SAWN*PMSN	DEFCM	47
H22=SAWN*H22+SNKN*722	DEFCM	48
IF(NCYCLE,GT,0)GOTO 44	DEFCM	49
DA2(N)=A22	DEFCM	50
IF(IP,FG,2)GOTO 40	DEFCM	51
IF(P(N)=FELGA/SMA	DEFCM	52
PA1(N)=H11	DEFCM	53
PH1(N)=H11	DEFCM	54
PH2(N)=H22	DEFCM	55
H7(N)=H(N)	DEFCM	56
72(N)=Z(1)	DEFCM	57
GOTO 40	DEFCM	58
44 AH11=A22/PA	DEFCM	59
AH22=A11/PA	DEFCM	60
HM11=AH11*P11	DEFCM	61
HM22=AH22*H22	DEFCM	62
HT=HM11+HM22	DEFCM	63
IF(IS,FG,1)CS211=-HMSA*H2*AH22	DEFCM	64
CS222=(H2*H22+72*722)*AH22	DEFCM	65
PA12=-(HM*H22+HMK*722)	DEFCM	66
DNH2=DNH2*AH22	DEFCM	67
DN=DNH2*DNH2/(1.0+PA*P*SAWN+HMK*SAKN)	DEFCM	68
PAW=H22*PA12+DN*SAWN	DEFCM	69
PAK=722*H22+PA*SAKN	DEFCM	70
IF(IS,FG,1)H11=(H11+HMSA*H2*H22)	DEFCM	71
PA22=H22*(H22-0.5*H22)+H22*(722-0.5*722)	DEFCM	72
IF(IS,FG,1)H11=-(HMSA*H2*H22+HMSA*PAK)	DEFCM	73
H22=H22*(H22+HMK*722+DNH2*H22+PAK*722)	DEFCM	74
DO 40 K=1,LAYER	DEFCM	75
DEFS1(K)=PA11-7*TA(K)*H11	DEFCM	76
DEFS2(K)=PA22-7*TA(K)*H22	DEFCM	77
IF(IM,GT,0)CALL FMSTES(N,K)	DEFCM	78
IF(IP,IF,0)CALL STRESS(N,K)	DEFCM	79
COOING 40 TO 100 STORES BEST ESTIMATE OF SURFACE STRAINS.	DEFCM	80
40 CONTINUE	DEFCM	81
IF(IP,FG,2)GOTO 140	DEFCM	82
PA1(N)=PA1(N)+PA11	DEFCM	83
UP1(N)=PA1(N)+PH11	DEFCM	84
PH2(N)=PA2(N)+PH22	DEFCM	85
FPSL1(N)=FPSL1(N)+PA11-7L*PH11	DEFCM	86
FPSL2(N)=FPSL2(N)-7L*PH22	DEFCM	87
FPSU1(N)=FPSU1(N)+PA11-7L*PH11	DEFCM	88
FPSU2(N)=FPSU2(N)-7L*PH22	DEFCM	89
GOTO 140	DEFCM	90

```

140 CONTINUE
  IF (M.EQ. N1A) PA22H = PA22
  PA2(N) = PA2(N) + PA22
  PA22H = 1.E-8*(PA22H+PA22)
  EPSL2(N) = EPSL2(N) + PA22H
  EPSU2(N) = EPSU2(N) + PA22H
  PA22H = PA22
  CALL RESULT(N)
  IF (M.EQ. N2A-1) GOTO 41
  GOTO 90
190 CONTINUE
  CALL RESULT(N)
90 CONTINUE
41 CONTINUE
  N=N2A
  EPSL2(N) = EPSL2(N) + PA22H
  EPSU2(N) = EPSU2(N) + PA22H
  CALL SYNTAX
  IF (NCYCLE.EQ.0) GOTO 420
  STREN = (4*STREN)
  IF (I=CF1.EQ.2) STREN = 2.0*STREN
  IF (I=CF2.EQ.2) STREN = 2.0*STREN
  IF (NCYCLE.NE. NCYCLP(NLP)) GOTO 420
  C
    WRITE (MATRIX
  NLP=NLP+1)
  WRITE (F,11) NCYCLE, TIME, (N,N=N1A,N2A)
  DO 402 K=1,LAYER
    WRITE (6,413) K, (LMAT(N,K),N=N1A,N2A)
  402 WRITE (6,413) K, (LMAT(N,K),N=N1A,N2A)
  420 IF (NCYCLE.NE. NCYCLP(NLP)) GOTO 700
  C
    WRITE SURFACE NORMAL VECTOR
    IF (JCHK(3)) 700,700,400
  400 WRITE (6,400) NCYCLE, TIME
    WRITE (6,410)
    WRITE (6,420) (N,SN(N),SNK(N),SNHM(N),SNKM(N),N=N1A,N2A)
  C
    TEMPORARY PRINT OF SURFACE STRAINS
    WRITE (6,770)
    DO 71 N=N1A,N2A
      EPCL2 = SCLT(1.0+2.0*EPSU2(N)*GL22(N))-1.0
      EPCL1 = SCLT(1.0+2.0*EPSL1(N)*GL11(N))-1.0
      EPCL2 = SCLT(1.0+2.0*EPSL2(N)*GL22(N))-1.0
      EPCL1 = SCLT(1.0+2.0*EPSL1(N)*GL11(N))-1.0
      WRITE (6,771) N, EPCL1, EPCL2, EPSU1(N), EPSL2(N)
    1
      EPCL1 = EPCL1/2, EPCL2 = EPCL2/2
  71 CONTINUE
    WRITE (6,770)
  770 FORMAT(//1 SURFACE STRAINS, DEFFH/LOFFH, (ECL,EC2,E11,E22),/,//)
  771 FORMAT(15,104F20.13,/,5X,4F20.13)
  700 RETURN
  C
  411 FORMAT(10H0TIME STEP,15.6X,4HTIME,1F16.7,1E+.10X,
    1 4H5H6DIVISIONS OF TIME INCREMENT IN STRESS//20A,
    2 4HLMAT(N,K)//9H K N=4013/(4X,4013))
  413 FORMAT(//16.5X,4013/(4X,4013))
  400 FORMAT(10H0TIME STEP,15.6X,5H TIME,F16.7/)
  410 FORMAT(13.72HSURFACE NORMAL VECTOR COMPONENTS //
    1 3X,16A,11X,4HSN(N),14X,6HSNK(N),16X,7HSNHM(N),16X,7HSNKM(N))
  420 FORMAT(//16.4X(2X,F20.14))
  END
  SUBROUTINE GMAP(ND)
  C
    CALL MAP HERE
  C
    T051=1.2,043, T052=1.2,043, T053=1.2,043, T054=1.2,043
  C
    I=1 I=NDOTES MORE POINTS, I=2 I=NDOTES LESS

```

DEFCM	91
DEFCM	92
DEFCM	93
DEFCM	94
DEFCM	95
DEFCM	96
DEFCM	97
DEFCM	98
DEFCM	99
DEFCM	100
DEFCM	101
DEFCM	102
DEFCM	103
DEFCM	104
DEFCM	105
DEFCM	106
DEFCM	107
DEFCM	108
DEFCM	109
DEFCM	110
DEFCM	111
DEFCM	112
DEFCM	113
DEFCM	114
DEFCM	115
DEFCM	116
DEFCM	117
DEFCM	118
DEFCM	119
DEFCM	120
DEFCM	121
DEFCM	122
DEFCM	123
DEFCM	124
DEFCM	125
DEFCM	126
DEFCM	127
DEFCM	128
DEFCM	129
DEFCM	130
DEFCM	131
DEFCM	132
DEFCM	133
DEFCM	134
DEFCM	135
DEFCM	136
DEFCM	137
DEFCM	138
DEFCM	139
DEFCM	140
DEFCM	141
DEFCM	142
DEFCM	143
DEFCM	144
DEFCM	145
DEFCM	146
DEFCM	147
DEFCM	148
DEFCM	149
GMAP	2
TAP	1
GAFF	4
GAFF	5

C	DIFFERENCES FOR FM (ENTERED) AT MESH POINTS. FOR FM IN MESHES	GFAD	6
	IF (10, 10, 2) GOTO 10	GFAD	7
	IF (10, 10, 3) GOTO 9	GFAD	8
	IF (10, 10, 4) GOTO 9	GFAD	9
C	SPECIAL CODING FOR N=NN (A FREE END)	GFAD	10
	R2=RT(2)*(W(N-2)-4.0*W(N-1)+3.0*W(N))	GFAD	11
	Z2=ZT(2)*(Z(N-2)-4.0*Z(N-1)+3.0*Z(N))	GFAD	12
	OW2=WT(2)*(O(N-2)-4.0*O(N-1)+3.0*O(N))	GFAD	13
	U2=UT(2)*(U(N-2)-4.0*U(N-1)+3.0*U(N))	GFAD	14
	R22=RD22*(2.0*W(N)-5.0*W(N-1)+4.0*W(N-2)-R(N-3))	GFAD	15
	Z22=ZD22*(2.0*Z(N)-5.0*Z(N-1)+4.0*Z(N-2)-Z(N-3))	GFAD	16
	OW22=OD22*(2.0*O(N)-5.0*O(N-1)+4.0*O(N-2)-O(N-3))	GFAD	17
	U22=UD22*(2.0*U(N)-5.0*U(N-1)+4.0*U(N-2)-U(N-3))	GFAD	18
	GOTO 20	GFAD	19
C	CONTINUE	GFAD	20
	R2=RT(2)*(R(N+1)-W(N-1))	GFAD	21
	Z2=ZT(2)*(Z(N+1)-Z(N-1))	GFAD	22
	OW2=WT(2)*(O(N+1)-O(N-1))	GFAD	23
	U2=UT(2)*(U(N+1)-U(N-1))	GFAD	24
	R22=RD22*(W(N-1)-2.0*W(N)+R(N+1))	GFAD	25
	Z22=ZD22*(Z(N-1)-2.0*Z(N)+Z(N+1))	GFAD	26
	OW22=OD22*(O(N-1)-2.0*O(N)+O(N+1))	GFAD	27
	U22=UD22*(U(N-1)-2.0*U(N)+U(N+1))	GFAD	28
	GOTO 20	GFAD	29
10	R2=RT(2)*(R(N+1)-R(N))	GFAD	30
	Z2=ZT(2)*(Z(N+1)-Z(N))	GFAD	31
	OW2=WT(2)*(O(N+1)-O(N))	GFAD	32
	U2=UT(2)*(U(N+1)-U(N))	GFAD	33
	IF (N, 10, 1) GOTO 10	GFAD	34
C	SPECIAL CODING FOR N=NN (A FREE END)	GFAD	35
	R22=RD22*(3.0*W(N+1)-7.0*W(N)+5.0*W(N-1)-R(N-2))	GFAD	36
	Z22=ZD22*(3.0*Z(N+1)-7.0*Z(N)+5.0*Z(N-1)-Z(N-2))	GFAD	37
	OW22=OD22*(3.0*O(N+1)-7.0*O(N)+5.0*O(N-1)-O(N-2))	GFAD	38
	U22=UD22*(3.0*U(N+1)-7.0*U(N)+5.0*U(N-1)-U(N-2))	GFAD	39
	GOTO 20	GFAD	40
10	CONTINUE	GFAD	41
	R22=RD22*(W(N-1)-R(N)-R(N+1)+W(N+2))	GFAD	42
	Z22=ZD22*(Z(N-1)-Z(N)-Z(N+1)+Z(N+2))	GFAD	43
	OW22=OD22*(O(N-1)-O(N)-O(N+1)+O(N+2))	GFAD	44
	U22=UD22*(U(N-1)-U(N)-U(N+1)+U(N+2))	GFAD	45
20	RETURN	GFAD	46
	END	GFAD	47
	SUBROUTINE STRESS(MD,KP)	STRESS	2
	CALL STRESS	STRESS	3
C	CONSTITUTIVE RELATION--LINEARLY ELASTIC, OR PLASTIC--(PERFECTLY	STRESS	4
C	PLASTIC OR--STRAIN HARDENING). OPTIONAL STRAIN RATE DEPENDENCE	STRESS	5
	N=ND	STRESS	6
	K=KD	STRESS	7
	SS11=0.0	STRESS	8
	SS22=0.0	STRESS	9
	LMNK=0	STRESS	10
	KN=(K-1)*NSFL	STRESS	11
	ZFTAK=2.0*ZETA(K)	STRESS	12
	G11=SS11-ZFTAK*H11	STRESS	13
	G22=SS22-ZFTAK*H22	STRESS	14
	(G=G11)*G22	STRESS	15
	SGG=SGRT(G)	STRESS	16
	RG=1.0/DG	STRESS	17
	GH11=RG*G22	STRESS	18
	GH22=RG*G11	STRESS	19
	DEPS11=GH11*DEPS1(K)	STRESS	20
	DEPS22=GH22*DEPS2(K)	STRESS	21
	DEPS11=DEPS11*DEPS1(K)+DEPS22*DEPS2(K)	STRESS	22

DSIG22 = PRAT*(DEFS22 + FNU*(DEFS11)	STRESS	23
IF (ISW.GT.0) EPSDGT=SQRT(DEFS11*(DEFS11-DEFS22)+	STRESS	24
1 DEFS22**2)/(ELTAT	STRESS	25
ELM03 U=1.45FL	STRESS	26
KJ=KA*J	STRESS	27
C IP=1 DENOTES MESH POINTS. IP=2 DENOTES MESHES	STRESS	28
IF (IP .EQ. 2) GOTO 22	STRESS	29
SIG111=G11*SIG1(N,KJ)	STRESS	30
SIG221=G22*SIG2(N,KJ)	STRESS	31
GOTO 23	STRESS	32
22 SIG111=G11*SIG1M(N,KJ)	STRESS	33
SIG221=G22*SIG2M(N,KJ)	STRESS	34
23 CONTINUE	STRESS	35
SIGYSQ=SIG2SQ(J)	STRESS	36
IF (ICD.GT.0) SIGYSQ=SIGYSQ*(1.0+(EPSDGT/DSR(J))*PSR(J))*2	STRESS	37
L=1	STRESS	38
LC=1	STRESS	39
SIG11L=SIG111+DSIG11	STRESS	40
SIG22L=SIG221+DSIG22	STRESS	41
IF (ISW.LT.0) GOTO 2	STRESS	42
PHIT=SIG11L*(SIG11L-SIG22L)+SIG22L**2-SIGYSQ	STRESS	43
IF (PHIT.LE.0.0) GOTO 2	STRESS	44
L=INT(YLPFAC*(SQRT((PHIT+SIGYSQ)/SIGYSQ)-1.0))+1	STRESS	45
100 SIG11=SIG111	STRESS	46
SIG22=SIG221	STRESS	47
IF (L.FU.1) GOTO 3	STRESS	48
LC=1	STRESS	49
FLOAT1=.0/FLOAT(L)	STRESS	50
DSG11L = DSIG11*FLOAT1	STRESS	51
DSG22L = DSIG22*FLOAT1	STRESS	52
101 SIG11L= SIG11+DSG11L	STRESS	53
SIG22L= SIG22+DSG22L	STRESS	54
PHIT=SIG11L*(SIG11L-SIG22L)+SIG22L**2-SIGYSQ	STRESS	55
IF (PHIT .GT. 0.0) GOTO 3	STRESS	56
C PLASTIC	STRESS	57
2 SIG11 = SIG11L	STRESS	58
SIG22 = SIG22L	STRESS	59
GO TO 9	STRESS	60
C PLASTIC	STRESS	61
3 SIG11D = (2.0-FNU)*SIG11-(1.0-2.0*FNU)*SIG22	STRESS	62
SIG22D = (2.0-FNU)*SIG22-(1.0-2.0*FNU)*SIG11	STRESS	63
AA=SIG11L*(SIG11D-SIG22D)+SIG22D**2	STRESS	64
W=- (SIG11L*(2.0*SIG11D-SIG22D)+SIG22L*(2.0*SIG22D-SIG11L))	STRESS	65
B=W**2-4.*AA*PHIT	STRESS	66
IF (L .GT. LMAX) LMAX=L	STRESS	67
IF (AA)A=16.4	STRESS	68
M WHITE(A,10)	STRESS	69
10 FORMAT(1H .4X.14MAA NEGATIVE AT)	STRESS	70
GOTO 12	STRESS	71
4 IF (D .LT. 0.0 .OR. F .GT. 0.0) GOTO 16	STRESS	72
TAMHDA=(-F-SQRT(D))/(2.*AA)	STRESS	73
SIG11 = SIG11L-TAMHDA*SIG11D	STRESS	74
SIG22 = SIG22L-TAMHDA*SIG22D	STRESS	75
4 LC=LC+1	STRESS	76
IF (LC-L) 101,101,102	STRESS	77
16 I=I+1	STRESS	78
IF (L .LE. 100) GOTO 100	STRESS	79
WRITE(A,104)	STRESS	80
104 FORMAT(1H .4X.36STRESS CALCULATION UNSATISFACTORY AT)	STRESS	81
12 WRITE(A,105) NCYCLE,N,K,J,L,LC	STRESS	82
105 FORMAT(10X,4HTIME STEF=14.5X,2HN=13.5X,2HN=12.5X,2HJ=13.5X,2HL=13.	STRESS	83
5X,2HLC=12)	STRESS	84
WRITE(A,106) EPS11(N),EPS22(N),EPS11(N),EPS22(N),DEFS1(N),	STRESS	85

```

1 FEP52(N).SIG1(N,KJ).SIG2(N,KJ).DSIG11(SIG22.SIG11.SIG22.
2 SIG11.SIG22L.SIG11C.SIG22D).AA.R.PHIT
100 FOMAT(10X.7HEP51=F15.R.3X.7HEP52=F15.R.3X.7HEP51=F15.R.
1 3X.7HEP52=F15.R.10X.7HEP51=F15.R.3X.7HEP52=F15.R.3X.
2 7H SIG1=F15.R.3X.7H SIG2=F15.R.10X.7HDSIG11=F15.R.3X.
3 7HSIG22=F15.R.3X.7HSIG11=F15.R.3X.7HSIG22=F15.R.10X.
4 7HSIG11=F15.R.3X.7HSIG22L=F15.R.3X.7HSIG11C=F15.R.3X.
5 7HSIG22D=F15.R.10X.4HAA=F15.R.7X.3HE=F15.R.4X.6HPHIT=F15.R)
NC3DP(N.3D)=NCYCLE
CALL PDATA (2)
CALL PDATA (3)
CALL PDATA (4)
STOP: STRESS:
102 SS11=SS11+SIG11*WT(J)
SS22=SS22+SIG22*WT(J)
IF(IP.FG. 2)GOTO 122
C IP=1 DENOTES MESH POINTS. IP=2 DENOTES MESHES
SIG1(N,KJ)=GR11+SIG11
SIG2(N,KJ)=GR22+SIG22
GOTO R03
122 SIG1M(N,KJ)=GR11+SIG11
SIG2M(N,KJ)=GR22+SIG22
R03 CONTINUE
IF(N.FG.NH.NP.N.FG.NP)SHG=0.5*SHG
IF(IGAUS.FG. 1) SFG = W(K)*SHG
IF(IP.FG. 1)STHEN=STHEN+((SS11+SS22)**2-(1.+FNU)**2.*SS11)*SS22)*SHG
SS1M(N)=(GR11+SS11
SS2M(N)=(GR22+SS22
IF(IP.FG. 1)LMNK(A,K)=LMNK
IF(IP.FG. 2)LMATM(A,K)=LMNK
PFTURN
END
SUBROUTINE HMSTHS(NC,KD)
C *CALL MAIN HERE
C STRESS FOR BEAMS
C CONSTITUTIVE RELATION--LINEARLY ELASTIC. OR ELASTIC-(=PERFECTLY
C PLASTIC OR -STRAIN HARDENING). OPTIONAL STRAIN RATE DEPENDENCE
N=ND
K=KD
SS22=0.0
LMNK=0
L=0
KA=(K-1)*ASFL
G22 = A22-2.0*ETA(K)*G22
IF(IP.FG. 1)SFG = SCRT(G22)
GR22 = 1.0/G22
DEPS22=GR22*DEPS2(K)
DSIG22=DEPS22
DO R03 J=1,ASFL
KJ=K+J
C IP=1 DENOTES MESH POINTS. IP=2 DENOTES MESHES
IF(IP.FG. 1)SIG22I=G22*SIG2(N,KJ)
IF(IP.FG. 2)SIG22I=G22*SIG2M(N,KJ)
SIGY7=SIG22(J)
IF(LS.GT.0)SIGYZ=SIGY7*(1.+(AHS(DEPS22)/(DELTAT*DSH(J)))*PSH(J))
SIG22=SIG22I + SIG22
IF(LS.LT.0)GOTO 102
IF(AHS(SIG22).GT.SIGYZ)(=1*INT(4.0*(AHS(SIG22)-SIGYZ)/SIGYZ))
IF(L.GT.LMNK)LMNK=1
IF(SIG22.LT. -SIGY7) SIG22=-SIGY7
IF(SIG22.GT. SIGY7) SIG22 = SIGY7
102 SS22=SS22+SIG22*WT(J)
IF(IP.FG. 1)SIG22I=G22*SS22

```

	IF (IP .EQ. 2) SIG2M(N,K) = (P22*SIG22	FASTHS	33
A03	CONTINUE	FASTHS	34
	IF (N.FG.NPH .AND. .EQ. N2H) SHG = 0.5*SHG	FASTHS	35
	IF (IP .EQ. 1) STET = STET + SHG*(SS22*ZETA(K)	FASTHS	36
	SS2M(K) = (P22*SS22	FASTHS	37
	IF (IP .EQ. 1) MAT(N,K) = LMNK	FASTHS	38
	IF (IP .EQ. 2) LMA(N,K) = LMNK	FASTHS	39
	RETURN	FASTHS	40
	END	FASTHS	41
	SUBROUTINE RESULT(ND)	RESULT	2
C	*CALL MAIN HERE	TAF	1
C	THE CALCULATION OF THE STRESS AND MOMENT RESULTANTS FOLLOW	RESULT	4
C	IPCF1=1.2*CH3. IPCF2=1.2*3.0H4. H/2R/74 1/7/76	RESULT	5
	I=ND	RESULT	6
	TP=TA*SHA	RESULT	7
	IF (IR .GT. 0) GOTO 7	RESULT	8
	IF (IGAUS5 .EQ. 1) GOTO 9	RESULT	9
	SUM411 = SS1M(1)	RESULT	10
	SUM422 = SS2M(1)	RESULT	11
	SUM411 = SS1M(1)*ZETA(1)	RESULT	12
	SUM422 = SS2M(1)*ZETA(1)	RESULT	13
	SUMC11 = SS1M(1)*ZETASQ(1)	RESULT	14
	SUMC22 = SS2M(1)*ZETASQ(1)	RESULT	15
	IF (LAYFR .EQ. 1) GOTO 5	RESULT	16
	DO 4 K=2,LAYFR	RESULT	17
	SUM411 = SUM411 + SS1M(K)	RESULT	18
	SUM422 = SUM422 + SS2M(K)	RESULT	19
	SUM411 = SUM411 + SS1M(K)*ZETA(K)	RESULT	20
	SUM422 = SUM422 + SS2M(K)*ZETA(K)	RESULT	21
	SUMC11 = SUMC11 + SS1M(K)*ZETASQ(K)	RESULT	22
	SUMC22 = SUMC22 + SS2M(K)*ZETASQ(K)	RESULT	23
4	CONTINUE	RESULT	24
5	CONTINUE	RESULT	25
	C11=SUM411-RT*SUMH11	RESULT	26
	C22=SUM422-RT*SUMH22	RESULT	27
	F11=SUMH11-(RT*FM11)*SUMC11	RESULT	28
	F22=SUMH22-(RT*FM22)*SUMC22	RESULT	29
	IF (IP .EQ. 2) GOTO 17	RESULT	30
C	IP=1 DENOTES MESH POINTS. IP=2 DENOTES MESHES	RESULT	31
C	DIFFERENCES FOR FM CENTERED AT MESH POINTS. FOR FM IN MESHES	RESULT	32
	IF (N .EQ. N1H .AND. IPCF1 .EQ. 3) F22=0.0	RESULT	33
	IF (N .EQ. N2H .AND. IPCF2 .EQ. 3) F22=0.0	RESULT	34
	IF (N .EQ. NN) GOTO 16	RESULT	35
C	N=NN. ASSUME IPCF2 = 4. COMPUTE ALL RESULTANTS.	RESULT	36
	FM11(N)=TR*F11*SNH(N)	RESULT	37
	FM22(N)=0.0	RESULT	38
	FM22(N)=0.0	RESULT	39
	FM11(N)=TR*G11*H(N)	RESULT	40
C	AT FREE END. FM2J(N) = (-CM22/DETA)*NJ	RESULT	41
	IF (SNH(N-1) .EQ. 0.0) F22=FM22H(N-1)/SNH(N-1)	RESULT	42
	IF (SNH(N-1) .EQ. 0.0) F22=FM22H(N-1)/SNH(N-1)	RESULT	43
	IF (SNH(N-2) .EQ. 0.0) F11=FM22H(N-2)/SNH(N-2)	RESULT	44
	IF (SNH(N-2) .EQ. 0.0) F11=FM22H(N-2)/SNH(N-2)	RESULT	45
	F22=RTD2*(4.0*F22-F11)	RESULT	46
	FM2H(N)=F22*SNH(N)	RESULT	47
	FM2K(N)=F22*SAK(N)	RESULT	48
	GOTO 14	RESULT	49
16	CONTINUE	RESULT	50
	FM11H(N)=TH*F11*SNH(N)	RESULT	51
C	FM11H(N)=TH*F11*SNH(N) IS NOT USED	RESULT	52
	FM22H(N)=TR*F22*SNH(N)	RESULT	53
	FM22H(N)=TR*F22*SAK(N)	RESULT	54
	GOTO 14	RESULT	55

