SC-RR-66-601

W O N D Y

A Computer Program for Calculating Problems of Motion in One Dimension

W. Herrmann
and
P. Holzhauser
Division 1116

with Appendix A

Stability of the Difference Equation

R. J. Thompson, 5262
Sandia Laboratory, Albuquerque

February 1967

ABSTRACT

Operational features of a FORTRAN computer code which solves the finite difference analogs to the Lagrangian equations of motion in one-dimensional rectangular, cylindrical, and spherical coordinates are described. Separate axial and transverse stress and strain components are carried in the code so that problems of elastic-plastic motion can be treated. FORTRAN listings, input instructions, and a number of check solutions are included.

DISCLAIMER

ACKNOWLEDGMENT

The authors are indebted for criticisms, corrections, and suggestions from staff at the Massachusetts Institute of Technology and at the Sandia Corporation too numerous to mention individually. The stability analysis is the work of Robert J. Thompson. Specially valuable contributions were made by B. J. Thorne, L. D. Bertholf, D. B. Hayes, L. H. Bakken, and F. J. Cupps.

Issued by Sandia Corporation,
a prime contractor to the
United States Atomic Energy Commission

ERRATA

```
      SUBROUTINE MOTION                                                  MOTIO  1
C PROGRAM WONDY                                                          MOTIO  2
      BANK, (0), MOTION    , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       MOTIO  3
C     INSERT COMMON CARDS HERE
      A=2.0*(SIGMA(J)-SIGMA(J+NVAR)+Q(J)-Q(J+NVAR))/(RHO(J)*(X(J)-XP)    MOTIO 27
     1+RHO(J+NVAR)*(X(J+NVAR)-X(J)))+2.0*(LPHA-1.0)*(PHI(J)+PHI(J+NVAR))MOTIO 28
     2/(RHO(J)*(X(J)+XP)+RHO(J+NVAR)*(X(J+NVAR)+X(J)))                   MOTIO 29
      U1=U(J)+0.5*(DELT(1)+DELT(2))*A                                    MOTIO 30
      X1=X(J)+U1*DELT(1)                                                 MOTIO 31
      IF (QFRACT(L)) GO TO 4011                                          MOTIO 32
      ENTRY MESHES                                                       MOTIO 33
      SIGMAP=SIGMA(J)                                                    MOTIO 34
      DELXJ=X1-X(J-NVAR)                                                 MOTIO 35
      IF (LPHA-2) 4018, 4019, 4020                                       MOTIO 36
 4018 CXIP=CXI=1.0                                                       MOTIO 37
      GO TO 4021                                                         MOTIO 38
 4019 CXIP=X(J-NVAR)+XP                                                  MOTIO 39
      CXI=X1+X(J)                                                        MOTIO 40
      GO TO 4021                                                         MOTIO 41
 4020 CXIP=X(J-NVAR)**2+X(J-NVAR)*XP+XP**2                               MOTIO 42
      CXI=X1**2+X1*X(J)+X(J)**2                                          MOTIO 43
 4021 CONTINUE                                                           MOTIO 44
      RHO1=1.0/(1.0/RHO(J)+DELT(1)/M(J)*(CXI*U1-CXIP*U(J-NVAR)))         MOTIO 45
C                                                                        MOTIO 46
C     CHECK FOR ROUNDOFF IN DENSITY                                      MOTIO 47
      IF (ABSF(RHO1-RHO(J)).LT.5.0E-10*RHO(J)) 4013, 4014                MOTIO 48
 4013 DELRJ=RHODOT=DELRHO=Q1=0.0                                         MOTIO 49
      GO TO 4003                                                         MOTIO 50
```

FOREWORD

The program described in this report has evolved over a period of many years.

Readers should be cautioned that some errors may remain in the program. It is customary for users to alter the program to fit their own requirements and tastes. Such users are urged to append a suitable designation to the program name, so that different versions may be distinguished, and to add distinctive identifying symbols to the last eight columns of the FORTRAN program cards which have been altered. Needless to say, the authors cannot take responsibility for any versions of the program which do not correspond exactly to the program listing in this report.

TABLE OF CONTENTS

June 12, 1967

TO: ALL WONDY AND TOODY USERS —SC-RR-66-601

Jany Bertholf

From: L. D. Bertholf - 1142

Re: Equation of State Stability

The Grueneisen Equation of State and the straight line U_s-U_p fit relation combination is subject to instabilities at high compressions.

For a constant Grueneisen Ratio this instability is possible for

$$\frac{\rho}{\rho_o} > \frac{2 + \Gamma_o - S}{1 + \Gamma_o - S}$$

or

$$\frac{\rho}{\rho_o} > \frac{1 + S}{S} \qquad \text{if} \qquad \Gamma_o = 2S - 1 \ .$$

However for $\Gamma = \frac{\Gamma_o \rho_o}{\rho}$, the equation state combination is stable for $\Gamma_o < S+1$ as long as the compression is less than the asymptotic value for the straight line fit.

Since Γ_o is usually less than S + 1, it is recommended that WONDY and TOODY runs be made with $\Gamma = (\Gamma_o \rho_o)/\rho$. This is easily accomplished by using NOH = 2.0 with h_1 = -1.0.

LDB:1142:sm

Copy to:
L. D. Bertholf, 1142
File 1142

1. INTRODUCTION

WONDY is a versatile FORTRAN code for computing wave propagation in one dimension in rectangular, cylindrical or spherical coordinates. The code is based on conventional finite difference analogs to the Lagrangian equations of motion and is similar in many respects to other such codes.*

Considerable effort has been expended to produce a very flexible code. Routines for equations of state or constitutive relations, special boundary routines, radiation energy addition, as well as the initializing routine are written as self-contained subroutines, and new routines are easily written to cover problems not handled by the original set. In this way most problems of motion in one dimension may be handled without difficulty.

Since many users may be unacquainted with previous work in the field, a very elementary treatment of the differencing technique is given. However, no attempt is made at a rigorous derivation or treatment.

There are several versions of WONDY. The code described in this report is designated WONDY 2. It is written in C.D.C. 3600 FORTRAN

---

*In particular that of Herrmann (1964)[4] and of Wilkins and Giroux (1963).[3]

and in its present configuration occupies two 32K banks on the C.D.C. 3600.* A maximum of 3,000 meshes and 20 different layers of material can be accomodated. While care has been taken to minimize computing time,+ the program has been written for flexibility rather than extreme efficiency, and some compromises on both program and storage size and on running time have been made to retain this flexibility. In this version no attempt has been made to minimize bank transfers on the C.D.C. 3600 since attempts to do so would make the program less adaptable to other computers. For special classes of problems for which much repetitive production running is required, it would be advisable to modify and specialize the code to minimize running times and/or optimize storage requirements for the particular computer on which the code is to be used.

The finite difference technique is an approximate method of solving the non-linear partial differential equations describing one-dimensional motion. The degree of approximation depends on the mesh sizes and artificial viscosity coefficients which are used.

---

*The main storage array is forced into Bank 1 and occupies approximately 31,100 decimal locations. The remainder of the storage arrays and the program including library subroutines, but excluding plot routines, occupy approximately 18,000 decimal locations in Bank 0. If the program and storage are to be accomodated in one 32K memory bank, the storage array must be reduced to approximately 8,000 decimal locations which allows a maximum of 800 meshes to be used.

+Running times depend on the complexity of the equation of state and the amount of output requested. For most problems, about 0.5 to 0.8 million mesh calculations (number of meshes times the number of cycles) can be done per hour on the C.D.C. 3600.

It is quite possible to obtain completely false or even random results by inappropriate choice of mesh sizes and viscosity coefficients. It is always advisable to run several problems with successively smaller mesh sizes and often with several choices of viscosity coefficients to insure that the solution is insensitive to choice of these parameters. A few problems cannot be handled adequately by finite difference techniques on present computers due to the fact that sufficiently small mesh sizes would entail prohibitively long computation times.

Results are, of course directly dependent on the constitutive equations or equations of state describing the materials involved. If physically realistic results are to be obtained, then physically realistic constitutive equations and material constants must be used. For some materials these are known less precisely than for others, and results will therefore have a greater uncertainty. The question of the sensitivity of the results to variation in any particular material constant connot be answered in any generality. If material constants have considerable uncertainty attached to them, it is always advisable to run at least three problems: one with the most probable value, one with the maximum, and one with the minimum probable value.

When the constitutive equations and material constants are known precisely and when the proper mesh sizes and viscosity coefficients are chosen, the finite difference technique is capable of great accuracy.

2. BASIC DIFFERENCE EQUATIONS

2.1 Differential Equations

The one-dimensional equation of motion expressing conservation of momentum is

$$\rho a = -\frac{\partial \sigma}{\partial x} - \frac{\partial q}{\partial x} + (\alpha - 1)\frac{\varphi}{x} \qquad (2.1)$$

where x is the spatial coordinate, ρ the density, a the acceleration, σ the stress in the x direction, and q the viscous stress, both taken positive in compression. The quantity φ is the difference between the stresses in the longitudinal and transverse directions (α = 1 for rectangular, 2 for cylindrical, 3 for spherical one-dimensional coordinates).

We will follow material particles in the motion, and thus the acceleration is given simply by

$$a = \frac{\partial u}{\partial t} \qquad (2.2)$$

where u is the velocity defined by

$$u \equiv \frac{\partial x}{\partial t} \qquad (2.3)$$

Mass conservation is expressed by

$$\frac{\rho}{\rho_o} = \frac{dV}{dv} \qquad (2.4)$$

where dV is an element of volume at time t = 0, when the density is ρ_o and dv is the current volume of the same element at time t.

These equations are supplemented by the equation expressing energy conservation and the equation of state or constitutive equation. It is convenient to defer consideration of these equations as they are solved separately in the equation of state subroutine.

2.2 Difference Equations

In the finite difference method all quantities are sampled at discrete material particles and at discrete times. The particles are labeled in order with an index j, and times are labeled in order with an index n. Thus the value of an arbitrary quantity Ψ at the j^{th} particle and n^{th} time is denoted Ψ_j^n. The differential equations are set into finite difference form by consistent use of simple, centered, second-order analogs

$$\left(\frac{\partial \Psi}{\partial x}\right)_{j+1/2}^{n} = \frac{\Psi_{j+1}^{n} - \Psi_j^{n}}{x_{j+1}^{n} - x_j^{n}}$$

$$\left(\frac{\partial \Psi}{\partial t}\right)_{j}^{n+1/2} = \frac{\Psi_j^{n+1} - \Psi_j^{n}}{t_j^{n+1} - t_j^{n}} \qquad (2.5)$$

and linear interpolation expressions.

$$\Psi_{j+1/2}^{n} = \frac{1}{2}\left(\Psi_{j+1}^{n} + \Psi_j^{n}\right)$$

$$\Psi_j^{n+1/2} = \frac{1}{2}\left(\Psi_j^{n+1} + \Psi_j^{n}\right) \qquad (2.6)$$

The difference equations used, corresponding to (2.1) through (2.4) are

$$a_j^n = 2\left\{\frac{\left(\sigma_{j-1/2}^n + q_{j-1/2}^n\right) - \left(\sigma_{j+1/2}^n + q_{j+1/2}^n\right)}{\rho_{j+1/2}^n\left(x_{j+1}^n - x_j^n\right) + \rho_{j-1/2}^n\left(x_j^n - x_{j-1}^n\right)}\right\}$$

$$+ 2(\alpha - 1)\left\{\frac{\left(p_{j+1/2}^n + p_{j-1/2}^n\right)}{\rho_{j+1/2}^n\left(x_{j+1}^n + x_j^n\right) + \rho_{j-1/2}^n\left(x_j^n + x_{j-1}^n\right)}\right\} \qquad (2.7)$$

$$u_j^{n+1/2} = u_j^{n-1/2} + \tfrac{1}{2}\left(\Delta t^{n+1/2} + \Delta t^{n-1/2}\right) a_j^n \qquad (2.8)$$

where $\Delta t^{n+1/2} = t^{n+1} - t^n$

$$x_j^{n+1} = x_j^n + \Delta t^{n+1/2}\, u_j^{n+1/2} \qquad (2.9)$$

$$\rho_{j-1/2}^{n+1} = \frac{m_{j-1/2}}{\left(x_j^{n+1}\right)^\alpha - \left(x_{j-1}^{n+1}\right)^\alpha} \qquad (2.10)$$

where m is a mesh constant initialized at t = 0 to

$$m_{j-1/2} = \rho_{j-1/2}^o\left\{\left(x_j^o\right)^\alpha - \left(x_{j-1}^o\right)^\alpha\right\} \qquad (2.11)$$

These equations are subject to excessive roundoff when $\alpha = 2$ and 3. The mesh constant m is written in the equivalent form

$$m_{j-1/2} = \rho_{j-1/2}^o\left(x_j^o - x_{j-1}^o\right)\xi_{j-1/2}^o$$

where

$$\varsigma^{0}_{j-1/2} = 1 \qquad \text{for } \alpha = 1$$

$$= x^{0}_{j} + x^{0}_{j-1} \qquad \text{for } \alpha = 2$$

$$= (x^{0}_{j})^{2} + x^{0}_{j}\, x^{0}_{j-1} + (x^{0}_{j-1})^{2} \qquad \text{for } \alpha = 3$$

The mass equation may be written in the alternate forms

$$m_{j-1/2} = \rho^{n+1}_{j-1/2} \left| (x^{n+1}_{j})^{\alpha} - (x^{n+1}_{j-1})^{\alpha} \right| = \rho^{n}_{j-1/2} \left| (x^{n}_{j})^{\alpha} - (x^{n}_{j-1})^{\alpha} \right|$$

Rearranging and subtracting leads to

$$\frac{1}{\rho^{n+1}_{j-1/2}} = \frac{1}{\rho^{n}_{j-1/2}} + \frac{1}{m_{j-1/2}} \left| (x^{n+1}_{j})^{\alpha} - (x^{n+1}_{j-1})^{\alpha} - (x^{n}_{j})^{\alpha} + (x^{n}_{j-1})^{\alpha} \right|$$

This may be written as

$$\rho^{n+1}_{j-1/2} = \left\{ \frac{1}{\rho^{n}_{j-1/2}} + \frac{\Delta t^{n+1/2}}{m_{j-1/2}} \left[\varsigma^{n+1/2}_{j}\, u^{n+1/2}_{j} - \varsigma^{n+1/2}_{j-1}\, u^{n+1/2}_{j-1} \right] \right\}^{-1}$$

where

$$\varsigma^{n+1/2}_{j} = 1 \qquad \text{for } \alpha = 1$$

$$= x^{n+1}_{j} + x^{n}_{j} \qquad \text{for } \alpha = 2$$

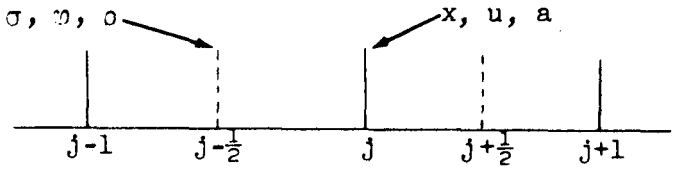
$$= (x^{n+1}_{j})^{2} + x^{n+1}_{j}\, x^{n}_{j} + (x^{n}_{j})^{2} \qquad \text{for } \alpha = 3$$

Two quantities are useful in later calculations.

$$\left(\frac{\Delta \rho}{2\rho^{2}} \right) \equiv \frac{2 \left(\rho^{n+1}_{j-1/2} - \rho^{n}_{j-1/2} \right)}{\left(\rho^{n+1}_{j-1/2} + \rho^{n}_{j-1/2} \right)^{2}} \qquad (2.12)$$

$$\left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right) \equiv \frac{2 \left(\rho^{n+1}_{j-1/2} - \rho^{n}_{j-1/2} \right)}{\Delta t^{n+1/2} \left(\rho^{n+1}_{j-1/2} + \rho^{n}_{j-1/2} \right)} \qquad (2.13)$$

Note that a, u, and x are centered at j, while all other quantities are centered at $j - \frac{1}{2}$ in space. This suggests the following interpretation:



If lines are drawn on the material at the initial instant to define a material coordinate mesh, which distorts with the material as the motion proceeds, the positions, velocities, and accelerations of these lines defining the mesh boundaries are found at discrete times. The same material particles are always contained in a given mesh. Stresses, densities, etc., are found which may be regarded as averages over each mesh between successive mesh boundaries.

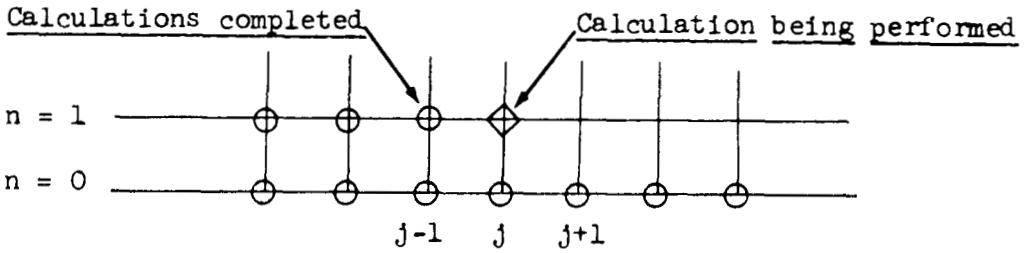
All quantities except velocity are centered at n in time, while u is centered at $n + \frac{1}{2}$. This occasions no difficulty except at the initial instant. Velocities are usually either zero or constant prior to the initial instant, so that $u_j^{-\frac{1}{2}} = u_j^o$, and starting the computation is not a problem.

In order to facilitate storage, mesh quantities, e.g., $\sigma_{j-\frac{1}{2}}$, $q_{j-\frac{1}{2}}$, $\rho_{j-\frac{1}{2}}$, etc., are indexed j. The velocity $u^{n+\frac{1}{2}}$ is stored at n + 1.

2.3 Order of Computation

The calculation proceeds as follows: at t = 0, all quantities are defined at all meshes by the initial data (via the initializing routine described later). The computation is performed successively at each mesh

starting with the left-hand boundary. At the j^{th} mesh the momentum equation (2.7) is used to compute the acceleration at the j^{th} mesh boundary. Velocity at a time $\frac{1}{2}\Delta t^{n+\frac{1}{2}}$ after the initial instant at the j^{th} mesh boundary follows from (2.8). Position at time $\Delta t^{n+\frac{1}{2}}$ after the initial instant at the j^{th} mesh boundary follows from (2.9).



The new position of the (j - 1)st mesh has already been found at this stage of the calculation. The mass equation (2.10) can therefore be used to determine the density in the mesh between j - 1 and j. For elastic-plastic materials the velocities at j - 1 and j can be used to determine strain rates at $j - \frac{1}{2}$. The energy equation and equation of state are then used to determine the energy and stresses at $j - \frac{1}{2}$. These calculations are accomplished in the equation of state subroutine and are discussed later.

The computation at the j^{th} mesh point is now complete, and the next mesh in sequence can be treated in the same way. Boundary calculations are described below. When all the mesh points have been treated, the solution for time $\Delta t^{n+\frac{1}{2}}$ after the initial instant has been constructed. The procedure can be repeated for the next time increment. Further repetition allows construction of the solution for the entire time of interest.

2.4 Units

No dimensional constants are coded into the program. Thus any self-consistent set of absolute units may be used. The user must be cautioned that nowhere does the acceleration due to gravity appear and absolute mass and force units must be used. Several sets of units which have been found useful are shown in the table below.

Quantity	c.g.s.	c.g.μs	S. I.	f.p.s.	i.p.s.
Time	sec	μsec	sec	sec	sec
Length	cm	cm	m	ft	ins
Mass	gm	gm	kg	slug	slug
Force	dyn	T dyn	Newton	lb	lb
Energy	erg	T erg	Joule	ft lb	ins lb
Energy Density	erg/gm	Mbar cm^3/gm	J/kg	ft lb/slug	ins lbs/slug
Power	erg/sec	T erg/sec	Watt	ft lb/sec	ins lb/sec
Density	gm/cm^3	gm/cm^3	kg/m^3	$slug/ft^3$	$slug/ins^3$
Pressure	dyn/cm^2	M bar	N/m^2	lb/ft^2	lb/ins^2

3. ARTIFICIAL VISCOSITY

Since materials are non-linear in the sense that they become stiffer as they are compressed, solutions to wave propagation problems which do not include viscosity may develop discontinuities or shock waves. Such discontinuities would have to be handled as internal floating boundaries since the difference analogs (2.5) are only approximately correct for small differences in all parameters. These internal boundaries are part of the solution, and it is extremely difficult to handle them.

The problem is resolved* by including viscosity, which renders the solution continuous and prevents formation of mathematical discontinuities. Shock waves are recognized as very steep but finite gradients in the solution. It is clear that a shock wave must occupy several mesh widths in order to satisfy the requirement that differences in quantities remain small.

Natural viscosity can be used. However, for most materials natural viscosity is so small that shocks would be extremely narrow. In order to insure that a shock occupies several meshes, it would then be necessary to use extremely small meshes. For the usual physical problems this would mean that an extremely large number of meshes would be required.

For this reason an artificially large viscosity is introduced. Care is necessary so that the viscous term does not affect the solution anywhere

---

*See Von Neumann and Richtmyer (1950)[1].

except near shocks. At shocks the solution is intentionally distorted to insure that gradients are much lower than in nature, so that a reasonable number of meshes can be used in a given problem. In effect, use of artificial viscosity broadens or smears shock waves.

The exact choice of form of artificial viscosity is somewhat arbitrary. We use a quadratic viscosity* in the form

$$q = \rho\, b_1^2 \left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right)^2 \qquad (3.1)$$

where b_1 is a constant with dimensions of length. Since $\left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right)$ essentially represents the volumetric strain rate, q is essentially a bulk viscosity.

The quadratic form is chosen so that the viscosity is very small except when rates become large, at which time the viscosity becomes very large. The quadratic form is therefore most effective in controlling gradients at shocks while introducing minimal disturbances elsewhere.

A linear viscosity is also used‡ in the form

$$q = b_2\, c \left(\frac{\partial \rho}{\partial t} \right) \qquad (3.2)$$

where c is the sound speed and b_2 is a constant with dimensions of length. The linear viscosity is effective in controlling small spurious oscillations in which gradients are insufficient to make the quadratic viscosity effective.

---

*Introduced by Von Neumann and Richtmyer (1950).[1]

‡Introduced by Landshoff (1955).[2]

Great care is necessary in the use of linear viscosity, as there is a much greater chance of distorting the solution in areas away from shocks.

The constants b_1 and b_2 determine the shock width.* Since it is desirable that the shock encompass a given number of meshes, independent of the choice of mesh size, b_1 and b_2 are non-dimensionalized by use of the mesh size.

$$b_1 = B_1 \Delta x \qquad b_2 = B_2 \Delta x \tag{3.3}$$

In finite difference form

$$q^{n+1}_{j-1/2} = \rho^{n+1}_{j-1/2} \left\{ B_2 \left(x^{n+1}_j - x^{n+1}_{j-1} \right) c^{n}_{j-1/2} \left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right) + B_1^2 \left(x^{n+1}_j - x^{n+1}_{j-1} \right)^2 \left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right) \left| \left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right) \right| \right\} \tag{3.4}$$

where $\left(\frac{1}{\rho} \frac{\partial \rho}{\partial t} \right)$ is given by (2.13).

Since rarefactions do not steepen into shocks, viscosity is set to zero when $\frac{\partial \rho}{\partial t} < 0$.

---

*See von Neumann and Richtmyer (1950)[1] for a discussion of the relation of the shock width to b_1 in the case of a perfect gas.

4. CONSTITUTIVE EQUATIONS

There are several options for constitutive equations in the program. Currently, six different constitutive equations can be accommodated, but it is very simple to increase this number. Individual constitutive equations are programmed in subroutines, an input parameter for each layer determining which subroutine is to be called for this layer. Only some of the available subroutines will be described in this report. Other special-purpose subroutines can be written as required.

The equation expressing conservation of energy is included in the constitutive subroutine. Before writing down the energy equation, it is useful to note a few results concerning stress and strain.

4.1 Stress and Strain

In one-dimensional motion shear strains are absent since there is no shearing of the material. It is more convenient to work in terms of the strain rate or stretching. In the direction of motion, i.e., the x direction, the stretching is defined as

$$d_x \equiv \frac{\partial u}{\partial x}$$

In rectangular coordinates there is no motion in the y and z coordinate direction so that

$$d_x = \frac{\partial u}{\partial x} \qquad d_y = 0 \qquad d_z = 0 \qquad \text{for } \alpha = 1 \tag{4.1}$$

In cylindrical coordinates there is no motion in the z direction, so that $d_z = 0$. However, motion in the x direction will induce a circumferential strain, so that (x is the radial direction)

$$d_x = \frac{\partial u}{\partial x} \qquad d_y = \frac{u}{x} \qquad d_z = 0 \qquad \text{for } \alpha = 2 \tag{4.2}$$

In spherical coordinates there is a hoop strain induced in mutually perpendicular circumferential directions when there is motion in the x direction, so that (x is the radial direction)

$$d_x = \frac{\partial u}{\partial x} \qquad d_y = \frac{u}{x} \qquad d_z = \frac{u}{x} \qquad \text{for } \alpha = 3 \tag{4.3}$$

The volumetric strain rate or dilatation is defined as

$$d = d_x + d_y + d_z \tag{4.4}$$

Thus it must be related to the rate at which the density is changing by

$$d = -\frac{1}{\rho}\frac{\partial \rho}{\partial t} \tag{4.5}$$

Stretching deviators are defined as

$$\overset{d}{d}_x = d_x - \frac{1}{3}d = d_x + \frac{1}{3\rho}\frac{\partial \rho}{\partial t} \tag{4.6}$$

and similarly for $\overset{d}{d}_y$, $\overset{d}{d}_z$. They are a measure of the rate of distortion independent of the volume change. From (4.4) it is evident that

$$\overset{d}{d}_x + \overset{d}{d}_y + \overset{d}{d}_z = 0 \tag{4.7}$$

Since the shear strains are zero in one-dimensional motion, shear stresses are zero. The stress components in the coordinate directions are

σ_x, σ_y, and σ_z. The pressure is defined as

$$(-p) = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z) \tag{4.8}$$

the minus sign appearing in agreement with the convention that stresses are considered positive in tension, while pressure is considered positive in compression. Stress deviators are defined as

$$\sigma_x{}^d = \sigma_x - (-p) = \sigma_x + p \tag{4.9}$$

and similarly for $\sigma_y{}^d$, $\sigma_z{}^d$. From (4.8) it is evident that

$$\sigma_x{}^d + \sigma_y{}^d + \sigma_z{}^d = 0 \tag{4.10}$$

The rate at which mechanical work is being done by the stresses, i.e., the stress power, is given by

$$P = \sigma_x d_x + \sigma_y d_y + \sigma_z d_z \tag{4.11}$$

Using (4.4), (4.5), (4.6), (4.8), and (4.9) the stress power may be expressed as

$$P = P_s + P_d \tag{4.12}$$

where P_s is given by

$$P_s = \frac{p}{\rho}\frac{\partial \rho}{\partial t} \tag{4.13}$$

and represents the rate at which work is being done by the pressure against a volume change, and P_d is given by

$$P_d = \sigma_x{}^d d_x^d + \sigma_y{}^d d_y^d + \sigma_z{}^d d_z^d \tag{4.14}$$

and represents the rate at which work is being done by the deviator stresses against distortion. Using (4.7) and (4.10) the components in the y direction can be eliminated.

$$P_d = 2\sigma_x{}^d d_x^d + \sigma_x{}^d d_z^d + \sigma_z{}^d d_x^d + 2\sigma_z{}^d d_z^d \tag{4.15}$$

In the momentum equation (2.7) we require

$$\varphi \equiv \sigma_x - \sigma_y \tag{4.16}$$

We note that, using definitions (4.9), φ can be written

$$\varphi = \sigma_x^d - \sigma_y^d \tag{4.17}$$

and using (4.10), this can be put into the more convenient form:

$$\varphi = 2\sigma_x^d + \sigma_z^d \tag{4.18}$$

Also the quantity σ in the momentum equation, from (4.9), is

$$\sigma = -\sigma_x = p - \sigma_x^d \tag{4.19}$$

where σ is taken positive in compression for convenience.

Considerable simplification arises when $\alpha = 1$ or 3. The symmetry inherent in rectangular and spherical one-dimensional motion implies that $\sigma_y = \sigma_z$. Thus (4.7) and (4.10) can be written

$$d_y^d = d_z^d = -\tfrac{1}{2} d_x^d \tag{4.20}$$

$$\sigma_y^d = \sigma_z^d = -\tfrac{1}{2} \sigma_x^d \tag{4.21}$$

Therefore, for $\alpha = 1$ or 3 (4.15) and (4.18) become

$$P_d = \tfrac{3}{2} \sigma_x^d \, d_x^d \tag{4.22}$$

$$\varphi = \tfrac{3}{2} \sigma_x^d \tag{4.23}$$

4.2 Conservation of Energy

The one-dimensional equation for conservation of energy expresses the fact that the rate of increase of internal energy per unit mass is equal to

the rate at which work is being done by the stresses and the rate at which heat is being added.

$$\rho \frac{\partial \mathcal{E}}{\partial t} = (p + q) \frac{1}{\rho} \frac{\partial \rho}{\partial t} + P_d + \left\{ \frac{\partial h}{\partial x} + (\alpha - 1)\frac{h}{x} \right\} + \rho Q \qquad (4.24)$$

where $\mathcal{E}$ is the internal energy per unit mass, Q is the heat added (say, by chemical reaction or radiation) per unit mass, and h is the heat flux due to heat conduction. We have included work done by the viscous stress q.