17	CONTINUE	RESULT	54
	CSM2=(CS211*F11+CS222*F22	RESULT	55
	FN1TH(N)=TR0G11*0.5*(F1(N)+F1(N+1))	RESULT	56
	FN1TH(N) AND FN112(N) ARE NOT USED IF IS=2.	RESULT	57
	FN2P(N)=T*(C22*W2+SANN(N)*CSM2)	RESULT	58
	FN2K(N)=TH*(C22*W2+SANN(N)*CSM2)	RESULT	59
18	CONTINUE	RESULT	60
	RETURN	RESULT	61
7	SUM422=0.0	RESULT	62
	SUMH22=0.0	RESULT	63
	SUMC22=0.0	RESULT	64
	DO K=1,LAYER	RESULT	65
	SUM422 = SUM422+W(K)*SS2MN(K)	RESULT	66
	SUMH22 = SUMH22 + SS2MA(K)*WZETA(K)	RESULT	67
	SUMC22 = SUMC22 + SS2MA(K)*WZETSO(K)	RESULT	68
	CONTINUE	RESULT	69
	GOTO 5	RESULT	70
9	SUM411 = 0.0	RESULT	71
	SUMH11 = 0.0	RESULT	72
	SUMC11 = 0.0	RESULT	73
	DO 10 K=1,LAYER	RESULT	74
	SUM411 = SUM411 + SS1MA(K)*W(K)	RESULT	75
	SUMH11 = SUMH11 + SS1MA(K)*WZETA(K)	RESULT	76
	SUMC11 = SUMC11 + SS1MA(K)*WZETSO(K)	RESULT	77
10	CONTINUE	RESULT	78
	GOTO 7	RESULT	79
	END	RESULT	80
	SUBROUTINE ACTION	RESULT	81
	*CALL MAIN HERE	MOTION	82
	IF (LOAD.FQ.0)GOTO 30	TAF	1
	FAS=0.0	MOTION	4
	CALL PWORK	MOTION	5
		MOTION	6
		MOTION	7
		MOTION	8
		MOTION	9
30	DO 130 N=N1V,N2V	MOTION	10
	IF (N.NF.NN)GOTO 31	MOTION	11
	SPECIAL CODING FOR N=NN (A FREE END)	MOTION	12
	VH=H022*(2.0*FM22H(N)-5.0*FM22H(N-1)+4.0*FM22H(N-2)-FM22H(N-3))	MOTION	13
	+2.0*F1(2)*F12H(N)-F12H(N-1))	MOTION	14
	V7=H022*(2.0*FM22K(N)-5.0*FM22K(N-1)+4.0*FM22K(N-2)-FM22K(N-3))	MOTION	15
	+2.0*F1(2)*F12K(N)-F12K(N-1))	MOTION	16
	IF (IS.FG.1)VR=VH-FV11H(N)-FN1TH(N)	MOTION	17
	GOTO 34	MOTION	18
31	CONTINUE	MOTION	19
	DIFFERENCES FOR FM (ENTERED AT MESH POINTS, FOR FN IN MESHES	MOTION	20
	VR=H022*(FM22H(N-1)-2.0*FM22H(N)+FM22H(N+1))	MOTION	21
	+W12H*(FN2H(N)-FN2H(N-1))	MOTION	22
	V7=H022*(FM22K(N-1)-2.0*FM22K(N)+FM22K(N+1))	MOTION	23
	+W12H*(FN2K(N)-FN2K(N-1))	MOTION	24
	IF (IS.FG.1)VR=VR-FV11H(N)-0.5*(FN1TH(N-1)+FN1TH(N))	MOTION	25
35	CONTINUE	MOTION	26
	IF (LOAD.FQ.0)GOTO 50	MOTION	27
	VR=VR-SNP(N)*P(N)	MOTION	28
	V7=V7-SNK(N)*P(N)	MOTION	29
50	GET=H(K)	MOTION	30
	DZ1=DZ(N)	MOTION	31
	DZS=DZ1+VH*TFMP(N)	MOTION	32
	DZS=DZ1+VZ*TEMP(N)	MOTION	33
	IF (TEMP.FQ.0.0)GOTO 115	MOTION	34
	VISCOSUS DAMPING	MOTION	35
	DZS=DZS-(DZS+DZ1)*C1	MOTION	36
		MOTION	37

115 DR(N)=DGS	MOTION	38
DZ(N)=DZS	MOTION	39
130 CONTINUE	MOTION	40
CALL RCHNGL	MOTION	41
CALL KINET	MOTION	42
IF (LOAD) 65,75,65	MOTION	43
65 CALL PTHK	MOTION	44
FN=0.5*(A*(FAS+FNH))	MOTION	45
FNH=FN	MOTION	46
IF (IRCF1 .EQ. 2) EN=2.0*FN	MOTION	47
IF (IRCF2 .EQ. 2) EN=2.0*FN	MOTION	48
TARG=TNRG+EN	MOTION	49
75 PLAST=TNRG-CINET-STHEN-TDAMP	MOTION	50
C	MOTION	51
IF (NCYCLE .NE. NCYCL(MNN)) GOTO 140	MOTION	52
WRITE (6,99991) NCYCLE,TIME,CINET,STHEN,PLAST,TNRG	MOTION	53
MNN=MNN+1	MOTION	54
140 RETURN	MOTION	55
C	MOTION	56
99991 FORMAT (/10H TIME STP,15.3X,5HTIME=F16.8,3X,MNN,1F10.3X,	MOTION	57
1HPELASTIC=F15.8,3X,AMPLASTIC=F15.8/14H TOTAL ENERGY=F15.8)	MOTION	58
END	MOTION	59
SUBROUTINE WRTAPE(KEY)	WETAPE	2
C	WETAPE	3
WETAPE HAS BEEN DELETED.	WETAPE	4
STOP, WETAPE DELETED.	WETAPE	5
END	WETAPE	6
SUBROUTINE STRAIN	STRAIN	2
C	STRAIN	3
*CALL MAIN HERE	STRAIN	4
C	STRAIN	5
PRINT STRAINS ON INNER OR OUTER FACES	STRAIN	6
C	STRAIN	7
DIMENSION EPSANG(6),EPSANG(6)	STRAIN	8
C	STRAIN	9
ARRAYS FOR EXTREME STRAINS --- IN STRAIN	STRAIN	10
DIMENSION EP(4),EPSXT(4),NEPSXT(4),LEPSX(4)	STRAIN	11
DATA (LEPSX(L),L=1,4)/HHEPSU1MAX,HHEPSU2MAX,HHEPSL1MAX,HHEPSL2MAX,	STRAIN	12
1 HHEPSU1MIN,HHEPSU2MIN,HHEPSL1MIN,HHEPSL2MIN/	STRAIN	13
DATA PI/3,141592653589793/	STRAIN	14
IF (NCYCLE .GT. 0) GOTO 25	STRAIN	15
C	STRAIN	16
----- INITIAL ENTRY -----	STRAIN	17
DO 20 I=1,NSTRN	STRAIN	18
J1=NT1(I)	STRAIN	19
J2=NT2(I)	STRAIN	20
J3=NT3(I)	STRAIN	21
A11 = DM1(J1) + DM1(I)*(DM1(J2)-DM1(J1))	STRAIN	22
M11 = DM1(J1) + DM1(I)*(DM1(J2)-DM1(J1))	STRAIN	23
P22 = DM2(J1) + DM1(I)*(DM2(J2)-DM2(J1))	STRAIN	24
A22 = DM2(J1) + DM1(I)*(DM2(J2)-DM2(J1))	STRAIN	25
7H = 7L	STRAIN	26
IF (METAG(I) .EQ. 1) 7H=7L	STRAIN	27
G11 = A11 - 2.0*2H*H11	STRAIN	28
G22 = A22 - 2.0*2H*H22	STRAIN	29
15 G111(I)=1.0/G11	STRAIN	30
G122(I)=1.0/G22	STRAIN	31
ANGFL=ANGLEF(I)*PI/180.0	STRAIN	32
SA=SIN(ANGFL)	STRAIN	33
SH=COS(ANGFL)	STRAIN	34
ASA(I)=2.0*SA**2	STRAIN	35
HSA(I)=2.0*SH**2	STRAIN	36
ANGF=ANGFL(I)*PI/180.0	STRAIN	37
SA=SIN(ANGF)	STRAIN	38
SH=COS(ANGF)	STRAIN	39
ASH(I)=2.0*SA**2	STRAIN	40
HSH(I)=2.0*SH**2	STRAIN	41

20	CONTINUE	STRAIN	39
	GOTO 71	STRAIN	40
C	-----	STRAIN	41
25	LINK=1	STRAIN	42
C	FIND EXTREMES OF STRAIN AND MAXIMUM DEFLECTION	STRAIN	43
	DO 401 L=1,4	STRAIN	44
	EPSX(L)=100000.	STRAIN	45
401	EPSX(L+4)=100000.	STRAIN	46
	DEFT=0.	STRAIN	47
	DO 905 N=M/4,N2H	STRAIN	48
	EPC(1)=EPSU1(N)*GU11(N) EPC(2)=EPSU2(N)*GU22(N)	STRAIN	49
	EPC(3)=EPSL1(N)*GL11(N) EPC(4)=EPSL2(N)*GL22(N)	STRAIN	50
	DO 906 L=1,4	STRAIN	51
	IF(L.GT.4)GOTO 402	STRAIN	52
	IF(EPSX(L).GE.EPC(L))GOTO 905	STRAIN	53
	EPSX(L)=EPC(L) NEPSX(L)=N GOTO 905	STRAIN	54
402	IF(EPSX(L).LE.EPC(L-4))GOTO 905	STRAIN	55
	EPSX(L)=EPC(L-4) NEPSX(L)=N	STRAIN	56
905	CONTINUE	STRAIN	57
	DO 910 L=1,4	STRAIN	58
	EPSX(L)=SQRT(1.0+2.0*EPSX(L))-1.0	STRAIN	59
	IF(L.GT.4)GOTO 906	STRAIN	60
	IF(EPSX(L).GE.EPSX(L))GOTO 910	STRAIN	61
	GOTO 907	STRAIN	62
906	IF(EPSX(L).LE.EPSX(L))GOTO 910	STRAIN	63
907	EPSX(L)=EPSX(L) TEPSX(L)=TIME	STRAIN	64
	NEPSX(L)=NEPSX(L)	STRAIN	65
910	CONTINUE	STRAIN	66
	DO 912 N=M/4,N2V	STRAIN	67
	AY3MN=(PZ(N)-Z(N))**2 + (ZZ(N)-Z(N))**2	STRAIN	68
	IF(DEFX.GE.AY3MN)GOTO 912	STRAIN	69
	DEFX=AY3MN NDEFX=N	STRAIN	70
912	CONTINUE	STRAIN	71
	DEFX = SQRT(DEFX)	STRAIN	72
	IF(DEFX.GE.DEFT) GOTO 413	STRAIN	73
	DEFX=DEFX NDEFX=NDEFX TDEFX=TIME	STRAIN	74
413	CONTINUE	STRAIN	75
C	CHECK FOR SURFACE STRAIN PRINT	STRAIN	76
	IF(NCYCLE.LT.NPRINT)GOTO 40	STRAIN	77
30	NPRINT=NPRINT+NHELP	STRAIN	78
	LINK=2	STRAIN	79
40	DO 46 I=1,NSTEP	STRAIN	80
	J1=M1(I)	STRAIN	81
	J2=M2(I)	STRAIN	82
	J3=M3(I)	STRAIN	83
	CA11 = PA1(J1) + PA1(I)*(CA1(J2)-PA1(J1))	STRAIN	84
	CB11 = PB1(J1) + PB1(I)*(CB1(J2)-PB1(J1))	STRAIN	85
	CH22 = PH2(J1) + PH1(I)*(CH2(J2)-PH2(J1))	STRAIN	86
	CA22 = PA2(J3) + PA2(I)*(CA2(J3+1)-PA2(J3))	STRAIN	87
	ZF=Z1	STRAIN	88
	IF(MFTAG(I).EQ.1)ZF=Z1	STRAIN	89
	EPSW1 = (CA11 - ZF*CH11)*G11(I)	STRAIN	90
	EPSW2 = (CA22 - ZF*CH22)*G12(I)	STRAIN	91
45	EPSS1(I)=SQRT(1.0+2.0*EPSW1)-1.0	STRAIN	92
	EPSS2(I)=SQRT(1.0+2.0*EPSW2)-1.0	STRAIN	93
	EPSAFC(I)=SQRT(1.0+PSA(I)*EPSS1+ASA(I)*EPSS2)-1.0	STRAIN	94
	EPSAFH(I)=SQRT(1.0+PSH(I)*EPSS1+ASH(I)*EPSS2)-1.0	STRAIN	95
46	CONTINUE	STRAIN	96
C	COMPONENTS OF VECTOR DISPLACEMENT	STRAIN	97
	D1=D1+(C2*DF(NU1)+C1*DF(NU2))	STRAIN	98
	D2=D2+(C2*DF(NU1)+C1*DF(NU2))	STRAIN	99
	WRITE(6,NO) NCYCLE,TIME	STRAIN	100
50		STRAIN	101

DO 70 I=1,NSTRN	STRAIN	102
ALFA=EPS1/PEW	STRAIN	103
IF (MFA1(I).EQ.1) GOTO 67	STRAIN	104
ALFA=EPS2/PEW	STRAIN	105
67 WRITE (6,65) ETAG2(I),PA(I),ALFA,EPSS1(I),EPSS2(I),	STRAIN	106
1ANG1(I),EPSANG(I),ANG1(I),EPSANG(I)	STRAIN	107
70 CONTINUE	STRAIN	108
C PRINTS OF EXTREME STRAINS AND MAXIMUM DEFLECTIONS	STRAIN	109
WRITE (6,930) TIME,NCYCLE	STRAIN	110
DO 421 L=1,4	STRAIN	111
921 WRITE (6,931) LEPSX(L),EPSXT(L),NEPSXT(L)	STRAIN	112
WRITE (6,931) ACONT,NCYCLE	STRAIN	113
DO 422 L=1,4	STRAIN	114
422 WRITE (6,933) LEPSX(L),EPSX(L),TEPSX(L),NEPSX(L)	STRAIN	115
WRITE (6,941) TIME,NCYCLE,DEFXT,NDEFXT	STRAIN	116
WRITE (6,942) ACONT,NCYCLE,UEFX,NUEFX,TUEFX	STRAIN	117
GOTO 71	STRAIN	118
930 FORMAT (//20X,'EXTREME STRAINS AT TIME=1,1PF13.6,	STRAIN	119
1 ' TIME CYCLE=1,15./41X,'STRAIN=7X,1N1)	STRAIN	120
931 FORMAT (20X,'EXTREME STRAINS BETWEEN CYCLES=15,1 AND 1,15./	STRAIN	121
1 32X,'STRAIN=9X,1TIME=9X,1N1)	STRAIN	122
932 FORMAT (25X,410.F13.6,2X,15)	STRAIN	123
933 FORMAT (16X,410.F13.6,1PF15.6,2X,15)	STRAIN	124
941 FORMAT (//10X,'MAXIMUM DEFLECTION AT TIME=1,1PF13.6,1, CYCLE=1,	STRAIN	125
1 15./13X,'DEFLECTION=1,15.6,4X,1(N = 1,12,1))	STRAIN	126
942 FORMAT (10X,'MAXIMUM DEFLECTION BETWEEN CYCLES=15,1 AND 1,15./13X,	STRAIN	127
1 'DEFLECTION=1,1PF15.6,5X,1(N = 1,12,1),10X, TIME=1,15.6)	STRAIN	128
71 RETURN	STRAIN	129
C	STRAIN	130
60 FORMAT (//10X TIME STEP,15.3X,5HTIME=1,15.6,14X,16FSURFACE STRAINS	STRAIN	131
1 .37X,19HSTRAIN GAGE READING//10X,4HETA2,14X,1HN,5X,	STRAIN	132
24HFACE,1X,7HANGLE 0,10X,5HANGLE 40,6X,5HANGLE,18X,5HANGLE/)	STRAIN	133
65 FORMAT (F15.3, F15.3, 4X ,A5,1X,2(2X,F15.6),2(2X,F6,2,2X,F15.6))	STRAIN	134
END	STRAIN	135
SUBROUTINE BOUND	BOUND	2
C CALL MAIN NEW	TAH	1
C INCE1=1,2,OR3, INCE2=1,2,3,OR4, 8/28/74 1/7/76	BOUND	4
C INSERT ONE EXTERNAL F AND Z, BOTH ENDS, 8/23/74	BOUND	5
C	BOUND	6
DELTH=Z(NIR+1) - F(NIR)	BOUND	7
DELTH=Z(NIR+1) - Z(NIR)	BOUND	8
IF (INCE1 .EQ. 1) GOTO 11	BOUND	9
IF (INCE1 .EQ. 3) GOTO 13	BOUND	10
IF (INCE1 .NE. 2) GOTO 5	BOUND	11
Z(NIR+1)=F(NIR+1)	BOUND	12
Z(NIR+1)=Z(NIR)-DELTH	BOUND	13
GOTO 20	BOUND	14
5 WRITE (6,949) INCE1,INCE2	BOUND	15
949 FORMAT (1 F40.0 IN BOUND, INCE1,INCE2 =1,2,15)	BOUND	16
STOP BOUND	BOUND	17
11 TEMPO=2.0*(DELTH*SAK(NIR)-DELTH*SAK(NIR))	BOUND	18
F(NIR+1) = F(NIR+1) + SAK(NIR)*TEMPO	BOUND	19
Z(NIR+1) = Z(NIR+1) - SAK(NIR)*TEMPO	BOUND	20
GOTO 20	BOUND	21
13 F(NIR+1) = F(NIR) - DELTH	BOUND	22
Z(NIR+1) = Z(NIR) - DELTH	BOUND	23
20 DELTH = F(NIR+1) - F(NIR)	BOUND	24
DELTH = Z(NIR+1) - Z(NIR)	BOUND	25
IF (INCE2 .EQ. 1) GOTO 21	BOUND	26
IF (INCE2 .EQ. 2) GOTO 22	BOUND	27
IF (INCE2 .EQ. 3) GOTO 23	BOUND	28
IF (INCE2 .EQ. 4) GOTO 30	BOUND	29
GOTO 5	BOUND	30
21 TEMPO=2.0*(DELTH*SAK(NIR)-DELTH*SAK(NIR))	BOUND	30

```

      6(N2H+1) = 6(N2H-1) + SNK(N2H)*TEMPU
      7(N2H+1) = 7(N2H-1) - SNK(N2H)*TEMPU
      GOTO 30
22 6(N2H+1) = 6(N2H-1)
      7(N2H+1) = 7(N2H-1)*DELTA
      GOTO 30
23 6(N2H+1) = 6(N2H) - DELTA
      7(N2H+1) = 7(N2H) - DELTA
30 RETURN
END
SUBROUTINE HQUADU
      *CALL MAIN HERE
      IHCE1=1.2,CH3. IHCE2=1.2,3,OR4. H/2H/74 1/1/74
      INSERT DR AND UZ AT ENDS. ONE EXTERNAL POINT. H/23/74
      IF (IHCE1 .EQ. 1) GOTO 11
      IF (IHCE1 .EQ. 3) GOTO 13
      IF (IHCE1 .NE. 2) GOTO 5
      DR(N1H-1)=DR(N1H+1)
      DZ(N1H-1)=-DZ(N1H+1)
      DZ(N1H)=0.0
      GOTO 20
5  WRITE(4,999) (IRCF1,IHCE2
999 FORMAT(' FROCK IN HQUADU. IHCE1,IHCE2 =',2I5)
      STOP, HQUADU
11 TEMPU=2.0*(SNK(N1H)*DZ(N1H+1)-SNK(N1H)*DR(N1H+1))
      DR(N1H-1) = DR(N1H+1) + SNK(N1H)*TEMPU
      DZ(N1H-1) = DZ(N1H+1) - SNK(N1H)*TEMPU
      DR(N1H)=0.0
      DZ(N1H)=0.0
      GOTO 20
13 DR(N1H-1) = -DR(N1H+1)
      DZ(N1H-1) = -DZ(N1H+1)
      DR(N1H)=0.0
      DZ(N1H)=0.0
20 IF (IHCE2 .EQ. 1) GOTO 21
      IF (IHCE2 .EQ. 2) GOTO 22
      IF (IHCE2 .EQ. 3) GOTO 23
      IF (IHCE2 .EQ. 4) GOTO 30
      GOTO 5
21 TEMPU = 2.0*(SNK(N2H)*DZ(N2H-1) - SNK(N2H)*DR(N2H-1))
      DR(N2H-1) = DR(N2H-1) + SNK(N2H)*TEMPU
      DZ(N2H-1) = DZ(N2H-1) - SNK(N2H)*TEMPU
      DR(N2H)=0.0
      DZ(N2H)=0.0
      GOTO 30
22 DR(N2H+1) = DR(N2H-1)
      DZ(N2H+1) = -DZ(N2H-1)
      DZ(N2H)=0.0
      GOTO 30
23 DR(N2H+1) = -DR(N2H-1)
      DZ(N2H+1) = -DZ(N2H-1)
      DR(N2H) = 0.0
      DZ(N2H) = 0.0
30 RETURN
END
SUBROUTINE SYNTHY
      *CALL MAIN HERE
      IF (IHCE2 .NE. 2) GOTO 50
      FM22P(N2H+1)=FM22P(N2H-1)
      FM22H(N2H+1)=-FM22H(N2H-1)
      FN1TH(N2H) = FN1TH(N2H-1)
      FN2K(N2H) = FN2K(N2H-1)

```

```

HQUADU 31
HQUADU 32
HQUADU 33
HQUADU 34
HQUADU 35
HQUADU 36
HQUADU 37
HQUADU 38
HQUADU 39
HQUADU 40
HQUADU 41
HQUADU 42
HQUADU 43
HQUADU 44
HQUADU 45
HQUADU 46
SYNTHY 1
SYNTHY 2
SYNTHY 3
SYNTHY 4
SYNTHY 5
SYNTHY 6
SYNTHY 7
SYNTHY 8