Heat addition Q may be assigned as required, while h will depend on the temperature gradient. Since the energy equation is included in the constitutive subroutine, Q and h are only included when necessary. They do not appear anywhere else in the program.

The energy equation using the difference analogs (2.5) becomes, in the absence of heat conduction

$$\mathcal{E}_{j-1/2}^{n+1} = \mathcal{E}_{j-1/2}^{n} + \left(p_{j-1/2}^{n+1} + p_{j-1/2}^{n} + q_{j-1/2}^{n+1} + q_{j-1/2}^{n} \right) \left(\frac{\Delta \rho}{2\rho^2} \right) + \Delta \mathcal{E}_{j-1/2}^{d} + \Delta Q_{j-1/2} \qquad (4.25)$$

Where $\Delta Q_{j-1/2}$ is the heat addition during the time increment $\Delta t^{n+1/2}$. $\Delta Q_{j-1/2}$ is ordinarily initialized to zero when there are no energy sources. For an example of how energy addition may be accommodated, see Section 6.7.

$\frac{\Delta \rho}{2\rho^2}$ has been defined in (2.12), and

$$\Delta \mathcal{E}^{d} = \frac{2\Delta t^{n+1/2} P_d}{\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^{n}} \qquad (4.26)$$

4.3 Fluid Equation of State

Liquids can support only a pressure, and the stress deviators vanish. The assumption is sometimes made that when very high pressures occur in solids,

the stress deviators are negligible, and the solid may be assumed to act like a liquid.

In this case it is evident that

$$\mathfrak{D} = 0 \qquad P_d = 0 \tag{4.27}$$

The equation of state of the material is usually taken in the form

$$p = f(\rho, \mathcal{E}) \tag{4.28}$$

This equation is centered at n+1 and $j-\frac{1}{2}$. It is therefore seen that the energy equation (4.25) and the equation of state (4.28) are two simultaneous equations for the two unknowns $p^{n+1}_{j-1/2}$, $\mathcal{E}^{n+1}_{j-1/2}$. If the equation of state has the form

$$p = f_1(\rho) + f_2(\rho)\,\mathcal{E} \tag{4.29}$$

then (4.25) and (4.29) can be solved explicitly

$$\mathcal{E}^{n+1}_{j-1/2} = \frac{\mathcal{E}^{n}_{j-1/2} + \left(f_{1\,j-1/2}^{n+1} + p^{n}_{j-1/2} + q^{n+1}_{j-1/2} + q^{n}_{j-1/2}\right)\left(\frac{\Delta\rho}{2\rho^2}\right) + \Delta\mathcal{E}^{d} + \Delta Q}{1 - f_{2\,j-1/2}^{n+1}\left(\frac{\Delta\rho}{2\rho^2}\right)} \tag{4.30}$$

$$p^{n+1}_{j-1/2} = f_{1\,j-1/2}^{n+1} + f_{2\,j-1/2}^{n+1}\,\mathcal{E}^{n+1}_{j-1/2} \tag{4.31}$$

where $\Delta\mathcal{E}^{d} = 0$ from (4.27) and (4.26). Then

$$\mathcal{E}^{n+1}_{j-1/2} = p^{n+1}_{j-1/2} \tag{4.32}$$

$$\mathfrak{D}^{n+1}_{j-1/2} = 0 \tag{4.33}$$

The functions f_1 and f_2 must be given. It is commonly assumed that a liquid or solid can be described by the Mie-Grueneisen equation, which can

be written

$$p - p_H = \Gamma\rho(\mathcal{E} - \mathcal{E}_H) \qquad (4.34)$$

where $p_H(\rho)$ and $\mathcal{E}_H(\rho)$ are the pressure and energy along some reference line and are functions of density only and where $\Gamma(\rho)$ is the Grueneisen ratio and is also a function of density only. The reference pressure $p_H(\rho)$ and energy $\mathcal{E}_H(\rho)$ are generally taken from experimental data along the Hugoniot. Two forms are common for p_H

$$p_H = \frac{\rho_o c_o^2 \eta}{(1 - s\eta)^2} \qquad (4.35)$$

where ρ_o is the initial density at zero pressure and ambient temperature, c and s are constants, and

$$\eta \equiv 1 - \frac{\rho_o}{\rho} \qquad (4.36)$$

This form follows from the observation that the shock velocity U is a linear function of particle velocity u for many materials, given by

$$U = c_o + s\,u \qquad (4.37)$$

where c_o and s are given constants. Alternately, p_H is given as a power series expansion in η

$$p_H = K_o \eta (1 + k_1\eta + k_2\eta^2 + k_3\eta^3 + \ldots.) \qquad (4.38)$$

where the K's are given constants. In order to match $\frac{dp_H}{d\eta}$ at $\eta = 0$, it is necessary to assume that

$$K_o = \rho_o c_o^2 \tag{4.39}$$

Note that c_o corresponds to the bulk sound speed and K_o to the adiabatic bulk modulus at zero pressure and room temperature.

The energy $\mathcal{E}_H$ is related to p_H by

$$\mathcal{E}_H = \frac{p_H \eta}{2\rho_o} \tag{4.40}$$

where $\mathcal{E} = 0$ at $p = 0$ at $\rho = \rho_o$.

The Grueneisen ratio is usually expressed as

$$\Gamma = \Gamma_o \left(1 + h_1\eta + h_2\eta^2 + \ldots . \right) \tag{4.41}$$

where the h's are given constants. Thus, rearranging (4.34) we have

$$p = p_H \left\{ 1 - \frac{\Gamma}{2}\left(\frac{\rho}{\rho_o} - 1\right)\right\} + \Gamma \rho \mathcal{E} \tag{4.42}$$

so that

$$f_1 = p_H \left(1 - \frac{\Gamma \mu}{2}\right) \tag{4.43}$$

$$f_2 = \Gamma \rho \tag{4.44}$$

where

$$\mu \equiv \frac{\rho}{\rho_o} - 1 \tag{4.45}$$

and p_H is given by (4.35) and (4.38) and Γ is given by (4.41).

The equation for p_H is selected by an equation of state constant NOK. If NOK = 0, then (4.35) is used, the first two succeeding equation of state constants being K_o and s. If NOK is a positive integer (≤ 6), then (4.38) is used. The first succeeding equation of state constant is K_o, which is computed internally using (4.39), and the next (NOK - 1) constants are the k's. Up to 5 k's can be used. If NOK = 1, then (4.38) becomes linear.

The Grueneisen ratio (4.41) is computed similarly. The number of terms used is selected by the equation of state constant NOH (≤ 6), the succeeding constant being Γ_o. The next (NOH - 1) constants are the h's. Note that if NOH = 1, then the Grueneisen ratio becomes a constant.

The sound speed is also computed in the constitutive subroutine. The sound speed is defined as

$$c \equiv \sqrt{\left(\frac{\partial p}{\partial \rho}\right)_s} \tag{4.46}$$

where $(\ )_s$ indicates that the differentiation is taken at constant entropy. Differentiating (4.29) at constant entropy

$$\left(\frac{\partial p}{\partial \rho}\right)_s = \frac{dj_1}{d\rho} + \mathcal{E}\frac{dj_2}{d\rho} + j_2\left(\frac{\partial \mathcal{E}}{\partial \rho}\right)_s \tag{4.47}$$

We note the thermodynamic relation

$$\left(\frac{\partial \mathcal{E}}{\partial \rho}\right)_s = \left(\frac{\partial \mathcal{E}}{\partial \upsilon}\right)_s \left(\frac{\partial \upsilon}{\partial \rho}\right)_s = \left(-p\right)\left(-\frac{1}{\rho^2}\right) = \frac{p}{\rho^2} \tag{4.48}$$

where $\upsilon = 1/\rho$ is the specific volume. Thus

$$c^2 = \frac{dj_1}{d\rho} + \mathcal{E}\frac{dj_2}{d\rho} + \frac{pj_2}{\rho^2} \tag{4.49}$$

Now, differentiating (4.43) and (4.44)

$$\frac{dj_1}{d\rho} = \frac{dp_H}{d\eta}\frac{d\eta}{d\rho}\left(1 - \frac{\Gamma\mu}{2}\right) - p_H\left(\frac{\Gamma}{2}\frac{d\mu}{d\rho} + \frac{\mu}{2}\frac{d\Gamma}{d\eta}\frac{d\eta}{d\rho}\right)$$

$$= \frac{\rho_0}{\rho^2}\left\{\frac{dp_H}{d\eta}\left(1 - \frac{\Gamma\mu}{2}\right) - \frac{p_H}{2}\left(\Gamma\left(\mu + 1\right)^2 + \frac{d\Gamma}{d\eta}\mu\right)\right\} \tag{4.50}$$

$$\frac{dj_2}{d\rho} = \Gamma + \frac{d\Gamma}{d\eta}(1 - \eta) \tag{4.51}$$

Where, differentiating (4.41)

$$\frac{d\Gamma}{d\eta} = \Gamma_0\left(h_1 + 2h_2\eta + 3h_3\eta^2 + \ldots\right) \tag{4.52}$$

and using (4.35)

$$\frac{dp_1}{d\eta} = \frac{\rho_0 c_0^2 (1 + s\eta)}{(1 - s\eta)^3} \tag{4.53}$$

or using (4.38)

$$\frac{dp_H}{d\eta} = K_o (1 + 2k_1\eta + 3k_2\eta^2 + \ldots.) \quad (4.54)$$

4.4 Elastic-Plastic Material

The constitutive equation is of general form

$$\sigma_x = f(d_x, d_y, d_z, \mathcal{E}) \quad (4.55)$$

with similar equations for σ_y and σ_z. However, (4.55) may be resolved into separate equations through the use of (4.6) and (4.9). Thus for the deviator stresses

$$\overset{d}{\sigma}_x = f(\overset{d}{d}_x, \rho, \mathcal{E}) \quad (4.56)$$

and similar equations for $\overset{d}{\sigma}_y$ and $\overset{d}{\sigma}_z$, and for the pressure

$$p = f(\rho, \mathcal{E}) \quad (4.57)$$

The latter equation is taken in identical form to (4.29). The deviator relations are specifically,

$$\frac{\partial \overset{d}{\sigma}_x}{\partial t} = 2G\,\overset{d}{d}_x \quad (4.58)$$

where $G(\rho, \mathcal{E})$ is the shear modulus and is taken as a function of the thermodynamic state. If the material exhibits plasticity, the deviator stresses have an upper limit determined by the yield condition. The Von Mises yield is

$$f_y = \left(\sigma_x^{\,d}\right)^2 + \left(\sigma_y^{\,d}\right)^2 + \left(\sigma_z^{\,d}\right)^2 \leq \frac{2}{3} Y^2 \quad (4.59)$$

where $Y(\rho,\mathcal{E})$ is a material constant known as the flow stress. It is more convenient to eliminate the y component by the use of (4.10).

$$J_y = 2\left\{\left(\sigma_x^d\right)^2 + \sigma_x^d \sigma_z^d + \left(\sigma_z^d\right)^2\right\} \le \frac{2}{3} Y^2 \tag{4.60}$$

When $\alpha = 1$ or 3, the symmetry condition (4.21) reduces (4.60) to

$$J_y = \frac{3}{2}\left(\sigma_x^d\right)^2 \le \frac{2}{3} Y^2 \tag{4.61}$$

Thus, putting these relations into finite difference form, d_x^d is given by (4.6) and (4.1) as

$$d_{x\ j-1/2}^{d\ n+1/2} = \frac{2\left(u_j^{n+1/2} - u_{j-1}^{n+1/2}\right)}{\left(x_j^{n+1} + x_j^{n}\right) - \left(x_{j-1}^{n+1} + x_{j-1}^{n}\right)} + \frac{1}{3}\left(\frac{1}{\rho}\frac{\partial \rho}{\partial t}\right) \tag{4.62}$$

where $\left(\frac{1}{\rho}\frac{\partial \rho}{\partial t}\right)$ is given by (2.13).

Then σ_x^d is given, if the material were entirely elastic, by

$$\sigma_x^d = \sigma_{x\ j-1/2}^{d\ n} + 2\Delta t^{n+1/2} G_{j-1/2}^{n+1/2} d_{x\ j-1/2}^{d\ n+1/2} \tag{4.63}$$

This value is limited by the yield condition. (See Ref. 4) For $\alpha = 1$ or 3

$$J_y = \frac{3}{2}\left(\sigma_x^d\right)^2 \tag{4.64}$$

Thus, if $J_y \le \frac{2}{3}\left(Y_{j-1/2}^{n+1}\right)^2$, then $\sigma_{x\ j-1/2}^{d\ n+1} = \sigma_x^d$

However, if $J_y > \frac{2}{3}\left(Y_{j-1/2}^{n+1}\right)^2$, then

$$\sigma_{x\ j-1/2}^{d\ n+1} = \left(\mathrm{Sgn}\ \sigma_x^d\right)\frac{2}{3} Y_{j-1/2}^{n+1} \tag{4.65}$$

The deviator stress work is, from (4.22) and (4.26)

$$\Delta\mathcal{E}^d = \frac{3}{2}\Delta t^{n+1/2} d_{x\ j-1/2}^{d\ n+1/2} \frac{\sigma_{x\ j-1/2}^{d\ n+1} + \sigma_{x\ j-1/2}^{d\ n}}{\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^{n}} \tag{4.66}$$

and

$$\omega_{j-1/2}^{n+1} = \frac{3}{2}\sigma_{x\ j-1/2}^{d\ n+1} \tag{4.67}$$

However, when $\alpha = 2$, it is necessary to use more complex equations. It is first necessary to compute d_z^d given by (4.2) and (4.6) as

$$d_{z\ j-1/2}^{d\ n+1/2} = \frac{1}{3}\left(\frac{1}{\rho}\frac{\partial \rho}{\partial t}\right) \tag{4.68}$$

where $\left(\frac{1}{\rho}\frac{\partial \rho}{\partial t}\right)$ is again given by (2.13).

Then σ_z^d is given, if the material were entirely elastic, by

$$\sigma_z^d = \sigma_{z\ j-1/2}^{d\ n} + 2\Delta t^{n+1/2} G_{j-1/2}^{n+1/2} d_{z\ j-1/2}^{d\ n+1/2} \tag{4.69}$$

The yield condition is therefore, from (4.60)

$$f_y = 2\left\{\left(\sigma_x^d\right)^2 + \sigma_x^d \sigma_z^d + \left(\sigma_z^d\right)^2\right\} \tag{4.70}$$

Then, if $f_y \le \frac{2}{3}\left(Y_{j-1/2}^{n+1}\right)^2$, $\sigma_{x\ j-1\ 2}^{d\ n+1} = \sigma_x^d$, $\sigma_{z\ j-1\ 2}^{d\ n+1} = \sigma_z^d$

However, if $f_y > \frac{2}{3}\left(Y_{j-1/2}^{n+1}\right)^2$, then

$$\left(\sigma_x^d\right)_{j-1/2}^{n+1} = \sqrt{\frac{\frac{2}{3}\left(Y_{j-1/2}^{n+1}\right)^2}{f_y}}\ \sigma_x^d \tag{4.71}$$

$$\left(\sigma_z^d\right)_{j-1/2}^{n+1} = \sqrt{\frac{\frac{2}{3}\left(Y_{j-1/2}^{n+1}\right)^2}{f_y}}\ \sigma_z^d \tag{4.72}$$

and the deviator stress work is, from (4.15) and (4.26)

$$\Delta \mathcal{E}^d = \frac{\Delta t^{n+1/2}}{\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^{n}}\left[\left(\sigma_{x\ j-1/2}^{d\ n+1} + \sigma_{x\ j-1/2}^{d\ n}\right)\left(2d_{x\ j-1/2}^{d\ n+1/2} + d_{z\ j-1/2}^{d\ n+1/2}\right)\right.$$

$$\left. + \left(\sigma_{z\ j-1/2}^{d\ n+1} + \sigma_{z\ j-1/2}^{d\ n}\right)\left(2d_{z\ j-1/2}^{d\ n+1/2} + d_{x\ j-1/2}^{d\ n+1/2}\right)\right] \tag{4.73}$$

and from (4.18)

$$\sigma_{j-1/2}^{n+1} = 2\sigma_{x\ j-1/2}^{d\ n+1} + \sigma_{z\ j-1/2}^{d\ n+1} \tag{4.74}$$

The energy $\mathcal{E}^{n+1}_{j-1/2}$ and pressure $p^{n+1}_{j-1/2}$ is then found from (4.30) and (4.31) as before. Then from (4.19)

$$\mathcal{T}^{n+1}_{j-1/2} = p^{n+1}_{j-1/2} - \mathcal{T}^{d\,n+1}_{x\,j-1/2} \tag{4.75}$$

The sound speed appropriate to the elastic-plastic case is

$$\left(c^{n+1}_{j-1/2}\right)^2 = \frac{3(1-\nu)}{(1+\nu)}\, c_b^2 \tag{4.76}$$

where c_b^2 on the right is given by (4.49), where ν is Poisson's Ratio, usually taken to be constant.

It now remains to specify the functions $G(\rho, \mathcal{E})$ and $Y(\rho, \mathcal{E})$ which appear above. The most common assumption is that the shear modulus G is related to the bulk modulus K by

$$G = \frac{3(1-2\nu)}{2(1+\nu)}\, K \tag{4.77}$$

where ν is Poisson's Ratio taken to be a constant. The bulk modulus is related to c_b^2 given by (4.49) by

$$K = \rho c_b^2 \tag{4.78}$$

Thus in finite difference form, in terms of (4.76) and (4.78)

$$G^{n+1/2}_{j-1/2} = \left\{\frac{(1-2\nu)}{2(1-\nu)}\right\} \rho^{n}_{j-1/2} \left(c^{n}_{j-1/2}\right)^2 \tag{4.79}$$

Note that (4.79) is not precisely centered.

An approximation which is used for G is to assume G is a function of ρ only, written as a power series in η

$$G = G_o (1 + g_1 \eta + g_2 \eta^2 + \ldots.) \qquad (4.80)$$

$$\eta = 1 - \frac{2\rho_o}{\rho^n + \rho^{n+1}}$$

where

$$G_o = \frac{3(1 - 2\nu)}{2(1 + \nu)} K_o \qquad (4.80a)$$

The form of G is selected by an equation of state constant NOG. If NOG = 0, the form (4.79) is used. If NOG is a positive integer (≤ 6), the form (4.80) is used, the first succeeding constant being G_o, which is computed internally using (4.80a), the next (NOG-1) constants being the g's. Up to 5 g's can be used. Note that if NOG = 1, G is a constant with value G_o.

One formulation for $Y(\rho, \varepsilon)$ is

$$Y_{j-1/2}^{n+1} = Y_o (1 + y_1 \eta) \left(1 - \frac{\varepsilon}{y_2}\right) \geq 0 \qquad (4.81)$$

where Y_o, y_1, and y_2 are given constants. The term y_2 may be taken to represent the melting or sublimation energy. An increase in the flow stress due to compression and a decrease in flow stress due to energy (or temperature) can thus be accommodated. Note that (4.81) does not represent strain hardening.

The form of equation to be used is selected by an equation of state constant NOY (≤ 6). If NOY = 0, then the elastic-plastic routine is bypassed and the material is a fluid. If NOY = 1, then a constant yield stress Y_o is used which appears as the next equation of state constant after NOY. If NOY = 2, then the yield test is omitted, and the material has an infinite yield strength. If NOY = 3, then (4.81) is used and y_1 and y_2 are the next equation of state constants after Y_o. (If other equations are added in place of (4.81), up to 6 constants can be supplied, and NOY can be used as an indicator to choose the appropriate equation.)

4.5 Vapor Equation of State

Under certain circumstances, especially when the material is heated by energy sources, the material may vaporize. Material strength disappears automatically if (4.81) is used. However, the HYDRO description is no longer applicable, and a vapor equation of state must be used. The vapor equation is only used for distended materials ($\eta < 0$). It is taken in the form

$$p = \rho \left| H + (\Gamma - H)\sqrt{\mu + 1} \right| \left| \mathcal{E} - \mathcal{E}_s \left[1 - \exp\left(N\eta(1 - \eta) \right) \right] \right| \qquad (4.82)$$

This form is chosen for the following reasons: When $\frac{\rho}{\rho_o} \ll 1$, i.e., for very distended materials, the equation essentially reduces to

$$p = H\rho(\mathcal{E} - \mathcal{E}_s) \qquad (4.83)$$

The material constant $\mathcal{E}_s$ represents the sublimation energy of the material. Equation (4.83) is therefore equivalent to the perfect gas law

$$p = (\gamma - 1)\,\rho\,(\mathcal{E} - \mathcal{E}_s) \qquad (4.84)$$

if $H = \gamma - 1$, where γ is the ratio of specific heat of the perfect gas, and the sublimation energy is subtracted from the internal energy.

When $\rho = \rho_o$, then the equation reduces to

$$p = \Gamma \rho \mathcal{E} \qquad (4.85)$$

and is therefore continuous with the Mie-Grueneisen equation (4.34) at this point.

Differentiating the vapor equation (4.82) and setting $\rho = \rho_o$ leads to

$$\frac{\partial p}{\partial \rho} = \Gamma_o \mathcal{E} + \tfrac{1}{2}(\Gamma_o - H)\,\mathcal{E} + \rho_o \frac{d\Gamma}{d\rho}\mathcal{E} + \Gamma_o \rho_o \frac{\partial \mathcal{E}}{\partial \rho} + N\,\Gamma_o \mathcal{E}_s \qquad (4.86)$$

Differentiating the Mie-Grueneisen equation (4.42) and setting $\rho = \rho_o$ leads to

$$\frac{\partial p}{\partial \rho} = c_o^2 + \rho_o \frac{d\Gamma}{d\rho}\mathcal{E} + \Gamma_o \mathcal{E} + \Gamma_o \rho_o \frac{\partial \mathcal{E}}{\partial \rho} \qquad (4.87)$$

for p_H given by (4.35) and (4.38) providing that (4.39) is satisfied. In order that the slopes match, equating (4.86) and (4.87), we have the condition

$$N = \frac{c_o^2}{\Gamma_o \mathcal{E}_s} - \tfrac{1}{2}\left(1 - \frac{H}{\Gamma_o}\right)\frac{\mathcal{E}}{\mathcal{E}_s}$$

The second term is much smaller than the first when $\mathcal{E} \ll \mathcal{E}_s$, and N is usually chosen as

$$N = \frac{c_o^2}{\Gamma_o \mathcal{E}_s} \tag{4.88}$$

so that the two equations (4.42) and (4.82) are approximately continuous in slope at $\rho = \rho_o$.

Equation (4.82) can be put into the form

$$p = f_1(\rho) + f_2(\rho)\,\mathcal{E}$$

where

$$f_1 = A\,(\exp B - 1)\,\rho\,\mathcal{E}_s \tag{4.89}$$

$$f_2 = A\,\rho \tag{4.90}$$

The sound speed will be given by (4.49). However,

$$\frac{df_1}{d\rho} = \left(A + \rho\frac{dA}{d\rho}\right)\left(\exp B - 1\right)\mathcal{E}_s + \rho A \exp B \frac{dB}{d\rho}\mathcal{E}_s \tag{4.91}$$

$$\frac{df_2}{d\rho} = A + \rho\frac{dA}{d\rho} \tag{4.92}$$

where

$$A = \left\{ H + (\Gamma - H)\sqrt{\mu + 1} \right\}$$

$$\frac{dA}{d\rho} = \frac{1}{\rho_o\sqrt{\mu + 1}}\left\{\frac{d\Gamma}{d\eta}(1 - \eta) + \tfrac{1}{2}(\Gamma - H)\right\} \tag{4.93}$$

$$B = N\eta(1-\eta) \tag{4.93}$$

$$\frac{dB}{d\rho} = N\frac{\rho_0}{\rho^2}(1-2\eta)$$

4.6 Tensile Stress Limit

A material cannot support an indefinitely large tensile stress. Provision is made to limit the tensile stress as one means of simulating fracture or cavitation. This is accomplished as follows:

If $\sigma_{j-1/2}^{n+1} < \sigma_{min}$

where σ_{min} is usually a negative quantity, then $\sigma_{j-1/2}^{n+1}$ is set equal to σ_{min} and the energy is recomputed by setting

$$p_{j-1/2}^{n+1} = \sigma_{min} + \sigma_{xj-1/2}^{dn+1} \tag{4.94}$$

and using (4.25) to recompute $\mathcal{E}_{j-1/2}^{n+1}$.

4.7 Solid Equation of State Subroutine

All of the previously described features are combined in a single subroutine STATE1. Thus, a solid material which supports a shear stress may be allowed to melt and vaporize, the correct equation of state being chosen automatically.

Certain features may be suppressed if they are not required. The vapor equations are normally used if $\rho < 0$. However a test on H is included so that if H = 0, then the vapor equation is by-passed, and the normal equations for the solid are used. Similarly, deviator stresses are normally computed if $\sigma < Y_2$. However, computation of deviator stresses is by-passed entirely if the indicator NOY is set to zero.

Since several options are included, a few unnecessary logical, and occasionally arithmetic, operations are performed when the simpler forms of the equations are used. Care has been taken to minimize unnecessary operations. Nevertheless, if extensive production runs are contemplated using a particular simplified form of the equation of state, it may be possible to reduce running times slightly by reprogramming the equation of state to incorporate only those features which are required.

4.8 High Explosives

High explosives are treated by considering that no pressures can appear in the undetonated explosive, and by forcing the detonation wave to move at the Chapman-Jouguet velocity from the initiation point.

If x_0 is the position of the point of initiation, then the time at which the detonation wave will reach a particular mesh is

$$t^b_{j-1/2} = \frac{\left| \frac{1}{2}(x^o_j + x^o_{j-1}) - x_0 \right|}{D} \tag{4.95}$$

where D is the Chapman-Jouguet detonation velocity. Then the equation of state of the detonation products is written.

$$p^{n+1}_{j-1/2} = F\ (f_1 + e^{n+1}_{j-1/2}\ f_2) \tag{4.96}$$

where F is a burn fraction given by

$$\begin{aligned} F &= 0 \qquad \text{if} \quad t^{n+1} \le t^b_{j-1/2} \\ F &= \frac{D(t^{n+1} - t^b_{j-1/2})}{B_5\,(x^{n+1}_j - x^{n+1}_{j-1})} \qquad \text{if} \quad t^{n+1} > t^b_{j-1/2} \end{aligned} \tag{4.97}$$

with the restriction $F \leq 1$. The constant B_5 is a factor, generally 2.5, which spreads the detonation front over several meshes.

Solving (4.96) and the energy equation (4.25) for the internal energy leads to

$$\varepsilon^{n+1}_{j-1/2} = \frac{\varepsilon^{n}_{j-1/2} + (F f_1 + p^{n}_{j-1/2} + g^{n+1}_{j-1/2} + g^{n}_{j-1/2}) \left(\frac{\Delta \rho}{2\rho^2}\right)}{1 - F f_2 \frac{\Delta \rho}{2\rho^2}} \tag{4.98}$$

the pressures being found from (4.96).

Since explosive gases cannot support a shear stress, it is unnecessary to distinguish between τ and p. Thus the pressure p is stored directly in the array named SIGMA. The burn time t^b, which is computed on the first call to this subroutine, is stored in the array named P.

Functions f_1 and f_2 appropriate to the explosion products must be supplied. For a perfect gas, the equation of state is

$$p = (\gamma - 1) \rho \varepsilon \tag{4.99}$$

where γ is the ratio of specific heats. Writing this in the form (4.29), the functions f_1 and f_2 for perfect gas explosion products are

$$\begin{aligned} f_1 &= 0 \\ f_2 &= (\gamma - 1) \rho^{n+1}_{j-1/2} \end{aligned} \tag{4.100}$$

Also

$$\frac{df_1}{d\rho} = 0 \tag{4.101}$$

$$\frac{df_2}{d\rho} = (\gamma - 1)$$

so that if these are inserted into (4.49), the expression for sound speed in a perfect gas is obtained, viz.,

$$c^2 = \frac{\gamma p}{\rho} \tag{4.102}$$

Equations for perfect gas explosion products are supplied in the subroutine STATE 2. The forms (4.96) and (4.49) for p and c are retained, although they lead to some inefficiencies in computation, in order to provide flexibility in accommodating more realistic equations of state.

If the equations for mass, momentum, and energy conservation across a shock, viz.

$$\begin{aligned} \rho\,(D - u) &= \rho_0\,(D - u_0) \\ p + \rho(D - u)^2 &= p_0 + \rho_0\,(D - u_0)^2 \\ \varepsilon - \varepsilon_0 - Q &= \tfrac{1}{2}\,(p + p_0)\,(v_0 - v) \end{aligned} \tag{4.103}$$

where Q is the chemical energy added in the detonation, are combined with the Chapman-Jouguet condition

$$D = c + u \tag{4.104}$$

and the expression for sound speed (4.102), then the pressure, density, and energy at the Chapman-Jouguet point immediately behind the detonation wave moving into undisturbed solid explosive ($p_0 = u_0 = 0$) are

$$\begin{aligned} p_{CJ} &= \frac{1}{\gamma+1}\,\rho_0\,D^2 \\ \rho_{CJ} &= \frac{\gamma+1}{\gamma}\,\rho_0 \\ \varepsilon_{CJ} &= \frac{\gamma}{(\gamma^2-1)(\gamma+1)}\,D^2 \end{aligned} \tag{4.105}$$

and the chemical energy added in the detonation is

$$Q = \frac{D^2}{2(\nu^2 - 1)} \tag{4.106}$$

For perfect gas explosion products the required equation of state input parameters are chosen to be ν and D. The pressure (SIGMA) is initialized to zero, the density is initialized to ρ_0, the density of the solid undetonated explosive, and the internal energy is initialized to Q.