```

50 IF (IPCF1 .NE. 2) GOTO 60	SYNTHY	10
FM22P(N1P-1) = FM22P(N1P+1)	SYNTHY	11
FM22K(N1P-1) = -FM22K(N1P+1)	SYNTHY	12
FA1TH(N1P-1) = -FA1TH(N1P+1)	SYNTHY	13
FA2P(N1P-1) = -FA2P(N1P+1)	SYNTHY	14
FN2K(N1P-1) = -FN2K(N1P+1)	SYNTHY	15
60 RETURN	SYNTHY	16
END	SYNTHY	17
SUBROUTINE AFINIT(N)	AFINIT	2
STOP "AFINIT DELETED"	AFINIT	3
END	AFINIT	4
SUBROUTINE KINET	KINET	2
C	TAF	1
CINET=0.	KINET	4
DO 43 N=1V,N2V	KINET	5
CA=1.0	KINET	6
IF (N .EQ. N1H) CA=0.5	KINET	7
IF (N .EQ. N2H) CA=0.5	KINET	8
CINET=CINET+(CF(N)**2+FZ(N)**2)/TEMP(N) *CA	KINET	9
43 CONTINUE	KINET	10
IF (IPCF1 .EQ. 2) CINET=2.0*CINET	KINET	11
IF (IPCF2 .NE. 2) GOTO 44	KINET	12
CINET=2.0*CINET	KINET	13
44 CINES=CINES	KINET	14
CINES=0.5*CA*CINET	KINET	15
CINET=0.5*(CINES+CINER)	KINET	16
RETURN	KINET	17
END	KINET	18
SUBROUTINE PCHRK	PCHRK	2
C	TAF	1
DO 20 N=1V,N2V	PCHRK	4
CA=0.5	PCHRK	5
IF (N .EQ. N1H) CA=0.25	PCHRK	6
IF (N .EQ. N2H) CA=0.25	PCHRK	7
PS=PH(N)*SNR(N)+DZ(N)*SAK(N)	PCHRK	8
ENS=ENS-CA*PH(N)	PCHRK	9
20 CONTINUE	PCHRK	10
RETURN	PCHRK	11
END	PCHRK	12
SUBROUTINE DAMP	DAMP	2
C	TAF	1
C	DAMP	4
C	DAMP	5
C	DAMP	6
C	DAMP	7
C	DAMP	8
C	DAMP	9
101 FORMAT(/ ' INITIATED DAMPING. NCYCLE = ',I6,10X,' TIME = ',1PF15.7)	DAMP	10
LEAD = 0	DAMP	11
LEWESS = 0	DAMP	12
IF 15 N=1,NN	DAMP	13
P(N)=0.0	DAMP	14
15 CONTINUE	DAMP	15
16 CONTINUE	DAMP	16
IF (CINES-CINER) 20,20,40	DAMP	17
20 TDAMP=TDAMP+CINET	DAMP	18
WRITE(6,101) NCYCLE,CINES,CINER,CINET,TDAMP,CINER	DAMP	19
102 FORMAT(/ ' STOPPING PROCEDURE NCYCLE,CINES,CINER,CINET,TDAMP,CINER	DAMP	20
1 / 116,11PF15.7)	DAMP	21
IF (CINET-CINER .LE. DEACT*TDAMP) GOTO 50	DAMP	22
DO 30 N=1,NN	DAMP	23
DEACT=0.0	DAMP	24
DZ(N)=0.0	DAMP	25

```

30 CONTINUE
  CIAEP=CIAPT
  CINES1 = CINEW
  CINES2 = LINES
  CIAFS=0.0
  NCYCLE = NCYCLE + 1
  TIME = TIME + DELTAT
  CALL MOTION
  IF(CINES .LE. LINES)GOTO 39
  CALLOFSTEP
39 CALL PDATA(2)
40 TDAMP=TDAMP+C2*CINES
45 RETURN
50 WRITE(6,100) NCYCLE
  MAXC=NCYCLE
  NC30P(NN30)=NCYCLE
  CALL PDATA (2)
  GOTO 45
100 FORMAT(1P1.10X,30HRUN SELF-TERMINATED TIME STEP,15)
  END
  SUBROUTINE DESTEP
  C          *CALL MAIN HERE
  C          DECREASE DELTAT (IT IS TOO LARGE FOR STOPPING PROCEDURE)
  DELTAT=SGHT(CIAFS1/CINES)*DELTAT
  DSGOLD=DELSQ
  CIOLD=C1
  IFLSU=DELTAT**2
  C2=2.0*DELTAT*DAWPF/GAM7
  C1=C2/(4.0+C2)
  (ELG=DELSQ/DSGOLD
  DELS=DELP*(1.0-C1)/(1.0-CIOLD)
  CINES=CINES*DELS**2/DELP
  CINEF=0.5*(CINES+CINER)
  PLAST=TNPG-CINEF-STHEN-TDAMP
  DO 10 A=1,NN
  TFMP(N)=DELP*TFMP(N)
  DR(N)=DELS*DR(N)
  DZ(N)=DELS*DZ(N)
10 CONTINUE
  RETURN
  END
  SUBROUTINE PDATA(LINK)
  C          *CALL MAIN HERE
  C          DIMENSION DAT(20)
  C          PDATA SELECTS AND WRITES DATA ON TAPE(INPLOT) FOR THE WPSII
  C          PLOTTING PROGRAM
  C
  C      GOTO (10,40,50,60),LINK
10 IN30=1
  II = 2*NSTN + 1
  C      INITIATE PLOT FILE. RESTART CAPABILITIES DELETED.
15 WRITE(INPLOT)ETAG2,GN,NSTN
  WRITE(INPLOT) (ETAG2(I),PA(I),NFTAG(I),I=1,NSTN)
  NPTS=N2H-N1H+1
  WRITE(INPLOT)NCYCLE,TIME,NPTS,(H(N),Z(N),NEN(H,N2H))
  C
  DO 25 I=1,11
  DAT(I)=0.0
25 CONTINUE
  IF (LOAD) 30,30,35
30 DAT(5)=TNPG
  DAT(6)=DELP
  DAT(7)=TAEP

```

DAMP	26
DAMP	27
DAMP	28
DAMP	29
DAMP	30
DAMP	31
DAMP	32
DAMP	33
DAMP	34
DAMP	35
DAMP	36
DAMP	37
DAMP	38
DAMP	39
DAMP	40
DAMP	41
DAMP	42
DAMP	43
DAMP	44
DAMP	45
DESTEP	2
TAF	1
DESTEP	4
DESTEP	5
DESTEP	6
DESTEP	7
DESTEP	8
DESTEP	9
DESTEP	10
DESTEP	11
DESTEP	12
DESTEP	13
DESTEP	14
DESTEP	15
DESTEP	16
DESTEP	17
DESTEP	18
DESTEP	19
DESTEP	20
DESTEP	21
DESTEP	22
PDATA	2
TAF	1
PDATA	4
PDATA	5
PDATA	6
PDATA	7
PDATA	8
PDATA	9
PDATA	10
PDATA	11
PDATA	12
PDATA	13
PDATA	14
PDATA	15
PDATA	16
PDATA	17
PDATA	18
PDATA	19
PDATA	20
PDATA	21
PDATA	22
PDATA	23

```

      DAT(I)=TIME
35  IFLAG=1
      WRITE(NPLOT) IFLAG
      WRITE(NPLOT) NCYCLE*(DAT(I)-I=1,II)
      GOTO 100
C
40  DAT(1)=TIME
      DAT(2)=R1
      DAT(3)=R2
      DAT(4)=INFT
      DAT(5)=STPHN+CINBT
      DAT(6)=TIMEG
      DAT(7)=DAT(6)+TCAMP
      J = 4
      DO 45 I=1,NSTHN
      DAT(J)=PFSS1(I)
      DAT(J+1)=PFSS2(I)
      J=J+2
45  CONTINUE
      IFLAG=1
      WRITE(NPLOT) IFLAG
      WRITE(NPLOT) NCYCLE*(DAT(I)-I=1,II)
C      CHECK FOR 3D PLOT
      IF(NCYCLE .NE. NC3DP(NN3D))GOTO 100
      NN3D=NN3D+1
      IFLAG=2
      WRITE(NPLOT) IFLAG
      NPTS=NPH-NIH+1
      WRITE(NPLOT)NCYCLE,TIME,NPTS,(R(N)-7(N),N=N:1h,N2H)
      GOTO 100
C
C 40 END FILE NPLOT * DELETED. MARKED RESTART POSITIONS FOR HPLESC
40  CONTINUE
      GOTO 100
C
40  IFLAG=00004
      WRITE(NPLOT) IFLAG
100  RETURN
      END
      SUBROUTINE PFSS
C      *CALL DATA HERE
C      CONSTANT PRESSURE = P0.
      DATA IPRESS/0/
      IF(IPRESS .GT. 1)GOTO 2
      IPRESS = 1
      READ(5,100)P0
      WRITE(6,105)P0
      DO 5 N=N1V,N2V
      P(N)=P0
      RETURN
100  FORMAT(F12.6)
105  FORMAT(//3AX,*,CONSTANT PRESSURE LOADING. P0=*,1PE15.6/)
      END
      SUBROUTINE INGEOM
C      SIGNALS TO PROGRAM FROM INGEOM.
C      REAR (IF=0).
C      SLAB SYMMETRY (RADIUS = 0.0 IF=0).
C      AXIAL SYMMETRY (RADIUS > 0.0 IF=0).
C      INGEOM FOR STRAIGHT REAR OF SLAB SYMMETRIC FLAT SLAB
C      EVALUATE THE INITIAL GEOMETRY
C      *CALL DATA HERE
C      SET PROGRAM TO USE GAUSSIAN INTEGRATION IN INGEOM.
      GAUSS = 1

```

PIATA	24
PIATA	25
PIATA	26
PIATA	27
PIATA	28
PIATA	29
PIATA	30
PIATA	31
PIATA	32
PIATA	33
PIATA	34
PIATA	35
PIATA	36
PIATA	37
PIATA	38
PIATA	39
PIATA	40
PIATA	41
PIATA	42
PIATA	43
PIATA	44
PIATA	45
PIATA	46
PIATA	47
PIATA	48
PIATA	49
PIATA	50
PIATA	51
PIATA	52
PIATA	53
PIATA	54
PIATA	55
PIATA	56
PIATA	57
PIATA	58
PIATA	59
PIATA	60
PIATA	61
PIATA	62
PFSS	2
TAH	1
PFSS	4
PFSS	5
PFSS	6
PFSS	7
PFSS	8
PFSS	9
PFSS	10
PFSS	11
PFSS	12
PFSS	13
PFSS	14
PFSS	15
INGEOM	2
INGEOM	3
INGEOM	4
INGEOM	5
INGEOM	6
INGEOM	7
INGEOM	8
TAH	1
INGEOM	10
INGEOM	11

C	RADIUS = 0.0	INCEM	12
C	RADIUS = 0.0 IS A SIGNAL TO USE SLAR SYMMETRY OR BEAM	INCEM	13
C		INCEM	14
C	HEAD(5,100) SLARL, SLARW, IS	INCEM	15
C		INCEM	16
C	DETA1 = 2.0*SLARW	INCEM	17
C	ENERGY IS COMPUTED FOR A WIDTH OF DETA1.	INCEM	18
C	DETA1 TENTATIVELY SET AT 2*SLARW. IF IF > 0 DETA1 SET TO 1.0 IN STANT	INCEM	19
C		INCEM	20
C	DETA2 = SLARL/FLOAT(NMF5H)	INCEM	21
C		INCEM	22
C	EVALUATE(P(N),Z(N),A=N1H,N2H)	INCEM	23
C		INCEM	24
C	DO 10 N=N1H,N2H	INCEM	25
C	W(N) = 0.0	INCEM	26
C	Z(N)=FLOAT(N-N1H)*DETA2	INCEM	27
C	10 CONTINUE	INCEM	28
C		INCEM	29
C	IF(IF .NE. 0) RETURN	INCEM	30
C	READ(5,9H)ZU	INCEM	31
C	WRITE(6,9H) SLARL,ZU	INCEM	32
C	DO 12 K=1,LAYFW	INCEM	33
C	READ(5,4H) WIDTHK,DZETAK,ZETA(K)	INCEM	34
C	W(K)=WIDTHK*DZETAK	INCEM	35
C	WRITE(6,4H) K,WIDTHK,DZETAK,ZETA(K),W(K)	INCEM	36
C	12 CONTINUE	INCEM	37
C	97 FORMAT(/' INCEM FOR BEAM OF LENGTH',E15.7/	INCEM	38
C	* ' DISTANCE OF CENTROID FROM TOP OF BEAM IS',E15.7/	INCEM	39
C	1 2X,'DIMENSIONS OF LAYERS'/	INCEM	40
C	2 2X,'K',2X,'WIDTHK', 9X,'DELH(IGHTR',6X,'ZETA(K)',9X,'AREAK'/)	INCEM	41
C	98 FORMAT(3F10.2)	INCEM	42
C	99 FORMAT(13,2X,4F15.7)	INCEM	43
C	RETURN	INCEM	44
C	100 FORMAT(2F10.4,15)	INCEM	45
C	END	INCEM	46
C	SUBROUTINE INNOVM	INCEM	47
C		INNOVM	2
C	*CALL MAIN HERE	INNOVM	1
C	NEW CONDITIONS ON FIXED ENDS 12/12/73	INNOVM	4
C	INNOVM COMPUTES THE INITIAL NORMAL AT CLAMPED ENDS	INNOVM	5
C	THIS VERSION ASSUMES A QUADRATIC THRU THE THREE END POINTS	INNOVM	6
C	IF(INCF1 .NE. 1) GOTO 5	INNOVM	7
C	END1 N=N1H, IS CLAMPED	INNOVM	8
C	W2=P(N1H+2)+4.0*P(N1H+1)+3.0*P(N1H)	INNOVM	9
C	Z2=7(N1H+2)+4.0*Z(N1H+1)+3.0*Z(N1H)	INNOVM	10
C	I=SQRT(W2**2+Z2**2)*SNR(N1H)=Z2/DA SNR(N1H)=W2/D	INNOVM	11
C	5 IF(INCF2 .NE. 1)GOTO 6	INNOVM	12
C	END2 N=N2H, IS CLAMPED	INNOVM	13
C	W2=P(N2H+2)+4.0*P(N2H+1)+3.0*P(N2H)	INNOVM	14
C	Z2=7(N2H+2)+4.0*Z(N2H+1)+3.0*Z(N2H)	INNOVM	15
C	D=SQRT(W2**2+Z2**2)*SNR(N2H)=Z2/DA SNR(N2H)=W2/D	INNOVM	16
C	6 RETURN	INNOVM	17
C	END	INNOVM	18

AD-A122 336

RPSL1D (A ONE-DIMENSIONAL VERSION OF REPSIL)(U) ARMY

242

UNCLASSIFIED

SBI-AD-F300 132

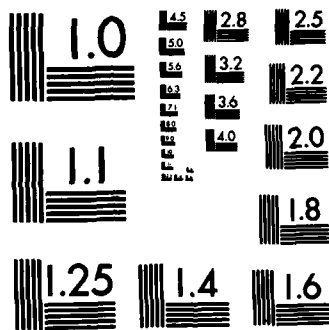
F/G 19/1.

NL

FILMED

1

DTAC



APPENDIX D

PLOTTING PROGRAM FOR RPSL1D

The plotting program is an independent program. The RPSL1D program stores data for plotting through unit 3 which is equivalenced to file BLOT on the PROGRAM card. The plotting program, which we have catalogued in file RPSL1DPLOT, reads data from unit 3 also equivalenced to file BLOT. For some runs it may be advantageous to catalogue BLOT and repeat the plotting program to reproduce plots that were unsatisfactory because of scaling (input card) or system problems.

This program uses the BRL plotting subroutines described in ARDC TR6 (BRLESC FORTRAN PLOTTING SUBROUTINES, Monte W. Coleman, John V. Lanahan, July 1970) as amended for CDC by local publication SPB-6-78, May 2, 1978. Conversion to SCOOP, the plotting system used in Reference 2, or another plotting system, would not be difficult.

The main program, RP1PLT, reads binary data from unit 3 (equated to BLOT) and controls the program flow. If the input variable IFLAG is 1, an array of data for one time point is read and stored. If IFLAG equals 2, different data is read and PLOT3D is called. If IFLAG equals 99999, or certain abnormal conditions occur, the program calls subroutine GRAPH.

Subroutine PLOT3D reads a control card on the initial entry containing DEFLM, SOFC, and SF with FORMAT (3E10.3), and sets up scaling for isometric plots of the "center line deflection profile" of Reference 2. This is a plot of the initial position of the reference curve, and the current position of the curve with the difference magnified by DEFLM. If SF is greater than zero, it is the reciprocal of the scale factor between the internal length measure and inches on the plotting surface. If the input SF is not greater than zero, the program assigns an SF that attempts to scale the plot of the initial curve into an SOFC by SOFC inch square which also includes the origin.

Subroutine GRAPH plots displacement increments DR and DZ vs. time at point (ETAD1, ETAD2) and the energy balance plot: time vs. kinetic energy, kinetic energy plus strain energy, total energy, and total energy less damping work. Subroutine GRAPH calls subroutine STRAIN to produce NSTRN plots of strains in the coordinate directions vs. time at points prescribed on input cards 13. If the PLOTP option is included, GRAPH also produces the NNPE prescribed plots of P(N) vs. time.

All the COMMON variables are included in COMDECK MAIN. The longer arrays are put in LEVEL 2 (a special version of FIXSCA was inserted to use them); if more than 3000 time cycles are recorded, these arrays and MAXC must be increased to plot them.

The following list is the COMPILE file image of RPSL1DPLOT formed through UPDATE. This listing gives the correct UPDATE card identifiers.

	PROGRAM WP1PLT(INPLT,OUTPUT,PL0T,TAPE13,TAPE5=INPUT,TAPE6=OUTPUT,	WP1PLT	2
	1 TAPE3=PL0T)	WP1PLT	3
C	PLOTING FOR 1-C DEFSIL.	WP1PLT	4
C	WPSID PLOTING PACKAGE (CALCOMP PLOTING, SPH-6-7A)	WP1PLT	5
C	IF NCYCL>MAXC, INCREASE MAXC AND ARRAYS IN PARRAY AND PLOT.	WP1PLT	6
C	PLOTING PIN). 9/30/75 (PD(3002,9))	WP1PLT	7
C		WP1PLT	8
	COMMON ETAG2(6),PN(6),NETAG(6),NSTRN,MAXC	MAIN	2
	COMMON RR(103),ZZ(103),Y(103),Z(103)	MAIN	3
	COMMON DAT(20),NCYCLE,TIME, ETAG2, QN,NCYCL	MAIN	4
	COMMON NNPE,NPE(9),PDAT(9)	MAIN	5
	COMMON /PARRAY/	MAIN	6
	1 TIM(3002), U2(3002), U3(3002), CIN(3002), STC(3002),	MAIN	7
	2 TNR(3002),DAMPLT(3002),EPSS1(1P012),EPSS2(1R012)	MAIN	8
	COMMON/PL0TP/PD(3002,9)	MAIN	9
	LEVEL2,TIM,PD	MAIN	10
	MAXC = 3002	WP1PLT	10
	NNPE = 0	WP1PLT	11
	NCYCL=0	WP1PLT	12
	NPL0T=3	WP1PLT	13
	NEWIND NPL0T	WP1PLT	14
C		WP1PLT	15
	READ(NPL0T) ETAG2,QN,NSTRN	WP1PLT	16
	READ(NPL0T) (ETAG2(I),PN(I),NETAG(I),I=1,NSTRN)	WP1PLT	17
C	READ(NPL0T)NNPE,(NPE(I),I=1,NNPE) ACTIVATE FOR PLOTP	WP1PLT	18
	READ(NPL0T) NCYCLE,TIME,N1,(RR(N),ZZ(N),N=1,N1)	WP1PLT	19
	II=2*NSTRN+8	WP1PLT	20
C		WP1PLT	21
C	SAVE INITIAL SHAPE NEEDED FOR DEFLECTION MAGNIFICATION IN PLOT3D	WP1PLT	22
	DO 5 N=1,N1	WP1PLT	23
	Y(N)=ZZ(N)	WP1PLT	24
	Z(N)=RR(N)	WP1PLT	25
	5 CONTINUE	WP1PLT	26
	CALL PLOT3D(N1)	WP1PLT	27
C		WP1PLT	28
	10 READ(NPL0T) IFLAG	WP1PLT	29
	IF(E0F(NPL0T) .NE. 0) GOTO 28	WP1PLT	30
	IF(IFLAG .EQ. 99999)GOTO 30	WP1PLT	31
	IF(IFLAG .EQ. 1)GOT 20	WP1PLT	32
	IF(IFLAG .EQ. 2)GOTO 25	WP1PLT	33
	WRITE(6,911)IFLAG,NCYCL,NCYCLE	WP1PLT	34
	911 FORMAT(///'HAD SIGNAL FROM TAPE. IFLAG,NCYCL,NCYCLE=*,3110)	WP1PLT	35
C	IF NUN FAILS, GET PARTIAL PLOT.	WP1PLT	36
	GOTO 28	WP1PLT	37
C		WP1PLT	38
	20 READ (NPL0T) NCYCLE,(DAT(I),I=1,II)	WP1PLT	39
C	1.(PDAT(J),J=1,NNPE) ACTIVATE FOR PLOTP	WP1PLT	40
	IF(E0F(NPL0T) .NE. 0) GOTO 28	WP1PLT	41
	NCYCL=NCYCL+1	WP1PLT	42
	IF(NCYCL .GT. MAXC)GOTO 28	WP1PLT	43
	TIM(NCYCL)=DAT(1)	WP1PLT	44
	U3(NCYCL)=DAT(2)	WP1PLT	45
	U2(NCYCL)=DAT(3)	WP1PLT	46
	CIN(NCYCL)=DAT(5)	WP1PLT	47
	STC(NCYCL)=DAT(6)	WP1PLT	48
	TNR(NCYCL)=DAT(7)	WP1PLT	49
	DAMPLT(NCYCL)=DAT(8)	WP1PLT	50
	DO 22 I=9,II,2	WP1PLT	51
	J=NCYCL + MAXC*(I-9)/2	WP1PLT	52
	EPSS1(J)=DAT(1)	WP1PLT	53
	EPSS2(J)=DAT(1+1)	WP1PLT	54
	22 CONTINUE	WP1PLT	55