4.9 Gasses

When $F = 1$, the above equations are appropriate for a gas, irrespective of whether the gasses are detonation products or not. Thus, equations (4.98) and (4.96) with $F = 1$ are used. Functions for f_1, f_2, $\frac{\partial f_1}{\partial \rho}$ and $\frac{\partial f_2}{\partial \rho}$ appropriate to the particular real gas being used must be supplied. If the gas is a perfect gas, then equations (4.100) and (4.101) are used, and a subroutine STATE3 is supplied using these equations.

The pressure is stored in the array named SIGMA. The array named P is not used.

The pressure, density, and internal energy must be initialized to appropriate values. Note that the pressure (stored in SIGMA) must never be initialized to zero. The initial values of pressure, density, and energy must exactly satisfy the equation of state. (For a perfect gas p, ρ and ε must exactly satisfy (4.99).)

4.10 Other Constitutive Equations

The program has been written expressly to allow the constitutive equations or equations of state to be changed easily. Any set of constitutive equations which compute the quantities $\sigma_{j-1/2}^{n+1}$ and $\varphi_{j-1/2}^{n+1}$ from the following quantities can be written:

x_{j-1}^{n} x_{j}^{n} x_{j-1}^{n+1} x_{j}^{n+1}

$u_{j-1}^{n+1/2}$ $u_{j}^{n+1/2}$

$\rho_{j-1/2}^{n}$ $\rho_{j-1/2}^{n+1}$

$q_{j-1/2}^{n}$ $q_{j-1/2}^{n+1}$

$\sigma_{j-1/2}^{n}$

$\varphi_{j-1/2}^{n}$

The quantities x, u, ρ, q, σ, and φ are saved in arrays. When the equation of state subroutine is entered, all of the above quantities are available. In addition, two other arrays are provided to save information at each mesh point for use internal to the equation of state. These are labeled $\mathcal{E}$ and p and are used in the routines described previously for the internal energy and pressure. However, they are not used anywhere else in the program except for output and may be used for storage of other quantities if internal energy and pressure are not required in the equation of state. (Note that $\mathcal{E}$ and p are initialized in the initializing routine.) Additional arrays may be added when required as described in Section 6.1. Such additional arrays are used, for example, when strain hardening is included and the plastic work done must be computed and saved at each mesh.

5. STABILITY

The computation is advanced each cycle by a time increment

$$\Delta t^{n+1/2} = t^{n+1} - t^{n} \tag{5.1}$$

The choice of time increment is not independent of the choice of mesh size. Without entering into a full discussion at this point, the numerical method becomes unstable if the time increment becomes too large. Instability leads to oscillations which grow very rapidly with time. The criterion for stability for the difference equations used here is (see Apendix A)

$$\begin{aligned} \Delta t &\leq \frac{\Delta x}{B_2 c + B_1^2 |\Delta u| + \sqrt{(B_2 c + B_1^2 |\Delta u|)^2 + c^2}} && \text{for } \Delta u < 0 \\ &\leq \frac{\Delta x}{c} && \text{for } \Delta u \geq 0 \end{aligned} \tag{5.2}$$

where $\Delta x = x_j - x_{j-1}$, and B_1 and B_2 are defined by (3.3).

The criterion (5.2) is applied to each mesh, the minimum value over all meshes being used to advance the calculation. The criterion (5.2) is actually computed at the conclusion of each mesh computation, the minimum value first being used in (2.8) and (2.9) on the next cycle. Thus (5.2) is written

$$\begin{aligned} \Delta t_{j-1/2}^{n+3/2} &= \frac{K_{t1}\left(x_j^{n+1} - x_{j-1}^{n+1}\right)}{B_2 c_{j-1/2}^{n+1} + B_2^2 |\Delta u| + \sqrt{(B_2 c_{j-1/2}^{n+1} + B_1^2 |\Delta u|)^2 + (c_{j-1/2}^{n+1})^2}} && \text{for } \Delta u < 0 \\ &= \frac{K_{t1}\left(x_j^{n+1} - x_{j-1}^{n+1}\right)}{c_{j-1/2}^{n+1}} && \text{for } \Delta u \geq 0 \end{aligned} \tag{5.3}$$

where $\Delta u = u_j^{n+1/2} - u_{j-1}^{n+1/2}$.

The factor K_{t1} is included so that the time increment may be reduced below that for stability. This factor is an input variable and is normally chosen to be 1. Occasionally, when a more stringent criterion is desired, it may be set to 0.95 or 0.9.

In order to limit the rate of increase of Δt, the value actually used on the next cycle is

$$\Delta t^{n+3/2} = \text{Min}\left(\Delta t^{n+3/2}_{j-1/2},\ K_{t2}\ \Delta t^{n+1/2}\right) \tag{5.4}$$

The factor K_{t2} is an input variable and is normally chosen to be 1.1 or 1.2. If this feature is not desired, K_{t2} may be made a very large number, say 100.

Occasionally, it is desirable to start a calculation with a smaller Δt than required for stability. Such a case arises, for instance, if there is an initial pressure or velocity discontinuity in the initial conditions. (See Section 6.5) The desired initial time increment may be read in as input in DELT (4). Then K_{t2} may be used to control the rate at which Δt increases until it is controlled entirely by stability. If this feature is not desired, DELT (4) may be left blank, which is read as zero. The program then automatically assigns a value of 10^5.

When energy sources are included, the energy added each cycle as ΔQ in (4.25) must be small. If the time during which energy is deposited is small, then the time increment for stability may be too large. The deposition time is called TDEP. This is normally initialized to zero. If it is non-zero, then if t is less than TDEP, the smaller of the time increment for stability and one hundredth of the deposition time is used to advance the computation. For an example of the use of TDEP, see Section 6.7.

6. INITIAL AND BOUNDARY CONDITIONS

6.1 Storage Arrangement

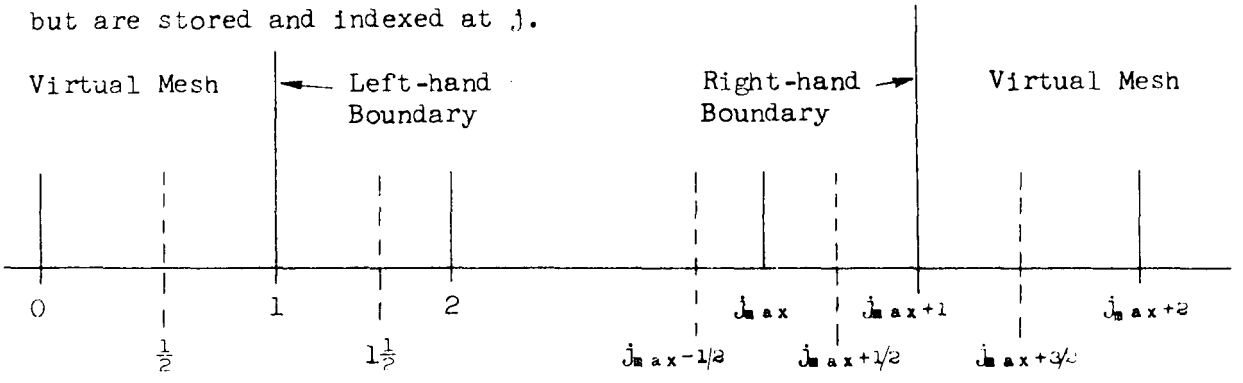
Initial and boundary conditions are very simple mathematically but require an understanding of the storage arrangement of variables in the program if the way they are treated in the program is to be understood.

Normally, ten quantities are stored at each mesh point. These are, in order, x, u, σ, ρ, q, φ, p, m, $\mathcal{E}$, c. The ten quantities are normally arranged in arrays overlaid in a single array named STORE in such a way that the first ten quantities are x, u, σ, etc., for j = 1, the next ten quantities are x, u, σ, etc., for j = 2, and so on, up to the maximum value of j. In this way storage is packed with no vacant locations interspersed with locations containing data. The number of variables may be increased by specifying an input quantity NVAR. Usually NVAR = 10 and the STORE array is arranged as described above. However, when extra quantities are required (for example, in the equation of state), then NVAR may be set to an integer greater than 10 to accommodate the extra arrays. Thus, for example, if one variable is to be added, NVAR = 11 and the eleventh quantity in STORE is this extra variable for j = 1, the 22nd quantity is this extra variable for j = 2, and so on. In the present version STORE is dimensioned 31,100. The maximum number of meshes which can be accommodated is

$$j_{max} = \frac{31,100}{NVAR} - 3 \leq 3,100 \qquad (6.1)$$

rounded to the next lowest integer. (Note that meshes are allocated to j = 0, $j_{max} + 1$, and $j_{max} + 2$)

Mesh quantities, τ, ρ, q, φ, p, m, $\mathcal{E}$, and c are centered at $j - \frac{1}{2}$ but are stored and indexed at j.



Thus at the left-hand boundary the position and velocity of the boundary itself are indexed with $j = 1$. The values of σ, ρ, q, φ, p, m, $\mathcal{E}$, and c indexed with $j = 1$ then actually refer to a location outside the left-hand boundary. These quantities appropriate to a virtual mesh outside the boundary are initialized to zero and are usually not used. However, they may be used to implement certain types of boundary conditions.

The maximum number of meshes in a problem is termed j_{max} (JMAX). Since we have started indexing at 1, the mesh boundary indexed j_{max} is actually one mesh short of the right-hand boundary. The right-hand boundary is indexed j_{max+1}. The position and velocity of the right-hand boundary are indexed j_{max+1}. The values of τ, ρ, q, φ, p, m, $\mathcal{E}$, and c indexed j_{max+1} refer to the mesh just inside the right-hand boundary. In the computer an additional virtual mesh indexed j_{max+2} is provided, which refers to a location outside the right-hand boundary. All quantities in this mesh are initialized to zero and are usually not used, but they may be used to implement certain types of boundary conditions.

When a mesh calculation is started at mesh j, the values of the ten variables stored in arrays have already been advanced at all meshes to the left, i.e., for smaller values of j, and are therefore appropriate to time n + 1. The values of the stored variables at j and at all meshes to the right, i.e., for larger values of j, have not yet been advanced and are appropriate to time n. Only after all calculations are complete at j are the variables in the arrays advanced to their new values appropriate to time n + 1.

6.2 Boundary Conditions

Three types of boundary conditions are provided at the left-hand and right-hand boundaries of the problem. They are: 1) a fixed boundary or reflection plane, 2) a free surface, and 3) a special boundary routine contained in a subroutine BOUNDARY which must be supplied by the user. Two indicators are used to determine the left-hand and right-hand boundary types called LHBT and RHBT, respectively. They are specified to be 1, 2, or 3 according to whether the boundary is fixed, free, or special, respectively.

The boundary condition affects only the calculation of acceleration, velocity, and position at the boundary in (2.7), (2.8), and (2.9).

For a fixed boundary, computation of acceleration a and velocity u are omitted, and these quantities are set to zero, while the position x is left unchanged.

In order to deal with a free surface, use is made of the virtual meshes outside the boundary. The values of σ, φ, and ρ are initialized to zero in

these virtual meshes outside the boundaries, and their values are not changed during the calculation. Use of the ordinary equations (2.7), (2.8), and (2.9) at the boundaries then leads to the correct acceleration, velocity, and position of the free surface.

A different type of boundary condition may be introduced via a subroutine BOUNDARY, which is called if either of the boundary indicators LHBT or RHBT is set to 3. This routine may be used to insert values of σ, ρ, e, or x in the virtual mesh outside the boundary. Note that these quantities are used only in the momentum equation (2.7) to calculate the acceleration at the boundary.

As one example of the use of the special boundary type, a BOUNDARY subroutine is included which applies a time-varying load on either boundary (but not both), given by

$$\sigma = \sigma_0 + \sigma_1 \exp(-K_\sigma t) \tag{6.2}$$

where σ_0, σ_1, and K_σ are constants. If $\sigma_0 = 0$, an exponentially decaying load is applied. If $\sigma_1 = 0$, a step function load is applied. Note that K_σ should be positive. A normally vacant input array ADDATA is provided which is used to input these three constants. The values of σ_0, σ_1, and K_σ are the 8th, 9th, and 10th quantities in ADDATA. (In order to use this feature, NOAD, the number of ADDATA variables read, must be set to at least 10.)

The work done at the boundary in a time cycle will be the applied stress times the distance moved by the boundary. In finite difference form

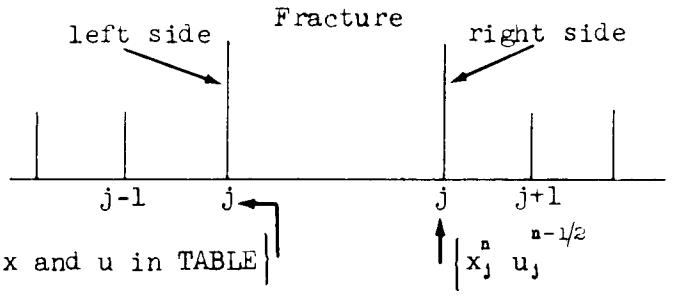
$$W^{n+1/2} = \frac{1}{2}\left(\sigma^{n+1} + \sigma^{n}\right)\left(x^{n+1} - x^{n}\right) \qquad (6.3)$$

where the appropriate index for the virtual mesh is used for σ, and for the boundary mesh for x. This work is added to the total energy in performing energy checks. See Section 8.1.

6.3 Spall and Join

A spall and join routine is provided for an alternate treatment of fracture to that discussed in Section 4.6. When the stress σ drops below the fracture stress, specified as input, the material is allowed to separate to form two free surfaces. If subsequently these surfaces collide, then the material is considered to rejoin, and the ordinary equations appropriate to an interior mesh are used. Subsequent fractures at a mesh which has fractured previously are considered to occur if the stress drops below a value which is effectively zero.

Since the material is considered to fracture only at mesh boundaries, σ is interpolated to the mesh boundary before testing for fracture. The logic is accomplished through two arrays of logical indicators called PFRACT and QFRACT. If the interpolated stress σ drops below the fracture stress, QFRACT is set to 1. This signals that two free surfaces occur at that mesh. It is now necessary to store extra values of x and u. The values of x and u for the right side of the fracture are stored in the X and U storage arrays, but the values of x and u for the left side of the fracture are stored in TABLE. In addition, mesh numbers of meshes currently fractured and separated are stored in ITABLE.



Different fracture stresses may be specified in each different layer of material (SIGMAF) and at each interface between materials (SIGMAIF).* The latter might represent a weak bond between layers.

During subsequent calculation at a mesh where a fracture has occurred, a test is made to see if the value of x at the left side exceeds the value of x at the right side of the fracture. If it does, the fractured surfaces have come together during that cycle. The values of x and u at the two sides are averaged and inserted into the X and U storage arrays, QFRACT is set to 0 and PFRACT is set to 1. This signals that the mesh is henceforth to be treated as an ordinary interior mesh. However, PFRACT signals that subsequent tests for fracture are to be made on a quantity SIGMASEP instead of on the fracture stress. SIGMASEP is an input quantity and may be set to zero or to a small negative value to prevent separation on small spurious oscillations about zero stress which occasionally occur in the solution.

Messages are printed on the standard output medium whenever a fracture occurs or fractured surfaces collide giving the cycle, time, and mesh number. A maximum of 50 fractures are allowed. If this number is exceeded, an error message is printed on the standard output medium and the run is terminated.

---

* The first SIGMAIF refers to the interface between the first and second material layers, etc. The last SIGMAIF has no significance.

6.4 Initial Conditions

Initial conditions are specified by assigning values to all quantities at all mesh points. This is accomplished in a subroutine GENERATE. Input cards are read and a detailed tabulation of input data is printed in GENERATE so that the main program need not be disturbed when these are altered.

The quantity q is always initialized to zero in all meshes. The following arrays are initialized according to information contained in the input data: x, u, ρ, p, σ, φ, $\mathcal{E}$, and c. The quantity m is computed once and for all and is usel only in (2.10) and must not be tampered with.

The quantities u, ρ, p, σ, φ, $\mathcal{E}$, and c are given constant values in each material layer but may be different in different material layers.* The way in which each one of these quantities is initialized is described below.

The sound speed c is given as the second equation of state constant and corresponds to the sound speed of the material in its natural uncompressed state. Initial values of ρ, p, σ, φ, and $\mathcal{E}$ are specified for each material layer in the input data. Note that values of ρ, p, σ, φ, and $\mathcal{E}$ must be fully compatible. Thus if the material is initially compressed to some pressure p, then values of σ, φ, ρ, and $\mathcal{E}$ appropriate to this compression from the initial uncompressed state must be used. In particular, note that the initial value of ρ will be different from ρ_o specified as the first equation of state constant, which is the reference density in the uncompressed state.

---

* For a way in which more complex initial conditions can be accommodated, see Section 7.4.

Values of c, ɔ, p, ʊ, ƿ, and $\mathcal{E}$ are initialized to zero in the virtual meshes outside the left-hand boundary (at $j = 1$) and right-hand boundary ($j = j_{max} + 2$).

6.5 Initial Velocity

Initialization of the velocity is a little more complicated since the velocity refers to mesh boundaries. The velocities of meshes within each material layer are initialized to values given in the input data for each material layer (UZERO). However, the velocities at interfaces between material layers is specified separately in the input data (UZEROI).* The velocity at the left-hand boundary ($j = 1$) is automatically set equal to that in the first material layer, while the velocity at the right-hand boundary ($j = j_{max} + 1$) is automatically set equal to that in the last material layer.

To illustrate the use of the initial interface velocities, consider a plate impact problem in which the first material layer has a positive velocity, while the second material layer has a zero velocity. The problem is considered to start at the moment of impact. The material at the interface will be compressed and begin to move with a velocity intermediate between that of the first and second material layers. In order to minimize starting transients, the correct interface velocity could be calculated from the shock impedances of the two materials, and this can be entered as the initial interface velocity.

For many purposes it is sufficiently accurate to initialize the interface velocity to the average of the velocities of the adjacent layers.

---

*The first UZEROI refers to the interface between the first and second material layers, etc. The last UZEROI has no significance.

Occasionally, no serious violence is done to the solution if either one of the velocities of the adjacent layers is used directly. After a number of cycles of computation, the interface velocity will automatically assume the correct value.

Some difficulties may arise in problems where the difference in velocity of adjacent material layers becomes very large. Since the time increment is determined by the stability criterion (5.3), on the first cycle the time increment reduces essentially to

$$\Delta t = \frac{\Delta x}{c}$$

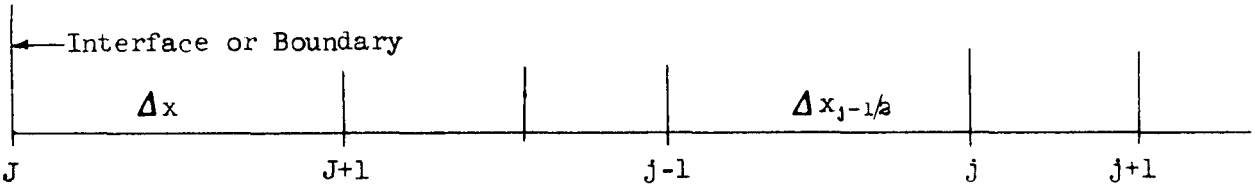
Thus when the initial velocity approaches the value of the sound speed c, the meshes adjacent to the interface will undergo very large compressions. In fact, it may happen that the interface moves beyond one of its neighboring mesh boundaries, leading to a negative density in that mesh. The code does not automatically check for this condition. It is necessary in such problems to choose an initial time increment Δt sufficiently small so that the meshes adjacent to the interface do not change volume by more than 10 per cent. This time increment may be entered in DELT(4) in the input.
KT2 (see Section 5) can then be used to control the rate of increase of Δt on successive cycles until Δt is controlled entirely by the stability criterion.

Similar problems may be encountered if a large initial pressure discontinuity is introduced. The difficulty makes itself felt by introducing large oscillations at an interface or boundary. Use of a sufficiently small initial time increment usually alleviates the difficulty.

The velocity in the virtual mesh outside the right-hand boundary ($j = j_{max} + 2$) is set equal to that in the last material layer.

6.0 Zoning

The positions x of the mesh boundaries are initialized in such a way that the mesh size may be constant, increasing or decreasing in each layer of material. We will denote the value of j at the left-hand boundary or interface of a material layer by J.*



Then the total number of meshes k between the left-hand boundary or interface J and a mesh boundary j will be

$$k \equiv j - J \tag{6.4}$$

The position of the jth mesh is computed from

$$x_j = x_{j-1} + \Delta x\, r^k \tag{6.5}$$

where Δx and r are constants for each layer.

For the first mesh in the layer, from (6.5)

$$\Delta x_0 = x_{J+1} - x_J = \Delta x\, r \tag{6.6}$$

---

*Note that J = 1 for the first material layer.

The value of r (XRATIO) and the size of the first mesh Δx_0 (DELTAX) are specified for each material layer in the input data.

Note that (6.5) is equivalent to

$$\Delta x_{j+1/2} = r\, \Delta x_{j-1/2} \tag{6.7}$$

where

$$\Delta x_{j-1/2} = x_j - x_{j-1} \tag{6.8}$$

etc. Thus, the mesh size progressively increases if $r > 1$ and progressively decreases if $r < 1$. If $r = 1$, the mesh size in a layer is constant. The total distance ℓ from the left-hand boundary or interface J to the jth mesh is given by the sum of (6.5).

$$\ell \equiv x_j - x_J = \Delta x_0 \frac{r^k - 1}{r - 1} \tag{6.9}$$

The size of the last mesh is, from (6.5), (6.6), and (6.8)

$$\Delta x_{j-1/2} = \Delta x\, r^k = \Delta x_0\, r^{k-1} \tag{6.10}$$

Thus the ratio of sizes of last mesh to first mesh is

$$R \equiv \frac{\Delta x_{j-1/2}}{\Delta x_0} = r^{k-1} \tag{6.11}$$

Substituting (6.10) into (6.9) and solving gives the useful relations

$$\ell = \frac{r\Delta x_{j-1/2} - \Delta x_0}{r - 1} \tag{6.12}$$

$$r = \frac{\ell - \Delta x_0}{\ell - \Delta x_{j-1/2}} \tag{6.13}$$

Solving (6.11) for the number of meshes k gives

$$k = \frac{\log \Delta x_{j-1/2} - \log \Delta x_0}{\log r} + 1 \tag{6.14}$$

The equations (6.9) through (6.14) are useful in determining the input quantities from the desired thickness of the layer ℓ and the ratio of sizes of the last mesh and first mesh R. If the first mesh size is specified, i.e., if ℓ, Δx_0, and R are given, then r and the total number of meshes k are given by (6.13)

$$r = \frac{\ell - \Delta x_0}{\ell - R\Delta x_0} \tag{6.15}$$

and (6.14)

$$k = \frac{\log R}{\log r} + 1 \tag{6.16}$$

Alternatively, if the total number of meshes is specified, i.e., if ℓ, k, and R are given, then r and the initial mesh size Δx_0 are given by (6.11)

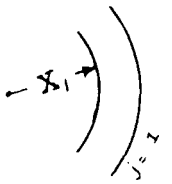
$$r = R^{\frac{1}{k-1}} \tag{6.17}$$

and (6.9)

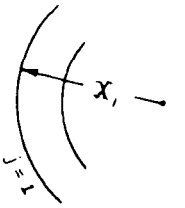
$$\Delta x_0 = \ell \frac{r - 1}{r^k - 1} \tag{6.18}$$

Some iteration is usually required to choose appropriate values of Δx_0, r, and R to give a suitable number of meshes for a given thickness.

The position x_1 of the left-hand boundary at j = 1, called XZERO, is specified as input. In the rectangular case, α = 1, the choice of this position does not affect the results and should normally be zero to minimize roundoff. However, in the cylindrical and spherical cases, α = 2 or 3, XZERO determines the radius at the left-hand boundary j = 1.



Concave



Convex

If this boundary is to be concave as shown, then XZERO must be positive. If this boundary is to be convex as shown, then XZERO must be negative.

It is also possible to introduce a gap between successive layers when setting up a problem. This is done easily by making use of the spall and join feature described in Section 6.3. It is only necessary to specify that an interface is initially fractured and separated. When the surfaces subsequently collide during the motion, the spall and join routine automatically computes the correct behavior at the interface.

Gaps may be introduced at each interface between material layers by specifying an input quantity XGAP for each layer.* If XGAP is vacant, read as zero, then zoning proceeds normally and no gap is introduced. However, if XGAP is non-zero, then initialization proceeds as follows: QFRACT for

---

*The first XGAP refers to the interface between the first and second material layers, etc. The last XGAP has no significance.

the interface mesh is set to 1, indicating that the interface is to be treated as two free surfaces, and the correct positions of the two sides of the gap are computed. The velocities of the two sides of the gap are set equal to the velocities in the adjoining layers. (Note that the input value of UZEROI is not used at this interface.)

6.7 Additional Arrays, Energy Sources

The subroutine GENERATE initializes the ten storage arrays x, u, ρ, p, σ, φ, $\mathcal{E}$, and c but does not initialize any of the extra storage arrays which may be provided by setting NVAR > 10 (see Section 6.1). This initialization may be accomplished in a special subroutine MORSTORE called from the main program after generate if NVAR > 10.

The subroutine MORSTORE must be written specifically for each application since all of the possible applications cannot be foreseen. The simplest use is to initialize one or more extra storage arrays to zero. Note that MORSTORE must be compatible with the number of extra variables specified by NVAR. Additional data required by MORSTORE may be read as ADDATA or may be read directly by READ statements in MORSTORE from additional data cards.

As an example of how an extra array may be used to specify energy sources, a very simple subroutine MORSTORE is included which specifies energy deposited in the material at a uniform rate for a given time interval. Much more complex subroutines which determine the energy deposition due to electromagnetic radiation, etc., may be programmed as required.

The total energy per unit mass deposited in each layer is read from additional input data cards. The energy is considered uniform in a given material layer, but may be different in each layer. This energy is stored for each mesh in an array SPEC which is overlaid with the normal storage arrays in STORE.

The energy is deposited at a constant rate for a total time TDEP, whereafter no further energy is deposited. TDEP is read as an input quantity on the additional data cards and placed in COMMON. Energy addition is accomplished in the equation of state via Q in (4.25), and this part of the computation is done in the equation of state subroutine.

At the beginning of each time cycle a quantity DEP is computed, given by

$$\text{DEP} = \frac{t^{n+1/2}}{\text{TDEP}} \tag{6.21}$$

Then the energy added at a particular mesh on that cycle, $Q_{j-1/2}$, is the product of DEP and SPEC for that mesh. The total energy added to all meshes in that cycle is summed (SUMQE). This energy sum is required to perform energy checks. (See Section 8.1) The above computation is performed by a number of special cards in the equation of state. If the way in which energy is to be added is changed, these cards must be altered.

It is important that the energy addition on each time cycle is small in order to avoid truncation errors in (4.25). In order to keep this energy addition small, the time increment used to advance the calculation must be kept small. The time increment required for stability may be too large. The time increment is therefore limited to one hundredth of the deposition time TDEP, while $t < \text{TDEP}$. (See Section 5)

7. OUTPUT ROUTINES

There are four output methods: 1) Binary Dump Tape, 2) Standard Editing, 3) Special Printed Output for a completed time cycle, 4) Special Printed Output during a time cycle. Several input parameters control frequency at which output is taken by each method. In each case the minimum time, time increment, and maximum time at which each type of output is required must be specified.

This is accomplished as follows: For the Binary Tape Dump, three input variables are provided, called TMIND, TDUMP, AND TMAXD. At each cycle a test is made to determine if the time t is greater than TMIND. On the first cycle on which t exceeds TMIND, the tape dump is called. Subsequent tests are made on TMIND + TDUMP, so that the second dump occurs after a further time equal to TDUMP. This is repeated adding TDUMP to the test time after each dump, so that the dump is called at time intervals TDUMP until TMAXD is exceeded, whereafter the tape dump is not called. If the output is to be inhibited, TMIND can be made larger than the maximum time in the program TMAX. If the output is to be called every cycle, TMIND and TDUMP can be set equal to zero. Exactly similar methods are used for calling the standard edit (via TMINP, TPRINT, and TMAXP) the output routine OUTPUT (via TMINPS, TPRINTS, and TMAXPS) and the output routine OUTL (via TMINPL, TPRINTL and TMAXPL).

In addition, a variety of messages and diagnostics are printed on the standard output medium during the computation, such as information concerning the occurrence or rejoining of fractures, overflow, occurrence of energy errors, normal exit, etc. Most messages include the mesh number, cycle, and time at which they were printed, and should be self-explanatory. For detailed information concerning origin of error messages, refer to the program listings and flow charts.

7.1 Binary Tape Dump

The primary output is via binary tape, which is written on logical tape unit 20. This tape may be used for subsequent plotting, listing, and tape storage of the results. The binary dump tape also contains sufficient information to restart the problem. (See Section 7.4)

The primary information on the dump tape is the STORE array, i.e., all quantities in the overlaid arrays in which values of x, u, ɔ, ɔ, q, ⁿ, p, m, ᵉ, and c and any additional variables are stored. (See Section 6.1) Also contained on the dump tape is information concerning fractures stored in TABLE and ITABLE. (See Section 6.3)

Each time the binary dump tape is written, a message appears on the standard output medium giving the cycle number, time, time step, and number of meshes written on the tape. If fractures have occurred, their mesh numbers are listed. If fractures have rejoined, their mesh numbers are also listed. This information is needed to restart the problem.

7.2 Standard Edit

Standard edit is written on logical tape unit 21 in B. C. D. by the main program and may be listed directly on a printer or the SC 4020. An input indicator W4020 should be set equal to 1. If only a small amount of printed output is required, the output may be put onto the standard output medium by equivalencing logical tape unit 21 to the standard output unit. In this case W4020 should be set equal to 0 to prevent duplication of error and input messages.

The primary information in the standard edit is the values of the following arrays at each mesh point: x, u, ρ, σ, η, q, $\mathcal{E}$, and c. In addition, the cycle number, time, and time increment are printed. At the end of the edit are printed the energy sums described in Sections 8.1 and values of x and u at the left-hand sides of any fractures. The edit is also called initially to check that the problem has been correctly set up, and when the problem terminates for any reason.

7.3 Special Printed Output and Plotting

Two special output subroutines OUTPUT and OUTL are provided which may be used to obtain information not contained in the standard edit. Since output requirements vary greatly depending on the problem, it is expected that these subroutines will be written as required. They may be written to print information not contained in the standard edit, to write binary data tapes for subsequent computations or for input to subsequent plotting programs, or they may be programmed to prepare plots directly. The special array ADDATA, which can be read as input, may be used to communicate with these subroutines.

OUTPUT is called at the completion of a cycle when all of the storage arrays overlaid in STORE have been advanced. It is chiefly useful for printing selected information in the storage arrays at more frequent intervals than the standard edit, or for writing selected information in the storage arrays on a binary tape for subsequent plotting. Calculations may be performed on the data before printing or writing.

A typical example of the use of OUTPUT is to print values of the principal stresses

$$\sigma_x = \sigma^{n+1}_{j-1/2}$$

$$\sigma_y = \sigma^{n+1}_{j-1/2} + \phi^{n+1}_{j-1/2} \tag{7.1}$$

$$\sigma_z = 2\sigma^{n+1}_{j-1/2} + \phi^{n+1}_{j-1/2} - 3p^{n+1}_{j-1/2}$$

(taken positive in compression) and position

$$x = \frac{1}{2}\left(x^{n+1}_{j+1} + x^{n+1}_{j}\right) \tag{7.2}$$

at a selected number of meshes at every cycle after some specified time. Another typical example is to write the three stresses above on a binary output tape at one or more particular meshes at every cycle for subsequent plotting of stress versus time by means of a suitable plot program. Alternatively, quantities could be written on tape at every mesh for one or more time cycles for subsequent plotting of stress versus position. The OUTPUT subroutine included in the listings writes $\sigma^{n+1}_{j-1/2}$ and x^{n+1}_{j} at each mesh on binary tape whenever the routine is called, together with the cycle number and time. The output from this subroutine is written on logical tape unit 23.

OUTL is called at the completion of each mesh calculation, providing that the cycle is one specified by input variables TMINPL, TPRINTL, and TMAXPL. At this stage of the calculation, intermediate quantities not stored in arrays are available. Thus, OUTL may be used to print or write on binary tape such quantities as

$$\left(\frac{1}{\rho}\frac{\partial \rho}{\partial t}\right),\ f_1,\ f_2,\ P_{\perp},\ \Gamma,\ f_y,\ Y,\ d^{d}_{x},\ \mathcal{C}^{d}$$

etc. etc. at selected times and meshes. It is also possible to perform calculations on the data before writing or printing. As an example, it is possible to compute the plastic work per unit mass done in a given mesh in a given cycle by

$$W^{n+1/2}_{j-1/2} = \frac{f_y}{G^{n+1/2}_{j-1/2}\left(\sigma^{n+1}_{j-1/2} + \sigma^{n}_{j-1/2}\right)}\left(\sqrt{\frac{2}{3}Y^2} - \frac{\frac{2}{3}Y^2}{f_y}\right) \qquad (7.3)$$

and to print this quantity at selected meshes and time cycles.

It is more efficient to use OUTPUT to print or write quantities contained in the storage arrays, since OUTPUT is called only once per cycle, while OUTL is called at each mesh calculation. However, it is sometimes convenient to use OUTL to write quantities such as stress on tape at selected meshes on each time cycle for subsequent plotting versus time, if OUTPUT is being used to write quantities such as stress at each mesh at selected times for subsequent plotting versus position. The OUTL subroutine included in the listings writes $\sigma^{n+1}_{j-1/2}$ at up to seven mesh numbers at each time cycle specified by the input. These mesh numbers are specified in the input as ADDATA (1 to 7). The output from this routine is written on logical tape unit 22.

While plotting routines may be programmed directly into the subroutines OUTPUT and OUTL, it is generally more desirable to have these subroutines write a binary data tape which can then be processed by a separate plotting program. If desired, the plotting program can be submitted as the next program after WONDY in the program batch, so that there is no delay in plotting. Since systems subroutines and hardware to accomplish plotting vary greatly, no plotting programs are included here. They must be written as required.

7.4 Restart Feature

The binary dump tape contains sufficient information to restart the problem. The tape must be equipped to logical tape unit 25. The input variable JTAPE is used to signal that the run is to be restarted from a dump tape. If JTAPE is non-zero, the calculation will be restarted from the dump tape.

For a normal restart the same input cards must be used with the following changes:

1) JTAPE must be set to the number of meshes on tape; NSTART must be set to the cycle number at which the calculation is to be restarted. Both these quantities will be included in the dump message on the original run.

2) If fractures have occurred, their mesh numbers must be entered in QMESH. If fractures have rejoined, their mesh numbers must be entered in PMESH. These are listed in the dump message. The number of fractures NOQM in the list QMESH, and the number of rejoined fractures NOPM in the list PMESH, must also be entered.

3) It may be necessary to change the maximum time TMAX and the quantities specifying times at which output are required. Output must not be requested at times prior to the restart time. This may require changes in TMIND, TMINP, TMINPS, and TMINPL.

Under certain conditions it is possible to change the problem slightly when restarting. To give an example of how this may be done, consider a plate impact problem in which the first layer of material has a positive velocity and the second layer has a zero velocity. The problem is run to a stage where

the shock wave originating from the interface has not yet reached the right-hand boundary of the second layer. At this stage a number of meshes adjacent to the right-hand boundary have not yet undergone any motion and are uncompressed. If a dump is taken at this stage and the problem is restarted, it is possible to add more meshes beyond the original right-hand boundary by suitably altering the input data. This feature is particularly useful when a parametric study is involved in which, say, a third material layer is to be added and the effects of the thickness or composition of this layer are to be investigated. It is unnecessary to rerun the first part of the problem which is unchanged. Very great care is necessary to ensure that changes are made only in or beyond undisturbed meshes. Under no circumstances may changes be made in meshes which have already undergone motion or compression.

It is possible under certain circumstances to use the restart feature to introduce complex initial conditions not allowed for in the present version of GENERATE. A binary tape may be prepared from a suitable program written for the purpose, with quantities in correct sequence, to initialize values of the storage arrays overlaid in STORE and values in TABLE and ITABLE. Very great care must be exercised to ensure that the information on tape is compatible with the input on cards. Also very great care must be exercised to ensure that the values of x, u, σ, ρ, q, $\mathfrak{p}$, p, m, $\breve{c}$, c, and any additional variables are completely compatible with each other and with the equation of state which is to be used. If these values are not completely compatible, totally false results will be obtained.

8. ERROR CHECKS

A number of features are included which permit checking for errors or to speed up the computation, and in some cases, to halt the calculation if errors become serious. These are described below.

8.1 Energy and Momentum Checks

The mass M in a mesh can be related to m given by (2.11) by

$$M_{j-1/2} = k' \, m_{j-1/2} \tag{8.1}$$

where

$k' = 1$ for $\alpha = 1$

$k' = \pi$ for $\alpha = 2$

$k' = \frac{4}{3}\pi$ for $\alpha = 3$

Note that m is not the mass in a mesh except in the rectangular case $\alpha = 1$. The momentum in a mesh may be written in finite difference form, within the factor k', as

$$H_{j-1/2}^{n+1/2} = \frac{1}{2} m_{j-1/2} \left(u_j^{n+1/2} + u_{j-1}^{n+1/2} \right) \tag{8.2}$$

The kinetic energy in a mesh is given within the factor k', as

$$K_{j-1/2}^{n+1/2} = \frac{1}{8} m_{j-1/2} \left(u_j^{n+1/2} + u_{j-1}^{n+1/2} \right)^2 \tag{8.3}$$

while the internal energy in a mesh is given, within the factor k', as

$$E_{j-1/2}^{n+1/2} = \frac{1}{2} m_{j-1/2} \left(\dot{c}_{j-1/2}^{n+1} + \dot{c}_{j-1/2}^{n} \right) \qquad (8.4)$$

These quantities (8.2), (8.3), and (8.4) are computed at each mesh and could be called out in the special printed output, if desired. Various sums over specified numbers of meshes are also occasionally of interest.

It is possible to check whether momentum and energy are conserved during the calculation. In particular, for the momentum

$$k' \sum_{j=2}^{j_{max}+1} H_{j-1/2}^{n+1/2} = \text{Constant} \qquad (8.5)$$

This sum is computed initially from the input data (HTOT). It is subsequently computed on each cycle (HT). A test is made to see if momentum is conserved by testing if

$$\left| HT - HTOT \right| \geq KH \qquad (8.6)$$

where KH is the allowable momentum error and is specified as an input variable. If this error is exceeded, the computation is terminated, an error message is printed, and standard printed output is initiated. If no value is inserted for KH, read as zero, a value of 10^{+00} is used to defeat this test.

The energy balance is more difficult since energy may be added by energy sources $\left(\Delta Q_{j-1/2} \text{ in (4.25)} \right)$ or by work done on boundaries by an applied load

(in subroutine BOUNDARY). The sum of kinetic and internal energy over all meshes is

$$E_{sum}^{n+1/2} = k' \sum_{j=2}^{j_{max}+1} \left(K_{j-1/2}^{n+1/2} + E_{j-1/2}^{n+1/2} \right) \qquad (8.7)$$

This sum is computed initially from the input data (ETOT). It is subsequently computed on each cycle (ET). In addition, if energy sources exist, these must be computed and stored in one of the additional storage arrays available in STORE. The energy added by these energy sources in each cycle must be summed over all the meshes and the result stored in SUMQE. This calculation can be done in the equation of state subroutine (see Section 6.7). If a load is applied to either the left-hand or right-hand boundary, the work done in each cycle must be computed and stored in WL or WR for the left-hand and right-hand boundary, respectively. This calculation can be done in the BOUNDARY subroutine (see Section 6.2).

A check is then made to determine if energy is conserved by testing if

$$\left| ET - \left| ETOT + \sum_t (SUMQE + WL + WR) \right| \right| \geq KE \qquad (8.8)$$

where KE is the allowable energy error and is specified as an input variable. If this error is exceeded, the computation is terminated, and an error message is printed, and standard printed output is initiated. If no value is inserted for KE, read as zero, a value of 10^{+00} is used to defeat this test.

The total energy, kinetic energy, internal energy, and momentum summed over all meshes is printed in the standard printed output, as is the energy error and momentum error.

Occasionally when complex energy sources or boundary loads are used for experimental runs, it is convenient to omit calculation of SUMQE, WL, and WR. The value of the energy error then indicates the amount of energy added from these sources since the beginning of the problem. The value of KE must be set to zero or a very large number to circumvent the energy check in this case.

The energy and momentum checks are very valuable in halting the computation if an error occurs and should normally be used.

8.2 Overflow Test

When instabilities occur, oscillations usually grow exponentially with time until overflow occurs in the computer. If the problem is terminated due to overflow, an abnormal exit occurs and no diagnostics are possible. For this reason an overflow test is incorporated. If the stress σ in any mesh exceeds a maximum pressure σ_{max} which is an input variable, the computation is terminated and a standard printed output is initiated together with an error message.

8.3 Activity Test

In many problems the motion initiates at or near the left-hand boundary. For a significant portion of the calculation, a large number of meshes may be inactive. In order to save computer time, an activity test is incorporated.

A quantity LACT is provided in the input. The computation is performed normally from j = 1 to j = LACT. If the value of σ in the last mesh to be computed, i.e., j = LACT, is less than a quantity SIGMAACT which is also an input variable, the computation is interrupted and advanced to the next time cycle. However, if σ is greater than SIGMAACT, then LACT is advanced by one and the computation is advanced normally.

Thus, meshes are activated as needed as a pulse propagates from left to right. The value of LACT should be specified in the input to be greater than any mesh number at which motion is expected in the first few cycles. To give an example of its use, consider a plate impact problem in which the first layer has a positive velocity, while the second layer has a zero velocity. Then LACT is given an integer value greater than the interface mesh number by, say 5. As the shock initiated at the interface moves to the right into the second layer, meshes are progressively activated just ahead of the shock.

The value of SIGMAACT should be a little greater than possible roundoff or spurious oscillations. Since considerable care has been taken to eliminate roundoff, SIGMAACT can be set to zero.

Note that meshes are activated from left to right. Under no circumstances must LACT be less than $j_{max} + 2$ unless it is absolutely certain that no disturbances originate in the non-active region.

When a standard printed output is called, only the active meshes will be printed.

REFERENCES

1. Von Neumann & Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks", Journal of Applied Physics, Vol. 21, p. 232, 1950.

2. Landshoff, A Numerical Method for Treating Fluid Flow in the Presence of Shocks, Los Alamos Scientific Laboratory, LA 1930, Jan., 1955.

3. Wilkins & Giroux, Calculation of Elastic-Plastic Flow, University of California Lawrence Radiation Laboratory, UCRL-7322, April, 1963.

4. Herrmann, A Lagrangian Finite Difference Method for Two-Dimensional Motion Including Material Strength, AFWL Technical Report No. WL-TR-64-107, Nov., 1964.

5. Richtmyer, R. D., "Difference Methods for Initial-Value Problems," Interscience Publishers, New York (1957)

APPENDIX A

Stability of the Difference Equations

R. J. Thompson

It is well known that when partial differential equations are replaced by difference equations, stability problems may arise. If the difference equations are not stable small rounding errors which occur in the computation are eventually magnified to such an extent that the computation becomes meaningless. In order to have a stable difference scheme, it is often necessary to place a restriction on the size of the time step.*

The purpose of this Appendix is to derive the stability criterion which is used in program WONDY. The program will accept such a wide variety of problems that it appears impossible to carry out a stability analysis which will cover every conceivable situation. On the other hand, it is impractical, if not impossible, to carry out separate stability analyses for each class of problems which the program will accept. In this Appendix an analysis will be made for a particular class of problems. The class is simple enough to be analysed. Nevertheless, it is a large class and incorporates many of the important features of the larger class of problems which can be studied with WONDY. Experience with the program indicates that the stability criterion works well with the larger class of problems.

The stability condition will place a restriction on the size of $\Delta t^{n+\frac{1}{2}}$. In the derivation it is assumed that $\Delta t^{n+\frac{1}{2}} = \Delta t^{n-\frac{1}{2}}$ and their common value will be denoted by Δt. In rectangular coordintes the equations (2.7) and (2.8) combine to give

---

* A rather thorough discussion of stability questions can be found in R. D. Richtmyer (1957)[5].

$$u_j^{n+1/2} = u_j^{n-1/2} + 2\Delta t \left\{ \frac{(p_{j-1/2}^n + q_{j-1/2}^n) - (p_{j+1/2}^n + q_{j+1/2}^n)}{\rho_{j+1/2}^n (x_{j+1}^n - x_j^n) + \rho_{j-1/2}^n (x_j^n - x_{j-1}^n)} \right\} \qquad (4.1)$$

Here it has been assumed that the stress deviators are negligible so that $\sigma = p$. (See Sec. 4.3 for a discussion of this case.) Equations (2.10) and (2.11) combine to give

$$\rho_{j-1/2}^{n+1} = \frac{\rho^o \Delta x}{x_j^{n+1} - x_{j+1}^{n+1}} \qquad (4.2)$$

where it has been assumed that the initial density ρ^o is constant and that the mesh spacing is initially uniform so that $x_j^o - x_{j-1}^o$ is a constant Δx. Using (4.2) in (4.1) one obtains

$$u_j^{n+1/2} = u_j^{n-1/2} + K(p_{j-1/2}^n - p_{j+1/2}^n + q_{j-1/2}^n - q_{j+1/2}^n) \qquad (4.3)$$

where $K = \dfrac{\Delta t}{\rho^o \Delta x}$. Putting (2.12) into (3.4) one gets

$$q_{j-1/2}^{n+1} = 0 \text{ if } \rho_{j-1/2}^{n+1} \leq \rho_{j-1/2}^n \qquad (4.4)$$

$$= \rho_{j-1/2}^{n+1} \left\{ B_2 (x_j^{n+1} - x_{j-1}^{n+1})\, c_{j-1/2}^n \, \frac{2(\rho_{j-1/2}^{n+1} - \rho_{j-1/2}^n)}{\Delta t(\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^n)} \right.$$

$$\left. + \left[B_1 (x_j^{n+1} - x_{j-1}^{n+1}) \, \frac{2(\rho_{j-1/2}^{n+1} - \rho_{j-1/2}^n}{\Delta t(\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^n)} \right]^2 \right\} \text{ otherwise.}$$

The pressure p is assumed to depend only on the density so the equation of state (4.28) becomes $p = f(\rho)$. In the computation this becomes

$$p_{j-1/2}^{n+1} = f(\rho_{j-1/2}^{n+1}) \tag{A5}$$

Equation (2.9) will also be used and is listed here for convenience:

$$x_j^{n+1} = x_j^n + \Delta t\, u_j^{n+1/2} \tag{A6}$$

The specific volume $v = 1/\rho$ is more convenient to work with than the density. In what follows it will be assumed that the computed values of ρ are positive. Heuristic arguments can be made in support of this. In any case negative values for ρ are physically meaningless. In terms of v equation (A2) can be written

$$v_{j-1/2}^{n+1} = \frac{x_j^{n+1} - x_{j-1}^{n+1}}{\rho^0 \Delta x}$$

and, with the aid of (A6), this becomes

$$v_{j-1/2}^{n+1} = v_{j-1/2}^n + K(u_j^{n+1/2} - u_{j-1}^{n+1/2}) \tag{A7}$$

(A5) becomes

$$p_{j-1/2}^{n+1} = g(v_{j-1/2}^{n+1}) \tag{A8}$$

where $g(z) = f(1/z)$.

(A4) can be written

$$g_{j-1/2}^{n+1} = \alpha(v_{j-1/2}^n - v_{j-1/2}^{n+1}) \tag{A9}$$

where $\alpha = \dfrac{B_2 c^{n}_{j-1/2}}{K\left(\dfrac{v^{n}_{j-1/2} + v^{n+1}_{j-1/2}}{2}\right)} + \dfrac{B_1^2 v^{n+1}_{j-1/2}\left(v^{n}_{j-1/2} - v^{n+1}_{j-1/2}\right)}{K^2\left(\dfrac{v^{n}_{j-1/2} + v^{n+1}_{j-1/2}}{2}\right)^2}$

if $v^{n}_{j-1/2} > v^{n+1}_{j-1/2}$

$\alpha = 0$ otherwise.

Equations (A3), (A7), (A8), and (A9) are the equations which will be analyzed. They prescribe how u, v, p, and g are to be computed at the $(n+1)^{st}$ time step when their values at the n^{th} time step are known.

For linear equations with constant coefficients techniques for making a stability analysis can be found in Richtnyer's book.[5] For more complicated situations where it may be impossible to make a rigorous stability analysis he suggests replacing the equations by analogous equations which can be analyzed. The result provides at least a tentative stability criterion which can be tested with the original system of equations.

(A3) and (A7) are linear equations with constant coefficients if the ratio $\Delta t/\Delta x$ is fixed. The other two are not; however they can be replaced by related equations which are linear. If α is regarded as a constant then (A9) becomes a linear equation. (A8) is replaced by a linear equation with the aid of Taylor's expansion:

$$p^{n+1}_{j-1/2} = p^{n}_{j-1/2} + g' {}^{n}_{j-1/2}\left(v^{n+1}_{j-1/2} + v^{n}_{j-1/2}\right)$$

Now

$$g' {}^{n}_{j-1/2} = -f' \frac{\rho^{n}_{j-1/2}}{\left(v^{n}_{j-1/2}\right)^2} = \frac{c^{n}_{j-1/2}}{v^{n}_{j-1/2}}^2$$

where c is the speed of sound, see (4.8) and (4.7).

Hence equation (A8) is replaced by

$$p^{n+1}_{j-1/2} = p^{n}_{j-1/2} - N(v^{n+1}_{j-1/2} - v^{n}_{j-1/2}) \tag{A10}$$

where $N = (c^{n}_{j-1/2}/v^{n}_{j-1/2})^2$. If N and α are regarded as constants, the equations (A3), (A7), (A9), and (A10) can be analyzed by the techniques discussed in Richtmyer's book[5]. The so-called amplification matrices can easily be found. They are given by

$$G(\Delta t,k) = \begin{Bmatrix} 1 & 0 & -i\tau & -i\tau \\ i\tau & 1 & \tau^2 & \tau^2 \\ -\alpha i\tau & 0 & -\alpha\tau^2 & -\alpha\tau^2 \\ -Ni\tau & 0 & -N\tau^2 & 1-N\tau^2 \end{Bmatrix}$$

where $\tau = 2K \sin\frac{k\Delta x}{2}$ and $i = \sqrt{-1}$. The eigenvalues λ of G satisfy the equation

$$(\lambda - 1)\lambda\,[(\lambda - 1)^2 + \tau^2(\alpha + N)(\lambda - 1) + \tau^2 N] = 0$$

The von Neumann criterion for stability is satisfied if $|\lambda| \le 1$ for all the roots for each integer k if Δt is sufficiently small, $\lambda = 0$ and $\lambda = 1$ are roots, and the other roots are given by

$$\lambda = \frac{2 - \tau^2(\alpha + N) \pm \sqrt{[\tau^2(\alpha + N)]^2 - 4\tau^2 N}}{2}$$

Since α and N are both nonnegative, it can be shown that $|\lambda| \le 1$ if and only if

$$\tau^2(2\alpha + N) \leq 4 \text{ or } 4K^2(2\alpha + N)\sin^2\frac{k\Delta x}{2} \leq 4.$$

This must hold for every integer k for all Δx sufficiently small, so the stability condition becomes $K^2(2\alpha + N) \leq 1$.

Now, using (A7), α can be written

$$\alpha = \frac{B_2 c^{n}_{j-1/2}}{Kv^*} + \frac{B_1^2 v^{n+1}_{j-1/2} |\Delta u|}{Kv^{*2}} \qquad \text{if } \Delta u < 0$$

$$= 0 \qquad \text{otherwise}$$

where

$$v^* = \frac{v^{n+1}_{j-1/2} + v^{n}_{j-1/2}}{2},$$

and

$$\Delta u = u^{n+1/2}_{j} - u^{n+1/2}_{j-1}.$$

Substituting α in N into the stability condition one obtains

$$K\frac{2B_2 c}{v^*} + \frac{2B_1^2 v|\Delta u|}{v^{*2}} + K\frac{c^2}{v^2} \leq 1$$

where $c = c^{n}_{j-1/2}$, $v = v^{n}_{j-1/2}$, and $v^{n+1}_{j-1/2}$ has been approximated by v. Since $K \geq 0$, the inequality is satisfied if and only if

$$K \leq \frac{1}{\frac{B_2 c}{v^*} + \frac{B_1^2 v|\Delta u|}{v^{*2}} + \sqrt{\left(\frac{B_2 c}{v^*} + \frac{B_1^2 v|\Delta u|}{v^{*2}}\right)^2 + \frac{c^2}{v^2}}}$$

If v^* is approximated by v, this can be written

$$\Delta t \leq \frac{\rho^\circ \Delta x v}{B_2 c + B_1^2 |\Delta u| + \sqrt{(B_2 c + B_1^2 |\Delta u|)^2 + c^2}}$$

Finally, using (A2), $\rho^\circ \Delta x v^n_{j-1/2} = x^n_j - x^n_{j-1}$ so the stability condition can be written

$$\Delta t \leq \frac{x^n_j - x^n_{j-1}}{B_2 c + B_1^2 |\Delta u| + \sqrt{(B_2 c + B_1^2 |\Delta u|)^2 + c^2}}$$

This is the condition when $\Delta u < 0$. When $\Delta u \geq 0$ the artificial viscosity is zero and the stability condition becomes

$$\Delta t \leq \frac{x^n_j - x^n_{j-1}}{c}$$

This is the classical stability condition for the equations of hydrodynamics. It says that Δt must be small enough that a disturbance cannot propagate from one mesh point to another in time Δt.

The Δu which appears in the stability criterion is, by definition, $u_j^{n+1/2} - u_{j-1}^{n+1/2}$. Since this quantity is not available until Δt has been chosen, Δu is approximated by $u^{n-1/2}_{j-1/2}$.

In the actual computation the stability criterion is evaluated for each j. The Δt used to advance to the next time step is chosen to be at least as small as the minimum of these.

APPENDIX B

- LIST OF SUBROUTINES AND TAPE UNITS -

These subroutines must be supplied whenever the program is run.

WONDY	Main Program.
GENERATE	Reads Input and Initializes Arrays.
MORSTORE	Initializes Extra Arrays - May Be Dummy.
BOUNDARY	Handles Special Boundary - May Be Dummy.
JLOOPING	Handles Logic to Advance Through Meshes.
MOTION	Computes Conservation of Mass and Momentum.
STATE1	Compute Equations of State - Up to 5 May Be Dummies.
STATE2	
STATE3	
STATE4	
STATE5	
STATE6	
OUTPUT	Handles Special Output. - May be Dummy
OUTL	Handles Special Output - May Be Dummy.

These logical tape units must be defined whenever the program is run.

20 Dump on binary tape, includes: N, T, DELT(1), DELT(4), (ITABLE(K), TABLE(1,K), TABLE(2,K), K=1,20), (STORE(J), J=1,JM)
Where JM=(LMAX+1)*NVAR.

25 Restart on binary tape, using same order of information as Tape 20.

21 BCD Tape of all regular output, including listing of input data, error messages, and information normally written in the normal edit. This tape may be equivalenced to the system output medium.