CALL PLTSYM (0.1,HEAD2(1),0.0, 0.0,-.3)	PLOT3D	30
ENCODE (12.15,STEP) DEFLM	PLOT3D	40
15 FORMAT (F10.3,2H >)	PLOT3D	41
CALL PLTSYM (0.1,STEP,0.0, 2.1,-.3)	PLOT3D	42
CALL PLTSYM (0.1,9+SCALE 1/2,0.0, 0.0,-.5)	PLOT3D	43
ISF=INT(SF)	PLOT3D	44
ENCODE (4.15,STEP) ISF	PLOT3D	45
16 FORMAT (I2,2H >)	PLOT3D	46
CALL PLTSYM (0.1,STEP,0.0, 0.9,-.5)	PLOT3D	47
SF=1.0/SF	PLOT3D	48
C *****	PLOT3D	49
20 YFARN=0.0	PLOT3D	50
I=1	PLOT3D	51
30 YHAF=4.0+YFARN	PLOT3D	52
XEAR=1.0	PLOT3D	53
C ----- CENTER-LINE DEFLECTION PROFILE -----	PLOT3D	54
YHAC=YHAF	PLOT3D	55
CALL PLTSCA (XPAR,YHAC,0.0,0.0,1.0,1.0)	PLOT3D	56
X1(1)=0.0	PLOT3D	57
X2(1)=-SF*Z(1)	PLOT3D	58
X1(2)=0.0	PLOT3D	59
X2(2)=SF*Z(1)	PLOT3D	60
CALL PLTDTS (1.0,X1(1),X2(1),2.0)	PLOT3D	61
X2(1)=0.0	PLOT3D	62
X1(2)=SF*Y(I2)	PLOT3D	63
X2(2)=0.0	PLOT3D	64
CALL PLTDTS (1.0,X1(1),X2(1),2.0)	PLOT3D	65
K=0	PLOT3D	66
DO 60 N=1,I2	PLOT3D	67
K=K+1	PLOT3D	68
X1(K)=SF*Y(N)	PLOT3D	69
X2(K)=SF*Z(N)	PLOT3D	70
60 CONTINUE	PLOT3D	71
CALL PLTDTS (4.0,X1(1),X2(1),K.0)	PLOT3D	72
CALL PLTDTS (3.0,X1(1),X2(1),K.0)	PLOT3D	73
K=0	PLOT3D	74
DO 70 N=1,I2	PLOT3D	75
K=K+1	PLOT3D	76
X1(K)=SF*(Y(N)+DEFLM*(Z7(N)-Y(N)))	PLOT3D	77
X2(K)=SF*(Z(N)+DEFLM*(R6(N)-Z(N)))	PLOT3D	78
70 CONTINUE	PLOT3D	79
CALL PLTDTS (2.0,X1(1),X2(1),K.0)	PLOT3D	80
C *****	PLOT3D	81
CALL PLTSYM (1.1,6+NCYCLE,0.0, .2,-3.0)	PLOT3D	82
ENCODE (6.101,STEP) NCYCLE	PLOT3D	83
101 FORMAT (IS,1H >)	PLOT3D	84
WRITE (6,109) NCYCLE	PLOT3D	85
109 FORMAT (' ISOMETRIC PLOT. CYCLE',I4)	PLOT3D	86
CALL PLTSYM (1.1,STEP,0.0, .8,-3.0)	PLOT3D	87
TI=TIME*1.0E6	PLOT3D	88
ENCODE (10.105,STEP) TI	PLOT3D	89
105 FORMAT (F8.1,2H >)	PLOT3D	90
CALL PLTSYM (1.1,STEP,0.0, .2,-3.2)	PLOT3D	91
CALL PLTSYM (1.1,HEAD3(1),0.0, 1.1,-3.2)	PLOT3D	92
C *****	PLOT3D	93
RETURN	PLOT3D	94
END	PLOT3D	95
SUBROUTINE GRAPH	GRAPH	2
COMMON ETAG2(A),PN(A),NETAG(A),NSTEN,MAXC	MAIN	2
COMMON RP(103),ZZ(103),Y(103),Z(103)	MAIN	3
COMMON DAT(20),NCYCLE,TIME	MAIN	4
COMMON ANDE,APL(9),PDAT(9)	MAIN	5
COMMON /PARRAY/	MAIN	6

```

1 TIM( 3002). U2( 3002). U3( 3002). CIN( 3002). STC( 3002). MATN      7
2 TNR( 3002).DAMPLT( 3002).EPSS1(18012).EPSS2(18012) MAIN      8
COMMON/PLTTP/PO( 3002,6) MAIN      9
LEVEL2.TIM,PO      10
DIMENSION SYM1(4),SYM2(2), ETA2(3),SY4(3) GRAFH      4
DATA(SYM1(I),I=1,4)/10*COMPONENT .10*OF VECTOR .10*DISPLACENT,3*NT GRAFH      5
1>/(SYM2(I),I=1,2)/10*TIME (MICR.10*HSECONDS)>/ GRAFH      6
DATA(ETA2(I),I=1,3)/10*ETA2 = .10H ( .5H )>/ GRAFH      7
2 (SY4(I),I=1,2)/10H ENERGY,9* BALANCE>/ GRAFH      8
C GRAFH      9
CALL PLTPGF GRAFH     10
XBAR=3.0 GRAFH     11
YBAR=2.0 GRAFH     12
XL=7.9 GRAFH     13
YL=6.4 GRAFH     14
N=NCYCL GRAFH     15
XS=1.0FA GRAFH     16
C ----- GRAPH ONE ----- GRAFH     17
CALL FIXSCA (TIM(1),N,XL,XS,XMIN1,XMAX1,DX1) GRAFH     18
CALL FIXSCA (U3(1),N,YL,YS1,YMIN1,YMAX1,DY1) GRAFH     19
CALL CONSCA (U2(1),N,YL,YS1,YMIN1,YMAX1,DY1) GRAFH     20
CALL PLTSCA (XBAR,YBAR,XMIN1,YMIN1,XS1,YS1) GRAFH     21
CALL PLTAXS (DX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,4) GRAFH     22
TFM1=XMAX1+DX1/3.0 GRAFH     23
CALL PLTDT2 (1.0,TIM(1),U2(1),N,0) GRAFH     24
U2N = U2(N) GRAFH     25
CALL PLTSYM (.1,3MDZ>,0.0, TEM1 ,U2,N ) GRAFH     26
CALL PLTDT2 (1.0,TIM(1),U3(1),N,0) GRAFH     27
U3N = U3(N) GRAFH     28
CALL PLTSYM (.1,3MDZ>,0.0, TEM1 ,U3,N ) GRAFH     29
CALL LABELA (DX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,XS,1.0) GRAFH     30
CALL PLTSCA (XBAR,YBAR,0.0,0.0,0.0,1.0,1.0) GRAFH     31
CALL PLTSYM (.1,SYM2(1),0.0, 3.0,-0.6) GRAFH     32
CALL PLTSYM (.1,SYM1(1),90.0, -1.2,1.4) GRAFH     33
CALL PLTSYM (.1,9HLOCATION>,0.0, 3.0,-1.0) GRAFH     34
CALL PLTSYM (.3,2H[>,0.0, 3.0,-1.1) GRAFH     35
CALL PLTSYM (.3,2H[>,0.0, 6.6,-1.1) GRAFH     36
CALL PLTSYM (.1,ETA2(1),0.0, 4.1,-1.1) GRAFH     37
FNCODE (R,50,STEP)ETAD2 GRAFH     38
CALL PLTSYM (.1,STFF,0.0, 4.0,-1.1) GRAFH     39
FNCODE (R,50,STEP)QA GRAFH     40
CALL PLTSYM (.1,STEP,0.0, 5.7,-1.1) GRAFH     41
C ----- GRAPH TWO ----- GRAFH     42
YBAR=12.0 GRAFH     43
CALL PLTSCA(XBAR,YBAR,0.0,0.0,0.0,1.0,1.0) GRAFH     44
CALL PLTSYM(.1,SY 4(1),90.0, -1.0,1.5) GRAFH     45
CALL PLTSYM(.1,SYM2(1),0.0, 3.0,-0.6) GRAFH     46
CALL FIXSCA (CIN(1),N,YL,YS4,YMIN4,YMAX4,DY4) GRAFH     47
CALL CONSCA (STC(1),N,YL,YS4,YMIN4,YMAX4,DY4) GRAFH     48
CALL CONSCA (TNR(1),N,YL,YS4,YMIN4,YMAX4,DY4) GRAFH     49
CALL CONSCA (DAMPLT(1),N,YL,YS4,YMIN4,YMAX4,DY4) GRAFH     50
CALL PLTSCA (XBAR,YBAR,XMIN1,YMIN4,XS1,YS4) GRAFH     51
CALL PLTAXS (DX1,DY4,XMIN1,XMAX1,YMIN4,YMAX4,4) GRAFH     52
CALL LABELA (DX1,DY4,XMIN1,XMAX1,YMIN4,YMAX4,XS,1.0) GRAFH     53
CALL PLTDT2(1.0,TIM(1),CIN(1),N,0) GRAFH     54
CALL PLTDT2(1.0,TIM(1),STC(1),N,0) GRAFH     55
CALL PLTDT2 (1.0,TIM(1),DAMPLT(1),N,0) GRAFH     56
CALL PLTDT2(1.0,TIM(1),TNR(1),N,0) GRAFH     57
C ----- STRAIN PLOTS ----- GRAFH     58
DO 100 I=1,NSTRN GRAFH     59
J=1+MAXC*(I-1) GRAFH     60
CALL STRAIN (TIM,EPSS1,EPSS2, ETASCA(1),EN(1),ETAG(1),J,N) GRAFH     61
100 CONTINUE GRAFH     62

```

C		PLOTTING P(N). 9/30/75	
	IF(NMPF.EQ.0)GOTO 35	GRAFH	63
	YPAR = 40.0	GRAFH	64
	DO 33 I=1,NMPF	GRAFH	65
	YPAR = YPAR + 10.0	GRAFH	66
	IF(YPAR.GT. 30.0)CALL PLTPGF	GRAFH	67
	IF(YPAR.GT. 30.0)YPAR = 2.0	GRAFH	68
	FNCODE(7,30,STEP)NPF(I)	GRAFH	69
30	FORMAT(2HP(1,13,2H)>)	GRAFH	70
	CALL PLTSCA(XHAP,YHAP,0.0,0.0,0.1,0.1,0)	GRAFH	71
	CALL PLTSYM(.1,SYM2(1),0.0, 3.0,-0.6)	GRAFH	72
	CALL PLTSYM(.1,STEP,90.0, -1.2,2.5)	GRAFH	73
	CALL FIXSCA(PD(1,1),N,YL,YS1,YMIN1,YMAX1,DY1)	GRAFH	74
	CALL PLTSCA(XHAP,YHAP,XMIN1,YMIN1,XS1,YS1)	GRAFH	75
	CALL PLTAXS(OX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,4)	GRAFH	76
	CALL LAELA(CX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,XS,0)	GRAFH	77
	CALL PLTOT2(1,0,TIM(1),PD(1,1),N,0)	GRAFH	78
33	CONTINUE	GRAFH	79
35	CONTINUE	GRAFH	80
	CALL PLTPGE	GRAFH	81
	RETURN	GRAFH	82
50	FORMAT (F7.3,1H>)	GRAFH	83
	END	GRAFH	84
	SUBROUTINE STRAIN (X,Y,Z, ETA2, PN1,META,J,N)	GRAFH	85
	DIMENSION X(1),Y(1),Z(1), SYM2(3),SYM3(2),SYM4(2),SYM5(2),	STRAIN	2
	1SYM6(2),X1(2),X2(2)	STRAIN	3
	LEVFL2,X,Y,Z	STRAIN	4
	DATA 1/0,YS/100,/,XS/1.0E6/	STRAIN	5
	DATA(SYM2(K),K=1,3)/10META2 = .10M (.5H)>/,	STRAIN	6
2	(SYM3(K),K=1,2)/10TIME (MICP,10H0SECONDS)>/,	STRAIN	7
3	(SYM4(K),K=1,2)/10MSTRAIN (H),1H>/,	STRAIN	8
4	(SYM5(K),K=1,2)/10META1 COMPO,5HNEAT>/,	STRAIN	9
5	(SYM6(K),K=1,2)/10META2 COMPO,5HNEAT>/	STRAIN	10
		STRAIN	11
		STRAIN	12
		STRAIN	13
		STRAIN	14
		STRAIN	15
		STRAIN	16
		STRAIN	17
10	CALL PLTPGF	STRAIN	18
	XL=7.9	STRAIN	19
	YL=6.4	STRAIN	20
	CALL FIXSCA(X(1),N,XL,XS1,XMIN,XMAX,DX)	STRAIN	21
20	YPARN=0.0	STRAIN	22
	I=1	STRAIN	23
25	YPAR=2.0*YPARN	STRAIN	24
	XHAP=2.0	STRAIN	25
	CALL PLTSCA (XHAP,YHAP,0.0,0.0,0.1,0.1,0)	STRAIN	26
	CALL PLTSYM (.1,SYM4(1),90.0, -1.2,2.5)	STRAIN	27
	CALL PLTSYM (.1,SYM3(1),0.0, 3.0,-0.6)	STRAIN	28
	CALL PLTSYM (.1,9HLOCATION>,0.0, 3.0,-1.0)	STRAIN	29
	CALL PLTSYM (.3,2H(>,0.0, 3.8,-1.1)	STRAIN	30
	CALL PLTSYM (.3,2H(>,0.0, 6.6,-1.1)	STRAIN	31
	IF(META.NE.0)GOTO 30	STRAIN	32
	CALL PLTSYM (.1,6HOUTER>,0.0, 6.8,-1.0)	STRAIN	33
	GOTO 35	STRAIN	34
30	CALL PLTSYM (.1,6HINNER>,0.0, 6.8,-1.0)	STRAIN	35
35	CALL PLTSYM (.1,SYM2(1),0.0, 4.1,-1.1)	STRAIN	36
	FNCODE(8,50,STEP)ETA2	STRAIN	37
	CALL PLTSYM (.1,STEP,0.0, 4.8,-1.1)	STRAIN	38
	FNCODE(8,50,STEP)PN1	STRAIN	39
	CALL PLTSYM (.1,STEP,0.0, 5.7,-1.1)	STRAIN	40
		STRAIN	41

X1(1)=2.95	STRAIN	42
X2(1)=-1.3	STRAIN	43
X1(2)=3.576	STRAIN	44
X2(2)=-1.3	STRAIN	45
CALL PLTDTS (1.0,X1(1),X2(1),2.0)	STRAIN	46
CALL PLTSYM (.1,SYMS(1),0.0, 3.7,-1.3)	STRAIN	47
X2(1)=-1.5	STRAIN	48
X2(2)=-1.5	STRAIN	49
CALL PLTDTS (4.0,X1(1),X2(1),2.0)	STRAIN	50
CALL PLTSYM (.1,SYMS(1),0.0, 3.7,-1.5)	STRAIN	51
CALL FIXSCA (Y(J),N,YL,YS1,YMIN,YMAX,DY)	STRAIN	52
CALL CONSCA (Z(J),N,YL,YS1,YMIN,YMAX,DY)	STRAIN	53
CALL PLTSCA (XPAR,YPAR,XMIN,YMIN,XS1,YS1)	STRAIN	54
CALL PLTAXS (DX,DY,XMIN,XMAX,YMIN,YMAX,4)	STRAIN	55
CALL PLTDY2 (1.0,X(1),Y(J),N,0)	STRAIN	56
CALL PLTDZ2 (4.0,X(1),Z(J),N,0)	STRAIN	57
CALL LABELA (DX,DY,XMIN,XMAX,YMIN,YMAX,XS,YS)	STRAIN	58
RETURN	STRAIN	59
50 FORMAT (F7.3,1H>)	STRAIN	60
END	STRAIN	61
SUBROUTINE FIXSCA (X,NPTS,SIZE,XSCALE,XMIN,XMAX,DX)	FIXSCA	2
DIMENSION X(1),T(2)	FIXSCA	3
C MUST READ IN FIXSCA TO PUT X IN LEVEL 2.	FIXSCA	4
LEVEL2,X	FIXSCA	5
C(I) X=ALINEAR ARRAY OF NUMBERS IN DATA UNITS	FIXSCA	6
C(1) NPTS=THE NUMBER OF X VALUES	FIXSCA	7
C(1) THE LENGTH OF X DIMENSION OF THE GRAPH IN PLOTTER UNITS	FIXSCA	8
C(R) XSCALE=THE SCALE IN DATA UNITS/PLOTTER UNIT	FIXSCA	9
C(R) XMIN=THE ADJUSTED MINIMUM IN DATA UNITS	FIXSCA	10
C(R) XMAX=THE ADJUSTED MINIMUM IN DATA UNITS	FIXSCA	11
C(R) DX=THE DELTA X FOR AXIS IN DATA UNITS	FIXSCA	12
LOGICAL CONT	FIXSCA	13
CONT=.FALSE.	FIXSCA	14
GOTO 100	FIXSCA	15
ENTRY CONSCA	FIXSCA	16
CONT=.TRUE.	FIXSCA	17
100 TXMI=X(1)	FIXSCA	18
TXMA=X(1)	FIXSCA	19
DO 120 I=1,NPTS	FIXSCA	20
IF(X(I).GE.TXMI)GOTO 110	FIXSCA	21
TXMI=X(I)	FIXSCA	22
GOTO 120	FIXSCA	23
110 IF(X(I).LE.TXMA)GOTO 120	FIXSCA	24
TXMA=X(I)	FIXSCA	25
120 CONTINUE	FIXSCA	26
IF(.NOT.CONT)GOTO 140	FIXSCA	27
IF(TXMI.LT.XMIN.OR.TXMA.GT.XMAX)GOTO 125	FIXSCA	28
TXMI=XMIN+ABS(XMIN)*0.000001	FIXSCA	29
TXMA=XMAX-ABS(XMAX)*0.000001	FIXSCA	30
GOTO 140	FIXSCA	31
125 IF(TXMI.LT.XMIN)GOTO 130	FIXSCA	32
TXMI=XMIN+ABS(XMIN)*0.000001	FIXSCA	33
130 IF(TXMA.GT.XMAX)GOTO 140	FIXSCA	34
TXMA=XMAX-ABS(XMAX)*0.000001	FIXSCA	35
140 DIFF=TXMA-TXMI	FIXSCA	36
IF(DIFF.LT.0.000001) DIFF=0.000001	FIXSCA	37
ENCODE (20, 150, T) DIFF	FIXSCA	38
150 FORMAT(1PF12.5)	FIXSCA	39
DECODE (20, 160, T) COFF, IEXP	FIXSCA	40
160 FORMAT(FR.5,1X,I3)	FIXSCA	41
170 IF(COFF.GT.2.0)GOTO 180	FIXSCA	42
DELTA=0.1	FIXSCA	43
GOTO 200	FIXSCA	44

180 IF (COFF.GT.4.0)GOTO 190

DELTA=0.2

GOTO 200

190 DELTA=0.5

200 DELTA=DELTA*10.0**IFXP

DX=DELTA

XMIN=AINT(ARS(TXMI)/DELTA)*DELTA

IF (TXMI.LT.0)XMIN=-(XMIN+DELTA)

XMAX=AINT(ARS(TXMA)/DELTA)*DELTA

IF (TXMA.LT.0)GOTO 210

XMAX=XMAX+DELTA

GOTO 220

210 XMAX=-XMAX

220 XSCALE=(XMAX-XMIN)/SIZE

RETURN

FNU

FIXSCA	45
FIXSCA	46
FIXSCA	47
FIXSCA	48
FIXSCA	49
FIXSCA	50
FIXSCA	51
FIXSCA	52
FIXSCA	53
FIXSCA	54
FIXSCA	55
FIXSCA	56
FIXSCA	57
FIXSCA	58
FIXSCA	59
FIXSCA	60

APPENDIX E

EXAMPLES

Two examples are given in this appendix. These were originally jobs run on the BRLESC computer, now defunct, and transferred to the CDC CYBER 70 Model 76 computer to check the program there. The two jobs test, and hence illustrate, many of the available options and internal variations of RPSLID. For both examples, the complete SCOPE 2.1 batch job is listed and a sample of the output is given. The sections of the batch job are separated in the listings by question marks which listed for the multipunch end of record and end of information signals.

E.1 Example 1: Pressure Loaded Flat Plate

The first example involves finding the response along the shorter symmetry line of a flat 4 by 15 inch soft aluminum plate. (Results from the standard REPSIL and the slab symmetric version of RPSLID are very similar for this elongated plate.) This example demonstrates the introduction of a new PRESS subroutine, the use of the optional coding PLOTP, and the activating of Gaussian integration and damping. Both the new PRESS routine and the option PLOTP are inserted by making changes in the program through UPDATE; both also require input data cards. With the catalogued INGEOM, Gaussian integration is automatically used with slab symmetry if IB = 0 on input card 14. Damping was activated from time Δt on by setting MDAMP = 0 on input card 5. The damping option was used here to help simulate displacement of the plate by quasi-static pressure loading. This is not the usual use of damping. Damping is a device normally used to bring a responding surface to rest at its final equilibrium configuration after the loading is removed. Since subroutine DAMP is not entered until cycle one, when NCYCLE = 1, the subroutine did not shut off pressure loading as it will if entered with NCYCLE = MDAMP. This run terminated at cycle 3546 with a displacement of .66 inches.

A listing of the batch job deck and samples of output for Example 1 follow. The first part of the job deck is the SCOPE 2.1 control statements to set up and run RPSLID and the plotting. The next set of cards is UPDATE directives and new FORTRAN statements for RPSLID. The input data cards for RPSLID are next. (The input and output for this example are in the now forbidden pound-inch-second system. If this is discomforting, pretend the units are SI units for some very exotic material.) After the data there are UPDATE changes for the plotting program, including changes for the PLOTP option and a change to permit up to 6000 time cycles. Finally, there is a data card to control plotting.

Following the job listing are sample listings of output. This starts with a summary of the input and the initial Cartesian coordinates as prescribed by INGEOM. All the output should be inspected to uncover errors, but these two sets should receive very close attention.

The output at cycle 3000 for this example is listed next. This includes:

- Displacement increments, coordinates, and pressure at each mesh point including the virtual external point at $N=1$ and the symmetry point at $N=13$.
- The LMAT matrix. A "1" signifies plasticity, and a "0" denotes an elastic response, on the current cycle. For each column labeled N , there are two rows for each integration station K . The first row is for mesh point N . The second row is for midmesh N .
- The components of the surface normal at each mesh point, the 2 columns on the left, and at each midmesh.
- Surface strains at all mesh points. For each mesh point, the strains on the upper surface are listed first, then the strains on the lower surface are listed. $EC1$ and $EC2$ are the elongation strains in the ξ^1 and ξ^2 directions, respectively. $E11$ and $E22$ are internal covariant components of strain. For slab symmetry, $EC1$ and $E11$ are zero.
- The tabulated output for the NSTRN prescribe surface strains accompanied by prints of extreme strains and maximum deflection for the current cycle, and for the entire run.
- The energy balance summary.

In our listing, this is followed by three prints noting the use of the kinetic energy annihilating procedure (a part of damping which stops motion at kinetic energy maxima) and finally a print of the prescribed surface strains, etc. at cycle 3500.

The occurrence and frequency of this output is controlled by input cards 8, 9, and 10.

Figures E.1.1 through E.1.4 are samples of plotting output. For most publication purposes, it is desirable to do some relabeling, but these plots have not been retouched. Figure E.1.1 shows displacement at selected time steps. Figure E.1.2 shows the displacement of the selected central mesh point and the energy balance as functions of time. The scallops in these plots occur when the stopping procedure was exercised. This energy balance plot shows the four possible curves. These are, from low to high: kinetic energy, kinetic energy plus elastic strain energy, kinetic energy plus elastic strain energy plus the work in damping, and the total energy. Figure E.1.3 is the first two of the six strain plots requested for this run. Figure E.1.4 is a plot of force per initial unit area at mesh point 3. The pressure increased linearly to a maximum of 80 psi and then remained constant, but the area increased throughout the entire run.

Listing of the Batch Job Cards for Example 1

```

VOETM,STAR7,TRD.    DUPLICATE RUN OF 1/2/77.  TEST GAUSS. PLOTP. LINPRS.
ACCTENT,*****.    ADETREN,4309,A3979
ATTACH(OLDF1,PSL10,IO=JFW)
UPDATE,F.
FTN(A,I,SI=0,P=0)
VAR,OFF.
LGO.
EXIT(0)          DATA TO PLOT TAPE
REFIND(PLOT)
RETURN,COMPILE.
RETURN,LGO.
RETURN,OLDPL.
ATTACH(OLDPL,PPSI10PLOT,IO=JFW)
BEGIN,ATTACH,PLOTLIR.
UPDATE(F)
FTN(A,I,F1=F,SI=0,R=0,LCM=1)
VAR,OFF.
LGO.
EXIT(0)
BEGIN,PLOT,CALCOMP,TAPE13.
?
*IDENT PLOTP
*F PSL10.5
C          PLOTTING P(N).  9/30/75
*F MAIN,28
COMMON NPPF,NPF( 9),PDAT( 9)
*F START,24
READ(5,110)NPPF,(NPF(I),I=1,NPPF)
*F PDATA,13
WRITE(6PLOT)NPPF,(NPF(I),I=1,NPPF)
*F PDATA,10
DO 24 I=1,NPPF
J=NPF(I)
24 PDAT(I)=F(J)
*F PDATA,27
1.(PDAT(J),J=1,NPPF)
*F PDATA,42
DO 44 I=1,NPPF
J=NPF(I)
44 PDAT(I)=F(J)
*F PDATA,45
1.(PDAT(J),J=1,NPPF)
*IDENT 1 LINPRS
*F PDRESS,4,PDRESS,14
C          PDRESS BY LINEAR INTERPOLATION WITH RESPECT TO TIME IN
C          TRIT TABLE TPR(I),PPH(I).  10/7/76
C          FORMAT (2F10.3)    TERMINATE WITH NEGATIVE TPR.
DIMENSION TPR(51),PPH(51)
DATA IPRESS/0/
IF(IPRESS.GT. 0)GOTO 4
WRITE(6,100)
1 = 0
2 I = I+1
READ( 5,101)TPR(I),PPH(I)
IF(TPR(I).LT. 0.0)GOTO 3

```

PLOTP

PLOTP

PLOTP

PLOTP

PLOTP

PLOTP

PLOTP

PLOTP

PLOTP

PLOTP

LINPRS

LINPRS

LINPRS

LINPRS

LINPRS

LINPRS

LINPRS

LINPRS

LINPRS

LINPRS

```

WRITE(5,122)I,T-P(I),PFF(I)
GO TO 2
3 IPRESS = I-1
4 CALL DVIINT(TIME,P0,TH,PFF,IPRESS,2)
DO 5 N=1,N2V
P(N) = P0
5 CONTINUE
RETURN
100 FORMAT(//) PRESSURE FROM A TABLE 1/4X,11.5X,TIME,9X,IPRESS(PF)
101 FORMAT(2F10.3)
102 FORMAT(15.2F15.7)
?
STRUCT. MODEL. .020 PLATE. GRADUALLY INCREASING PRESSURE TO HOFSI 1/3/77
5 10 4 2.0
4000 0 0000 0.500000E-6
2 2 1 1
1 0000 0 0.1 0.001
10.000000E-6 0.33 5000. 2.530000E-4 .020 4 0
5000. .0005H
0700. .035
11000. .10
13050. .25
500 1 1 1
6 500 1000 1500 2000 3000 3545
6 500 1000 1500 2000 3000 3545
2 500 3545
1 3
0.0 2.0 45.0 135.0 0
0.0 0.0 45.0 135.0 0
0.0 1.0 45.0 135.0 0
0.0 2.0 45.0 135.0 0
0.0 0.0 45.0 135.0 1
0.0 1.0 45.0 135.0 1
0.0 2.0 45.0 135.0 1
2.0 7.5 0
0.0 0.0
0.00135 00.0
1.0 00.0
-1.0
?
*START PLOT
*P P1PLT,12
PFAF(NPLOTIANPF,(NPF(I),I=1,NPFF)
*P P1PLT,40
1.(PRAT(J),J=1,NPFF)
*TC P002
*P MAIN,7,MAIN,9
1 TIM( P002), L2( P002), U3( P002), CIN( P002), STC( P002), P002
2 TNP( P002),DAMPIT( P002),EPSS1(36012),EPSS2(36012)
COMMON/PL0TP/P0( P002,9)
*P P1PLT,10
MAXI = P002
?
1.0 3.0
?