22 BCD or Binary Tape of all information written in the sample Subroutine OUTL.

23 BCD or Binary Tape of all information written in the Sample Subroutine OUTPUT.

APPENDIX C

GLOSSARY OF VARIABLE NAMES

VARIABLES IN COMMON

Fortran Name	Type	Description	
A	R	a_j^n	acceleration
ADDATA(14)	R		dummy array for additional input data
B1	R	B_1	quadratic viscosity coefficient
B11	R		$4 B_1^2 + 1$
B2	R	B_2	linear viscosity coefficient
B22	R		$2 B_2 + 1$
CAPE	R	$E_{j-1/2}^{n+1/2}$	internal energy
CAPH	R	$H_{j-1/2}^{n+1/2}$	momentum
CAPK	R	$K_{j-1/2}^{n+1/2}$	kinetic energy
C(31000)	R	$c_{j-1/2}^n$	sound speed
CES(42,20)	R		equation of state constants per plate
DELE	R		$\Delta^d \mathcal{E}$
DELRHO	R		$\Delta\rho / 2\rho^2$
DELRJ	R		$2\left(\rho_{j-1/2}^{n+1} - \rho_{j-1/2}^{n}\right) \Big/ \left(\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^{n}\right)$
DELT(4)	R		$\Delta t^{n+1/2}$; $\Delta t^{n-1/2}$; $\Delta t_{j-1/2}^{n+1/2}$; $\Delta t^{n+3/2}$
DELTAX(20)	R		initial mesh size per plate
DELXJ	R		$X_j^{n+1} - X_{j-1}^{n+1}$
DEP	R		$\Delta t^{n+1/2} / \text{tdep}$

Fortran Name	Type	Description	
E(31000)	R	$\mathcal{E}^{n}_{j-1/2}$	internal energy
E1	R	$\mathcal{E}^{n+1}_{j-1/2}$	latest calculation of E(J)
EERROR	R		energy error, ET-ETOT
ET	R		total present energy
ETOT	R		total initial energy plus added energy
EXIT	I		exit indicator
EZERO(20)	R		initial energy in each mesh per plate
GOIND	I		computed go to index in MOTION
HERROR	R		momentum error, HT-HTOT
HT	R		total present momentum
HTOT	R		total initial momentum
IND	I		indicates an interface
ITABLE(50)	I		storage for mesh numbers at fractures
J	I		index for STORE arrays
JONE	I		indicates first mesh
JTAPE	I		number of data in STORE on tape
K	I		index for TABLE array
KE	R		shutoff value for energy error
KH	R		shutoff value for momentum error
KM(3)	R		symmetry constants 1; π; $\frac{4}{3}\pi$
KT1	R		time constant in stability criterion
KT2	R		maximum increase in time step per cycle
L	I		mesh number

Fortran Name	Type	Description	
LACT	I		activity test
LHBT	I		left-hand boundary type
LMAX	I		maximum number of meshes -1
LPHA	I	α	symmetry coefficient
M(31000)	R	$m_{j-1/2}$	mesh constant
N	I	n	cycle number
NIL	I		not used
NUL	I		not used
NOAD	I		number of addition data on Card 8
NOMESHES(20)	R		number of meshes per plate
NONE	I		indicates first cycle
NOP	I		number of plates
NOPM	I		number of PMESH on Card 4
NOQM	I		number of QMESH on Card 3
NSTART	I		restart from dump tape at this cycle
NTWO	I		indicates first cycle after restart
NVAR	I		number of variables in STORE array $\geq$ 10
P(31000)	R	$p^{n}_{j-1/2}$	pressure
P1	R	$p^{n+1}_{j-1/2}$	latest calculation of P(J)
PFRACT(3100)	L		indicates fracture which has rejoined
PHI(31000)	R	$\varphi^{n}_{j-1/2}$	difference in principal stresses
PHIZERO(20)	R		initial value of φ
PLATE	I		indicates plate number

Fortran Name	Type	Description	
PMESH(50)	I		mesh number for fracture which has rejoined
PRINTL	I		indicates call OUTL
PRINTR	I		indicates call editing
PRINTS	I		indicates call OUTPUT
PZERO(20)	R		initial pressure in each mesh per plate
Q(31000)	R	$q^{n}_{j-1/2}$	artificial viscosity
Q1	R	$q^{n+1}_{j-1/2}$	latest calculation of Q(J)
QFRACT(3100)	L		indicates fracture
QMESH(50)	I		mesh number for fracture
RHBT	I		right-hand boundary type
RHO(31000)	R	$\rho^{n}_{j-1/2}$	density
RHO1	R	$\rho^{n+1}_{j-1/2}$	latest calculation for RHO(J)
RHODOT	R		$\dot{\rho}/\rho$
RHOZERO(20)	R		initial density of each mesh per plate
SIGMA(31000)	R	$\sigma^{n+1}_{j-1/2}$	stress
SIGMAACT	R		stress used in activity test
SIGMAF(20)	R		fracture stress per plate
SIGMAIF(20)	R		fracture stress at interface
SIGMAL	R		left boundary stress
SIGMAMAX	R		maximum stress
SIGMAP	R	$\sigma^{n}_{j-1/2}$	previous stress
SIGMAR	R		right boundary stress
SIGMASEP	R		separation stress
SIGZERO(20)	R		initial stress in each mesh per plate

Fortran Name	Type	Description	
STATE(20)	R		equation of state type per plate
STORE(31100)	R		STORE array in BANK 1
SUMH	R		total momentum
SUMIE	R		total internal energy
SUMKE	R		total kinetic energy
SUMQE	R		total energy sources
T	R	t	time
TABLE(2,50)	R		storage of U and X at fractures
TDEP	R		time duration of energy sources
TDUMP	R		time interval between dumps
TITLE(10)	A		title of run
TMAX	R		maximum time
TMAXD	R		time of last dump
TMAXP	R		time of last edit
TMAXPL	R		time oflast special output via OUTL
TMAXPS	R		time of last special output via OUTPUT
TMIND	R		time of first dump
TMINP	R		time of first edit
TMINPL	R		time of first special output via OUTL
TMINPS	R		time of first special output via OUTPUT
TPRINT	R		time interval between edits
TPRINTL	R		time interval between special output via OUTL
TPRINTS	R		time interval between special output via OUTPUT
U(31000)	R	$u_j^{n-1/2}$	velocity
U1	R	$u_j^{n+1/2}$	latest calculation of U(J)
UZERO(20)	R		initial velocity per plate

Fortran Name	Type	Description	
UZEROI(20)	R		initial velocity at interface
W4020			indicates write tape 21
WL	R		work at left boundary
WR	R		work at right boundary
X(31000)	R	x_j^n	position of mesh boundary
X1	R	x_j^{n+1}	latest calculation of X(J)
XP	R	x_{j-1}^n	former position of previous mesh
XGAP(20)	R		initial distance between plates
XRATIO(20)	R		ratio between successive mesh sizes
XZERO	R		initial position of left boundary

VARIABLES IN JLOOPING ONLY

Fortran Name	Type	Description
LOR	I	mesh on right interface of plate
PHIE	R	temporary storage of φ at a fracture
QE	R	temporary storage of q at a fracture
RHOE	R	temporary storage of ρ at a fracture
SIGMAA	R	average stress
SIGMAE	R	temporary storage of σ at a fracture
TEST	R	test stress for fractures
UE	R	temporary storage of u at a fracture
XE	R	temporary storage of x at a fracture
XPE	R	temporary storage of XP at a fracture

VARIABLES IN STATE1 (HVEP) ONLY:

Fortran Name	Type	Description	
A	R		intermediate quantities in vapor equation
AP	R		
B	R		
BP	R		
DEP*	R		$\Delta t^{n+1/2}/\mathrm{Tdep}$
DX	R	d_x^d	stretching deviator
DZ	R	d_z^d	stretching deviator
ETA	R	η	$1 - \rho_o/\rho_{j-1/2}^{n+1}$
ETA1	R	η'	$1 - 2\rho_o/\left(\rho_{j-1/2}^{n+1} + \rho_{j-1/2}^{n}\right)$
ETAP	R		intermediate quantity in sums
F1	R	f_1	
F2	R	f_2	
FP1	R	f_1'	
FP2	R	f_2'	
G	R	G	shear modulus
GAMMA	R	Γ	Grueneisen ratio
GAMMAP	R	$d\Gamma/d\eta$	
GCONST(20)	R		$(1 - 2\nu)/(2 - 2\nu)$
KCONST(20)	R		$3(1 - \nu)/(1 + \nu)$
MU	R	μ	$\rho/\rho_o - 1$
NCONST(20)	R		$(c_o)^2/(\Gamma_o \mathcal{E}_s)$
PE	R		temporary storage for P1

Fortran Name	Type	Description	
PH	R	P_H	reference pressure
PHP	R	$dP_H/d\eta$	
QDEP	R	$\Delta Q_{J-1/2}$	energy source strength
RTMU	R		$\sqrt{\mu + 1}$
SPEC(31000)*	R		special array for energy sources
SUMDEP*	R		Σ DEP ≤ 1
SUMG	R		intermediate quantities in polynomial expansions
SUMGAM	R		
SUMGAP	R		
SUMPH	R		
SUMPHP	R		
TX	R	σ_x^d	stress deviator
TXP	R		stress deviator of previous cycle
TZ	R	σ_z^d	stress deviator
Y	R	Y	flow stress
YIELDF	R	f_y	yield function

* These quantities used only with Energy Absorption Program in MORSTORE.

APPENDIX D

INPUT INSTRUCTIONS

Normal Input Cards

CARD 1 FORMAT(10A8)

TITLE - date, run title, run number.

CARD 2 FORMAT(14I5)

LPHA - Symmetry Coefficient
- 1 rectangular symmetry
- 2 cylindrical symmetry
- 3 spherical symmetry

NOP - number of layers, maximum of 20.

NVAR - number of variables in STORE array, normally 10. ($10 \leq NVAR \leq 100$.)

LHBT - left-hand boundary type
- 1 fixed boundary
- 2 free boundary
- 3 calls SUBROUTINE BOUNDARY

RHBT - right-hand boundary type, same as above.

LACT - mesh number to begin activity test.

JTAPE* - number of words of STORE to be read in from a binary dump tape for restart. Set to zero if not restarting from tape.

NSTART* - cycle number to be read in from a binary dump tape for restart.

NOQM* - number of QMESH data to be read in, maximum of 50. If zero, omit CARD 3.

NOPM* - number of PMESH data to be read in, maximum of 50. If zero, omit CARD 4.

---

*Used for restarting from binary dump tape only. If not restarting from tape, set to zero.

NOAD - number of ADDATA data to be read in, maximum of 14. If zero, omit CARD 9.

W4020 - set = 0 if equivalencing unit 21 to 61.
set = 1 if writing output on user tape 21.

NIL not used, but included for user's convenience

NUL for including additional input indicators.

CARD 3* FORMAT(14I5)

QMESH - mesh numbers where there are fractures. Omit CARD 3 if NOQM = 0.

CARD 4* FORMAT(14I5)

PMESH - mesh numbers where fractures have rejoined. Omit CARD 4 if NOPM = 0.

CARD 5 FORMAT(7E10.3)

XZERO - initial position of the left boundary.

B1 - quadratic viscosity coefficient, usually 2.0.

B2 - linear viscosity coefficient, usually 0.1.

KE - energy error, if exceeded - program calls exit. Set = 0 to avoid using this feature.

SIGMAACT - activity is tested for this value of stress. If stress is less than SIGMAACT at a mesh, further meshes are not computed for that time cycle.

SIGMAMAX - maximum stress, if exceeded - program calls exit.

SIGMASEP - separation stress for a mesh which has already fractured, set equal to roundoff of SIGMA.

CARD 6 FORMAT(7E10.3)

KT1 - constant used in stability criterion, usually 1.0.

KT2 - maximum allowable increase in time step, usually 1.1 or 1.2.

---

*If there are more than 14 of these quantities, use more cards.

TMAX - maximum time, program has "normal exit" when this is exceeded.

TMIND - the time to write the first binary dump on tape 20. Set greater than TMAX if tape is not desired.

TDUMP - the time increment for additional binary dumps on tape 20. Set to 0.0 to dump every time cycle.

TMAXD - the time of the last binary dump on tape 20.

DELT(4) - maximum initial time step.

CARD 7 FORMAT(7E10.3)

TMINP - the first time to call EDIT. Set greater than TMAX to avoid calling EDIT.

TPRINT - the time increment for additional calls to EDIT. SET = 0.0 to call EDIT every time cycle.

TMAXP - the final time to call EDIT.

TMINPS

TPRINTS - same as above, for OUTPUT

TMAXPS

KH - momentum error, if exceeded - program calls exit. Set = 0 to avoid using this feature.

CARD 8 FORMAT(7E10.3)

TMINPL

TPRINTL - same as card 7, for OUTL

TMAXPL

CARD 9* FORMAT(7E10.3)

ADDATA - additional data may be added here. This array may be used for input to subroutines BOUNDARY, OUTPUT, OUTL Omit CARD 9 if NOAD = 0.

---

* If there are more than 7 of these quantities, use as many cards as required.

For Layer 1:

CARD 10 FORMAT(7E10.3)

NOMESHES - number of meshes in this layer.

STATE - indicator for equation of state to be used for this layer.

DELTAX - initial mesh size for this layer.

XRATIO - ratio of successive mesh sizes. Set = 1.0 for constant mesh size.

XGAP - distance between right boundary of this layer and left boundary of the next. Set = 0.0 if no gap is desired.

UZERO - initial velocity for this layer.

UZEROI - initial velocity of interface between this layer and the next to the right

CARD 11 FORMAT(7E10.3)

RHOZERO - initial density in this layer.

PZERO - initial pressure in this layer.

SIGZERO - initial stress in this layer

EZERO - initial energy in this layer.

PHIZERO - initial value of ϕ in this layer.

SIGMAF - fracture stress in this layer.

SIGMAIF - fracture stress at interface between this layer and the next to the right.

CARDS 12 - 16 FORMAT(7E10.3)

CES - equation of state constants for this layer. There are 35 of these constants.

REPEAT CARDS 10 THROUGH 16 FOR EACH ADDITIONAL LAYER.

EQUATION OF STATE CONSTANTS FOR HVEP

1.	ρ_o	- density of uncompressed material
2.	c_o	- bulk sound speed of uncompressed material
3.	$\mathcal{E}_s$	- energy of sublimation
4.	σ_{min}	- minimum stress
5.		- must be left blank; used internally
6.	ν	- Poisson's ratio
7.	$H = \gamma - 1$ *	- where γ = ratio of specific heats of distended vapor
8.	NOK †	- number of K constants, including K_o
9.	K_o	- must be left blank; computed internally from c_o, ρ_o
10.	k_1	(if NOK = 0, s appears here)
11.	k_2	
12.	k_3	
13.	k_4	
14.	k_5	
15.	NOH †	- number of H constants, including Γ_o
16.	Γ_o †	(if NOH = 1, $\Gamma = \Gamma_o$)
17.	h_1	
18.	h_2	
19.	h_3	
20.	h_4	
21.	h_5	

---

* To suppress vaporization, set H = 0.

† For linear elastic material, set NOK = NOH = 1 NOG = 0 NOY = 2 Γ_o = 0.

22.	NOG*	- number of G constants, including G_0
23.	G_0	- must be left blank; computed internally from c_0, ν
24.	g_1	(if NOG = 0, G is computed from K and ν internally)
25.	g_2	
26.	g_3	
27.	g_4	
28	g_5	
29.	NOY*	- if NOY = 0, the material is a fluid with zero strength
30.	Y_0	- initial yield if NOY = 1, Y is constant at Y_0 if NOY = 2, yield strength is infinite if NOY = 3, yield strength varies
31.	y_1	- rate of increase of yield with compression. Use with NOY = 3
32.	y_2	- energy at which the yield strength vanishes. Use with NOY = 3
33.		
34.		
35.		

*For linear elastic material, set NOK = NOH = 1, NOG = 0, NOY = 2; Γ_0 = 0.

EQUATION OF STATE CONSTANTS FOR HE

1.	ρ_0	- density of solid explosive
2.	c_0	- initial sound speed
3.	γ	- ratio of specific heats
4.	D	- detonation wave velocity
5.	x_D	- detonation point
6.	B_5	- wave width constant
7. to 35.		- not used

EQUATION OF STATE CONSTANTS FOR GAS

1.	ρ_0	- initial density
2.	c_0	- initial sound speed
3.	γ	- ratio of specific heats
4. to 35.		- not used

Special Input Read by MORSTORE

The following cards follow the normal input cards.

Card 17 FORMAT(E10.3)

TDEP - deposition time in seconds

CARD 18* FORMAT(7E10.3)

SPECZERO(PLATE) PLATE = 1, NOP

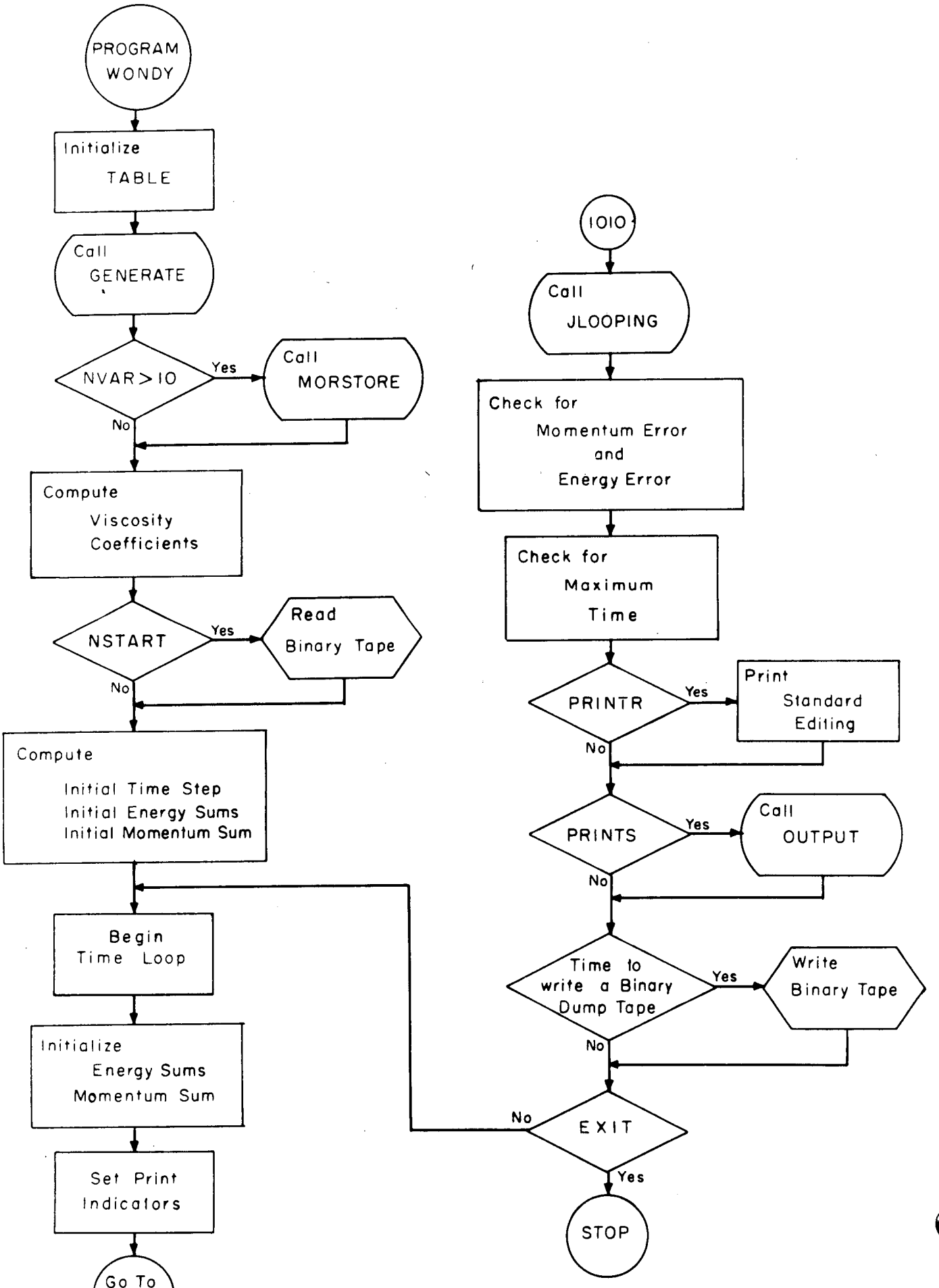
- energy deposited per mesh for each material layer

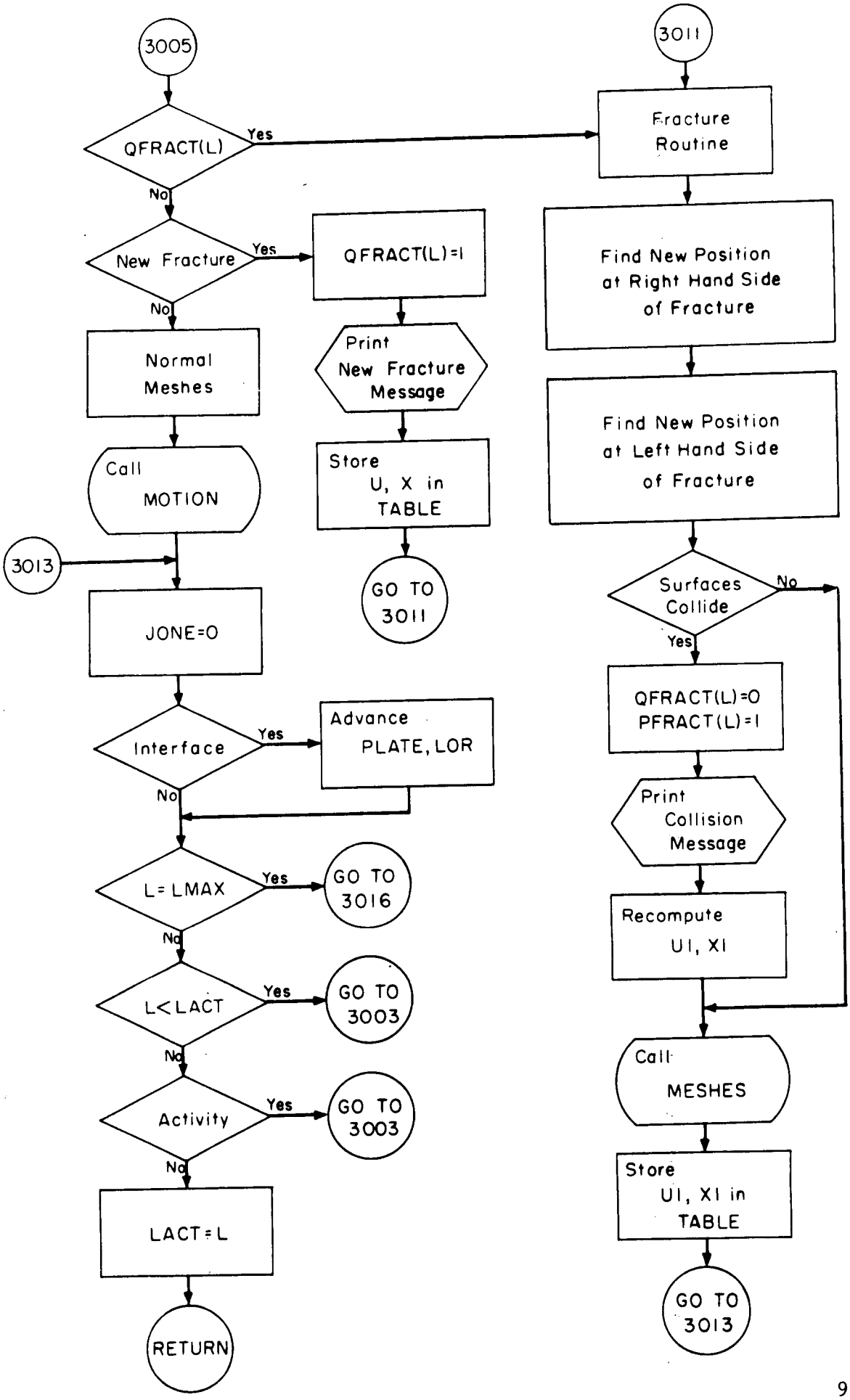
---

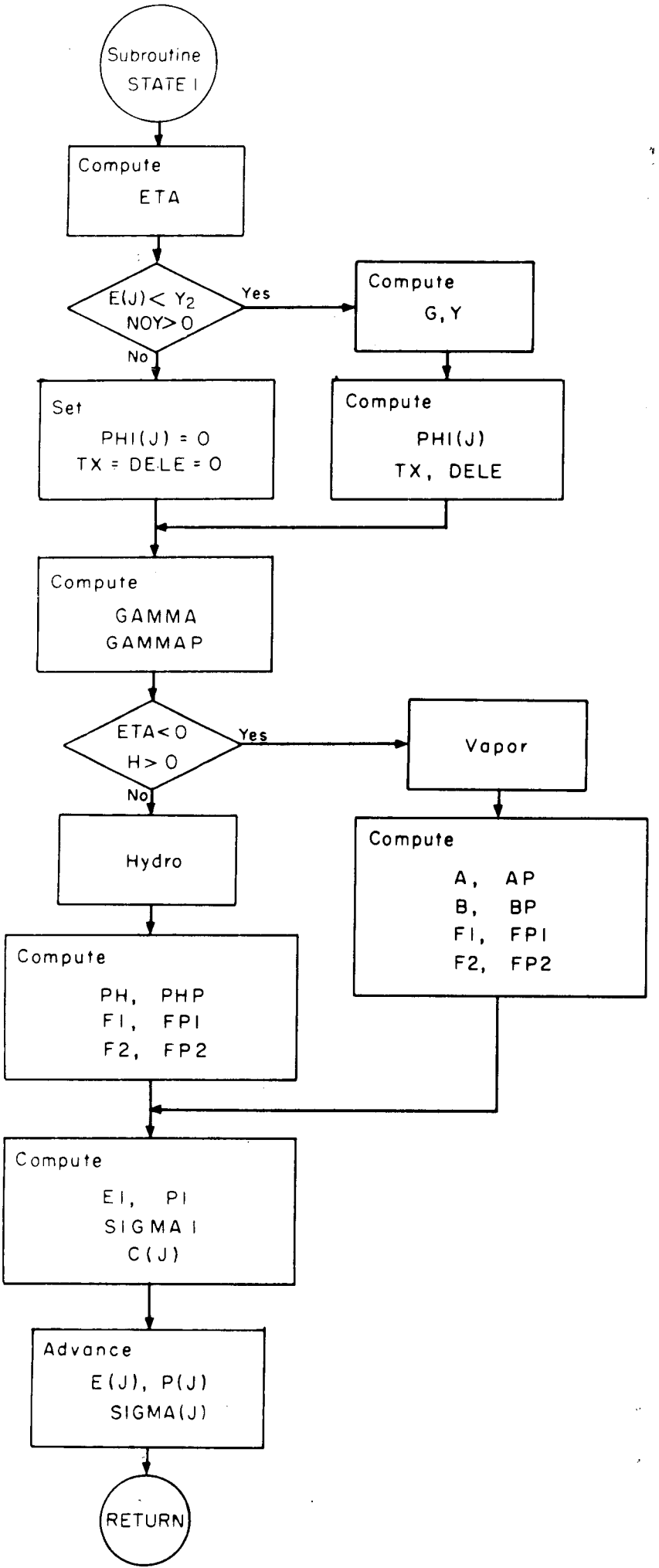
*If there are more than 7 layers, additional cards may be used.

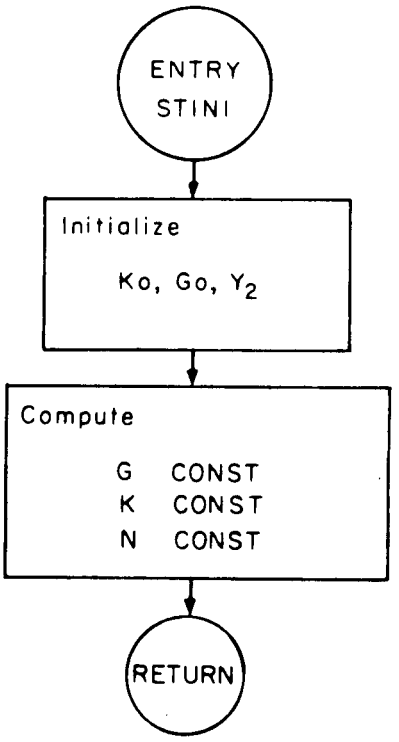
APPENDIX E

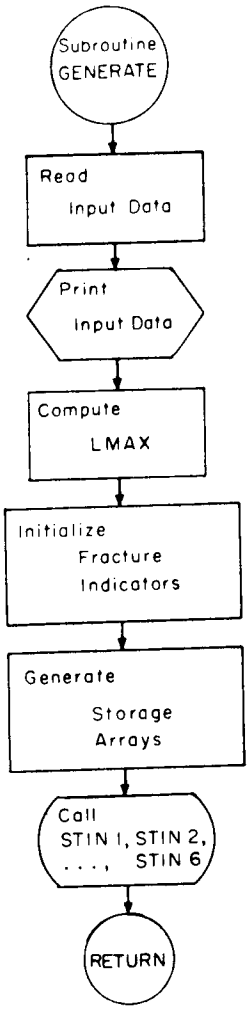
FLOW CHARTS

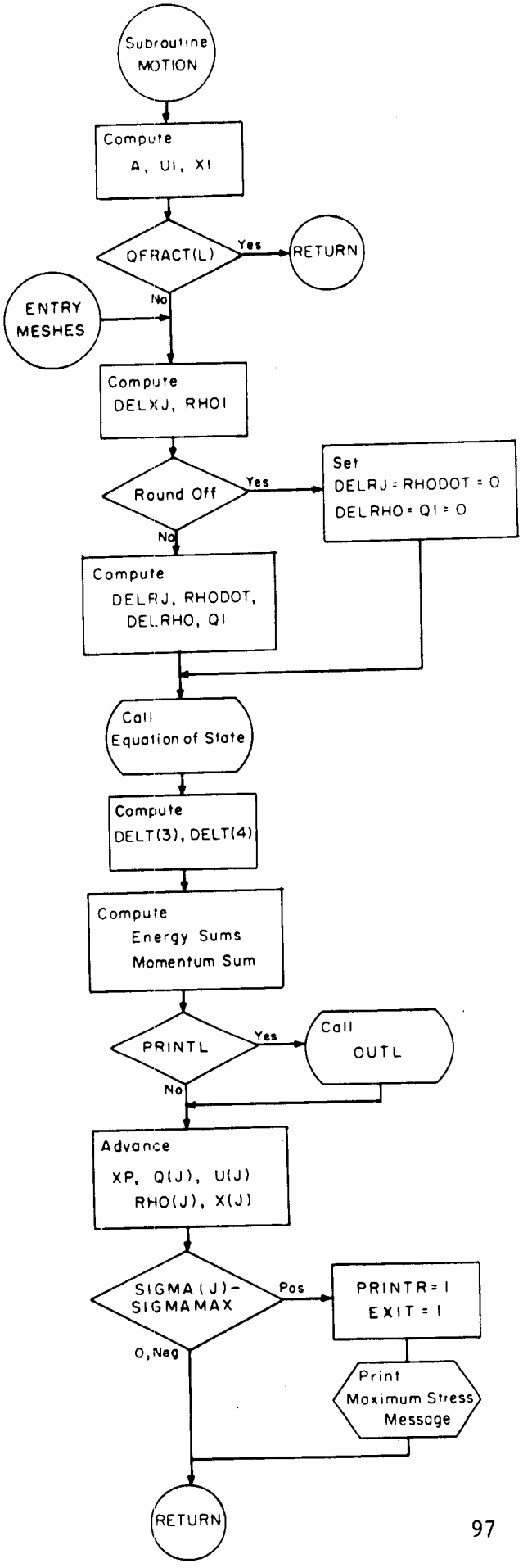

PROGRAM WONDY
Initialize TABLE
Call GENERATE
NVAR > 10
Yes
Call MORSTORE
No
Compute Viscosity Coefficients
NSTART
Yes
Read Binary Tape
No
Compute Initial Time Step Initial Energy Sums Initial Momentum Sum
Begin Time Loop
Initialize Energy Sums Momentum Sum
Set Print Indicators
Go To 1010
1010
Call JLOOPING
Check for Momentum Error and Energy Error
Check for Maximum Time
PRINTR
Yes
Print Standard Editing
No
PRINTS
Yes
Call OUTPUT
No
Time to write a Binary Dump Tape
Yes
Write Binary Tape
No
EXIT
No
Yes
STOP