```

ORL NPSIL CDPF
STRUCT. MODEL. .070 PLATE. GRADUALLY INCREASING PRESSURE TO NPSI 1/3/77
COMPUTATIONS FOR TANGENT USE DETAIL .150000E-02
10 RESHES IN ETAS DIRECTION (INITIAL .200000E-00)

ENDING TIME INCREMENT .164400E-04
MEMBRANE TIME INCREMENT .000000E-00
INPUT TIME INCREMENT .000000E-00

TIME INCREMENT USED BY NPSIL .000000E-00

YOUNG'S MODULUS = .100000E+08
POISSON'S RATIO = .330000E+00
MASS DENSITY = .250000E-03
YIELD STRESS = .500000E+04
THICKNESS = .700000E-01

START AT TIME STEP 0
FINAL TIME STEP 4000
SURFACE STRAINS EVERY 500 TIME STEP
RESTART UNTIL EVERY 9999 TIME STEP

LAYER = 4
LOAD = 1
MSTEN = 6
LPRSS = 9999

1/2/3/4/ = CLAMPED/SYMMETRY/MINOR/FREE/
BOUNDARY CONDITIONS
END1 (IBCE4) = 1
END2 (IBCE2) = 2

PRINT OPTION CONTROL CARD
0/1 = NO PRINT/PRINT
1 DISPLACEMENT INCREMENTS
1 CARTESIAN COORDINATES. PRESSURE
1 SURFACE NORMAL VECTOR COMPONENTS

PRINT INFORMATION AT THE FOLLOWING TIME STEPS

500 1000 1500 2000 3000 3545
PRINT L MATRIX (LMAT) AT THE FOLLOWING TIME STEPS
500 1000 1500 2000 3000 3545
3-D PLOTS FOR THE FOLLOWING TIME STEPS
500 3545

CONSTITUTIVE RELATION ELASTOPLASTIC-VOON HARDENING-STRAIN RATE INDEPENDENT
STRESS-STRAIN APPROXIMATION HAS 4 SUPERVARS

J	SSIG(J)	SEPS(J)	STRESS-STRAIN AND STRAIN RATE PARAMETERS	1/PSR(J)
1	5.000000E+03	5.000000E-04	0.	0.
2	7.700000E+03	2.500000E-02	0.	0.
3	1.100000E+04	1.000000E-01	0.	0.
4	1.305000E+04	2.500000E-01	0.	0.

START DAMPING AFTER TIME STEP 0 TIME 00.
DAMP = .1000E-02

PROSCOPY FROM A TABLE
1 TIME PRESSURE
1 0.
2 .130000E-02 .000000E+02
3 .100000E+01 .000000E+02

Selected Output from Example 1

[illegible]

VINE = 2.709070F-03

TIME STEP 1000	TIME = .2700000E+02	PLASTIC = .0702637E+00	PLASTIC = .000012E+01	PLASTIC = .0109760E+03
TOTAL ENH:VE	.1977710E+04			

STOPPING INFORMATION ACCT# C01695-C16P0.C16F1.TDAMP.C16P

STOPPAGE IN COUNTRIES OF THE TROPICS

1214	4.115432E-01	4.015459E-01	4.015474E-01	7.533374E-02	6.705700E-01
------	--------------	--------------	--------------	--------------	--------------

SINOP INC FORT WORTH ACYCLE.CINCS.CINFMCJMT.TODAP.CIMAP
V002 1.0978627-01 3.0978626-01 3.0978604F-01 7.5445520F.02 4.0157597F-01

1145 0021 2115 2411 30-1000051C-20

SURFACE STRAINS		STRAIN GAUGE READING			
FLAT	N	FACE	ANGLE 0	ANGLE 90	ANGLE
0.000	2.000	OUTER	0.	-12654401E+00	45.00
1.000	7.000	OUTER	0.	-68055300E+01	05.00
2.000	12.000	OUTER	0.	-60559107E+01	45.00
3.000	2.000	INNER	0.	-11564164E+01	45.00
4.000	7.000	INNER	0.	-7466800E+01	45.00
5.000	12.000	INNER	0.	-74032761E+01	45.00
6.000	2.000	OUTER	0.	-65154300E+01	135.00
7.000	7.000	OUTER	0.	-34507330E+01	135.00
8.000	12.000	OUTER	0.	-34847400E+01	135.00
9.000	2.000	INNER	0.	-57907600E+01	135.00
10.000	7.000	INNER	0.	-37912400E+01	135.00
11.000	12.000	INNER	0.	-30049000E+01	135.00

EXHIBIT STAINING AT RATE = 3.15000E-03. TIME CYCLE = 3500

EXTREME STAINS NETWORK CYCLES		TIME	
STRAIN	N	STRAIN	N
PSU1PAR	0.0000000	0.0000000	-07
PSU2PAR	-1765461	-1265461	2
PSL1PAR	0.0000000	0.0000000	2
PSL2PAR	-0748376	-0748376	2
PSU1PIN	0.0000000	0.0000000	12
PSU2PIN	-0674160	-0674160	2
PSL1PIN	0.0000000	0.0000000	2
PSL2PIN	-0119617	-0119617	2
EXTREME STAINS NETWORK CYCLES			
STRAIN		TIME	
PSU1MAX	0.0000000	0.0000000	-07
PSU2MAX	-1265461	-1265461	2
PSL1MAX	0.0000000	0.0000000	2
PSL2MAX	-0748376	-0748376	2
PSU1MIN	0.0000000	0.0000000	-03
PSU2MIN	-0.0000000	-0.0000000	-07
PSL1MIN	-0.0004305	-0.0004305	2
PSL2MIN	-0.0000000	-0.0000000	2
PSU1MAX	-0.0103760	-0.0103760	-03

```
***** REFLECTION AT TIME = 3.08090E-07 ***** CYCLE# = 3500  
REFLECTANCE    6.80509E-01   (N = 12)  
***** REFLECTION AFTER CYCLES 9 AND 3500 ****  
REFLECTANCE    6.80509E-01   (N = 12)      TIME= 3.08090E-07
```

2 0. 1.1003422625149E-01 0. 1.2712633554400E-01
 3 0. 7.110645730706E-03 0. 7.1603095714023E-03
 4 0. 6.2125720206107E-07 0. 6.4055531374525E-02
 5 0. 6.8123052906272E-02 0. 7.1004061906500E-02
 6 0. 6.2336445696370E-02 0. 6.4279400420054E-02
 7 0. 6.8038107050816E-02 0. 7.1313339712104E-02
 8 0. 6.2501243005085E-02 0. 6.4540074055974E-02
 9 0. 6.9067561047574E-02 0. 7.1652725042100E-02
 10 0. 6.2016237013704E-02 0. 6.4052941604300E-02
 11 0. 6.9232204550245E-02 0. 7.1620039170064E-02
 12 0. 6.316524137445E-02 0. 6.5101653904251E-02
 13 0. 6.941009400400E-02 0. 7.1019939059670E-02
 14 0. 6.3435053024030E-02 0. 6.564705000104E-02
 15 0. 6.9504610439671E-02 0. 7.200562001374E-02
 16 0. 6.3065970049100E-02 0. 6.569765740333E-02
 17 0. 6.973096144600E-02 0. 7.210503305990E-02
 18 0. 6.3042320554910E-02 0. 6.500250012570E-02
 19 0. 6.9055133697140E-02 0. 7.2295003549075E-02
 20 0. 6.3052753043672E-02 0. 6.5997731205167E-02
 21 0. 6.9020027030521E-02 0. 7.2374917393330E-02
 22 0. 6.3990366759561E-02 0. 6.5037776070701E-02
 23 0. 6.9055300105561E-02 0. 7.2402746035200E-02

SURFACE STRAINS. UPPER/LOWER. IEC1.EC2.F11.E22)

TIME STEP 1000 TIME = .27000000E-02

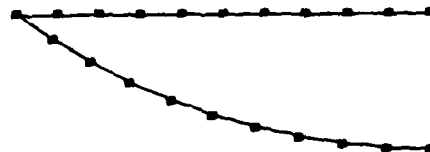
SURFACE STRAINS			STRAIN GAGE READING		
STEP	N	FAC	ANGLE 90	ANGLE	
0.000	2.000	OUTER	0.	0.	0.6162063E-01
1.000	7.000	OUTER	0.	45.00	.3206596E-01
2.000	12.000	OUTER	0.	45.00	.32491030E-01
0.000	2.000	INNER	0.	45.00	.30627244E-02
1.000	7.000	INNER	0.	45.00	.35207370E-01
2.000	12.000	INNER	0.	45.00	.35560562E-01

EMPTY STRAINS AT TIME = 2.700000E-03. TIME CYCLE = 3000

PRIME STRAINS BETWEEN CYCLES 0 AND 3000			TIME	N
STEP	STRAIN	TIME		
EPSUMAX	0.0000000	0.00000E-07	2	2
EPSIMAX	.11993423	2.70000E-03	2	2
EPSLMAX	0.0000000	0.00000E-07	2	2
EPSLMAX	.06995537	2.70000E-03	12	12
EPSUMIN	0.0000000	0.00000E-07	2	2
EPSUMIN	.06712573	2.70000E-04	5	5
EPSLMIN	0.0000000	0.00000E-07	2	2
EPSLMIN	.00711004	1.073300E-03	2	2



CYCLE 500
450.0 MICROSECONDS



CYCLE 3546
3191.4 MICROSECONDS



CYCLE 0
0.0 MICROSECONDS
DEFLECTION MAGNIFIED 1.000
SCALE 1/ 0



CYCLE 3546
3190.5 MICROSECONDS

Figure E.1.1. Plots of the reference curve at selected times.

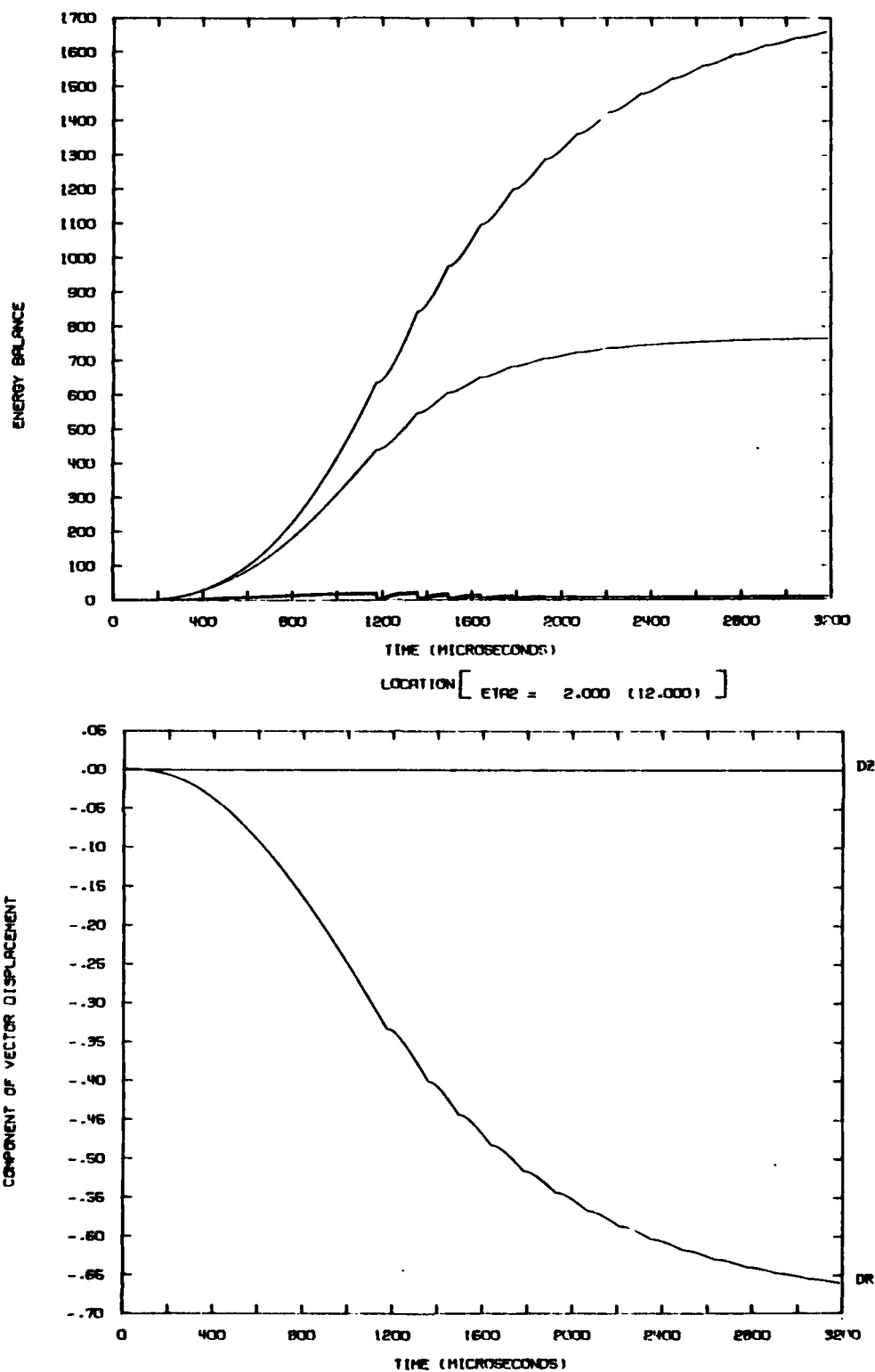


Figure E.1.2. Displacement history of the central point and energy balance curves.

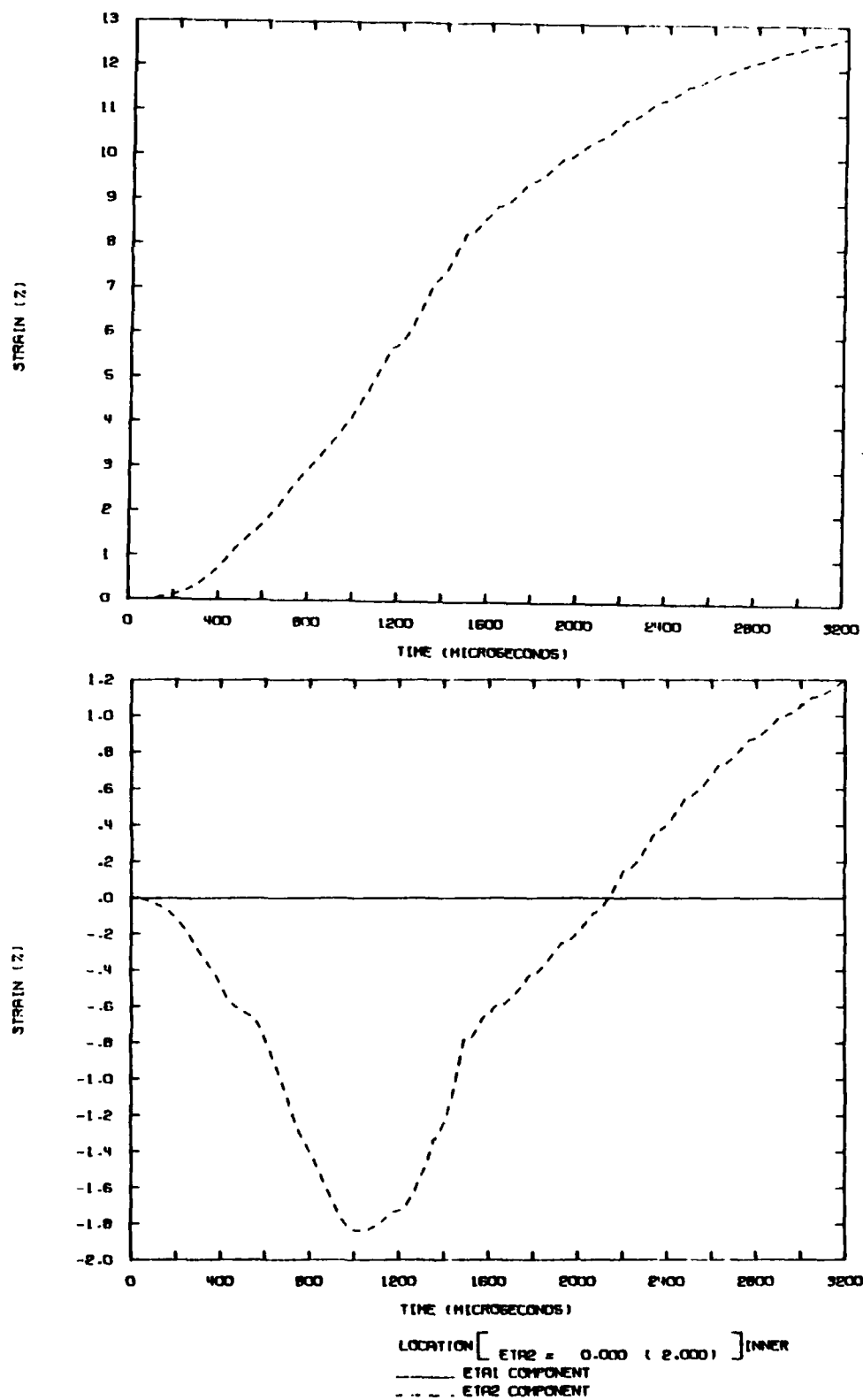


Figure E.1.3. Strain histories at the edge on the top and bottom surfaces

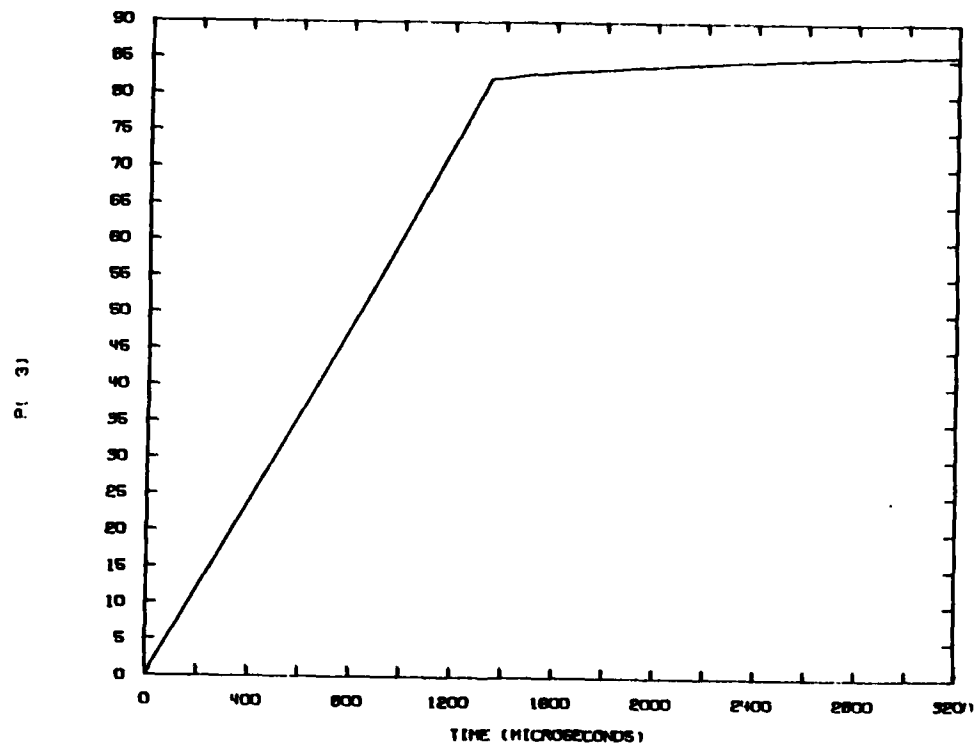


Figure E.1.4. Force per initial unit area at mesh point B.

E.2 Example 2: Eroding Rod

This example uses the beam option of RPSL1D and utilizes all seven of the options listed in Section 5. This program was designed to illustrate that the test strain-time records at two locations along a steel rod penetrating a steel plate could be reproduced by varying the force and the rate of erosion on the end. Two quite different tables of erosion and force were used, and both returned strains that were close to the test values at three locations. This example used a fast erosion rate. Mesh points 2 and 3 were eroded by time cycle 5.

The batch job cards and some sample output are listed. The batch job for this example is rather long. The first section of the run stream is the SCOPE 2.1 control statements. The next section is the UPDATE directives and new FORTRAN statements for RPSL1D. Breaking this section into the nine groups with different UPDATE card identifiers we have: PLOTP, SHR3/1, MSQSVS, BSTRS, APLFRC (including a subroutine ENDFRC which is adjusted for erosion), EAPFRC, 11/2/76 (a modification for EAPFRC), INVELC (an insert into INVEL which creates a uniform initial velocity toward end 1), and ERODE.

The next section in the job cards is the input data. The input, and output, for this example are in SI units: meters, seconds, kilograms, Pascals, etc. This leads to some awkward scaling on both input and output, but assures a consistent, unambiguous system. Along with the usual input data, this section contains a long table of strain vs. stress for the BSTRS option, input to approximate a solid cylindrical beam for INGEOM, no input for PRESS, a card with initial velocity for INVEL, three tables for ENDFRC, and a table for ERODE.

The next section is UPDATE changes for the file RPSL1D PLOT for options MSQSVS and PLOTP. Finally, there is an input data card to control the plotting.

The tabular output listed is two of the seven pages which reflect input (most of the omitted tabulations are simply copies of input tables), and the output at cycle 130. The output at cycle 130 includes:

- The displacement and coordinates. R and DR are effectively zero.
- The output from SHR3/1. The moment, MS, and shear, VS, are zero. Only axial force, QS, is significant.
- The table of plastic stress occurrence.
- The components of the unit normal vector. The rod has not bent.
- The surface strains.
- The energy balance.

Six of the output plots are shown next. (An isometric plot of the rod and plots of MS, QS, and VS were made at cycles 70 and 130. The isometric plot is uninteresting, and MS and VS are zero.) A plot of QS vs. mesh point number at cycle 130 is included to illustrate output of the MSQSVS option. We also include plots of several functions of time: the displacement of mesh point 5, strain plots at 19.26 mm and 39.67 mm, a plot of force on end 1, EFZ1, which was stored in P(53), and the erosion at end 1 which was stored in P(54).

Listing of the batch job cards for Example 2

```

WORTM.STMFZ.T30.    DUPLICATE RUN OF 6/8/77 AND 10/12/77.    ERUDING RGD.
ACCOUNT.            WORTMAN.B309.X3979
ATTACH.OLDPL.WPSLID.ID=JOW.
UPDATE.F.
FTN(A.I.SL=0.R=0)
MAP.OFF.
LGO.
EXIT(U)            DATA TO PLOT TAPE
REWIND(PLOT)
RETURN.COMPILE.
RETURN.LGO.
RETURN.OLDPL.
ATTACH(OLDPL.WPSLIDPLOT.ID=JOW)
HEGIN.ATTACH.PLOTLP.
UPDATE(F)
FTN(A.I.EL=F.SL=0.R=0)
MAP.OFF.
LGO.
EXIT(U)
HEGIN.PLOT.CALCOMP.TAPE13.
?
*IDENT PLOTP
*I WPSLID.5
C            PLOTTING P(N). 9/30/75
*J MAIN.28
COMMON NNPE,NPE( 9),PDAT( 9)
*J START.24
READ(5.110)NNPE,(NPE(I),I=1,NNPE)
*J PDATA.13
WRITE(NPLOT)NNPE,(NPE(I),I=1,NNPE)
*J PDATA.19
DO 26 I=1,NNPE
J=NPE(I)
26 PDAT(I)=P(J)
*J PDATA.27
1.(PDAT(J),J=1,NNPE)
*J PDATA.42
DO 46 I=1,NNPE
J=NPE(I)
46 PDAT(I)=P(J)
*J PDATA.45
1.(PDAT(J),J=1,NNPE)
*IDENT SHF3/1
*I WPSLID.5
C            ESTIMATE SHEAR FORCE VS. AND MS AND QS. 3/1/76 (FOR A PEAM) SHF 3/1
*J MAIN.28
COMMON AMS(103),QS(103),VS(103)
*J RESULT.48
MS AT FREE END IS ZERO (4/5/76).    CHANGE WITH APLFRC.
AMS(N2R) = 0.0
*J RESULT.54
AMS(N) = A22*F22
*J RESULT.62
VS(N) = (AMS(N-1) - AMS(N))/(DET12*FRJ)
209 QS(N) = A22*(C22-RW22*F22)

```

```

C      VS,MS,QS PRINT CONTROLLED BY CODING.   SHR 3/1/76.
C      TO PRINT EVERY 10 CYCLES.
C      IF(MOD(NCYCLE,10) .EQ. 0) GOTO 212
C      IF(NCYCLE .EQ. 70 .OR. NCYCLE .EQ. 130) GOTO 212
C      GOTO 218
210 FORMAT(///' CYCLE  N',F10.0,'MS',F10.0,'VS',F10.0,'QS',F10.0,'A22',F10.0,
1  'M22',F10.0,'G22',F10.0,'M22',F10.0)
211 FORMAT(1A,13,1P7E15.7)
212 IF(N .EQ. N1P )WRITE(6,210)
      WRITE(6,211)NCYCLE,N,AMS(N),VS(N),QS(N),A22,BM22,G22,F22
C      OUTPUT OF MS(N2B)   4/5/76
      IF(N .EQ. N2B-1)WRITE(6,211)NCYCLE,N2B,AMS(N2B)
218 CONTINUE
*IDENT MSQSVS
*I RPSLID,5
C      PLOTTING MS, QS, AND VS. 3/19/76
*I PDATA,52
      WRITE(NPLOT)(AMS(N),QS(N),VS(N),N=N1B,N2B)
*IDENT HSTRS
*I RPSLID,5
C      NEW UNIAXIAL STRESS-STRAIN CURVE FOR BEAMS. 4/1/77
C      NOT COMPATIBLE WITH STREN OPTION.
*O MAIN,5
      3 SIG(103,6),SIG2(103,6),LMAT(103,6),
*O MAIN,26
      COMMON SNRM(103),SNKM(103),SIG1M(103,6),SIG2M(103,6),LMATM(103,6)
*I MAIN,2P
      COMMON EPSR(103,6),EPSRM(103,6),EPLB(103,6),EPLBM(103,6),
      2 EFR(100),SSH(100),NEST,EPSZ
*I RPSLID,60
      EPSR (N,K) = 0.0
      EPSRM(N,K) = 0.0
      EPLR (N,K) = 0.0
      EPLRM(N,K) = 0.0
*O START,16,START,20
      DO 700 I=1, 99
      READ(5,702)EFR(I),SSH(I)
      IF(EFR(I) .LT. 0.0) GOTO 701
700 CONTINUE
701 NEST = I
      NSFL = 1
      SSR(I) = SSH(I-1)
      FER(I) = 1000.
      SSR(I) = SIGZ
      EPSZ = SIGZ/E
      EFR(I) = EPSZ
      WRITE(6,703)(I,EEB(I),SSB(I),I=1,NEST)
      IF(SSR(NEST) .GT. 2.0*SIGZ)WRITE(6,704)NEST
702 FORMAT(2F15.7)
703 FORMAT(/3X,'STRESS-STRAIN TABLE FOR HSTRS',F10.0,'I',F10.0,'EPS(I)',F10.0,
2  'SIG(I)',F10.0,'(15,1P2F15.7)')
704 FORMAT(///' ***WARNING*** STRAIN ENERGY SUSPICIOUS.  SIG',F10.0,
2  ' ',F10.0,'GT. 2*SIGZ',F10.0)
*O HMSTRS,23,HMSTRS,30
C      CHANGES IN RMSTRS FOR HSTRS 4/1/77
      IF(IP .EQ. 1)GOTO 10
      REPS = EPSRM(N,K)
      HEPI = EPLRM(N,K)
      GOTO 11
10 REPS = EPSR(N,K)
      HEPI = EPLR(N,K)
11 REPI = HEPI
      IF(REPLM .GE. 0.0) GOTO 13