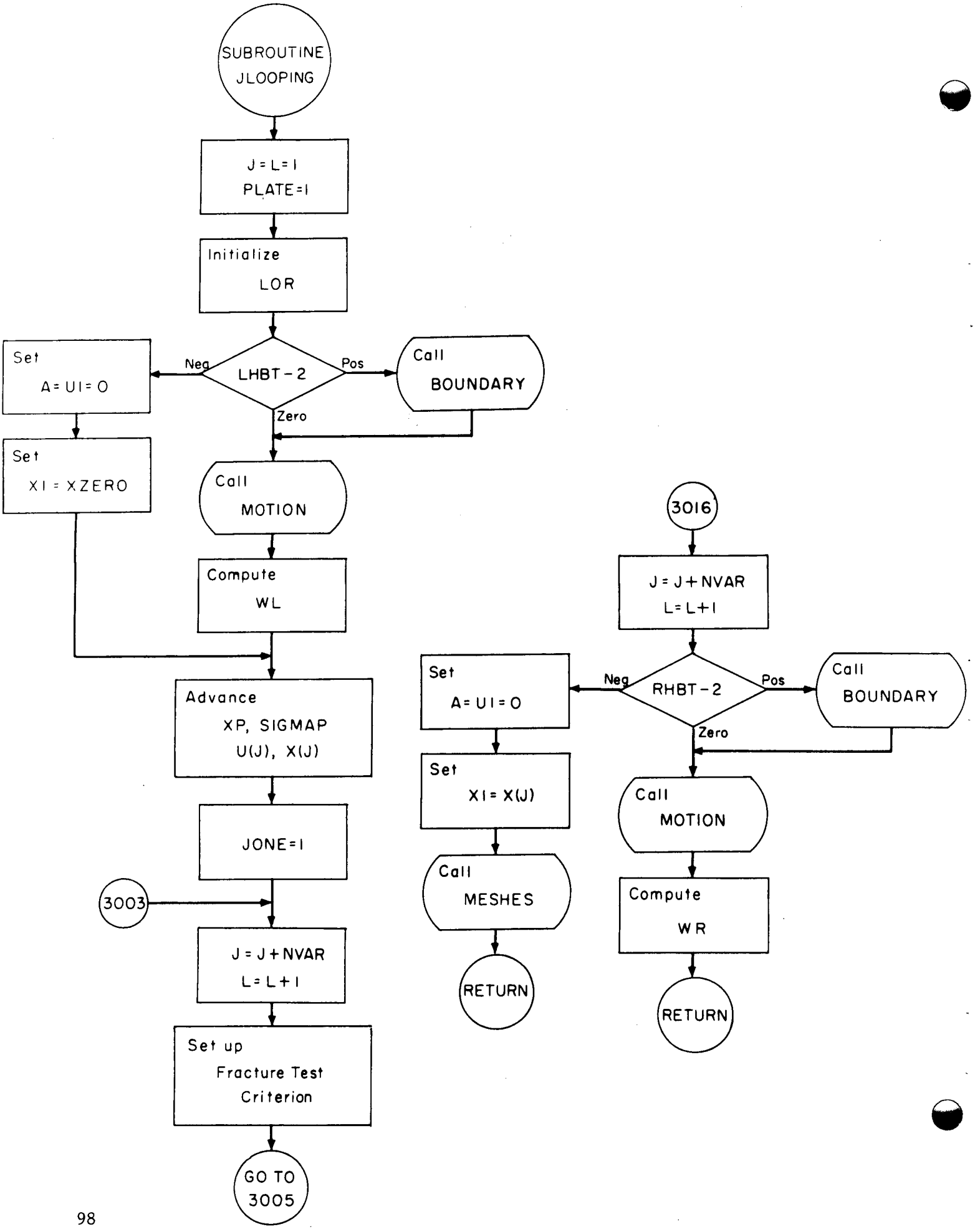

3005
QFRACT(L)
Yes
No
New Fracture
Yes
No
QFRACT(L)=1
Print
New Fracture
Message
Store
U, X in
TABLE
GO TO
3011
Normal
Meshes
Call
MOTION
3013
JONE=0
Interface
Yes
No
Advance
PLATE, LOR
L= LMAX
Yes
No
GO TO
3016
L<LACT
Yes
No
GO TO
3003
Activity
Yes
No
GO TO
3003
LACT=L
RETURN
3011
Fracture
Routine
Find New Position
at Right Hand Side
of Fracture
Find New Position
at Left Hand Side
of Fracture
Surfaces
Collide
No
Yes
QFRACT(L)=0
PFRACT(L)=1
Print
Collision
Message
Recompute
UI, XI
Call
MESHES
Store
UI, XI in
TABLE
GO TO
3013


Subroutine
STATE I
Compute
ETA
E(J) < Y2
NOY > 0
Yes
No
Compute
G, Y
Set
PHI(J) = 0
TX = DELE = 0
Compute
PHI(J)
TX, DELE
Compute
GAMMA
GAMMAP
ETA < 0
H > 0
Yes
No
Vapor
Hydro
Compute
A, AP
B, BP
F1, FP1
F2, FP2
Compute
PH, PHP
F1, FP1
F2, FP2
Compute
EI, PI
SIGMA I
C(J)
Advance
E(J), P(J)
SIGMA(J)
RETURN


ENTRY
STINI
Initialize
Ko, Go, Y2
Compute
G CONST
K CONST
N CONST
RETURN


Subroutine GENERATE
Read Input Data
Print Input Data
Compute LMAX
Initialize Fracture Indicators
Generate Storage Arrays
Call STIN 1, STIN 2, . . . , STIN 6
RETURN


Subroutine MOTION
Compute A, UI, XI
QFRACT(L)
Yes
RETURN
No
ENTRY MESHES
Compute DELXJ, RHOI
Round Off
Yes
Set DELRJ = RHODOT = 0 DELRHO = QI = 0
No
Compute DELRJ, RHODOT, DELRHO, QI
Call Equation of State
Compute DELT(3), DELT(4)
Compute Energy Sums Momentum Sum
PRINTL
Yes
Call OUTL
No
Advance XP, Q(J), U(J) RHO(J), X(J)
SIGMA(J)- SIGMAMAX
Pos
PRINTR = 1 EXIT = 1
0, Neg
Print Maximum Stress Message
RETURN


SUBROUTINE
JLOOPING
J = L = 1
PLATE = 1
Initialize
LOR
Set
A = U1 = 0
Neg
LHBT - 2
Pos
Call
BOUNDARY
Zero
Set
X1 = XZERO
Call
MOTION
Compute
WL
Advance
XP, SIGMAP
U(J), X(J)
JONE = 1
3003
J = J + NVAR
L = L + 1
Set up
Fracture Test
Criterion
GO TO
3005
3016
J = J + NVAR
L = L + 1
Set
A = U1 = 0
Neg
RHBT - 2
Pos
Call
BOUNDARY
Zero
Set
X1 = X(J)
Call
MESHES
RETURN
Call
MOTION
Compute
WR
RETURN