```

```

SIG221 = -SIG221
DEPS22 = -DEPS22
HEPS = -HEPS
REPL = -REPL
13 REPS = HEPS + DEPS22
   IF(DEPS22 .GE. 0.0)GOTO 14
   IF(HEPS .GE. REPL-EPSZ)GOTO 15
   IF(REPS .GT. -EPSZ)GOTO 16
   CALL DVDINT(-HEPS,SIG22,EEH,SSH,NEST,2)
   SIG22 = -SIG22
   GOTO 17
14 IF(REPS .LE. REPL+EPSZ)GOTO 15
   CALL DVDINT(HEPS,SIG22,EEH,SSH,NEST,2)
   HEPL = REPS - EPSZ
   LMNK = 1
   GOTO 14
15 SIG22 = SIG221 + E*DEPS22
   GOTO 14
16 CALL DVDINT(REPS+2.0*EPSZ,SIG22,EEH,SSH,NEST,2)
   SIG22 = SIG22 - 2.0*SIG2
17 LMNK = 1
   REPL = REPS + EPSZ
18 IF(HEPLM .GE. 0.0)GOTO 19
   SIG22 = -SIG22
   HEPS = -HEPS
   REPL = -REPL
19 IF(IP .EQ. 1) GOTO 20
   FPSHM(N,K) = HEPS
   EPLRM(N,K) = REPL
   GOTO 21
20 EPSB(N,K) = REPS
   FPLR(N,K) = REPL
21 CONTINUE
*IDENT APLFRC
*I RPSLID.5
C   APPLIED FORCE (AND MOMENT) AS END CONDITION 4. 4/12/76
C   APL FRC ASSUMES SHF 3/1 IS INCLUDED IN THE DECK.
C   THE APPLIED FORCE CHANGE TO (1-D) REPSIL FOR BEAMS REQUIRES
C   A USER SUBROUTINE. ENDFRC. TO SUPPLY END FORCES AND MOMENTS.
*I MAIN.28
   COMMON EFR1,EFZ1,EM1,FFR2,EFZ2,FM2
*I START.54
   IF(IPCF1 .EQ. 4)N1V=N1R
*H START.230
   IF(IPCF1 .LT. 5 .AND. IBCE2 .LT. 5)RETURN
*H GRAD.9,GRAD.10
   IF(IPCF1 .NE. 4 .OR. N .GT. N1R)GOTO 8
C   IPCF1=4 AND N=N1R. USE FORWARD DIFFERENCES
   R2 = RTD2*(-3.0* R(N)+4.0* R(N+1)- R(N+2))
   Z2 = RTD2*(-3.0* Z(N)+4.0* Z(N+1)- Z(N+2))
   DR2 = RTD2*(-3.0*DR(N)+4.0*DR(N+1)-DR(N+2))
   DZ2 = RTD2*(-3.0*DZ(N)+4.0*DZ(N+1)-DZ(N+2))
   R22 = RD22*(2.0* R(N)-5.0* R(N+1)+4.0* R(N+2)- R(N+3))
   Z22 = RD22*(2.0* Z(N)-5.0* Z(N+1)+4.0* Z(N+2)- Z(N+3))
   DR22 = RD22*(2.0*DR(N)-5.0*DR(N+1)+4.0*DR(N+2)-DR(N+3))
   DZ22 = RD22*(2.0*DZ(N)-5.0*DZ(N+1)+4.0*DZ(N+2)-DZ(N+3))
   GOTO 20
R IF(IBCE2 .NE. 4 .OR. N .LT. N2R)GOTO 9
C   IBCE2=4 AND N=N2R. USE BACKWARD DIFFERENCES.
*H GRAD.34,GRAD.35
   IF(IPCF1 .NE. 4 .OR. N .GT. N1R )GOTO 18
C IPCF1=4 AND N=N1R (1/2). USE NON-CENTRAL 4 POINT PAD DIFFERENCES
   R22 = RD22*(3.0* R(N)-7.0* R(N+1)+5.0* R(N+2)- R(N+3))

```

```

      Z22 = HD22*(3.0*7(N)-7.0*Z(N+1)+5.0*Z(N+2)-7(N+3))
      DW22 = HD22*(3.0*DR(N)-7.0*DW(N+1)+5.0*DW(N+2)-DW(N+3))
      DZ22 = HD22*(3.0*DZ(N)-7.0*DZ(N+1)+5.0*DZ(N+2)-DZ(N+3))
      GOTO 20
14 IF(IRCE2.NE.4.OR.N.LT.N2H-1)GOTO 19
C IRCE2=4 AND N=N2H-1 (+1/2). USE NON-CENTRAL 4 POINT 2ND DIFFERENCES
*P RESULT.34,RESULT.49
      IF(N.EQ.N2H.AND.IRCE2.EQ.4)GOTO 13
      IF(N.NE.N1H.OR.IRCE1.NE.4)GOTO 16
C      N=N1H AND IRCE1=4. APPLIED FORCE FND1.
      CALL FNDFRC
      AMS(N1H) = EM1
      GOTO 1A
C      N = N2H AND IRCE2=4. APPLIED FORCE FND2.
13 IF(IRCE1.NE.4)CALL ENDFRC
      AMS(N2H) = EM2
      GOTO 1A
*P MOTION.11,MOTION.18
C      APPLIED FORCE (AND MOMENT) AS FND CONDITION 4. 4/12/74
      IF(IRCE1.NE.4.OR.N.GT.N1H+1)GOTO 13
      IF(N.GT.N1H)GOTO 11
C      N=N1H AND IRCE1=4. USE DIFFERENCE (F(N+1/2)-F(N))/(DETA2/2).
      VR = 2.0*RTD2M*(VS(N)*SNRM(N)-QS(N)*SNKM(N)-EFR1)
      V7 = 2.0*RTD2M*(VS(N)*SNKM(N)+QS(N)*SNRM(N)-EFZ1)
      GOTO 35
11 M=N-1
C      M=N1H+1 AND IRCE2=4. USE DIFFERENCE (F(N+1/2)-F(N-1/2))/DETA2
      VR=RTD2M*(VS(N)*SNRM(N)-QS(N)*SNKM(N)-VS(M)*SNRM(M)+QS(M)*SNKM(M))
      VZ=RTD2M*(VS(N)*SNKM(N)+QS(N)*SNRM(N)-VS(M)*SNKM(M)-QS(M)*SNRM(M))
      GOTO 35
13 IF(IRCE2.NE.4.OR.N.LT.N2H-1)GOTO 31
      IF(N.LT.N2H)GOTO 16
C      N=N2H AND IRCE2=4. USE DIFFERENCE (F(N)-F(N-1/2))/(DETA2/2)
      M=N2H-1
      VR = 2.0*RTD2M*(EFR2-VS(M)*SNRM(M)+QS(M)*SNKM(M))
      VZ = 2.0*RTD2M*(EFZ2-VS(M)*SNKM(M)-QS(M)*SNRM(M))
      GOTO 35
16 M = N2H-2
C      N=N2H-1 AND IRCE2=4. USE DIFFERENCE (F(N+1/2)-F(N-1/2))/DETA2
      VP=RTD2M*(VS(N)*SNRM(N)-QS(N)*SNKM(N)-VS(M)*SNRM(M)+QS(M)*SNKM(M))
      VZ=RTD2M*(VS(N)*SNKM(N)+QS(N)*SNRM(N)-VS(M)*SNKM(M)-QS(M)*SNRM(M))
      GOTO 35
*1 HOUNDR.9
      IF(IRCE1.EQ.4)GOTO 20
*1 HOUNDI.7
      IF(IRCE1.EQ.4)GOTO 20
*AF
*DFCK ENDFRC
      SUBROUTINE ENDFRC
C      THE APL FRC CHANGE TO (1-0) REFSIL FOR BEAMS REQUIRES A USER
C      SUBROUTINE ENDFRC TO SUPPLY FORCES AND MOMENTS AT BOTH ENDS.
C      FOR AN INITIALLY STRAIGHT BEAM. R(N)=0.0, Z(N)=(N-2)UETA2.
C      EFZ1>0 DECREASES DZ(2). EFR1>0 DECREASES DW(2)
C      EFZ2>0 INCREASES DZ(N2H). EFR2>0 INCREASES DR(N2H)
C      EM1>0 DECREASES DR(2) AND INCREASES DR(3)
C      EM2>0 INCREASES DR(N2H-1) AND DECREASES DW(N2H)
C
C      CALL MAIN
C      TABULAR INPUT FOR EFZ1, EFTH, AND EFMU. 10/27/74
C      MODIFIED (12/7/74) FORCE RELATIVE TO NORMAL
C      EFR1 = EFMU*EFZ1. EM1 = 2U*(EFZ1*SIN(EFTH)+EFR1*COS(EFTH))
C      DIMENSION LAPELV(7),EFZ1(40),TMF7(40),EFTH(40),TMTH(40),
C      1 EFMU(40),TMMU(40)

```

```

DATA IENDFR/0/
IF(IENDFR.GT. 0)GOTO 1
IENDFR = 1
10/27/76
READ (5,100)NFZPTS.(LABELV(I),I=1,7)
10/27/76
WRITE(6,101) (LABELV(I),I=1,7),NFZPTS
10/29/76
READ (5,102) ( EFZT (I),TMFZ(I),I=1,NFZPTS)
10/27/76
WRITE(6,103) (I,EFZT (I),TMFZ(I),I=1,NFZPTS)
10/27/76
READ (5,100)NFTHS.(LABELV(I),I=1,7)
10/27/76
WRITE(6,104) (LABELV(I),I=1,7),NFTHS
10/29/76
READ (5,102) ( EFTHT(I),TMTH(I),I=1,NFTHS)
10/27/76
WRITE(6,103) (I,EFTHT(I),TMTH(I),I=1,NFTHS)
10/27/76
READ (5,100)NFMUS.(LABELV(I),I=1,7)
10/27/76
WRITE(6,105) (LABELV(I),I=1,7),NFMUS
10/29/76
READ (5,102) ( EFMUT(I),TMMU(I),I=1,NFMUS)
10/27/76
WRITE(6,103) (I,EFMUT(I),TMMU(I),I=1,NFMUS)
10/27/76
EFR2 = 0.0
10/27/76
EFZ2 = 0.0
10/27/76
EM2 = 0.0
10/27/76
1 CALL DVDINT(TIME,EFZ1,TMFZ,EFZT,NFZPTS,2)
10/27/76
CALL DVDINT(TIME,EFTH,TMTH,EFTHT,NFTHS,2)
10/27/76
CALL DVDINT(TIME,EFMU,TMMU,EFMUT,NFMUS,2)
10/27/76
FFR1 = EFMU*EFZ1
10/27/76
EM1 = -70*(SIN(EFTH)*EFZ1 + (1.0-COS(EFTH))*EFR1)
10/29/76
C CODE CHECKING PLOTS. 10/28/76
CC/10/76
P(53) = EFZ1
7/20
C REPLACE COMPUTATION OF FFZ1 AND EFZ2 IN ENDFRC 12/7/76 CARDS
EFOD21
C INTERPOLATE FOR NORMAL AT END WITH QUADRATIC INTERPOLATION
EFOD21
XMX1 = DE - DELTA2
FRCOD21
XMX2 = DE - 2.0*DELTA2
FFC021
C X-X0 = DE. X2-X1 = X1-X0 = DELTA2.
EFOD21
SNRFND=(XMX2*(XMX1*SNR(N1R)-2.0*DE*SNR(N1R+1))+DE*XMX1*SNR(N1R+2))
FRCOD21
1 )*RD22M
EFOD21
SNKFND=(XMX2*(XMX1*SNK(N1R)-2.0*DE*SNK(N1R+1))+DE*XMX1*SNK(N1R+2))
EFOD21
1 )*RD22M
FRCOD21
AAA = SQRT(SNRFND**2 + SNKFND**2)
EFOD21
SNRFND = SNRFND/AAA
FRCOD21
SNKFND = SNKFND/AAA
EFOD21
TFFR1 = EFR1*SNRFND - EFZ1*SNKFND
EFOD21
EFZ1 = FFR1*SNKFND + EFZ1*SNRFND
FRCOD21
EFR1 = TFFR1
EFOD21
C TFFR1 = FFR1*SNR(N1R) - EFZ1*SNK(N1R)
12/7/76
C FFZ1 = FFR1*SNK(N1R) + EFZ1*SNR(N1R)
12/7/76
C EFR1 = TFFR1
12/7/76
PFTURN
10/27/76
100 FORMAT(I10,7A10)
10/27/76
101 FORMAT('1SUBROUTINE ENDFRC 10/27/76. TABULAR DATA FFZ1,EFTH,EFMU',10/27/76
1 /' TABLE 1.',7A10,15,' POINTS',4X,'1',6X,'EFZT ',9X,'TIME')
10/27/76
102 FORMAT(2E15,7)
10/27/76
103 FORMAT(15,1P2F15,7)
10/27/76
104 FORMAT('/' TABLE 2.',7A10,15,' POINTS',4X,'1',6X,'EFTHT',9X,'TIME')
10/27/76
105 FORMAT('/' TABLE 3.',7A10,15,' POINTS',4X,'1',6X,'EFMUT',9X,'TIME')
10/27/76
END
10/27/76
*IDENT EAPFRC
*I RPSL10.5
C EXTERNAL WORK FOR APL FRC OF 4/12/76. (6/3/76)
EAP FRC
*I MAIN.2R
COMMON DWF1P,DTH1P,DWF2P,DTH2P
EAP FRC
*I RPSL10.43
DTH1P = 0.0
EAP FRC
DWF1P = 0.0
EAP FRC
DWF2P = 0.0
EAP FRC
DTH2P = 0.0
EAP FRC
*0 MOTION.40

```

```

IF(N.GT.N1R)GOTO 130
DRI1 = DRI
DZI1 = DZI
130 CONTINUE
IF(IRCE1.NE.4)GOTO 131
N=N1R
DWF = -0.5*(EFW1*(DRI1+DR(N))+EFZ1*(DZI1+DZ(N)))
R2=RTD2*(-3.0*(R(N)+DW(N))+4.0*(R(N+1)+DW(N+1))-(R(N+2)+DR(N+2)))
Z2=RTD2*(-3.0*(Z(N)+DZ(N))+4.0*(Z(N+1)+DZ(N+1))-(Z(N+2)+DZ(N+2)))
D = SQRT(Z2**2 + R2**2)
SNRN = Z2/D
SNKN = -R2/D
DTHFT = ARCSIN(SNK(N)*SNRN - SNR(N)*SNKN)
DWM = -0.5*EM1*(DTH1P + DTHET)
TNRG = TNRG + 0.5*(DWF + DWM + DWE1P)
DWE1P = DWF + DWM
DTH1P = DTHET
131 IF(IRCE2.NE.4)GOTO 132
N=N2R
DWF = 0.5*(EFW2*(DRI+DRS) + EFZ2*(DZI+DZS))
R2=RTD2*(R(N-2)+DR(N-2)-4.0*(R(N-1)+DW(N-1))+3.0*(R(N)+DR(N)))
Z2=RTD2*(Z(N-2)+DZ(N-2)-4.0*(Z(N-1)+DZ(N-1))+3.0*(Z(N)+DZ(N)))
D = SQRT(Z2**2 + R2**2)
SNRN = Z2/D
SNKN = -R2/D
DTHFT = ARCSIN(SNK(N)*SNKN - SNR(N)*SNKN)
DWM = -0.5*EM2*(DTH2P + DTHET)
TNRG = TNRG + 0.5*(DWF + DWM + DWE2P)
DWE2P = DWF + DWM
DTH2P = DTHET
132 CONTINUE
*IDENT 11/2/76
*I WPSLID.5
C INSERT TO REMOVE FAILURE HALT AT FREE OR FORCED END. 11/2/76 11/2/76
*I RMSTRS.13
C INSERT TO REMOVE FAILURE HALT AT FREE OR FORCED END. 11/2/76 11/2/76
IF(G22.GT.0.0)GOTO 2
IF(IP.EQ.2)GOTO 2
IF(N.EQ.N1R.AND. IRCE1.EQ.4)G22=1.0E-10
IF(N.EQ.N2R.AND. IRCE2.EQ.4)G22=1.0E-10
2 CONTINUE
*IDENT INVELC
*I INVEL.6
C LATERAL VELOCITY . VM. TOWARD END 1. 1/8/76
C ENTIRE ROD MOVING WITH CONSTANT VELOCITY. 7/20/76
WRITE(6,333) 1/8/76
333 FORMAT('/// LATERAL VELOCITY TOWARD END 1. 1/8/76///') 1/8/76
DO 29 N=N1V,N2V 7/20/76
DZ(N)=-VR 1/8/76
29 WRITE(6,666)N,DZ(N),DR(N) 1/8/76
RETURN 1/8/76
666 FORMAT(' N= ',I3,' DZ(N)= ',1P13.6,' DR(N)= ',F13.6)
*IDENT EROD26
*I RPSLID.5
C EROSION OF END1. 12/26/76. PUT AFTER APLFRC AND SHR3/1. EROD
C INCLUDE SUBROUTINE ERODE AND CORRECT SUBROUTINE ENDFMC.
*I MAIN.2P
COMMON DE FFGD21
*I WPSLID.120
CALL ERODE EFGD21
*I RESULT.61
IF(N.EQ.N1R)VS(N)=(AMS(N+1)-AMS(N))/(SRA*(DETA2-DE))
IF(N.EQ.N1R)GOTO 200

```

```

*1 MOTION.26
  IF(N.NE.N1H.OF.IRCE1.NE.4)GOTO36
C   REPLACES VP,VZ COMPUTATION IN MOTION WHEN N=N1H AND IRCE1=4
  V6 = (VS(N)*SNRM(N)-QS(N)*SNRM(N)-FFR1)/(0.5*DETA2 -DF)
  V7 = (VS(N)*SNRM(N)+QS(N)*SNRM(N)-FFZ1)/(0.5*DETA2 -UF)
36 CONTINUE
*1 STRAIN.81
  IF(J1.GE.N1H)GOTO 44
  EPSS1(I) = EPSS2(I) = 0.0
  EPSANG(I) = EPSANH(I) = -1.0E20
  GOTO 46
44 CONTINUE
*1 STRAIN.97
  IF(NQ1.GE.N1H)GOTO 49
  D1 = D2 = 0.0
  GOTO(71,50).LINK
49 CONTINUE
*1 PDATA.7
  N1H = 7
*1 PDATA.61
  100 N1R = N1A
  RETURN
*AF
*DECK ERODE
  SUBROUTINE FRODE
C   INSERT ANY INPUT FOR ERODE AFTER INPUT FOR ENDFRC.
C   THIS USER SUBROUTINE COMPUTES THE EROSION ON END1 OF A ROD.
C   THIS VERSION ASSUMES TABLE OF FROSION FROM END OF ROD VS TIME.
C   THE TABLE PRESUMABLY AGREES WITH ENDFRC VS TIME TABLE IN ENDFRC
C   AND STRESS-STRAIN TABLE READ IN INPUT.
*CALL MAIN
  DIMENSION LARLTE(7),TEROD(100),XEND(100)
  DATA IERODE/0/
  IF(IERODE.EQ.1)GOTO 10
  DECOM = DETA2/4.0
  IERODE = 1
  FRODP = 0.0
  DF = 0.0
  READ(5,100)NTEPTS,(LARLTE(I),I=1,7)
  WRITE(6,101)NTEPTS,(LARLTE(I),I=1,7)
  READ(5,102)(TEROD(I),XEND(I),I=1,NTEPTS)
  WRITE(6,103)((TEROD(I),XEND(I),I=1,NTEPTS)
100 FORMAT(110,7A10)
101 FORMAT('11SUBROUTINE FRODE 5/26/77, TABULAR DATA TIME VS XEND'/
2  '15,' POINTS',5X,7A10,7X,'1',7X,'TIME',9X,'XEND')
102 FORMAT(2E15,7)
103 FORMAT(15,2E15,7)
10 CALL DVDINT(TIME,EROD,TEROD,XEND,NTEPTS,2)
  DELDF = FROD - ERODP
  IE = DF + DELDF
  ERODP = FROD
  P(54) = FROD
C   CODE CHECKING PLOT 5/27/77
10 IF(DF.LT.DECOM)GOTO 20
  DF = DF -DETA2
  R(N1R) = Z(N1R) = -1.0E20
C   THE FOLLOWING CARD IS NEEDED IF THE DECK INCLUDES MSQSVS.
  AMS(N1H) = QS(N1H) = VS(N1H) = 0.0
  N1R = N1V = N1A = N1R+1
  GOTO 19
20 RETURN
  END

```

```

      PRODING ROD (254 BY 8.12H MM)      REPEAT OF 6/8/77 RUN
      36      6 2.0
130      0 9999 .5000000E-6
      4
      0      0 9999 68.44757 .001
1.945714E11 .27      1.39000000F9 7790.      .008128      4      0
0.007      1.340      F9
0.008      1.514      F9
0.009      1.564      F9
0.010      1.597      F9
0.011      1.624      F9
0.012      1.648      F9
0.013      1.669      F9
0.014      1.688      F9
0.015      1.705      F9
0.016      1.721      F9
0.017      1.736      F9
0.018      1.750      F9
0.019      1.763      F9
0.020      1.776      F9
0.022      1.799      F9
0.024      1.820      F9
0.026      1.840      F9
0.028      1.860      F9
0.030      1.879      F9
0.032      1.897      F9
0.034      1.915      F9
0.036      1.932      F9
0.038      1.949      F9
0.040      1.966      F9
0.045      2.008      F9
0.050      2.048      F9
0.055      2.086      F9
0.060      2.126      F9
0.065      2.164      F9
0.070      2.202      F9
0.075      2.239      F9
0.080      2.276      F9
0.085      2.312      F9
0.090      2.348      F9
0.095      2.384      F9
0.100      2.419      F9
0.105      2.454      F9
0.110      2.489      F9
0.115      2.523      F9
0.120      2.557      F9
0.125      2.590      F9
0.130      2.624      F9
0.135      2.656      F9
0.140      2.689      F9
0.145      2.721      F9
0.150      2.753      F9
0.30      3.713      F9
-1.0
20      1      1      1
2      70      130
2      70      130
2      70      130
2      53      54
      0.021167      4
      0.005      45.0      135.0      0
      0.010      45.0      135.0      0
      0.015      45.0      135.0      0

```

.01926	45.0	135.0	0
.03967	45.0	135.0	0
.06	45.0	135.0	0
.254	.004064	1	
.004064			
4.19606E-31.35467E-3-3.3547E-3			
6.98023E-31.35467E-3-2.0124E-3			
7.97487E-31.35467E-3-6.7086E-4			
7.97487E-31.35467E-3 6.7086E-4			
6.48023E-31.35467E-3 2.0126E-3			
4.19606E-31.35467E-3 3.3543E-3			

1000.