APPENDIX F

FORTRAN LISTINGS

```
      COMMON /CONSTS/ B1, B11, B2, B22, EXIT, IND, J, JONE, LACT, LMAX, WONDY  5
     1JTAPE, KE, KH, KM(3), KT1, KT2, L, LHBT, LPHA, N, NOAD, NONE, NOP,WONDY  6
     2NOPM, NOQM, NSTART, NTWO, NVAR, PRINTR, PRINTS, RHBT, SIGMAACT,   WONDY  7
     3SIGMAMAX, SIGMASEP, T, TDEP, TDUMP, TITLE(10), TMAX, TMAXD, TMAXP,WONDY  8
     4TMAXPS, TMIND, TMINP, TMINPS, TPRINT, TPRINTS, W4020, XZERO        WONDY  9
     5, NIL, NUL, LOL, LOR, TMINPL, TPRINTL, TMAXPL, PRINTL              WONDY 10
      COMMON /INTERM/ A, ADDATA(14), CAPE, CAPH, CAPK, DELE, DEP, DELRHOWONDY 11
     1, DELRJ, DELT(4), DELXJ, E1, EERROR, ET, ETOT, GOIND, HERROR, HT, WONDY 12
     2HTOT, ITABLE(50), K, P1, PMESH(50), Q1, QMESH(50), RHO1, RHODOT,  WONDY 13
     3SIGMAL, SIGMAP, SIGMAR, SUMH, SUMIE, SUMKE, SUMQE, TABLE(2,50),   WONDY 14
     4U1, WL, WR, X1, XP                                                WONDY 15
      COMMON /PLATES/ PLATE, CES(42,20), DELTAX(20), EZERF(20), NOMESHESWONDY 16
     1(20), PHIZERO(20), PZERO(20), RHOZERO(20), SIGMAF(20), SIGMAIF(20)WONDY 17
     2,SIGZERO(20), STATE(20), UZERO(20), UZEROI(20),XGAP(20),XRATIO(20)WONDY 18
      COMMON /LOGIC/ PFRACT(3100), QFRACT(3100)                          WONDY 19
      COMMON /ARRAYS/ STORE(31100)                                       WONDY 20
      EQUIVALENCE (STORE(10), X(10), U(9), SIGMA(8), RHO(7), Q(6),       WONDY 21
     1PHI(5), P(4), M(3), E(2), C(1))                                    WONDY 22
      DIMENSION C(31000), E(31000), M(31000), P(31000), PHI(31000),      WONDY 23
     1Q(31000), RHO(31000), SIGMA(31000), U(31000), X(31000)             WONDY 24
      TYPE INTEGER EXIT, GOIND, PLATE, PRINTR, PRINTS, PMESH, QMESH,RHBTWONDY 25
      TYPE REAL HERROR, HT, HTOT, KE, KH, KM, KT1, KT2, M, NOMESHES      WONDY 26
      TYPE LOGICAL PFRACT, QFRACT                                        WONDY 27
```

```
      PROGRAM WONDY                                                    WONDY  1
C     VERSION  II                                                      WONDY  2
      BANK,(1), /ARRAYS/                                               WONDY  3
      BANK, (0), WONDY     , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/     WONDY  4
C     INSERT COMMON CARDS HERE
      DATA ((KM(IC),IC=1,3)= 1.0, 3.1415926535, 4.1887902029 )         WONDY 28
C                                                                      WONDY 29
C     INITIALIZE INDICATORS                                            WONDY 30
      IND=0                                                            WONDY 31
      XP=0.0                                                           WONDY 32
      DO 1001 K=1,50                                                   WONDY 33
 1001 ITABLE(K)=TABLE(1,K)=TABLE(2,K)=0.0                              WONDY 34
      TDEP=0.0                                                         WONDY 35
      EXIT=NTWO=0                                                      WONDY 36
      CALL GENERATE                                                    WONDY 37
      ASSIGN 1002 TO IOUT                                              WONDY 38
      GO TO 1021                                                       WONDY 39
 1002 IF (NVAR.GT.10) CALL MORSTORE                                    WONDY 40
      B11=2.0*B1**2                                                    WONDY 41
      B22=2.*B2+1.                                                     WONDY 42
      N=T=0                                                            WONDY 43
      NONE=1                                                           WONDY 44
      SIGMAL=SIGMAR=0.0                                                WONDY 45
      IF (DELT(4).EQ.0.0) DELT(4)=1.0E05                               WONDY 46
      IF (NSTART) 1003, 1004                                           WONDY 47
C                                                                      WONDY 48
C     READ BINARY DUMP TAPE HERE                                       WONDY 49
 1003 L=JTAPE/NVAR                                                     WONDY 50
```

```
 1501 READ (25) N, T, DELT(1), DELT(4), (ITABLE(K), TABLE(1,K), TABLE(2,WONDY 51
     1K), K=1,50), (STORE(J), J=1, JTAPE)                              WONDY 52
      IF (N-NSTART) 1501, 1502, 1502                                   WONDY 53
 1502 PRINT 1701, N, L                                                 WONDY 54
      IF (W4020) WRITE (21, 1701) N, L                                 WONDY 55
      NTWO=1                                                           WONDY 56
      ASSIGN 1004 TO IOUT                                              WONDY 57
      GO TO 1021                                                       WONDY 58
C                                                                      WONDY 59
C     COMPUTE INITIAL TIME STEP AND INITIAL ENERGY SUMS                WONDY 60
 1004 SUMIE=SUMKE=SUMH=0.0                                             WONDY 61
      LM=LMAX+1                                                        WONDY 62
      J=1                                                              WONDY 63
      DO 1005 L=2,LM                                                   WONDY 64
      J=J+NVAR                                                         WONDY 65
      DELU=U(J)-U(J-NVAR)                                              WONDY 66
      IF (DELU) 1025, 1026, 1026                                       WONDY 67
 1025 BCBU=B2*C(J)-B1**2*DELU                                          WONDY 68
      DELT(3)=KT1*(X(J)-X(J-NVAR))/(BCBU+SQRTF(BCBU**2+C(J)**2))       WONDY 69
      GO TO 1027                                                       WONDY 70
 1026 DELT(3)=KT1*(X(J)-X(J-NVAR))/C(J)                               WONDY 71
 1027 IF (TDEP) DELT(3)=MIN1F(DELT(3),0.01*TDEP)                       WONDY 72
      DELT(4)=MIN1F(DELT(4),DELT(3))                                   WONDY 73
      CAPH=0.5*M(J)*(U(J)+U(J-NVAR))                                   WONDY 74
      CAPK=M(J)/8.0*(U(J)+U(J-NVAR))**2                                WONDY 75
      CAPE=M(J)*E(J)                                                   WONDY 76
      SUMH=SUMH+CAPH                                                   WONDY 77
```

```
      SUMIE=SUMIE+CAPE                                                  WONDY 78
 1005 SUMKE=SUMKE+CAPK                                                  WONDY 79
      SUMH=SUMH*KM(LPHA)                                                WONDY 80
      SUMIE=SUMIE*KM(LPHA)                                              WONDY 81
      SUMKE=SUMKE*KM(LPHA)                                              WONDY 82
      ETOT=SUMIE+SUMKE                                                  WONDY 83
      HTOT=SUMH                                                         WONDY 84
      PRINT 1201, SUMIE, SUMKE, ETOT, HTOT                              WONDY 85
      IF (W4020) WRITE (21, 1201) SUMIE, SUMKE, ETOT, HTOT              WONDY 86
      IF (KE.EQ.0.0) KE=1.0E100                                         WONDY 87
      IF (KH.EQ.0.0) KH=1.0E100                                         WONDY 88
      ASSIGN 1022 TO IOUT                                               WONDY 89
      DELT(2)=DELT(1)                                                   WONDY 90
      DELT(1)=DELT(4)                                                   WONDY 91
      IF (JTAPE) GO TO 1006                                             WONDY 92
      DELT(2)=DELT(1)                                                   WONDY 93
C                                                                       WONDY 94
C     BEGIN TIME LOOP                                                   WONDY 95
 1006 T=T+DELT(1)                                                       WONDY 96
      N=N+1                                                             WONDY 97
      PRINTR=PRINTS=0                                                   WONDY 98
      PRINTL=0                                                          WONDY 99
      SUMIE=SUMKE=SUMQE=WL=WR=0.0                                       WONDY100
      SUMH=0.0                                                          WONDY101
      DELT(4)=KT2*DELT(1)                                               WONDY102
C                                                                       WONDY103
C     SET PRINT INDICATORS                                              WONDY104
      IF (T-TMINPL) 1020, 1019, 1019                                    WONDY105
```

```
 1019 TMINPL=TMINPL+TPRINTL                                              WONDY106
      PRINTL=1                                                           WONDY107
      IF (TMINPL.GT.TMAXPL) TMINPL=2.0*TMAX                              WONDY108
 1020 IF (T-TMINP) 1008, 1007, 1007                                      WONDY109
 1007 TMINP=TMINP+TPRINT                                                 WONDY110
      PRINTR=1                                                           WONDY111
      IF (TMINP.GT.TMAXP) TMINP=2.0*TMAX                                 WONDY112
 1008 IF (T-TMINPS) 1010, 1009, 1009                                     WONDY113
 1009 TMINPS=TMINPS+TPRINTS                                              WONDY114
      PRINTS=1                                                           WONDY115
      IF (TMINPS.GT.TMAXPS) TMINPS=2.0*TMAX                              WONDY116
 1010 CALL JLOOPING                                                      WONDY117
C                                                                        WONDY118
C     COMPUTE MOMENTUM ERROR                                             WONDY119
      HT=SUMH=SUMH*KM(LPHA)                                              WONDY120
      HERROR=HT-HTOT                                                     WONDY121
      IF (ABSF(HERROR).GT.KH) 1017, 1018                                 WONDY122
 1017 PRINTR=EXIT=1                                                      WONDY123
C                                                                        WONDY124
C     MESSAGE SAYS MOMENTUM ERROR EXCEEDED                               WONDY125
      PRINT 1204, N, T                                                   WONDY126
      IF (W4020) WRITE (21, 1204) N, T                                   WONDY127
C                                                                        WONDY128
C     COMPUTE ENERGY ERROR, THE DIFFERENCE BETWEEN PRESENT AND INITIAL   WONDY129
C     ENERGIES                                                           WONDY130
 1018 SUMIE=SUMIE*KM(LPHA)                                               WONDY131
      SUMKE=SUMKE*KM(LPHA)                                               WONDY132
```

```
      SUMQE=SUMQE*KM(LPHA)                                          WONDY133
      ET=SUMIE+SUMKE                                                WONDY134
      ETOT=ETOT+SUMQE+WL+WR                                         WONDY135
      EERROR=ET-ETOT                                                WONDY136
      IF (ABSF(EERROR).GT.KE) 1011, 1012                            WONDY137
 1011 PRINTR=EXIT=1                                                 WONDY138
C                                                                   WONDY139
C     MESSAGE SAYS ENERGY ERROR EXCEEDED                            WONDY140
      PRINT 1202, N, T                                              WONDY141
      IF (W4020) WRITE (21, 1202) N, T                              WONDY142
 1012 IF (T-TMAX) 1014, 1014, 1013                                  WONDY143
 1013 PRINTR=EXIT=1                                                 WONDY144
C                                                                   WONDY145
C     MESSAGE SAYS NORMAL EXIT                                      WONDY146
      PRINT 1203, N, T                                              WONDY147
      IF (W4020) WRITE (21, 1203) N, T                              WONDY148
 1014 IF (PRINTR) 1021, 1022                                        WONDY149
 1021 CONTINUE                                                      WONDY150
C                                                                   WONDY151
C     BEGIN STANDARD EDITING                                        WONDY152
      J=1-NVAR                                                      WONDY153
      LEND=0                                                        WONDY154
 9000 LBEGIN=LEND+1                                                 WONDY155
      LEND=LEND+55                                                  WONDY156
      IF (LEND.GT.LACT) LEND=LACT                                   WONDY157
      WRITE (21, 9101) N, T, DELT(1)                                WONDY158
      WRITE (21, 9102)                                              WONDY159
```

```
      DO 9002 L=LBEGIN, LEND                                              WONDY160
      J=J+NVAR                                                            WONDY161
      WRITE (21, 9103) L, X(J), U(J), RHO(J), SIGMA(J), PHI(J), Q(J),     WONDY162
     1E(J), C(J)                                                          WONDY163
 9002 CONTINUE                                                            WONDY164
      IF (LEND.EQ.LACT) 9003, 9000                                        WONDY165
 9003 WRITE (21, 9105) WL, WR, SUMIE, SUMKE, SUMQE, SUMH, EERROR, HERRORWONDY166
      IF (ITABLE(1)) WRITE (21, 9106)                                     WONDY167
      DO 9004 K=1,50                                                      WONDY168
      IF (ITABLE(K).EQ.0) GO TO 9005                                      WONDY169
      WRITE (21, 9104) ITABLE(K), TABLE(1,K), TABLE(2,K)                  WONDY170
 9004 CONTINUE                                                            WONDY171
 9005 CONTINUE                                                            WONDY172
      GO TO IOUT                                                          WONDY173
C                                                                         WONDY174
C       END STANDARD EDITING                                              WONDY175
 1022 IF (PRINTS) CALL OUTPUT                                             WONDY176
      IF (T-TMIND) 1016, 1015, 1015                                       WONDY177
 1015 TMIND=TMIND+TDUMP                                                   WONDY178
C                                                                         WONDY179
C     WRITE BINARY DUMP TAPE HERE                                         WONDY180
C     BINARY DUMP TAPE MAY BE USED TO RESTART THE RUN OR FOR PLOTTING     WONDY181
      JM=(LMAX+1)*NVAR                                                    WONDY182
      WRITE (20) N, T, DELT(1), DELT(4), (ITABLE(K), TABLE(1,K),          WONDY183
     1TABLE(2,K), K=1,50), (STORE(J), J=1,JM)                             WONDY184
      PRINT 9201, N, T, DELT(1), JM                                       WONDY185
      IF (ITABLE(1)) PRINT 9202, (ITABLE(K), K=1,50)                      WONDY186
```

```
      LM=LMAX+1                                                          WONDY187
      DO 9001 L=1,LM                                                     WONDY188
      IF (PFRACT(L))PRINT 9203, L                                        WONDY189
 9001 CONTINUE                                                           WONDY190
      IF (TMIND.GT.TMAXD) TMIND=1.0                                      WONDY191
 1016 IF (EXIT) GO TO 1050                                               WONDY192
      DELT(2)=DELT(1)                                                    WONDY193
      DELT(1)=DELT(4)                                                    WONDY194
      NONE=NTWO=0                                                        WONDY195
      GO TO 1006                                                         WONDY196
 1050 STOP                                                               WONDY197
 1201 FORMAT ( 27H0 INITIAL INTERNAL ENERGY  , E15.5 / 27H  INITIAL KINEWONDY198
     1TIC ENERGY    , E15.5 / 27H  INITIAL TOTAL ENERGY      , E15.5 /  WONDY199
     227H  INITIAL TOTAL MOMENTUM    , E15.5 )                          WONDY200
 1202 FORMAT (32H0 ENERGY ERROR EXCEEDED ON CYCLE   , I4, 10H AT TIME = WONDY201
     1 , E15.5)                                                         WONDY202
 1203 FORMAT (22H0 NORMAL EXIT ON CYCLE   , I4,  10H AT TIME =  , E15.5)WONDY203
 1204 FORMAT (34H0 MOMENTUM ERROR EXCEEDED ON CYCLE    , I5, 10H AT TIMEWONDY204
     1 =  , E15.5 )                                                     WONDY205
 1701 FORMAT (28H0 RESTART FROM TAPE AT CYCLE  ,I5, 10X, 9H READ IN ,I5,WONDY206
     17H MESHES)                                                        WONDY207
 9101 FORMAT (7H1 CYCLE, I5, 8H   TIME , E15.5, 18H   TIME INCREMENT ,  WONDY208
     1 E15.5)                                                           WONDY209
 9102 FORMAT (113H0  L          X              U              RHO          SWONDY210
     1IGMA          PHI            Q             E             C     )  WONDY211
 9103 FORMAT (I5, 8E14.4)                                               WONDY212
 9104 FORMAT (10X, I5, 7X, E15.4, 11X, E15.4   )                        WONDY213
```

```
9105 FORMAT ( 20H0 WORK AT LEFT      , E15.5,10X, 20H  WORK AT RIGHT   WONDY214
     1    , E15.5 / 20H  INTERNAL ENERGY   , E15.5, 10X, 20H  KINETIC ENEWONDY215
     2RGY      , E15.5 / 20H  ADDED ENERGY       , E15.5, 10X, 20H  TOTALWONDY216
     3 MOMENTUM      , E15.5 / 20H  ENERGY ERROR       , E15.5, 10X,     WONDY217
     4 20H  MOMENTUM ERROR      , E15.5 )                                 WONDY218
9106 FORMAT ( 68H0 FRACTURE AT L       U AT LEFT HAND SIDE       X AT LWONDY219
     1EFT HAND SIDE      )                                              WONDY220
9201 FORMAT (12H0 DUMP AT N= , I5, 3H T=, E15.5, 9H DELT(1)=, E15.5,    WONDY221
     120H POINTS ON TAPE      , I10)                                    WONDY222
9202 FORMAT (23H  FRACTURES AT STATIONS    , 14I7 / (23X, 14I7))        WONDY223
9203 FORMAT (30H  PREVIOUS FRACTURE AT STATION , I5)                    WONDY224
      END                                                               WONDY225
```

```
      SUBROUTINE GENERATE                                                GENER   1
C PROGRAM WONDY                                                          GENER   2
      BANK, (0), GENERATE , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/        GENER   3
C     INSERT COMMON CARDS HERE
C                                                                        GENER  27
C     READ INPUT DATA                                                    GENER  28
      READ 1101, (TITLE(IC), IC=1,10)                                    GENER  29
      READ 1102, LPHA, NOP, NVAR, LHBT, RHBT, LACT, JTAPE, NSTART,       GENER  30
     1NOQM, NOPM, NOAD, W4020, NIL, NUL                                  GENER  31
      IF (NOQM) READ 1102, (QMESH(K), K=1,NOQM)                          GENER  32
      IF (NOPM) READ 1102, (PMESH(K), K=1,NOPM)                          GENER  33
      READ 1103, XZERO, B1, B2, KE, SIGMAACT, SIGMAMAX, SIGMASEP         GENER  34
      READ 1103, KT1, KT2, TMAX, TMIND, TDUMP, TMAXD, DELT(4)            GENER  35
      READ 1103, TMINP, TPRINT, TMAXP, TMINPS, TPRINTS, TMAXPS, KH       GENER  36
      READ 1103, TMINPL, TPRINTL, TMAXPL                                 GENER  37
      IF (NOAD) READ 1103, (ADDATA(K), K=1,NOAD)                         GENER  38
      DO 2001 PLATE=1,NOP                                                GENER  39
      READ 1103, NOMESHES(PLATE), STATE(PLATE), DELTAX(PLATE), XRATIO(PLGENER  40
     1ATE), XGAP(PLATE), UZERO(PLATE), UZEROI(PLATE)                     GENER  41
      READ 1103, RHOZERO(PLATE), PZERO(PLATE), SIGZERO(PLATE), EZERO(PLAGENER  42
     1TE), PHIZERO(PLATE), SIGMAF(PLATE), SIGMAIF(PLATE)                 GENER  43
      READ 1103, (CES(IC,PLATE),IC=1, 35)                                GENER  44
 2001 CONTINUE                                                           GENER  45
C                                                                        GENER  46
C     PRINT INPUT DATA                                                   GENER  47
      PRINT 1201, (TITLE(IC), IC=1, 10)                                  GENER  48
      PRINT 1202, LPHA, NOP, NVAR, LHBT, RHBT, LACT, JTAPE, NSTART,      GENER  49
```

```
     1NOQM, NOPM, NOAD, W4020, NIL, NUL                                  GENER 50
      IF (NOQM) PRINT 1203, (QMESH(K), K=1,NOQM)                         GENER 51
      IF (NOPM) PRINT 1204, (PMESH(K), K=1,NOPM)                         GENER 52
      PRINT 1205, XZERO, B1, B2, KE, SIGMAACT, SIGMAMAX, SIGMASEP        GENER 53
      PRINT 1206, KT1, KT2, TMAX, TMIND, TDUMP, TMAXD, DELT(4)           GENER 54
      PRINT 1207, TMINP, TPRINT, TMAXP, TMINPS, TPRINTS, TMAXPS, KH      GENER 55
      PRINT 1223, TMINPL, TPRINTL, TMAXPL                                GENER 56
      IF (NOAD) PRINT 1208, (ADDATA(K), K=1,NOAD)                        GENER 57
      NOP2=0                                                             GENER 58
 2002 NOP1=NOP2+1                                                        GENER 59
      NOP2=NOP2+6                                                        GENER 60
      IF (NOP2.GT.NOP) NOP2=NOP                                          GENER 61
      PRINT 1209, (NOMESHES(PLATE), PLATE=NOP1,NOP2)                     GENER 62
      PRINT 1210, (STATE(PLATE), PLATE=NOP1,NOP2)                        GENER 63
      PRINT 1211, (DELTAX(PLATE), PLATE=NOP1,NOP2)                       GENER 64
      PRINT 1212, (XRATIO(PLATE), PLATE=NOP1,NOP2)                       GENER 65
      PRINT 1213, (XGAP(PLATE), PLATE=NOP1,NOP2)                         GENER 66
      PRINT 1214, (UZERO(PLATE), PLATE=NOP1,NOP2)                        GENER 67
      PRINT 1215, (UZEROI(PLATE), PLATE=NOP1,NOP2)                       GENER 68
      PRINT 1216, (RHOZERO(PLATE), PLATE=NOP1,NOP2)                      GENER 69
      PRINT 1217, (PZERO(PLATE), PLATE=NOP1,NOP2)                        GENER 70
      PRINT 1218, (SIGZERO(PLATE), PLATE=NOP1,NOP2)                      GENER 71
      PRINT 1219, (EZERO(PLATE), PLATE=NOP1,NOP2)                        GENER 72
      PRINT 1234, (PHIZERO(PLATE), PLATE=NOP1,NOP2)                      GENER 73
      PRINT 1220, (SIGMAF(PLATE), PLATE=NOP1,NOP2)                       GENER 74
      PRINT 1221, (SIGMAIF(PLATE), PLATE=NOP1,NOP2)                      GENER 75
      DO 2003 IC=1,35                                                    GENER 76
```

```
2003 PRINT 1222,      IC, (CES(IC, PLATE), PLATE=NOP1,NOP2)                  GENER 77
     IF (NOP.NE.NOP2) GO TO 2002                                             GENER 78
     IF (W4020) 2004, 2005                                                   GENER 79
2004 CONTINUE                                                                GENER 80
     WRITE (21, 1201) (TITLE(IC), IC=1,10)                                   GENER 81
     WRITE (21, 1202) LPHA, NOP, NVAR, LHBT, RHBT, LACT, JTAPE, NSTART,GENER 82
    1NOQM, NOPM, NOAD, W4020, NIL, NUL                                       GENER 83
     IF (NOQM) WRITE (21, 1203) (QMESH(K), K=1,NOQM)                         GENER 84
     IF (NOPM) WRITE (21, 1204) (PMESH(K), K=1,NOPM)                         GENER 85
     WRITE (21, 1205) XZERO, B1, B2, KE, SIGMAACT, SIGMAMAX, SIGMASEP        GENER 86
     WRITE (21, 1206) KT1, KT2, TMAX, TMIND, TDUMP, TMAXD, DELT(4)           GENER 87
     WRITE (21, 1207) TMINP, TPRINT, TMAXP, TMINPS, TPRINTS, TMAXPS, KHGENER 88
     WRITE (21, 1223) TMINPL, TPRINTL, TMAXPL                                GENER 89
     IF (NOAD) WRITE (21, 1208) (ADDATA(K), K=1,NOAD)                        GENER 90
     NOP2=0                                                                  GENER 91
2019 NOP1=NOP2+1                                                             GENER 92
     NOP2=NOP2+6                                                             GENER 93
     IF (NOP2.GT.NOP) NOP2=NOP                                               GENER 94
     WRITE (21, 1209) (NOMESHES(PLATE), PLATE=NOP1,NOP2)                     GENER 95
     WRITE (21, 1210) (STATE(PLATE), PLATE=NOP1,NOP2)                        GENER 96
     WRITE (21, 1211) (DELTAX(PLATE), PLATE=NOP1,NOP2)                       GENER 97
     WRITE (21, 1212) (XRATIO(PLATE), PLATE=NOP1,NOP2)                       GENER 98
     WRITE (21, 1213) (XGAP(PLATE), PLATE=NOP1,NOP2)                         GENER 99
     WRITE (21, 1214) (UZERO(PLATE), PLATE=NOP1,NOP2)                        GENER100
     WRITE (21, 1215) (UZEROI(PLATE), PLATE=NOP1,NOP2)                       GENER101
     WRITE (21, 1216) (RHOZERO(PLATE), PLATE=NOP1,NOP2)                      GENER102
     WRITE (21, 1217) (PZERO(PLATE), PLATE=NOP1,NOP2)                        GENER103
```

```
      WRITE (21, 1218) (SIGZERO(PLATE), PLATE=NOP1,NOP2)                GENER104
      WRITE (21, 1219) (EZERO(PLATE), PLATE=NOP1,NOP2)                  GENER105
      WRITE (21, 1234) (PHIZERO(PLATE), PLATE=NOP1,NOP2)                GENER106
      WRITE (21, 1220) (SIGMAF(PLATE), PLATE=NOP1,NOP2)                 GENER107
      WRITE (21, 1221) (SIGMAIF(PLATE), PLATE=NOP1,NOP2)                GENER108
      DO 2006 IC=1,35                                                   GENER109
 2006 WRITE (21, 1222) IC, (CES(IC, PLATE), PLATE=NOP1,NOP2 )           GENER110
      IF (NOP.NE.NOP2) GO TO 2019                                       GENER111
 2005 CONTINUE                                                          GENER112
      LMAX=0                                                            GENER113
      DO 2007 PLATE=1, NOP                                              GENER114
 2007 LMAX=LMAX+NOMESHES(PLATE)                                         GENER115
      LM=LMAX+1                                                         GENER116
      DO 2008 L=1,LM                                                    GENER117
 2008 QFRACT(L)=PFRACT(L)=0                                             GENER118
      DO 2009 K=1, NOQM                                                 GENER119
      L=QMESH(K)                                                        GENER120
 2009 QFRACT(L)=1                                                       GENER121
      DO 2010 K=1, NOPM                                                 GENER122
      L=PMESH(K)                                                        GENER123
 2010 PFRACT(L)=1                                                       GENER124
      QFRACT(1)=1                                                       GENER125
C                                                                       GENER126
C     GENERATE LARGE ARRAYS                                             GENER127
      J=PLATE=L=1                                                       GENER128
      LOL=1                                                             GENER129
      LOR=NOMESHES(PLATE)+1                                             GENER130
```

```
      DX=DELTAX(PLATE)/XRATIO(PLATE)                                    GENER131
      X(J)=XZERO                                                        GENER132
      U(J)=UZERO(PLATE)                                                 GENER133
      C(J)=RHO(J)=Q(J)=PHI(J)=E(J)=P(J)=SIGMA(J)=0.0                    GENER134
      M(J)=0.0                                                          GENER135
 2011 J=J+NVAR                                                          GENER136
      L=L+1                                                             GENER137
      U(J)=UZERO(PLATE)                                                 GENER138
      DX=DX*XRATIO(PLATE)                                               GENER139
      X(J)=X(J-NVAR)+DX                                                 GENER140
      C(J)=CES(2,PLATE)                                                 GENER141
      RHO(J)=RHOZERO(PLATE)                                             GENER142
      P(J)=PZERO(PLATE)                                                 GENER143
      SIGMA(J)=SIGZERO(PLATE)                                           GENER144
      E(J)=EZERO(PLATE)                                                 GENER145
      PHI(J)=PHIZERO(PLATE)                                             GENER146
      Q(J)=0.0                                                          GENER147
      IF (LPHA-2) 2028, 2029, 2030                                      GENER148
 2028 CXI=1.0                                                           GENER149
      GO TO 2031                                                        GENER150
 2029 CXI=X(J)+X(J-NVAR)                                                GENER151
      GO TO 2031                                                        GENER152
 2030 CXI=X(J)**2+X(J)*X(J-NVAR)+X(J-NVAR)**2                           GENER153
 2031 M(J)=RHO(J)*(X(J)-X(J-NVAR))*CXI                                  GENER154
      IF (L-LOR) 2011, 2012, 2012                                       GENER155
 2012 GOIND=STATE(PLATE)                                                GENER156
      GO TO (2021, 2022, 2023, 2024, 2025, 2026) GOIND                  GENER157
```

```
2021 CALL STIN1                                                    GENER158
     GO TO 2027                                                    GENER159
2022 CALL STIN2                                                    GENER160
     GO TO 2027                                                    GENER161
2023 CALL STIN3                                                    GENER162
     GO TO 2027                                                    GENER163
2024 CALL STIN4                                                    GENER164
     GO TO 2027                                                    GENER165
2025 CALL STIN5                                                    GENER166
     GO TO 2027                                                    GENER167
2026 CALL STIN6                                                    GENER168
2027 IF (L-LMAX+1) 2013, 2018, 2018                                GENER169
2013 U(J)=UZEROI(PLATE)                                            GENER170
     IF (XGAP(PLATE)) 2014, 2017                                   GENER171
2014 QFRACT(L)=1                                                   GENER172
     DO 2015 K=1, 50                                               GENER173
     IF (ITABLE(K)) 2015, 2016                                     GENER174
2015 CONTINUE                                                      GENER175
2016 ITABLE(K)=L                                                   GENER176
     TABLE(1,K)=UZERO(PLATE)                                       GENER177
     TABLE(2,K)=X(J)                                               GENER178
     X(J)=X(J)+XGAP(PLATE)                                         GENER179
     U(J)=UZERO(PLATE+1)                                           GENER180
2017 PLATE=PLATE+1                                                 GENER181
     LOL=LOR                                                       GENER182
     LOR=LOR+NOMESHES(PLATE)                                       GENER183
     DX=DELTAX(PLATE)/XRATIO(PLATE)                                GENER184
```

```
      GO TO 2011                                                        GENER185
 2018 J=J+NVAR                                                          GENER186
      U(J)=UZERO(PLATE)                                                 GENER187
      X(J)=X(J-NVAR)                                                    GENER188
      C(J)=RHO(J)=Q(J)=PHI(J)=E(J)=P(J)=SIGMA(J)=0.0                    GENER189
      M(J)=RHO(J)*(X(J)**LPHA-X(J-NVAR)**LPHA)                          GENER190
      RETURN                                                            GENER191
 1101 FORMAT (10A8)                                                     GENER192
 1102 FORMAT (14I5)                                                     GENER193
 1103 FORMAT (7E10.3)                                                   GENER194
 1201 FORMAT (1H1, 10A8)                                                GENER195
 1202 FORMAT(115H0    LPHA     NOP     NVAR     LHBT     RHBT     LACT   JTAGENER196
     1PE  NSTART     NOQM     NOPM     NOAD   W4020      NIL      NUL      / GENER197
     11X, 14I8)                                                         GENER198
 1203 FORMAT ( 29H0 MESHES WHICH HAVE FRACTURED  / (1X, 14I8))          GENER199
 1204 FORMAT ( 26H0 MESHES WHICH HAVE JOINED  / (1X, 14I8))             GENER200
 1205 FORMAT (114H0            XZERO                B1                B2  GENER201
     1          KE         SIGMAACT         SIGMAMAX         SIGMASEP   /  GENER202
     2 1X, 7E16.3)                                                      GENER203
 1206 FORMAT (114H0              KT1              KT2              TMAX   GENER204
     1       TMIND            TDUMP            TMAXD          DELT(4)   /  GENER205
     2 1X, 7E16.3)                                                      GENER206
 1207 FORMAT (114H0            TMINP            TPRINT            TMAXP   GENER207
     1      TMINPS          TPRINTS           TMAXPS              KH   /  GENER208
     2 1X, 7E16.3)                                                      GENER209
 1208 FORMAT ( 17H0 ADDITIONAL DATA  / 1X, 7E16.3 / 1X, 7E16.3)         GENER210
 1209 FORMAT (17H0 NOMESHES       , 6E16.3)                             GENER211
```

```
 1210 FORMAT (17H  STATE EQ        , 6E16.3)                              GENER212
 1211 FORMAT (17H  DELTAX          , 6E16.3)                              GENER213
 1212 FORMAT (17H  XRATIO          , 6E16.3)                              GENER214
 1213 FORMAT (17H  XGAP            , 6E16.3)                              GENER215
 1214 FORMAT (17H  UZERO           , 6E16.3)                              GENER216
 1215 FORMAT (17H  UZEROI          , 6E16.3)                              GENER217
 1216 FORMAT (17H  RHOZERO          , 6E16.3)                             GENER218
 1217 FORMAT (17H  PZERO           , 6E16.3)                              GENER219
 1218 FORMAT (17H  SIGZERO         , 6E16.3)                              GENER220
 1219 FORMAT (17H  EZERO           , 6E16.3)                              GENER221
 1220 FORMAT (17H  SIGMAF          , 6E16.3)                              GENER222
 1221 FORMAT (17H  SIGMAIF          , 6E16.3)                             GENER223
 1222 FORMAT ( I8, 9X, 6E16.3)                                            GENER224
 1223 FORMAT (50HO          TMINPL          TPRINTL          TMAXPL     / GENER225
     1 1X, 3E16.3)                                                        GENER226
 1234 FORMAT (17H  PHIZERO          , 6E16.3)                             GENER227
      END                                                                 GENER228
```

```
      SUBROUTINE MORSTORE                                              MORST   1
C PROGRAM WONDY II                                                     MORST   2
      BANK, (0), MORSTORE , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/      MORST   3
C     INSERT COMMON CARDS HERE
      EQUIVALENCE (STORE(11), SPEC(1))                                 MORST  27
      DIMENSION SPEC(31000)                                            MORST  28
      DIMENSION SPECZERO(20)                                           MORST  29
      READ 2701, TDEP                                                  MORST  30
      READ 2701, (SPECZERO(PLATE), PLATE=1,NOP)                        MORST  31
      PRINT 2702, TDEP                                                 MORST  32
      PRINT 2703, (SPECZERO(PLATE), PLATE=1,NOP)                       MORST  33
      L=J=LOR=1                                                        MORST  34
      PLATE=0                                                          MORST  35
 2401 PLATE=PLATE+1                                                    MORST  36
      LOR=LOR+NOMESHES(PLATE)                                          MORST  37
 2402 L=L+1                                                            MORST  38
      J=J+NVAR                                                         MORST  39
      SPEC(J)=SPECZERO(PLATE)                                          MORST  40
      IF (L-LOR) 2402, 2403                                            MORST  41
 2403 IF (PLATE-NOP) 2401, 2404                                        MORST  42
 2404 CONTINUE                                                         MORST  43
      RETURN                                                           MORST  44
 2701 FORMAT (7E10.3)                                                  MORST  45
 2702 FORMAT (*0 DEPOSITION TIME = *, E16.3)                           MORST  46
 2703 FORMAT (*0 TOTAL ENERGY DEPOSITED PER MESH FOR EACH PLATE * /    MORST  47
     1 (7E16.3))                                                       MORST  48
      END                                                              MORST  49
```

```
      SUBROUTINE BOUNDARY                                                BOUND  1
C PROGRAM WONDY                                                          BOUND  2
      BANK, (0), BOUNDARY , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/        BOUND  3
C     INSERT COMMON CARDS HERE
C     SAMPLE CODING OF A SPECIAL BOUNDARY CONDITION                      BOUND 27
C     BY CHOOSING PROPER INPUT FOR THE ADDATA ARRAY, THIS CODING WILL    BOUND 28
C     GIVE A CONSTANT OR AN EXPONENTIALLY VARYING SURFACE LOAD FOR       BOUND 29
C     EITHER THE LEFT OR THE RIGHT BOUNDARY.                             BOUND 30
      IF (LHBT-3) 6002, 6001                                             BOUND 31
 6001 SIGMAL=SIGMA(1)                                                    BOUND 32
      SIGMA(1)=ADDATA(8)+ADDATA(9)*EXPF(-ADDATA(10)*T)                   BOUND 33
      SIGMAL=0.5*(SIGMA(1)+SIGMAL)                                       BOUND 34
      GO TO 6999                                                         BOUND 35
 6002 JN=J+NVAR                                                          BOUND 36
      SIGMAR=SIGMA(JN)                                                   BOUND 37
      SIGMA(JN)=ADDATA(8)+ADDATA(9)*EXPF(-ADDATA(10)*T)                  BOUND 38
      SIGMAR=0.5*(SIGMA(JN)+SIGMAR)                                      BOUND 39
 6999 RETURN                                                             BOUND 40
      END                                                                BOUND 41
```

```
      SUBROUTINE JLOOPING                                              JLOOP  1
C PROGRAM WONDY                                                        JLOOP  2
      BANK, (C), JLOOPING , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/      JLOOP  3
C     INSERT COMMON CARDS HERE
      J=L=PLATE=1                                                      JLOOP 27
      LOR=NOMESHES(PLATE)+1                                            JLOOP 28
C                                                                      JLOOP 29
C     LEFT HAND BOUNDARY                                               JLOOP 30
C     BOUNDARY TYPES ARE IN ORDER - FIXED, FREE, SPECIAL               JLOOP 31
      IF (LHBT-2) 3000, 3002, 3001                                     JLOOP 32
 3000 A=U1=0.0                                                         JLOOP 33
      X(J)=X1=XZERO                                                    JLOOP 34
      GO TO 3030                                                       JLOOP 35
 3001 CALL BOUNDARY                                                    JLOOP 36
 3002 CALL MOTION                                                      JLOOP 37
      WL=SIGMAL*(X1-X(J))*KM(LPHA)                                     JLOOP 38
 3030 XP=X(J)                                                          JLOOP 39
      SIGMAP=SIGMA(J)                                                  JLOOP 40
      U(J)=U1                                                          JLOOP 41
      X(J)=X1                                                          JLOOP 42
      JONE=1                                                           JLOOP 43
C                                                                      JLOOP 44
C     INTERNAL MESHES                                                  JLOOP 45
 3003 J=J+NVAR                                                         JLOOP 46
      L=L+1                                                            JLOOP 47
      TEST=SIGMAF(PLATE)                                               JLOOP 48
      IF (L-LOR) 3005, 3004, 3004                                      JLOOP 49
```

```
 3004 TEST=SIGMAIF(PLATE)                                          JLOOP 50
      IND=1                                                        JLOOP 51
C                                                                  JLOOP 52
C     TEST FOR PRESENT FRACTURE                                    JLOOP 53
 3005 IF (QFRACT(L)) 3011, 3006                                    JLOOP 54
C     TEST FOR PREVIOUS FRACTURE WHICH HAS COME TOGETHER           JLOOP 55
 3006 IF (PFRACT(L)) TEST=SIGMASEP                                 JLOOP 56
C     TEST FOR NEW FRACTURE                                        JLOOP 57
      SIGMAA=0.5*(SIGMA(J+NVAR)+SIGMA(J))                          JLOOP 58
      IF (SIGMAA-TEST) 3008, 3007, 3007                            JLOOP 59
C                                                                  JLOOP 60
C     LOOK AHEAD FOR FRACTURE                                      JLOOP 61
 3007 IF (QFRACT(L+1)) 3401, 3404                                  JLOOP 62
 3401 XAHEAD=X(J+NVAR)                                             JLOOP 63
      DO 3402 K=1,50                                               JLOOP 64
      IF (L+1.EQ.ITABLE(K)) 3403, 3402                             JLOOP 65
 3402 CONTINUE                                                     JLOOP 66
 3403 X(J+NVAR)=TABLE(2,K)                                         JLOOP 67
      CALL MOTION                                                  JLOOP 68
      X(J+NVAR)=XAHEAD                                             JLOOP 69
      GO TO 3013                                                   JLOOP 70
C                                                                  JLOOP 71
C     CONTINUE WITH NORMAL MESHES                                  JLOOP 72
 3404 CALL MOTION                                                  JLOOP 73
      GO TO 3013                                                   JLOOP 74
C                                                                  JLOOP 75
C     NEW FRACTURE OCCURRED, STORE VALUES IN TABLE                 JLOOP 76
 3008 QFRACT(L)=1                                                  JLOOP 77
```

```
      NP=N-1                                                          JLOOP 78
      PT=T-DELT(1)                                                    JLOOP 79
      PRINT 3201, L, NP, PT                                           JLOOP 80
      IF (W4020) WRITE(21, 3201) L, NP, PT                            JLOOP 81
      DO 3009 K=1,50                                                  JLOOP 82
      IF (ITABLE(K).EQ.0.OR.ITABLE(K).EQ.L) GO TO 3010                JLOOP 83
 3009 CONTINUE                                                        JLOOP 84
      PRINT 3202, N, T                                                JLOOP 85
      IF (W4020) WRITE (21, 3202) N, T                                JLOOP 86
      PRINTR=EXIT=1                                                   JLOOP 87
      GO TO 3020                                                      JLOOP 88
 3010 ITABLE(K)=L                                                     JLOOP 89
      TABLE(1,K)=U(J)                                                 JLOOP 90
      TABLE(2,K)=X(J)                                                 JLOOP 91
C                                                                     JLOOP 92
C     FRACTURE ROUTINE BEGINS HERE                                    JLOOP 93
 3011 JN=J+NVAR                                                       JLOOP 94
C                                                                     JLOOP 95
C     FIND NEW POSITION AT RIGHT HAND SIDE OF FRACTURE                JLOOP 96
C     STORE VALUES FOR LEFT SIDE OF FRACTURE TEMPORARILY AND TREAT    JLOOP 97
C     FRACTURE AS A FREE SURFACE                                      JLOOP 98
      SIGMAE=SIGMA(J)                                                 JLOOP 99
      PHIE=PHI(J)                                                     JLOOP100
      RHOE=RHO(J)                                                     JLOOP101
      QE=Q(J)                                                         JLOOP102
      SIGMA(J)=PHI(J)=RHO(J)=Q(J)=0.0                                 JLOOP103
C     LOOK AHEAD FOR FRACTURE                                         JLOOP104
      IF (QFRACT(L+1)) 3405, 3408                                     JLOOP105
```

```
 3405 XAHEAD=X(J+NVAR)                                                  JLOOP106
      DO 3406 K=1,50                                                    JLOOP107
      IF (L+1.EQ.ITABLE(K)) 3407, 3406                                  JLOOP108
 3406 CONTINUE                                                          JLOOP109
 3407 X(J+NVAR)=TABLE(2,K)                                              JLOOP110
      CALL MOTION                                                       JLOOP111
      X(J+NVAR)=XAHEAD                                                  JLOOP112
      GO TO 3409                                                        JLOOP113
 3408 CALL MOTION                                                       JLOOP114
 3409 XPE=X(J)                                                          JLOOP115
      U(J)=U1                                                           JLOOP116
      X(J)=X1                                                           JLOOP117
      SIGMA(J)=SIGMAE                                                   JLOOP118
      PHI(J)=PHIE                                                       JLOOP119
      RHO(J)=RHOE                                                       JLOOP120
      Q(J)=QE                                                           JLOOP121
C                                                                       JLOOP122
C     FIND NEW POSITION AT LEFT HAND SIDE OF FRACTURE                   JLOOP123
C     STORE VALUES FOR RIGHT SIDE TEMPORARILY, SAME AS ABOVE            JLOOP124
      SIGMAE=SIGMA(JN)                                                  JLOOP125
      PHIE=PHI(JN)                                                      JLOOP126
      RHOE=RHO(JN)                                                      JLOOP127
      QE=Q(JN)                                                          JLOOP128
      UE=U(J)                                                           JLOOP129
      XE=X(J)                                                           JLOOP130
      SIGMA(JN)=PHI(JN)=RHO(JN)=Q(JN)=0.0                               JLOOP131
      DO 3501 K=1,50                                                    JLOOP132
```

```
      IF (L.EQ.ITABLE(K)) 3502, 3501                                    JLOOP133
 3501 CONTINUE                                                          JLOOP134
      PRINT 3701, L                                                     JLOOP135
      IF (W4020) WRITE (21, 3701) L                                     JLOOP136
      PRINTR=EXIT=1                                                     JLOOP137
      GO TO 3020                                                        JLOOP138
 3502 U(J)=TABLE(1,K)                                                   JLOOP139
      X(J)=TABLE(2,K)                                                   JLOOP140
      CALL MOTION                                                       JLOOP141
C                                                                       JLOOP142
C     DID FRACTURE SURFACES COLLIDE                                     JLOOP143
      IF (X1-XE) 3504, 3504, 3503                                       JLOOP144
 3503 QFRACT(L)=0                                                       JLOOP145
      PFRACT(L)=1                                                       JLOOP146
      PRINT 3702, L, N, T                                               JLOOP147
      IF (W4020) WRITE (21, 3702) L, N, T                               JLOOP148
      U1=UE=0.5*(U1+UE)                                                 JLOOP149
      X1=XE=0.5*(X1+XE)                                                 JLOOP150
C                                                                       JLOOP151
C     CONTINUE WITH MESH CALCULATIONS FOR MESH TO LEFT OF FRACTURE      JLOOP152
 3504 CALL MESHES                                                       JLOOP153
      TABLE(1,K)=U(J)                                                   JLOOP154
      TABLE(2,K)=X(J)                                                   JLOOP155
      SIGMA(JN)=SIGMAE                                                  JLOOP156
      PHI(JN)=PHIE                                                      JLOOP157
      RHO(JN)=RHOE                                                      JLOOP158
      Q(JN)=QE                                                          JLOOP159
```

```
      U(J)=UE                                                          JLOOP160
      X(J)=XE                                                          JLOOP161
      XP=XPE                                                           JLOOP162
 3701 FORMAT (39HO CANNOT FIND FRACTURE ENTRY AT STATION , I4)         JLOOP163
 3702 FORMAT (40HO FRACTURED SURFACES COLLIDED AT STATION , I4,        JLOOP164
     17H CYCLE , I4, 6H TIME , E15.5)                                  JLOOP165
C                                                                      JLOOP166
C     CONTINUE WITH JLOOPING                                           JLOOP167
 3013 JONE=0                                                           JLOOP168
      IF (IND) 3014, 3015                                              JLOOP169
C                                                                      JLOOP170
C     IF THIS IS AN INTERFACE, ADVANCE PLATE INDICATOR                 JLOOP171
 3014 PLATE=PLATE+1                                                    JLOOP172
      LOR=LOR+NOMESHES(PLATE)                                          JLOOP173
      IND=0                                                            JLOOP174
C                                                                      JLOOP175
C     HAS JMAX BEEN REACHED                                            JLOOP176
 3015 IF (L.LT.LMAX) 3021, 3016                                        JLOOP177
C                                                                      JLOOP178
C     ACTIVITY TESTED HERE                                             JLOOP179
 3021 IF (L.GE.LACT) 3022, 3003                                        JLOOP180
 3022 IF (SIGMA(J).GE.SIGMAACT) 3003, 3023                             JLOOP181
 3023 LACT=L                                                           JLOOP182
      GO TO 3020                                                       JLOOP183
C                                                                      JLOOP184
C     RIGHT HAND BOUNDARY                                              JLOOP185
C     BOUNDARY TYPES ARE IN ORDER - FIXED, FREE, SPECIAL               JLOOP186
 3016 J=J+NVAR                                                         JLOOP187
```

```
      L=L+1                                                          JLOOP188
      LACT=L                                                         JLOOP189
      IF (RHBT-2) 3019, 3018, 3017                                   JLOOP190
 3017 CALL BOUNDARY                                                  JLOOP191
 3018 CALL MOTION                                                    JLOOP192
      WR=SIGMAR*(X1-X(J))*KM(LPHA)                                   JLOOP193
      GO TO 3020                                                     JLOOP194
 3019 A=U1=0.0                                                       JLOOP195
      X1=X(J)                                                        JLOOP196
      CALL MESHES                                                    JLOOP197
 3020 RETURN                                                         JLOOP198
 3201 FORMAT (30H0 FRACTURE OCCURRED AT STATION , I4, 7H CYCLE , I4,  JLOOP199
     16H TIME , E15.5)                                               JLOOP200
 3202 FORMAT (46H0 MAXIMUM NUMBER OF FRACTURES EXCEEDED, CYCLE , I4,  JLOOP201
     16H TIME , E15.5)                                               JLOOP202
      END                                                            JLOOP203
```

```
      SUBROUTINE MOTION                                                  MOTIO  1
C PROGRAM WONDY                                                          MOTIO  2
      BANK, (U), MOTION    , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       MOTIO  3
C     INSERT COMMON CARDS HERE
      A=2.0*(SIGMA(J)-SIGMA(J+NVAR)+Q(J)-Q(J+NVAR))/(RHO(J)*(X(J)-XP)    MOTIO 27
     1+RHO(J+NVAR)*(X(J+NVAR)-X(J)))+2.0*(LPHA-1.0)*(PHI(J)+PHI(J+NVAR))MOTIO 28
     2/(RHO(J)*(X(J)+XP)+RHO(J+NVAR)*(X(J+NVAR)+X(J)))                   MOTIO 29
      U1=U(J)+0.5*(DELT(1)+DELT(2))*A                                    MOTIO 30
      X1=X(J)+U1*DELT(1)                                                 MOTIO 31
      CXIP=CXI                                                           MOTIO 32
      IF (LPHA-2) 4018, 4019, 4020                                       MOTIO 33
 4018 CXI=1.0                                                            MOTIO 34
      GO TO 4021                                                         MOTIO 35
 4019 CXI=X1+X(J)                                                        MOTIO 36
      GO TO 4021                                                         MOTIO 37
 4020 CXI=X1**2+X1*X(J)+X(J)**2                                          MOTIO 38
 4021 CONTINUE                                                           MOTIO 39
      IF (QFRACT(L)) GO TO 4011                                          MOTIO 40
      ENTRY MESHES                                                       MOTIO 41
      SIGMAP=SIGMA(J)                                                    MOTIO 42
      DELXJ=X1-X(J-NVAR)                                                 MOTIO 43
      RHO1=1.0/(1.0/RHO(J)+DELT(1)/M(J)*(CXI*U1-CXIP*U(J-NVAR)))         MOTIO 44
C                                                                        MOTIO 45
C     CHECK FOR ROUNDOFF IN DENSITY                                      MOTIO 46
      IF (ABSF(RHO1-RHO(J)).LT.5.0E-10*RHO(J)) 4013, 4014                MOTIO 47
 4013 DELRJ=RHODOT=DELRHO=Q1=0.0                                         MOTIO 48
      GO TO 4003                                                         MOTIO 49
```

```
 4014 DELRJ=2.*(RHO1-RHO(J))/(RHO1+RHO(J))                          MOTIO 50
      RHODOT=DELRJ/DELT(1)                                          MOTIO 51
      DELRHO=DELRJ/(RHO1+RHO(J))                                    MOTIO 52
      IF (RHODOT) 4001, 4001, 4002                                  MOTIO 53
 4001 Q1=0.0                                                        MOTIO 54
      GO TO 4003                                                    MOTIO 55
 4002 Q1=RHO1*(B2*DELXJ*C(J)*RHODOT+(B1*DELXJ*RHODOT)**2)           MOTIO 56
C                                                                   MOTIO 57
C     CALL EQUATION OF STATE ROUTINE FOR CORRECT PLATE              MOTIO 58
 4003 GOIND=STATE(PLATE)                                            MOTIO 59
      GO TO (4004, 4005, 4006, 4007, 4008, 4009)GOIND               MOTIO 60
 4004 CALL STATE1                                                   MOTIO 61
      GO TO 4010                                                    MOTIO 62
 4005 CALL STATE2                                                   MOTIO 63
      GO TO 4010                                                    MOTIO 64
 4006 CALL STATE3                                                   MOTIO 65
      GO TO 4010                                                    MOTIO 66
 4007 CALL STATE4                                                   MOTIO 67
      GO TO 4010                                                    MOTIO 68
 4008 CALL STATE5                                                   MOTIO 69
      GO TO 4010                                                    MOTIO 70
 4009 CALL STATE6                                                   MOTIO 71
C                                                                   MOTIO 72
C     COMPUTE NEW TIME STEP                                         MOTIO 73
 4010 DELU=U1-U(J-NVAR)                                             MOTIO 74
      IF (DELU) 4015, 4016, 4016                                    MOTIO 75
 4015 BCBU=B2*C(J)-B1**2*DELU                                       MOTIO 76
```

```
      DELT(3)=KT1*DELXJ/(BCBU+SQRTF(BCBU**2+C(J)**2))                    MOTIO 77
      GO TO 4017                                                        MOTIO 78
 4016 DELT(3)=KT1*DELXJ/C(J)                                            MOTIO 79
 4017 CONTINUE                                                          MOTIO 80
      IF (TDEP.GT.T) DELT(3)=MIN1F(DELT(3), 0.01*TDEP)                  MOTIO 81
      DELT(4)=MIN1F(DELT(4),DELT(3))                                    MOTIO 82
C                                                                       MOTIO 83
C     COMPUTE ENERGY SUMS                                               MOTIO 84
      CAPH=0.5*M(J)*(U(J-NVAR)+U1)                                      MOTIO 85
      CAPK=M(J)/8.0*(U(J-NVAR)+U1)**2                                   MOTIO 86
      CAPE=0.5*M(J)*(E1+E(J))                                           MOTIO 87
      SUMH=SUMH+CAPH                                                    MOTIO 88
      SUMKE=SUMKE+CAPK                                                  MOTIO 89
      SUMIE=SUMIE+CAPE                                                  MOTIO 90
      IF (PRINTL) CALL OUTL                                             MOTIO 91
      XP=X(J)                                                           MOTIO 92
      Q(J)=Q1                                                           MOTIO 93
      U(J)=U1                                                           MOTIO 94
      RHO(J)=RHO1                                                       MOTIO 95
      X(J)=X1                                                           MOTIO 96
C                                                                       MOTIO 97
C     CHECK IF MAXIMUM STRESS IS EXCEEDED                               MOTIO 98
      IF(SIGMA(J)-SIGMAMAX) 4011, 4011, 4012                            MOTIO 99
 4012 PRINTR=EXIT=1                                                     MOTIO100
      PRINT 4101, L, N, T                                               MOTIO101
      IF (W4020) WRITE (21, 4101) L, N, T                               MOTIO102
 4101 FORMAT (36H0 MAXIMUM STRESS EXCEEDED AT STATION , I4, 7H CYCLE ,  MOTIO103
```

```
      114, 6H TIME ,E15.5)                                          MOTIO104
 4011 RETURN                                                        MOTIO105
      END                                                           MOTIO106
```

```
      SUBROUTINE STATE1                                                  STATE  1
C PROGRAM WONDY                                                          STATE  2
      BANK, (0), STATE1    , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       STATE  3
C     INSERT COMMON CARDS HERE
C                                                                        STATE 27
C     ELASTIC-PLASTIC HYDRO VAPOR EQUATIONS                              STATE 28
      DIMENSION GCONST(20), KCONST(20), NCONST(20)                       STATE 29
      TYPE REAL KCONST, NCONST, MU                                       STATE 30
C                                                                        STATE 31
C     SPECIAL CARDS FOR ENERGY DEPOSITION - THROUGH 5006  +2             STATE 32
      EQUIVALENCE (STORE(11), SPEC(1))                                   STATE 33
      DIMENSION SPEC(31000)                                              STATE 34
      IF (JONE.AND.TDEP) 5001, 5006                                      STATE 35
 5001 IF (NONE) SUMDEP=0.0                                               STATE 36
      IF (NTWO) SUMDEP=1.0                                               STATE 37
      IF (SUMDEP-1.0) 5002, 5005                                         STATE 38
 5002 DEP=DELT(1)/TDEP                                                   STATE 39
      IF (SUMDEP+DEP-1.0) 5004, 5003, 5003                               STATE 40
 5003 DEP=1.0-SUMDEP                                                     STATE 41
 5004 SUMDEP=SUMDEP+DEP                                                  STATE 42
      GO TO 5006                                                         STATE 43
 5005 DEP=0.0                                                            STATE 44
 5006 QDEP=DEP*SPEC(J)                                                   STATE 45
      SUMQE=SUMQE+QDEP*M(J)                                              STATE 46
C     END OF SPECIAL CARDS                                               STATE 47
      ETA=1.0-CES(1,PLATE)/RHO1                                          STATE 48
      IF (ABSF(ETA).LT.5.0E-10) ETA=0.0                                  STATE 49
```

```
      IF ((E(J).LT.CES(32,PLATE)).AND.CES(29,PLATE)) 5008, 5007          STATE 50
C                                                                        STATE 51
C     MATERIAL HAS NO STRENGTH                                           STATE 52
 5007 PHI(J)=DELE=TX=0.0                                                 STATE 53
      GO TO 5021                                                         STATE 54
C                                                                        STATE 55
C     MATERIAL HAS STRENGTH - ELASTIC-PLASTIC YIELDING                   STATE 56
C                                                                        STATE 57
C     COMPUTE G AND Y                                                    STATE 58
 5008 ETA1=1.0-2.0*CES(1,PLATE)/(RHO1+RHO(J))                            STATE 59
      IF (ABSF(ETA1).LT.5.0E-10) ETA1=0.0                                STATE 60
      IF (CES(22,PLATE)) 5010, 5009                                      STATE 61
 5009 G=GCONST(PLATE)*RHO(J)*C(J)**2                                     STATE 62
      GO TO 5012                                                         STATE 63
 5010 SUMG=ETAP=1.0                                                      STATE 64
      NO=CES(22,PLATE)-0.9                                               STATE 65
      DO 5011 IE=1,NO                                                    STATE 66
      ETAP=ETAP*ETA1                                                     STATE 67
 5011 SUMG=SUMG+CES(IE+23,PLATE)*ETAP                                    STATE 68
      G=SUMG*CES(23,PLATE)                                               STATE 69
 5012 Y=CES(30,PLATE)                                                    STATE 70
C                                                                        STATE 71
C     YIELD EQUATION MAY BE CHANGED HERE                                 STATE 72
      IF (CES(29,PLATE).EQ.3.0)  Y=CES(30,PLATE)*(1.0+CES(31,PLATE)*ETA)STATE 73
     1*(1.0-E(J)/CES(32,PLATE))                                          STATE 74
C                                                                        STATE 75
C     COMPUTE DEVIATORS                                                  STATE 76
      IF (LPHA-2) 5013, 5017, 5013                                       STATE 77
```

```
C                                                                       STATE 78
C     ALPHA = 1 OR 3, RECTANGULAR OR SPHERICAL SYMMETRY                 STATE 79
 5013 TXP=P(J)-SIGMA(J)                                                 STATE 80
      DX=2.0*(U1-U(J-NVAR))/(DELXJ+X(J)-XP)+RHODOT/3.0                  STATE 81
      TXI=TXP+2.0*DELT(1)*G*DX                                          STATE 82
      IF (CES(29,PLATE).EQ.2.0) GO TO 5015                              STATE 83
      YIELDF=1.5*TXI**2                                                 STATE 84
      IF (YIELDF.GT.Y**2/1.5) 5014, 5015                                STATE 85
 5014 TX=TXI*Y/(1.5*ABSF(TXI))                                          STATE 86
      GO TO 5016                                                        STATE 87
 5015 TX=TXI                                                            STATE 88
 5016 DELE=1.5*DELT(1)*(TX+TXP)*DX/(RHO1+RHO(J))                        STATE 89
      PHI(J)=1.5*TX                                                     STATE 90
      GO TO 5021                                                        STATE 91
C                                                                       STATE 92
C     ALPHA = 2, CYLINDRICAL SYMMETRY                                   STATE 93
 5017 TXP=P(J)-SIGMA(J)                                                 STATE 94
      DX=2.0*(U1-U(J-NVAR))/(DELXJ+X(J)-XP)+RHODOT/3.0                  STATE 95
      DZ=RHODOT/3.0                                                     STATE 96
      TXI=TXP+2.0*DELT(1)*G*DX                                          STATE 97
      TZI=PHI(J)-2.0*TXP+2.0*DELT(1)*G*DZ                               STATE 98
      IF (CES(29,PLATE).EQ.2.0) GO TO 5019                              STATE 99
      YIELDF=2.0*(TXI**2+TXI*TZI+TZI**2)                                STATE100
      IF (YIELDF.GT.Y**2/1.5) 5018, 5019                                STATE101
 5018 TZ=Y/SQRTF(1.5*YIELDF)                                            STATE102
      TX=TZ*TXI                                                         STATE103
      TZ=TZ*TZI                                                         STATE104
```

```
      GO TO 5020                                                        STATE105
 5019 TX=TXI                                                            STATE106
      TZ=TZI                                                            STATE107
 5020 DELE=DELT(1)/(RHO1+RHO(J))*((TX+TXP)*(2.0*DX+DZ)+(TZ+PHI(J)-2.0*  STATE108
     1TXP)*(2.0*DZ+DX))                                                 STATE109
      PHI(J)=2.0*TX+TZ                                                  STATE110
C                                                                       STATE111
C     FOR ALL THREE EQUATIONS                                           STATE112
C                                                                       STATE113
C     COMPUTE GAMMA                                                     STATE114
 5021 MU=RHO1/CES(1,PLATE)-1.0                                          STATE115
      IF (CES(15,PLATE)) 5023, 5022                                     STATE116
C     GAMMA IS A CONSTANT                                               STATE117
 5022 GAMMA=CES(16,PLATE)                                               STATE118
      GAMMAP=0.0                                                        STATE119
      GO TO 5025                                                        STATE120
C     GAMMA IS A POLYNOMIAL IN ETA                                      STATE121
 5023 SUMGAM=ETAP=1.0                                                   STATE122
      SUMGAP=0.0                                                        STATE123
      NO=CES(15,PLATE)-0.9                                              STATE124
      DO 5024 IE=1,NO                                                   STATE125
      SUMGAP=SUMGAP+IE*CES(IE+16,PLATE)*ETAP                            STATE126
      ETAP=ETAP*ETA                                                     STATE127
 5024 SUMGAM=SUMGAM+CES(IE+16,PLATE)*ETAP                               STATE128
      GAMMA=CES(16,PLATE)*SUMGAM                                        STATE129
      GAMMAP=CES(16,PLATE)*SUMGAP                                       STATE130
 5025 IF (ETA.LE.0.0.AND.CES(7,PLATE).GT.0.0) 5026, 5027                STATE131
```

```
C                                                                       STATE132
C       VAPOR EQUATION                                                  STATE133
 5026 RTMU=SQRTF(MU+1.0)                                                STATE134
      A=CES(7,PLATE)+(GAMMA-CES(7,PLATE))*RTMU                          STATE135
      AP=(GAMMAP*(1.0-ETA)+0.5*(GAMMA-CES(7,PLATE)))/(CES(1,PLATE)*RTMU)STATE136
      B=NCONST(PLATE)*ETA*(1.0-ETA)                                     STATE137
      BP=NCONST(PLATE)*CES(1,PLATE)*(1.0-2.0*ETA)/RHO1**2               STATE138
      F1=A*RHO1*CES(3,PLATE)*(EXPF(B)-1.0)                              STATE139
      F2=A*RHO1                                                         STATE140
      FP2=A+RHO1*AP                                                     STATE141
      FP1=CES(3,PLATE)*(FP2*(EXPF(B)-1.0)+A*BP*RHO1*EXPF(B))            STATE142
      GO TO 5032                                                        STATE143
C                                                                       STATE144
C       HYDRO EQUATION                                                  STATE145
 5027 IF (CES(8,PLATE)) 5029, 5028                                      STATE146
 5028 PHP=CES(9,PLATE)/(1.0-CES(10,PLATE)*ETA)**2                       STATE147
      PH=PHP*ETA                                                        STATE148
      PHP=PHP*(1.0+2.0*CES(10,PLATE)*ETA/(1.0-CES(10,PLATE)*ETA))       STATE149
      GO TO 5031                                                        STATE150
C       PH IS A POLYNOMIAL IN ETA                                       STATE151
 5029 SUMPH=SUMPHP=ETAP=1.0                                             STATE152
      NO=CES(8,PLATE)-0.9                                               STATE153
      DO 5030 IE=1,NO                                                   STATE154
      ETAP=ETAP*ETA                                                     STATE155
      SUMPH=SUMPH+CES(IE+9,PLATE)*ETAP                                  STATE156
 5030 SUMPHP=SUMPHP+(IE+1)*CES(IE+9,PLATE)*ETAP                         STATE157
      PH=SUMPH*CES(9,PLATE)*ETA                                         STATE158
```

```
      PHP=SUMPHP*CES(9,PLATE)                                          STATE159
 5031 F1=PH*(1.0-0.5*GAMMA*MU)                                         STATE160
      F2=GAMMA*RHO1                                                    STATE161
      FP1=CES(1,PLATE)/RHO1**2*(PHP*(1.0-0.5*GAMMA*MU)-0.5*PH*(GAMMA*  STATE162
     1(MU+1.0)**2+GAMMAP*MU))                                          STATE163
      FP2=GAMMA+GAMMAP*(1.0-ETA)                                       STATE164
C                                                                      STATE165
C     ENERGY EQUATION                                                  STATE166
 5032 E1=(E(J)+(F1+P(J)+Q1+Q(J))*DELRHO+DELE+QDEP)/(1.0-F2*DELRHO)     STATE167
      P1=F1+F2*F1                                                      STATE168
      SIGMA1=P1-TX                                                     STATE169
      IF (SIGMA1.GE.CES(4,PLATE)) 5034, 5033                           STATE170
 5033 SIGMA1=CES(4,PLATE)                                              STATE171
      PE=SIGMA1+TX                                                     STATE172
      E1=E1+(PE-P1)*DELRHO                                             STATE173
      P1=PE                                                            STATE174
C                                                                      STATE175
C     COMPUTE SOUND SPEED                                              STATE176
 5034 C(J)=FP1+FP2*E1+F2*P1/RHO1**2                                    STATE177
      IF (C(J)) 5035, 5035, 5036                                       STATE178
 5035 C(J)=CES(2,PLATE)                                                STATE179
      GO TO 5039                                                       STATE180
 5036 IF (E(J).LT.CES(3,PLATE)) 5038, 5037                             STATE181
 5037 C(J)=SQRTF(C(J))                                                 STATE182
      GO TO 5039                                                       STATE183
 5038 C(J)=SQRTF(KCONST(PLATE)*C(J))                                   STATE184
C                                                                      STATE185
C     ADD ANY SPECIAL CALCULATIONS HERE                                STATE186
 5039 E(J)=E1                                                          STATE187
```

```
      P(J)=P1                                                          STATE188
      SIGMA(J)=SIGMA1                                                  STATE189
      GO TO 5999                                                       STATE190
      ENTRY STIN1                                                      STATE191
C                                                                      STATE192
C     INITIALIZE EQUATION OF STATE CONSTANTS                           STATE193
      CES(9,PLATE)=CES(1,PLATE)*CES(2,PLATE)**2                        STATE194
      CES(23,PLATE)=1.5*CES(9,PLATE)*(1.0-2.0*CES(6,PLATE))/           STATE195
     1(1.0+CES(6,PLATE))                                               STATE196
C                                                                      STATE197
C     GCONST IS USED IN THE CONSTANT G EQUATION                        STATE198
C     KCONST IS USED IN THE SOUND SPEED EQUATION IFF EPP               STATE199
C     NCONST IS USED IN THE VAPOR EQUATION                             STATE200
      GCONST(PLATE)=(1.0-2.0*CES(6,PLATE))/(2.0-2.0*CES(6,PLATE))      STATE201
      KCONST(PLATE)=3.0*(1.0-CES(6,PLATE))/(1.0+CES(6,PLATE))          STATE202
      NCONST(PLATE)=CES(2,PLATE)**2/(CES(16,PLATE)*CES(3,PLATE))       STATE203
      IF (CES(32,PLATE).EQ.0.0) CES(32,PLATE)=1.0E 30                  STATE204
      QDEP=0.0                                                         STATE205
      DEP=0.0                                                          STATE206
      PRINT 5201, PLATE                                                STATE207
      IF (W4020) WRITE (21, 5201) PLATE                                STATE208
 5201 FORMAT (66HC THE HYDRO VAPOR ELASTIC PLASTIC EQUATION WILL BE USEDSTATE209
     1 FOR PLATE     , I5)                                             STATE210
 5999 RETURN                                                           STATE211
      END                                                              STATE212
```

```
      SUBROUTINE STATE2                                                  STATE  1
C PROGRAM WONDY                                                          STATE  2
      BANK, (0), STATE2    , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       STATE  3
C     INSERT COMMON CARDS HERE
C     HIGH EXPLOSIVE WITH BURN TIME                                      STATE 27
C                                                                        STATE 28
C     PERFECT GAS DETONATION PRODUCTS                                    STATE 29
 5002 F=CES(4,PLATE)*(T-P(J))/(CES(6,PLATE)*DELTAX(PLATE))               STATE 30
      IF (P(J)    .GE.T) F=0.0                                           STATE 31
      IF (F.GT.1.0)  F=1.0                                             STATE 32
      E1=(E(J)+(SIGMA(J)+Q1+Q(J))*DELRHO)/(1.0-F*RHO1*DELRHO*          STATE 33
     1 CES(9,PLATE))                                                   STATE 34
      SIGMA(J)=F*RHO1*E1*CES(9,PLATE)                                    STATE 35
      C(J)=CES( 3,PLATE)*SIGMA(J)/RHO1                                   STATE 36
      IF (C(J).LT.CES(8,PLATE)) 5003, 5004                             STATE 37
 5003 C(J)=CES(2,PLATE)                                                  STATE 38
      GO TO 5005                                                         STATE 39
 5004 C(J)=SQRTF(C(J))                                                   STATE 40
 5005 E(J)=E1                                                            STATE 41
      GO TO 5999                                                         STATE 42
      ENTRY STIN2                                                        STATE 43
      PRINT 5201, PLATE                                                  STATE 44
      IF (W4020) WRITE (21, 5201) PLATE                                  STATE 45
      LOM=LOL+1                                                          STATE 46
      DO 5001 LF=LOM,LOR                                                 STATE 47
      JF=(LF-1)*NVAR+1                                                   STATE 48
 5001 P(JF)=ABSF(0.5*(X(JF)+X(JF-NVAR))-CES(5,PLATE))/CES(4,PLATE)       STATE 49
```

```
      CES(8,PLATE)=CES(2,PLATE)**2                                      STATE 50
      CES(9,PLATE)=CES(3,PLATE)-1.0                                     STATE 51
 5201 FORMAT (53H0 THE HIGH EXPLOSIVE EQUATION WILL BE USED FOR PLATE    STATE 52
     1    , I5)                                                         STATE 53
 5999 RETURN                                                            STATE 54
      END                                                               STATE 55
```

```
      SUBROUTINE STATE3                                                  STATE  1
C PROGRAM WONDY                                                          STATE  2
      BANK, (0), STATE3    , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       STATE  3
C     INSERT COMMON CARDS HERE
      E1=(E(J)+(SIGMA(J)+Q1+Q(J))*DELRHO)/(1.-RHO1*DELRHO*CES(9,PLATE))  STATE 27
      SIGMA(J)=RHO1*E1*CES(9,PLATE)                                      STATE 28
      C(J)=CES( 3,PLATE)*SIGMA(J)/RHO1                                   STATE 29
      IF (C(J).LT.CES(8,PLATE)) 5001, 5002                               STATE 30
 5001 C(J)=CES(2,PLATE)                                                  STATE 31
      GO TO 5003                                                         STATE 32
 5002 C(J)=SQRTF(C(J))                                                   STATE 33
 5003 E(J)=E1                                                            STATE 34
      GO TO 5999                                                         STATE 35
      ENTRY STIN3                                                        STATE 36
      PRINT 5201, PLATE                                                  STATE 37
      IF (W4020) WRITE (21, 5201) PLATE                                  STATE 38
      CES(8,PLATE)=CES(2,PLATE)**2                                       STATE 39
      CES(9,PLATE)=CES(3,PLATE)-1.0                                      STATE 40
 5201 FORMAT (50HO THE PERFECT GAS EQUATION WILL BE USED FOR PLATE     ,STATE 41
     1 I5)                                                               STATE 42
 5999 RETURN                                                             STATE 43
      END                                                                STATE 44
```

```
      SUBROUTINE OUTPUT                                                  OUTPU  1
C PROGRAM WONDY                                                          OUTPU  2
      BANK, (0), OUTPUT    , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       OUTPU  3
C     INSERT COMMON CARDS HERE
      JMAX=LMAX*NVAR+1                                                   OUTPU 27
      WRITE (23) N, T, (X(J), SIGMA(J), J=1,JMAX,NVAR)                   OUTPU 28
 9005 RETURN                                                             OUTPU 29
      END                                                                OUTPU 30
```

```
      SUBROUTINE OUTL                                                    OUTL    1
C PROGRAM WONDY                                                          OUTL    2
      BANK, (0), OUTL      , /CONSTS/, /INTERM/, /PLATES/, /LOGIC/       OUTL    3
C     INSERT COMMON CARDS HERE
C     SAMPLE CODING FOR A SPECIAL OUTPUT ROUTINE.  ARRAYS MAY BE PRINTEDOUTL   27
C     EVERY TIME CYCLE FOR VARIOUS MESH POINTS.  MESH POINTS MAY BE      OUTL   28
C     LISTED IN THE ADDATA ARRAY (MAXIMUM OF 7)                          OUTL   29
      EQUIVALENCE (LPR,ADDATA)                                           OUTL   30
      DIMENSION LPR(7)                                                   OUTL   31
      IF(JONE) II=1                                                      OUTL   32
      IF(JONE.AND.NONE) WRITE(22,9700)(TITLE(IC),IC=1,10)                OUTL   33
 9502 IF (IFIX(ADDATA(II)).EQ.L) 9503, 9504                              OUTL   34
 9503 WRITE(22,9701) L,T, X(J), U(J), RHO(J), SIGMA(J), PHI(J), Q(J),    OUTL   35
     1E(J)                                                               OUTL   36
      II=II+1                                                            OUTL   37
      IF (II.EQ.8) PRINTL=0                                              OUTL   38
 9504 RETURN                                                             OUTL   39
 9700 FORMAT (70H0 SPECIAL OUTPUT IS PRINTED IN THE ORDER - L T X U RHO OUTL   40
     1SIGMA PHI Q E      / 1X, 10A8)                                     OUTL   41
 9701 FORMAT (I5,8E14.4)                                                 OUTL   42
      END                                                                OUTL   43
```

Distribution:

Lawrence Radiation Laboratories
Livermore, California
Attn: R. J. Wasley

Lawrence Radiation Laboratories
Livermore, California
Attn: J. M. Duff

Lawrence Radiation Laboratories
Livermore, California
Attn: M. Williams

LASL
GMX - 6
Los Alamos, New Mexico
Attn: R. G. McQueen

LASL, GMX - 6
Los Alamos, New Mexico
Attn: J. W. Taylor

LASL, GMX - 6
Los Alamos, New Mexico
Attn: W. Deal

LASL, GMX - 11
Los Alamos, New Mexico
Attn: D. Venable

LASL, T - 5
Los Alamos, New Mexico
Attn: C. L. Mader

LASL, T - 5
Los Alamos, New Mexico
Attn: G. N. White

LASL, T - 3
Los Alamos, New Mexico
Attn: F. H. Harlow

WLRP, Physics Branch
Kirtland Air Force Base, New Mexico
Attn: Chief of Physics Branch

WLRT, Theoretical Branch
Kirtland Air Force Base, New Mexico
Attn: Chief of Theoretical Branch

WLREX
Kirtland Air Force Base
Albuquerque, New Mexico
Attn: A. H. Guenther

AVCO - RAD
Wilmington, Massachusetts
Attn: M. Atkins

AVCO - RAD
Wilmington, Massachusetts
Attn: J. P. Averill

Poulter Lab
Stanford Research Institute
Menlo Park, Calif.
Attn: D. G. Doran

Poulter Lab
Stanford Research Institute
Menlo Park, Calif.
Attn: G. R. Abrahamson

Poulter Lab
Stanford Research Institute
Menlo Park, Calif.
Attn: J. Erkman

General Atomic
San Diego, Calif.
Attn: J. M. Walsh

Aerospace Corporation
San Bernardino, Calif.
Attn: D. Singer

GE Valley Forge Space Tech. Center
Box 8555
Philadelphia 1, Pennsylvania
Attn: Dr. A. F. Lucy

GE Valley Forge Space Tech. Center
Box 8555
Philadelphia 1, Pennsylvania
Attn: T. D. Riney

Distribution: (con't)

Physics International
2700 Mercedes St.
San Leandro, Calif.
Attn: D. Bernstein

General Motors Defense Research Labs.
Goleta, Calif.
Attn: C. Maiden

Gen. Motors Defense Research Labs.
Goleta, Calif.
Attn. A. H. Jones

Applied Theory Inc.
1950 Cotner Ave.
Los Angeles, Calif.
Attn: J. G. Trulio

Shock Hydrodynamics Inc.
2444 Wilshire Blvd.
Santa Monica, Calif.
Attn: R. L. Bjork

Special Project Lab.
Bldg. 61
Picatinney Arsenal
Dover, N. Jersey
Attn. Arcadio Garcia

Naval Ordinance Lab.
White Oak, Maryland
Attn. Julius W. Enig

Kaman Nuclear
Attn: D. C. Williams
Garden of Gods Road
Colorado Springs, Colorado

Kaman Nuclear
Attn. L. Bothell
Garden of Gods Road
Colorado Springs, Colorado

C. Glass
BRL, Aberdeen Proving Ground
Aberdeen, Maryland

W. Troutman, BTL,
Murray Hill, New Jersey

ASRL, Bldg. 41
Massachusetts Institute of Technology
Cambridge, Mass.
Attn: Prof. E. A. Witmer

Dept. of Physics
Washington State Univ.
Pullman, Washington
Attn: Prof. G. Duvall

Missile and Information Systems Division
Boeing Corporation
Aerospace Group
P. O. Box 3985
Seattle, Washington
Attn: D. Young

Missile and Information Systems Division
Boeing Corporation
Aerospace Group
P. O. Box 3985
Seattle, Washington
Attn: G. Butler

U. S. Atomic Energy Commission (3)
Division of Technical Information,
Reports Section
Headquarters Library, G-017
Washington, D. C. 20546

J. E. McDonald, 1110
R. K. Traeger, 1111
G. W. Dyckes, 1113
O. L. Burchett, 1114
L. G. Lee, 1114
T. R. Guess, 1114
C. D. Lundergan, 1115
L. M. Barker, 1115
B. M. Butcher, 1115
C. H. Karnes, 1115
R. P. May, 1115
D. E. Munson, 1115
F. R. Tuler, 1115
Mrs. Emily Young, 1115
W. Herrmann, 1116 (30)
L. D. Bertholf, 1116
Mrs. Pat Holzhauser, 1116
S. W. Key, 1116
W. C. Lyons, 1116
R. B. Reinman, 1116
R. C. Heckman, 1134

Distribution: (con't)

G. A. Carlson, 1134
C. G. Scott, 1316
F. B. Brumley, 1316
A. W. Battaglia, 1325
J. P. Shoup, 1330 (2)
P. J. Komen, 1422
C. S. Williams, 1425
L. M. Melick, 1425
J. A. Cooper, 1425
N. J. Pollard, 1433
J. Duncan, 1433
R. T. Othmer, 1541
D. M. Ellett, 1541
D. W. Doak, 1541
M. J. Forrestal, 1541
W. B. Murfin, 1541
L. T. Wilson, 1541
J. T. Risse, 1542

D. W. Ballard, 2564
E. P. Quigley, 2565
G. E. Seay, 5130
O. E. Jones, 5133
M. Cowan, 5141
E. C. Cnare, 5141
E. H. Beckner, 5142
E. L. Patterson, 5142
C. R. Mehl, 5231
K. G. Adams, 5231
F. Biggs, 5231
D. A. Dahlgren, 5231
J. A. Mogford, 5231

J. H. Renken, 5231
M. L. Merritt, 5232
R. C. Bass, 5232
R. R. Boade, 5232
P. C. Lysne, 5232
W. R. Perret, 5232
W. D. Weart, 5232
H. H. Wicke, 5261
R. J. Thompson, 5262
F. W. Neilson, 5620
S. D. Spray, 5621
G. S. Kimoshita, 5621
A. J. Chabai, 5623
Miss G. V. Barton, 5623
A. Goodman, 5623
D. B. Hayes, 5623
Mrs. L. Kennedy, 5623
R. G. Clem, 5630
A. V. Engel, 5632
R. Wardlaw, 5632

F. H. Mathews, 7343
D. E. Gregson, 8130
J. E. Marion, 8131
G. L. Ludwig, 8131
G. E. Brandvold, 8132
J. M. Brierly, 8133
A. W. McKinney, 8144
J. V. Almstad, 8144
F. J. Cupps, 8144
V. K. Gabrielson, 8144
W. D. Zinke, 8147
L. H. Bakken, 8147, (5)
P. D. Gildea, 8148
A. N. Blackwell, 8149, (5)
Technical File 1100, (2)
Department File 1100
Central Technical File 3428-1

R. S. Gillespie, 3413, (4) for DTIE
W. F. Carstens, 3410
B. R. Allen, 3421
C. H. Sproul, 3428-2 (20)