29 FFZ1 AND TIME 6/3/77 CASE B

0.0	0.0	
72160.00	1.080	E-6
78750.00	1.590	E-6
81140.00	1.790	F-6
82860.00	2.143	E-6
85550.00	2.715	E-6
87650.00	2.915	F-6
89380.00	3.115	F-6
90890.00	3.315	F-6
92270.00	3.515	E-6
95090.00	3.890	F-6
97600.00	4.372	F-6
99940.00	4.847	F-6
102160.00	5.257	E-6
104310.00	5.579	E-6
106390.00	5.968	E-6
110430.00	6.580	E-6
114350.00	7.123	E-6
118200.00	7.574	E-6
121460.00	7.943	F-6
125650.00	8.240	E-6
129260.00	8.495	F-6
132790.00	8.717	E-6
136250.00	8.917	E-6
139630.00	9.117	F-6
142940.00	9.317	F-6
159310.00	9.817	E-6
192050.00	19.817	F-6
192050.00	1.00000	

2 THETA = 0.0. 3/8/77

0.0 0.0

0.0 1.0

2 MI = 0.0. 3/8/77

0.0 0.0

0.0 1.0

29 TIME AND XEND 6/3/77 CASE B

0.000	F-6	3.000	E-3
1.080	F-6	6.000	E-3
1.590	E-6	9.000	F-3
1.790	F-6	9.256	E-3
2.143	F-6	9.256	E-3
2.715	F-6	9.256	F-3
2.915	E-6	9.403	E-3
3.115	F-6	9.642	E-3
3.315	E-6	9.644	E-3
3.515	F-6	9.855	E-3
3.890	E-6	9.855	E-3
4.372	F-6	9.855	F-3
4.847	F-6	9.855	F-3
5.257	F-6	9.855	F-3

```

5.579      F-6  9.855      E-3
5.968      F-6  9.855      F-3
6.580      F-6  9.855      F-3
7.123      F-6  9.855      F-3
7.574      F-6  9.855      F-3
7.943      F-6  9.855      F-3
8.240      F-6  9.855      F-3
8.495      F-6  9.855      E-3
8.717      F-6  9.855      F-3
8.917      F-6  9.855      E-3
9.117      F-6  9.893      E-3
9.317      F-6  9.915      F-3
9.817      F-6  9.915      F-3
19.817     F-6  9.915      E-3
1.00000    9.915      F-3

```

?

*IDFNT PLOTP

```

*O RP1PLT.18
  READ(NPLOT)NNPF.(NPF(I),I=1,NNPF)

```

PLOTP

```

*O RP1PLT.40
  1.(PDAT(J),J=1,NNPF)

```

PLOTP

*ID MSQSVS

*O MAIN.10

```

C      PLOTTING MS. QS. AND VS. 3/19/76
      COMMON/MSQSVS/ AMS(103),QS(103),VS(103),XSN(103)
      LFVFL2,TIM,PC,AMS

```

MSQSVS

MSQSVS

MSQSVS

*I RP1PLT.61

```

C      PLOTTING MS. QS. AND VS. 3/19/76
      READ (NPLOT)(AMS(N),QS(N),VS(N),N=1,N1)
      IF(EOF(NPLOT).NE.0)GOTO 2H

```

MSQSVS

MSQSVS

MSQSVS

*O PLOT3D.53

XMAR = 3.0

MSQSVS

*I PLOT3D.93

C PLOTTING MS. QS. AND VS. 3/19/76

MSQSVS

IF(NCYCLE.EQ.0)PFTLPM

MSQSVS

ENCODE(10,711,M0H)NCYCLE

MSQSVS

711 FORMAT(5HCYCLE,I4,1M>)

MSQSVS

MDF1 = 10MMESH PTS.>

MSQSVS

CALL PLTPGE

MSQSVS

DO 72 I=1,I2

MSQSVS

XSN(I) = FLOAT(I)+1.0

MSQSVS

72 CONTINUE

MSQSVS

CALL FIXSCA(XSN(1), 12,7.9,DSYS,XMINS,XMAXS,DELDXS)

MSQSVS

CALL FIXSCA(AMS(1), 12,6.4,DSYS,YMINS,YMAXS,DELDYS)

MSQSVS

CALL PLTSCA(3.0, 2.0,0.0,0.0,0.0,1.0,1.0)

MSQSVS

MD = 5H M>

ANYUNITS

CALL PLTSYM(.1,MD, 90.0, -1.2,2.5)

MSQSVS

CALL PLTSYM(.1,MDF,0.0, 3.5,-1.0)

MSQSVS

CALL PLTSYM(.1,MDF1,0.0, 3.5,-0.6)

MSQSVS

CALL PLTSCA(3.0, 2.0,XMINS,YMINS,DSYS,DSYS)

MSQSVS

CALL PLTTP2(1.0,XSN(1),AMS(1), 12,0)

MSQSVS

CALL PLTAXS(DFLUXS,DFLDYS,XMINS,XMAXS,YMINS,YMAXS,4)

MSQSVS

CALL LABELA(DFLUXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,1.0,1.0)

MSQSVS

I2M1 = I2-1

MSQSVS

DO 73 I=1,I2M1

MSQSVS

XSN(I) = XSN(I)+0.5

MSQSVS

73 CONTINUE

MSQSVS

CALL FIXSCA(QS(1),I2M1,6.4,DSYS,YMINS,YMAXS,DELDYS)

MSQSVS

CALL PLTSCA(3.0,I2,0.0,0.0,0.0,1.0,1.0)

MSQSVS

MD = 5H Q>

ANYUNITS

CALL PLTSYM(.1,MD, 90.0, -1.2,2.5)

MSQSVS

CALL PLTSYM(.1,MDF,0.0, 3.5,-1.0)

MSQSVS

CALL PLTSYM(.1,MDF1,0.0, 3.5,-0.6)

MSQSVS

CALL PLTSCA(3.0,12.0,XMINS,YMINS,DSXS,DSYS)	MSGSVS
CALL PLTDT2(1.0,XSN(1),QS(1),I2M1.0)	MSGSVS
CALL PLTAXS(DFLUXS,DELFYS,XMINS,XMAXS,YMINS,YMAXS,4)	MSGSVS
CALL LAHFLA(DFLUXS,DELFYS,XMINS,XMAXS,YMINS,YMAXS,1.0,1.0)	MSGSVS
CALL FIXSCA(VS(1),I2M1,6.6,DSYS,YMINS,YMAXS,DELDYS)	MSGSVS
CALL PLTSCA(3.0,22.0,0.0,0.0,0.0,1.0,1.0)	MSGSVS
HD = 5H V>	ANYUNITS
CALL PLTSYM(.1,HD,90.0,-1.2,2.5)	MSGSVS
CALL PLTSYM(.1,HDP,0.0,3.5,-1.0)	MSGSVS
CALL PLTSYM(.1,HDP1,0.0,3.5,-0.6)	MSGSVS
CALL PLTSCA(3.0,22.0,XMINS,YMINS,DSXS,DSYS)	MSGSVS
CALL PLTDT2(1.0,XSN(1),VS(1),I2M1.0)	MSGSVS
CALL PLTAXS(DFLUXS,DELFYS,XMINS,XMAXS,YMINS,YMAXS,4)	MSGSVS
CALL LAHFLA(DFLUXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,1.0,1.0)	MSGSVS
I = 4	MSGSVS

?
1.0 3.0
?

HRL REVSIL CONF

```

ENDING 000 (254 BY 0.128 MM)  MPFAT OF 6/8/77 RUN
COMPUTATIONS FOR ENERGY USE DTAL= .100000E+01
%0 RESSES IN F102 DIRECTION  IN102= .705555E-02)

PENDING TIME INCREMENT= .040610E-05
REPRNAME TIME INCREMENT= .130747E-05
INPUT TIME INCREMENT= .500000E-04

TIME INCREMENT USED BY REPSIL= .500000E-04

YOUNG'S MODULUS = .190871E+12
POISSON'S RATIO = .270000E+00
MASS DENSITY = .770000E+04

YIELD STRESS = .130000E+10
THICKNESS = .012400E-02

START AT TIME STEP 0
FINAL TIME STEP 130
SURFACE STRAINS EVERY 20 TIME STEP
RESTART WRITE EVERY 9999 TIME STEP

LAYER = 0
LOAD = 0
NSTEN = 0
LPRESS = 0

1/2/2/4/ = CLAMPER/SYMMETRY/MINERD/FREE/
END1 (INCE1) = 0
END2 (INCE2) = 4

BOUNDARY CONDITIONS
PRINT OPTION CONTINUAL CARD
0/1 = NO PRINT/PRINT
1 DISPLACEMENT INCREMENTS
1 CARTESIAN COORDINATES, PRESSURE
1 SURFACE NORMAL VECTOR COMPONENTS

PRINT INFORMATION AT THE FOLLOWING TIME STEPS
70 130
PRINT L MATRIN (LPAT) AT THE FOLLOWING TIME STEPS
70 130
3-0 PLOTS FOR THE FOLLOWING TIME STEPS
70 130

CONSTITUTIVE RELATION ELASTOPLASTIC-NO WORK HARDENING-STRAIN RATE INDEPENDENT

J 55101J1 SEPS1J1 PSM1J1 1/PSR1J1
1 1.300000E+00 7.000000E-03 0. 0.

STRESS-STRAIN AND STRAIN RATE PARAMETERS
PSM1J1 SEPS1J1 PSM1J1 1/PSR1J1
1 1.300000E+00 7.000000E-03 0. 0.

START DAMPING AFTER TIME STEP 9999 TIME = .5000E-02
DAMPF = .0000E+02
  
```

LATERAL VELOCITY INCHES PER 1. 1/4/76

N= 2	07101=	-1.000000E+03	DM1N1=	0.
N= 3	07101=	-1.000000E+03	DM1N1=	0.
N= 4	07101=	-1.000000E+03	DM1N1=	0.
N= 5	07101=	-1.000000E+03	DM1N1=	0.
N= 6	07101=	-1.000000E+03	DM1N1=	0.
N= 7	07101=	-1.000000E+03	DM1N1=	0.
N= 8	07101=	-1.000000E+03	DM1N1=	0.

N= 9	07/01=	-1.000000E+03	DR(N)=	0.
N= 10	07/01=	-1.000000E+03	DR(N)=	0.
N= 11	07/01=	-1.000000E+03	DR(N)=	0.
N= 12	07/01=	-1.000000E+03	DR(N)=	0.
N= 13	07/01=	-1.000000E+03	DR(N)=	0.
N= 14	07/01=	-1.000000E+03	DR(N)=	0.
N= 15	07/01=	-1.000000E+03	DR(N)=	0.
N= 16	07/01=	-1.000000E+03	DR(N)=	0.
N= 17	07/01=	-1.000000E+03	DR(N)=	0.
N= 18	07/01=	-1.000000E+03	DR(N)=	0.
N= 19	07/01=	-1.000000E+03	DR(N)=	0.
N= 20	07/01=	-1.000000E+03	DR(N)=	0.
N= 21	07/01=	-1.000000E+03	DR(N)=	0.
N= 22	07/01=	-1.000000E+03	DR(N)=	0.
N= 23	07/01=	-1.000000E+03	DR(N)=	0.
N= 24	07/01=	-1.000000E+03	DR(N)=	0.
N= 25	07/01=	-1.000000E+03	DR(N)=	0.
N= 26	07/01=	-1.000000E+03	DR(N)=	0.
N= 27	07/01=	-1.000000E+03	DR(N)=	0.
N= 28	07/01=	-1.000000E+03	DR(N)=	0.
N= 29	07/01=	-1.000000E+03	DR(N)=	0.
N= 30	07/01=	-1.000000E+03	DR(N)=	0.
N= 31	07/01=	-1.000000E+03	DR(N)=	0.
N= 32	07/01=	-1.000000E+03	DR(N)=	0.
N= 33	07/01=	-1.000000E+03	DR(N)=	0.
N= 34	07/01=	-1.000000E+03	DR(N)=	0.
N= 35	07/01=	-1.000000E+03	DR(N)=	0.
N= 36	07/01=	-1.000000E+03	DR(N)=	0.
N= 37	07/01=	-1.000000E+03	DR(N)=	0.
N= 38	07/01=	-1.000000E+03	DR(N)=	0.

TIME STEP 130 TIME .4500000E-04

DISPLACEMENT INCREMENTS BETWEEN U.S. 129 AND 130

PRESSURE

(COORDINATES

N	P	COORDINATES				PRESSURE			
		07 (N)	08 (N)	09 (N)	10 (N)	07 (N)	08 (N)	09 (N)	10 (N)
1	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.	0.	0.	0.
11	0.	0.	0.	0.	0.	0.	0.	0.	0.
12	0.	0.	0.	0.	0.	0.	0.	0.	0.
13	0.	0.	0.	0.	0.	0.	0.	0.	0.
14	0.	0.	0.	0.	0.	0.	0.	0.	0.
15	0.	0.	0.	0.	0.	0.	0.	0.	0.
16	0.	0.	0.	0.	0.	0.	0.	0.	0.
17	0.	0.	0.	0.	0.	0.	0.	0.	0.
18	0.	0.	0.	0.	0.	0.	0.	0.	0.
19	0.	0.	0.	0.	0.	0.	0.	0.	0.
20	0.	0.	0.	0.	0.	0.	0.	0.	0.
21	0.	0.	0.	0.	0.	0.	0.	0.	0.
22	0.	0.	0.	0.	0.	0.	0.	0.	0.
23	0.	0.	0.	0.	0.	0.	0.	0.	0.
24	0.	0.	0.	0.	0.	0.	0.	0.	0.
25	0.	0.	0.	0.	0.	0.	0.	0.	0.
26	0.	0.	0.	0.	0.	0.	0.	0.	0.
27	0.	0.	0.	0.	0.	0.	0.	0.	0.
28	0.	0.	0.	0.	0.	0.	0.	0.	0.
29	0.	0.	0.	0.	0.	0.	0.	0.	0.
30	0.	0.	0.	0.	0.	0.	0.	0.	0.
31	0.	0.	0.	0.	0.	0.	0.	0.	0.
32	0.	0.	0.	0.	0.	0.	0.	0.	0.
33	0.	0.	0.	0.	0.	0.	0.	0.	0.
34	0.	0.	0.	0.	0.	0.	0.	0.	0.
35	0.	0.	0.	0.	0.	0.	0.	0.	0.
36	0.	0.	0.	0.	0.	0.	0.	0.	0.
37	0.	0.	0.	0.	0.	0.	0.	0.	0.
38	0.	0.	0.	0.	0.	0.	0.	0.	0.

N	P	COORDINATES				PRESSURE			
		07 (N)	08 (N)	09 (N)	10 (N)	07 (N)	08 (N)	09 (N)	10 (N)
139	0.	0.	0.	0.	0.	0.	0.	0.	0.
140	0.	0.	0.	0.	0.	0.	0.	0.	0.
141	0.	0.	0.	0.	0.	0.	0.	0.	0.
142	0.	0.	0.	0.	0.	0.	0.	0.	0.
143	0.	0.	0.	0.	0.	0.	0.	0.	0.
144	0.	0.	0.	0.	0.	0.	0.	0.	0.
145	0.	0.	0.	0.	0.	0.	0.	0.	0.
146	0.	0.	0.	0.	0.	0.	0.	0.	0.
147	0.	0.	0.	0.	0.	0.	0.	0.	0.
148	0.	0.	0.	0.	0.	0.	0.	0.	0.
149	0.	0.	0.	0.	0.	0.	0.	0.	0.
150	0.	0.	0.	0.	0.	0.	0.	0.	0.
151	0.	0.	0.	0.	0.	0.	0.	0.	0.
152	0.	0.	0.	0.	0.	0.	0.	0.	0.
153	0.	0.	0.	0.	0.	0.	0.	0.	0.
154	0.	0.	0.	0.	0.	0.	0.	0.	0.
155	0.	0.	0.	0.	0.	0.	0.	0.	0.
156	0.	0.	0.	0.	0.	0.	0.	0.	0.
157	0.	0.	0.	0.	0.	0.	0.	0.	0.
158	0.	0.	0.	0.	0.	0.	0.	0.	0.
159	0.	0.	0.	0.	0.	0.	0.	0.	0.
160	0.	0.	0.	0.	0.	0.	0.	0.	0.
161	0.	0.	0.	0.	0.	0.	0.	0.	0.
162	0.	0.	0.	0.	0.	0.	0.	0.	0.
163	0.	0.	0.	0.	0.	0.	0.	0.	0.
164	0.	0.	0.	0.	0.	0.	0.	0.	0.
165	0.	0.	0.	0.	0.	0.	0.	0.	0.
166	0.	0.	0.	0.	0.	0.	0.	0.	0.
167	0.	0.	0.	0.	0.	0.	0.	0.	0.
168	0.	0.	0.	0.	0.	0.	0.	0.	0.
169	0.	0.	0.	0.	0.	0.	0.	0.	0.
170	0.	0.	0.	0.	0.	0.	0.	0.	0.
171	0.	0.	0.	0.	0.	0.	0.	0.	0.
172	0.	0.	0.	0.	0.	0.	0.	0.	0.
173	0.	0.	0.	0.	0.	0.	0.	0.	0.
174	0.	0.	0.	0.	0.	0.	0.	0.	0.
175	0.	0.	0.	0.	0.	0.	0.	0.	0.
176	0.	0.	0.	0.	0.	0.	0.	0.	0.
177	0.	0.	0.	0.	0.	0.	0.	0.	0.
178	0.	0.	0.	0.	0.	0.	0.	0.	0.
179	0.	0.	0.	0.	0.	0.	0.	0.	0.
180	0.	0.	0.	0.	0.	0.	0.	0.	0.

130	28	1.4001113F	-13	P.4R331700E	-17	P.3648150F	04	1.0210773F	00	-7.30333015E	-14	N.1735395F	04	-9.753215E	01
130	29	7.2729307F	-13	4.0876513F	-11	P.6233782F	04	1.8195369E	00	1.0615109F	-14	P.1010577F	04	1.0053344E	-13
130	21	-7.130114F	-13	4.491513F	-11	P.1064530F	04	1.0190227E	00	2.6023007F	-14	N.0330800F	04	3.2379564E	-13
130	27	3.293459F	-13	1.0246209F	-11	P.1409429F	04	1.0194735E	00	-5.1770809E	-15	7.9028737F	04	3.3040104E	-15
130	24	7.4530772F	-13	4.3117204F	-11	P.8689101F	04	1.0170733E	00	-2.7749115F	-14	7.0275399F	04	3.5922843E	-13
130	25	7.0564784F	-13	1.5191567F	-10	7.8159041F	04	1.0161103E	00	1.4574280E	-14	7.0496110F	04	3.3200850E	-13
130	26	3.7480208F	-13	4.4.11435E	-11	6.0500564F	04	1.0133055E	00	2.6435310F	-14	6.7542360E	04	9.2325725E	-13
130	21	-6.1172204F	-13	1.6475345F	-10	5.3636633F	04	1.0117118F	00	-2.2736404E	-14	5.4592937F	04	3.5126293E	-13
130	24	4.7743767F	-13	7.0466508E	-11	3.1664709F	04	1.0033942E	00	-1.4502710E	-14	7.0325307F	04	3.4612104E	-13
130	20	-1.1800294F	-13	1.5649064F	-11	2.9154167E	04	1.0000061F	00	0.6407511E	-15	2.0095717E	04	1.0070910E	-13
130	30	-2.7630464F	-13	4.4032037F	-11	6.2064531F	03	1.0020005E	00	0.4077025F	-15	4.2733513F	03	0.5579464E	-14
130	31	5.5074727F	-14	1.9237742F	-11	1.5770110F	03	1.0011093E	00	-2.5111937E	-15	1.5756532E	03	3.3095005E	-13
130	32	7.2009704F	-13	7.1248910F	-11	3.3149020F	03	1.0013084E	00	-0.1517957E	-16	1.0160365E	03	3.1264817E	-13
130	33	7.0631307E	-14	6.0467007F	-11	5.1711424F	03	1.0013725E	00	0.6731293E	-15	5.1643242E	03	7.5359510E	-13
130	34	4.9063964E	-13	3.7397701F	-11	7.2591712F	03	1.0001741E	00	7.6004110E	-10	2.2503214E	03	1.7093000E	-13
130	35	2.0074719F	-13	7.3196426E	-12	7.4969064F	02	1.0001741E	00	4.3644068F	-15	7.0955003E	02	0.3517220E	-14
130	36	1.5016027F	-13	3.6750209F	-11	6.1003351F	02	1.0001303F	00	0.2645637E	-17	6.1799005E	02	7.4600510E	-14
130	37	1.0177742F	-13	1.6367004F	-11	-5.9712204F	02	9.9900740E	01	9.1000370E	-15	-5.9710901F	02	3.6994369E	-13

[illegible]

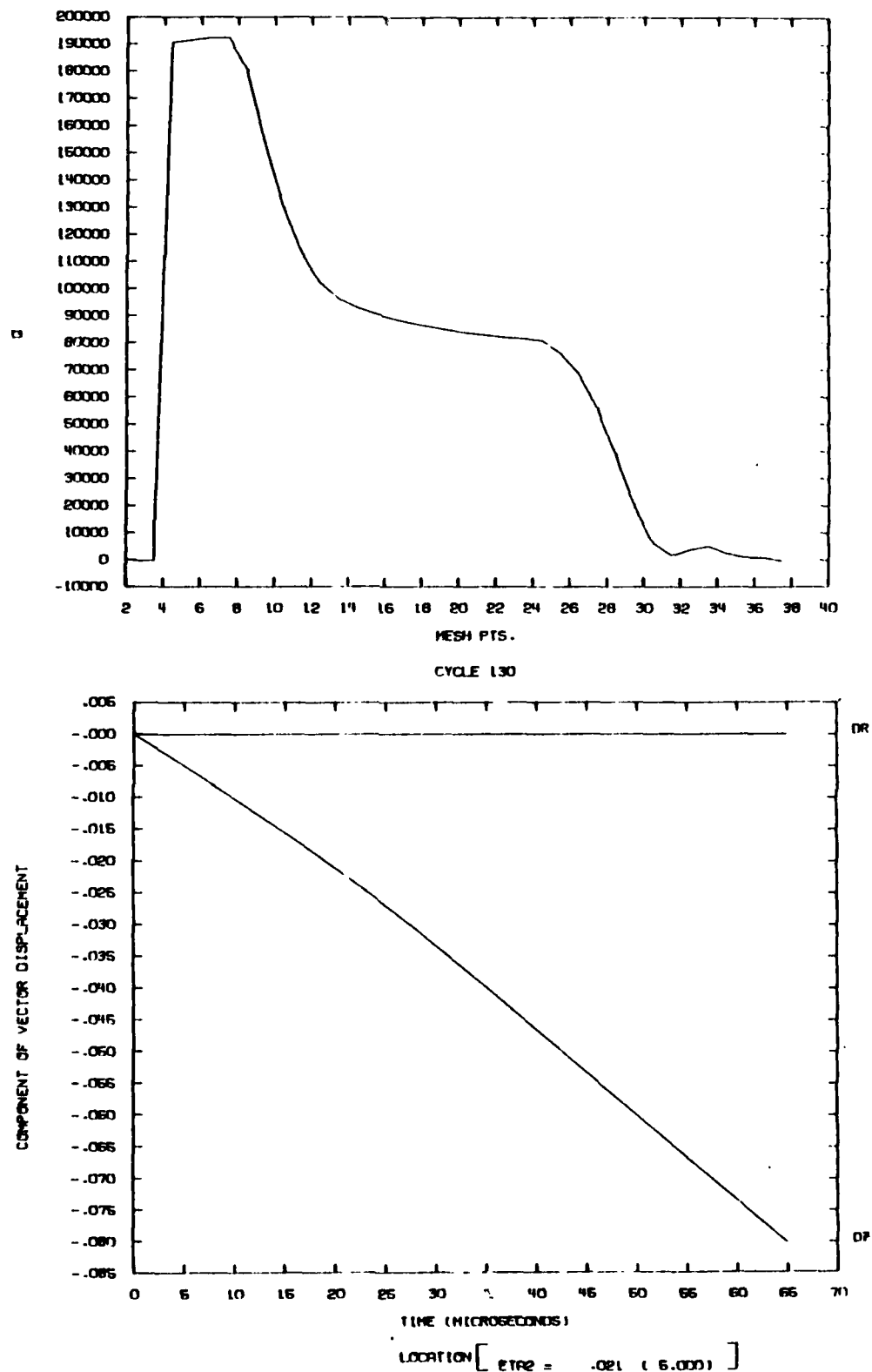
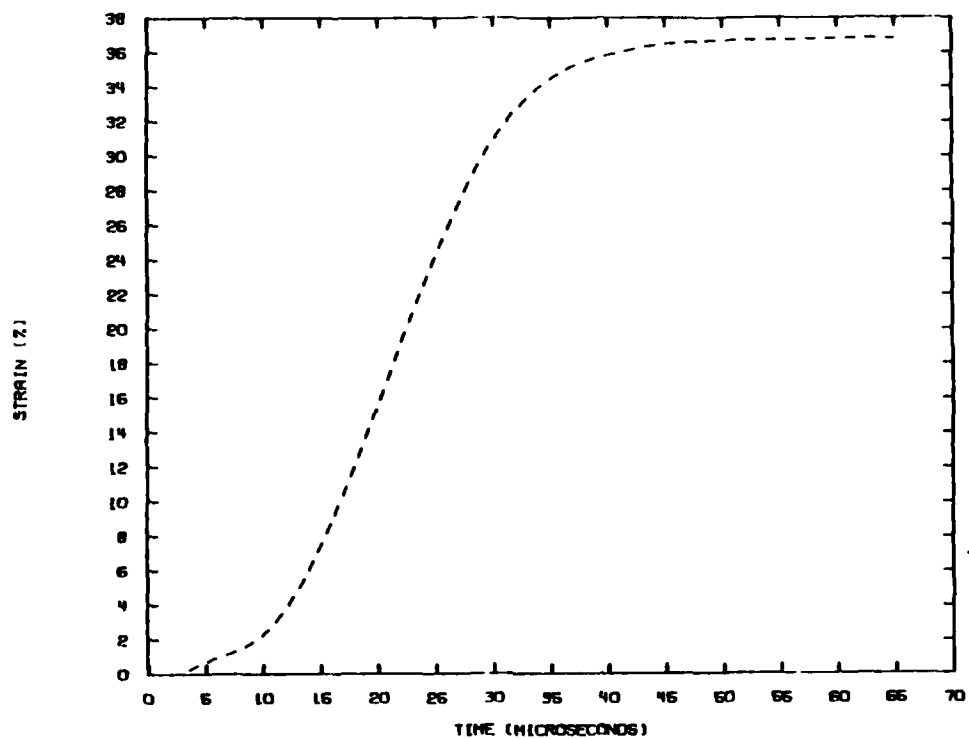
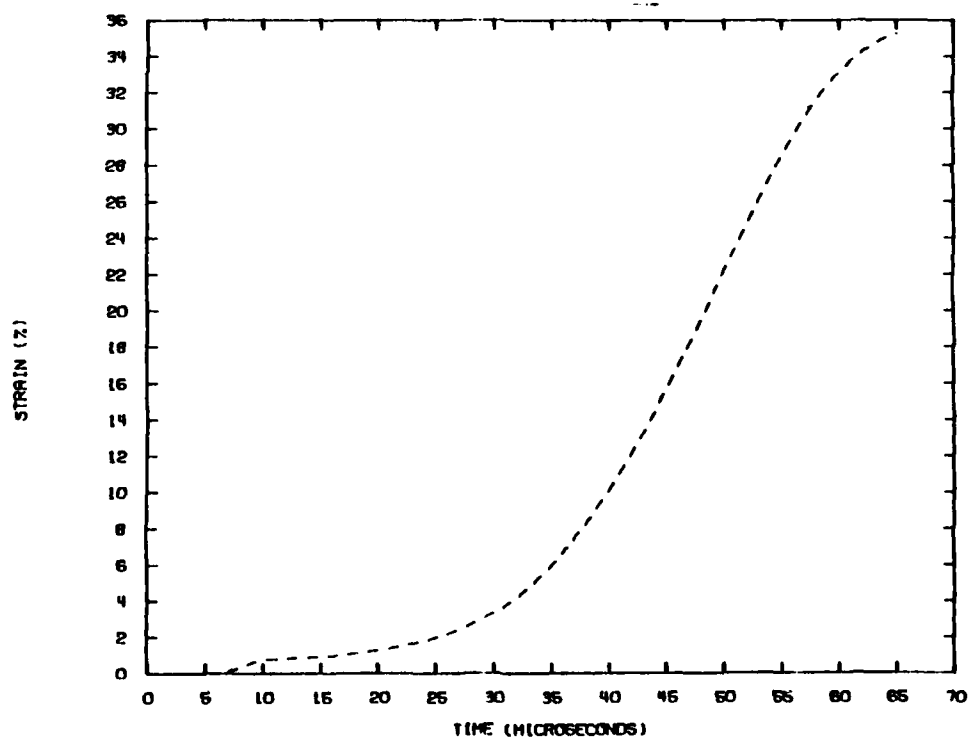


Figure E.2.1. Plot of axial force, Q , at cycle 130 and the displacement history at mesh point 5.



LOCATION [ETR2 = .019 (4.730)] OUTER



LOCATION [ETR2 = .040 (7.623)] OUTER

— ETR1 COMPONENT

- - - ETR2 COMPONENT

Figure E.2.2. Strain plots at 19.26 mm and 39.67 mm.

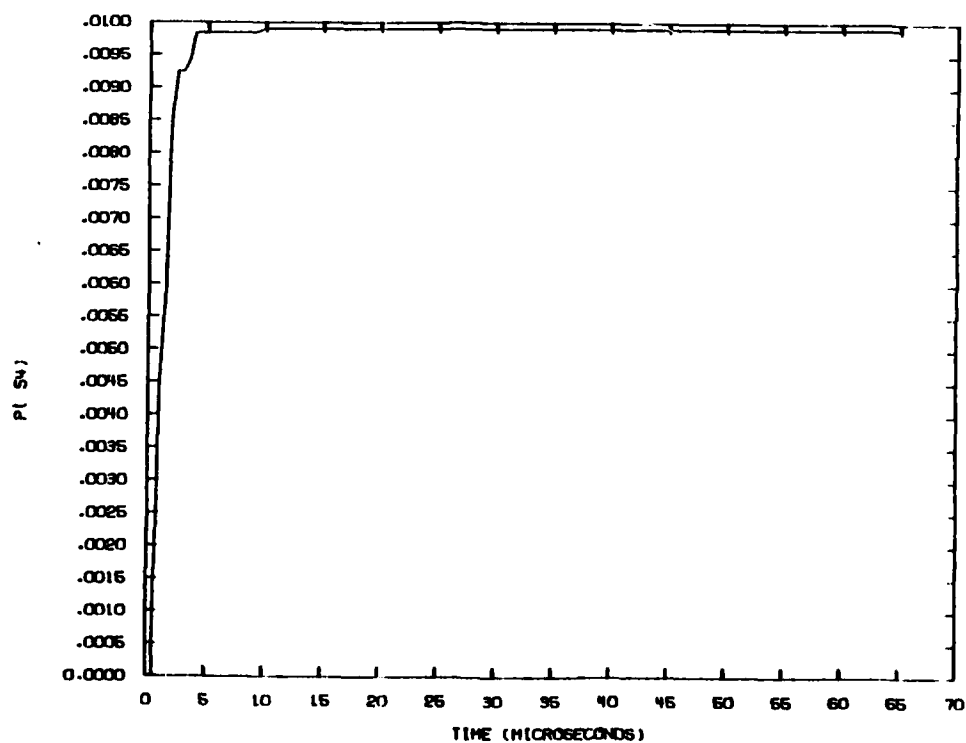
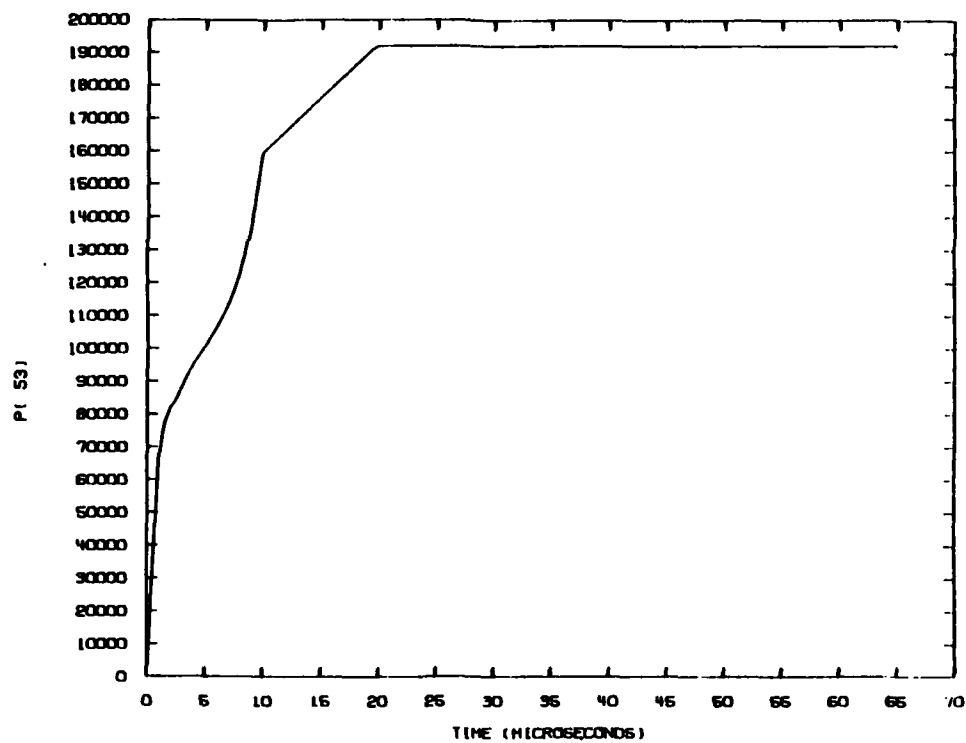
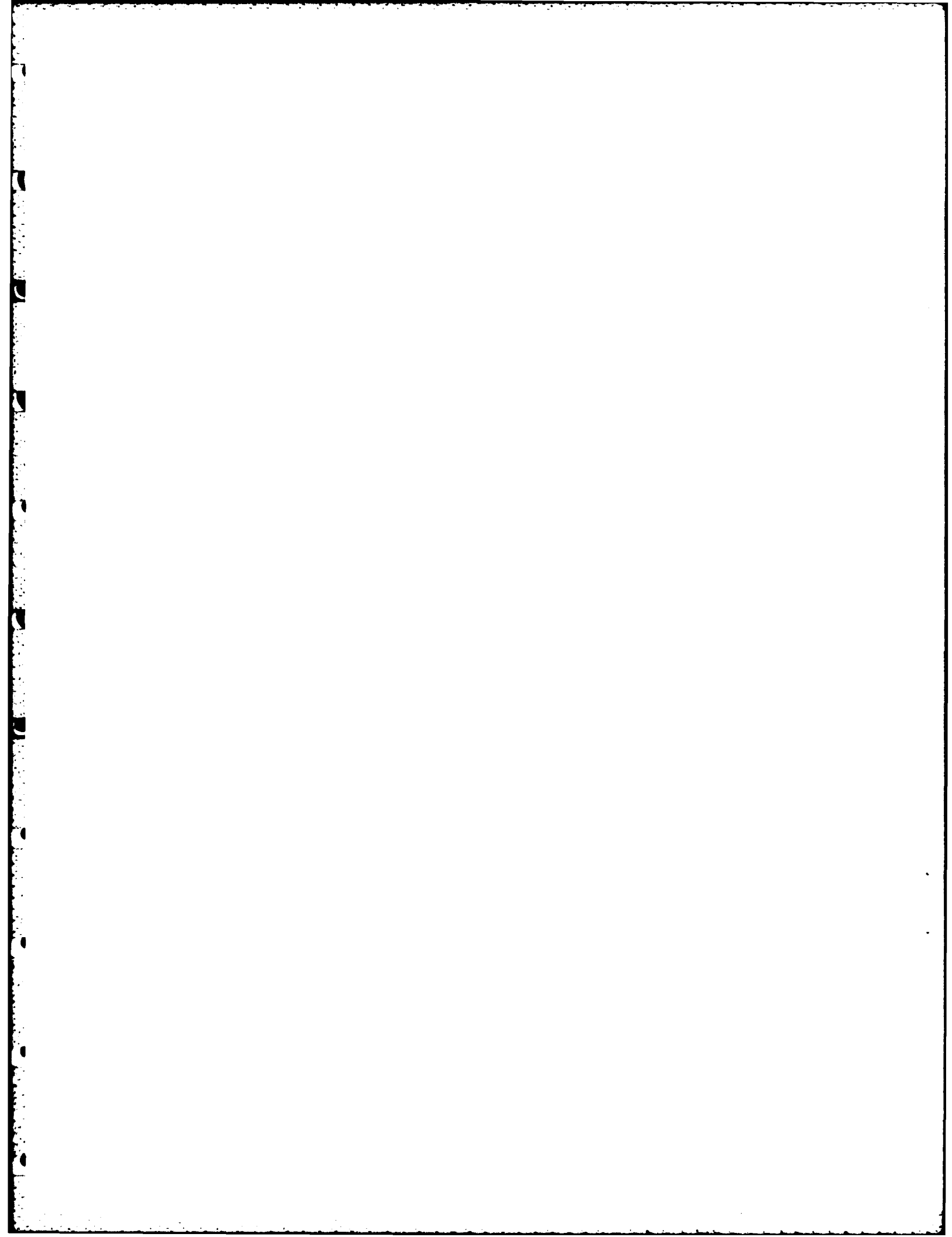


Figure E.2.3. Plots of force on endl, P(53), and erosion at endl, P(54), as functions of time.



APPENDIX F

SOME ALTERNATE USER SUBROUTINES

UPDATE input cards for alternate versions of the user subroutines INGEOM, PRESS, and ENDFRC are listed in this appendix. Other versions of the user subroutines are listed in Appendices C and E. Descriptions for user subroutines are given in Section 4.

The following are listed in the order given:

PRESS for Two Phase Pressure Decay in a Cylinder. This is described in Section 4.2.3.

INGEOM for an axisymmetric cylinder. This is described in Section 4.1.2.

ENDFRC for one or both ends free. This is described in Section 4.5.1.

```

*00 PRESS.4,PRESS.14
C      PRESS SUBROUTINE, 3/10/76. TABULAR DATA.
C      POSITION ZSI = 0 CORRESPONDS TO THE 1-D REPSIL POINT N=2.
C      THIS PRESS SET IS FOR A SYMMETRIC AXISYMMETRIC CYLINDER
      DIMENSION ZSI(50),TSI(50),PSI(50),TS(103),PS(103),TSH(103),
1 PSR(103),PTU(N)
      DATA IPRESS/0/
      IF(IPRESS.GT.0)GOTO 30
      IPRESS = 1
      READ(5,100) PHAR,BETA,(PID(I),I=1,6)
      READ(5,105) CON CZ,CONCT,CONCP,TDIF
      WRITE(6,102) PHAR,BETA,(PID(I),I=1,6),TDIF,CON CZ,CONCT,CONCP
      READ(5,103) ALPHAS,ROUSE,ISM
      READ(5,101) (ZSI(I),TSI(I),PSI(I),I=1,ISM)
      WRITE(6,104) ALPHAS,(ZSI(I),TSI(I),PSI(I),I=1,ISM)
C      CONVERT DATA TO Z=IN, T=SEC, P=LB/SGIN.
      DO 2 I=1,ISM
      ZSI(I) = CON CZ*ZSI(I)
      TSI(I) = CONCT*(TSI(I)-TDIF)
      PSI(I)=CONCP*PSI(I)
2 CONTINUE
      WRITE(6,106) (ZSI(I),TSI(I),PSI(I),I=1,ISM)
C      INITIATE DATA FOR COMPUTING THE PRESSURE
      DO 20 N=N1V,N2V
      DEN = FLOAT(N-2)*CETA2
      CALL DVCINT(DEN,TS(N),ZSI,TSI,ISM,2)
      CALL DVCINT(DEN,PS(N),ZSI,PSI,ISM,2)
      TSH(N) = TS(N) - ALOG(PHAR/PS(N))/ALPHAS
      PS(N) = PS(N)*EXP(ALPHAS*TS(N))
      PSR(N) = PHAR*EXP(BETA*TSH(N))
20 CONTINUE
      WRITE(6,111) (Z(N),TS(N),PS(N),TSH(N),PSR(N),N=N1V,N2V)
C      COMPUTE THE PRESSURE
30 EALPHT = EXP(-ALPHAS*TIME)
      FHETAT = EXP(-BETA*TIME)
      DO 30 N=N1V,N2V
      IF(TIME.LT. TS(N))GOTO 40
      IF(TIME.GT. TSH(N))GOTO 31
      P(N) = -EALPHT*PS(N)
      GOTO 30
31 P(N) = -FHETAT*PSR(N)
30 CONTINUE
40 RETURN
100 FORMAT(2F10.3,6A10)
101 FORMAT(3E10.3)
102 FORMAT(4E10.3)
103 FORMAT(// 3X,PRESS SUBROUTINE FOR 1-D STRUCTURE,3/10/76,
1 // FROM TABLES OF ARRIVAL TIME AND PEAK PRESSURE VS POSITION,
2 // POSITION = 0 CORRESPONDS TO N=2 OF 1-D REPSIL,
3 // THE QUASI-STATIC PRESSURE IS 1E15.7,5X, WITH EXPONENTIAL,
4 // DAMPING COEFFICIENT = 1E15.7/10X,6A10/
5 // INPUT TIME, TSI, REDUCED BY THIF= 1E15.7/
6 // CONVERSION COEFFICIENTS FOR Z, T, AND P ARE 1E15.7)
104 FORMAT(2E10.3,110)
105 FORMAT(//20X, 'INPUT DATA' //7X, 'ZSI',12X, 'TSI',12X, 'PSI'/(3E15.7))
106 FORMAT(//5X, 'TABULAR DATA AFTER CONVERSION'
1 7X, 'ZSI',12X, 'TSI',12X, 'PSI'/(1PE3E15.7))
107 FORMAT(//5X, 'FINAL DATA' //7X, 'Z',14X, 'TS',13X, 'PS',13X
1 'TSH',12X, 'PSR'/(1PE15.7))
      END

```

12/2/75

12/2/75

12/2/75

12/2/75

```

*P INGEOM.7.INGEOM.47
*CALL MAIN
C
C INGEOM FOR ASYMMETRIC CYLINDER. 12/ 2/75.
C
C SET PROGRAM TO USE GAUSSIAN INTEGRATION
  IGAUSS = 1
  WFAI(5.100)CYLL*RADIUS
  WRTF(6.101)CYLL*RADIUS
  DETA1 = 2.0*3.141592653589793*RADIUS
C   DETA1 = 2*PI*RADIUS GIVES ENERGY FOR ENTIRE CYLINDER
  DETA2 = CYLL/FLOAT(NMESH)
  IB = 0
C   ***** EVALUATE R(N) AND Z(N), N=N1R,N2R *****
  DO 10 N=N2R,N1R
    R(N)=RADIUS
    Z(N)=FLOAT(N-2)*DETA2
  10 CONTINUE
  RETURN
100 FORMAT(2E12.4)
101 FORMAT(//' INGEOM FOR CYLINDER 12/3/75. LENGTH =*1PF15.6,5X,
1   * RADIUS =*.F15.6/)
  END

```

```

*AF
*DECK ENDFRC
SUBROUTINE ENDFRC
*CALL MAIN
C
C THE APL FRC CHANGE TO (1-0) WPSIL FOR BEAMS REQUIRES A USER
C SUBROUTINE ENDFRC TO SUPPLY FORCES AND MOMENTS AT BOTH ENDS.
C FOR AN INITIALLY STRAIGHT BEAM, R(N)=0.0, Z(N)=(N-2)DETA2.
C FF71>0 DECREASES DZ(2), EFR1>0 DECREASES DR(2)
C FF72>0 INCREASES DZ(N2H), EFR2>0 INCREASES DR(N2R)
C FM1>0 DECREASES DR(2) AND INCREASES DH(3)
C FM2>0 INCREASES DR(N2H-1) AND DECREASES DR(N2H)
C
C THIS VERSION IS FOR A BEAM WITH ONE, TWO, OR NO FREE ENDS.
DATA IFFER/0/
IF(IFFER.GT.0)RETURN
IFFER = 1
FF01 = 0.0
FF71 = 0.0
FM1 = 0.0
FF02 = 0.0
FF72 = 0.0
FM2 = 0.0
RETURN
END

```

```

*10 STREN
*1 RPELID.5
C NEW STRAIN ENERGY COMPUTATION FOR A BEAM. 1/20/76 STREN
*1 RMSTRS.7
C NEW STRAIN ENERGY COMPUTATION 1/20/76 STREN
DIMENSION SST(5),DEFT(5),LST(5) STREN
*1 RMSTRS.33
C NEW STRAIN ENERGY COMPUTATION 1/20/76 STREN
IF(IP .EQ. 2)GOTO #03 STREN
SST(J) = SIG22 STREN
LST(J) = J STREN
*0 RMSTRS.36
C NEW STRAIN ENERGY COMPUTATION 1/20/76 STREN
*1 RMSTRS.39
C NEW STRAIN ENERGY COMPUTATION 1/20/76 STREN
IF(IP .EQ. 2)GOTO 35 STREN
IF(NSFL .GT. 1)GOTO 10 STREN
DSTREN = (SS22)**2 STREN
GOTO 30 STREN
10 IF(SS22 .LE. 0.0)GOTO 12 STREN
SS22 = -SS22 STREN
DO 11 L=1,NSFL STREN
11 SST(L) = -SST(L) STREN
12 DO 13 L=1,NSFL STREN
13 DEFT(L) = (SIGZZ(L) - SST(L))/F STREN
LLAST=NSFL-1 STREN
DO 15 L=1,LLAST STREN
JSTART = L+1 STREN
DO 14 J=JSTART,NSFL STREN
IF(DEFT(L) .LT. DEFT(J))GOTO 14 STREN
DSTEMP = DEFT(L) STREN
DEFT(L) = DEFT(J) STREN
DEFT(J) = DSTEMP STREN
LSTEMP = LST(L) STREN
LST(L) = LST(J) STREN
LST(J) = LSTEMP STREN
14 CONTINUE STREN
15 CONTINUE STREN
FM = 1.0 STREN
FSTREN = 0.0 STREN
DO 25 L=1,NSFL STREN
FSTAR = FM*E STREN
DEPSTR = -SS22/FSTAR STREN
DSTREN = DSTREN + (SS22)**2/FM STREN
IF(DEPSTR .LE. DEFT(L)) GOTO 30 STREN
CORNER AT DEFT(L) IN UNLOADING CURVE STREN
SS22 = -FSTAR*(DEPSTR-DEFT(L)) STREN
DSTREN = DSTREN - (SS22)**2/FM STREN
L1 = LST(L) STREN
EM = FM - 4T(L1) STREN
DEFTL = DEFT(L) STREN
DO 20 J=L,NSFL STREN
DEFT(J) = DEFT(J) - DEFTL STREN
20 CONTINUE STREN
25 CONTINUE STREN
30 STREN = STREN + SPG*(K)*DSTREN STREN
35 CONTINUE STREN

```

LIST OF SYMBOLS

[FORTRAN name in brackets]

a	determinant of surface metric [DA]
E	Young's modulus of elasticity [E]
h	shell thickness [THICKN]
$\underline{i}_1, \underline{i}_2, \underline{i}_3$	orthonormal vector basis
$\hat{\underline{M}}^{*\alpha\beta}$	Normal vectors of stress moment resultant (See Reference 1). $\hat{\underline{M}}^{*\alpha\beta} = \hat{M}^{*\alpha\beta} \underline{n} = a^{\frac{1}{2}} \hat{M}^{\alpha\beta} \underline{n} = a^{\frac{1}{2}} (M^{\alpha\beta} + M^{\beta\alpha}) \underline{n}$.
$\hat{\underline{N}}^{*\alpha\beta}$	Modified stress resultant tensor (See Reference 1).
n or N	index associating variables with the N 'th mesh point (e.g. $R(N)$) [N]
$\underline{n}$	unit normal vector to reference surface, $\underline{n} = n_k \underline{i}_2 + n_r \underline{i}_3$
$\underline{r}$	position vector to a point on the reference surface
R, Z	position components. $\underline{r} = R \underline{i}_3 + Z \underline{i}_2$ [$R(N), Z(N)$]
t	time [TIME]
W_k	area or weight associated with ζ_k [$W(K)$]
Δt	time increment [DELTAT]
$\Delta \xi, \Delta \xi^2$	spacing of ξ^2 between mesh points [DETA2]
ξ	ξ is ξ^2
ξ^1, ξ^2	material (Lagrangian) coordinates (called η^1, η^2 in Reference 2)
ζ	normal distance from reference surface
ζ_k	ζ at k 'th integration station [ZETA(K)]
$\zeta_\ell, (\zeta_u)$	ζ on lower (upper) surface of shell [ZL, (ZU)]
ν	Poisson's ratio [FNU]
ρ	mass density [RHO]

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314	1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35898
3	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS (2 cys) DRDAR-TDC Dover, NJ 07801	1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35898
1	Commander US Army Armament Materiel and Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299	1	Commander US Army Tank Automotive Rsch and Development Command ATTN: DRDTA-UL Warren, MI 48090
1	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Watervliet, NY 12189	1	Director US Army TRACOD Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range NM 88002
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Blvd. St. Louis, MO 63120	1	AFELM, The Rand Corporation ATTN: Library-D 1700 Main Street Santa Monica, CA 90406
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035	1	Massachusetts Institute of Technology Aeroelastic & Structures Rsch Lab 77 Massachusetts Ave ATTN: Dr. E. A. Witmer Cambridge, MA 02139
1	Commander US Army Communications Rsch and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703		<u>Aberdeen Proving Ground</u> Dir, USAMSAA ATTN: DRXSY-D DRXSY-MP, H. Cohen Cdr, USATECOM ATTN: DRSTE-TO-F Dir, USACSL, Bldg. E3516, EA ATTN: DRDAR-CLB-PA

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number _____

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) _____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) _____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: _____

Telephone Number: _____

Organization Address: _____

END

FILMED