Internal Report file

Shock Dynamics Laboratory
Physics Department
Washington State University
Pullman, Wa.

Internal Report 73-01

G. E. Duvall

PROGRAM DESCRIPTION FOR "POT"*

The purpose of this program is to enable the operator to calculate the mechanical and thermodynamic states of matter produced by impact of a plane flier plate on a target consisting of a sample and possibly a quartz gage and potting compound. Variables are controlled by the equations for conservation of mass, momentum and energy and by the constitutive relations of the various materials. Experimental control is simulated by varying the impact velocity and flier plate material.

The equations of plane, time-dependent flow in Lagrangian coordinates are

Continuity: $\partial V/\partial t = \partial u/\partial m$ (1)

Motion: $\partial u/\partial t = -\partial(p + q)/\partial m$ (2)

Energy: $\partial E/\partial t = -(p + q)\partial V/\partial t$ (3)

p = compressive stress acting in the x-direction = $-\sigma_{xx}$, megabars. Referred to as "pressure."

V = specific volume in cc/g

u = particle velocity in cm/μsec

E = internal energy in cc/g

q = artificial viscous stress, megabars

$dm = \rho dx$ = mass contained between planes at x, $x + dx$

Four regions are considered. (Figure 1)

Region I is the flier plate.

Region II is the sample.

Region III is quartz.

Region IV is potting compound.

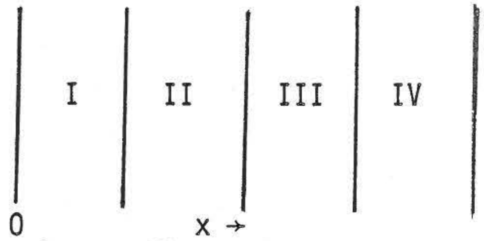


Figure 1

Boundary and Interface Conditions

Pressures at the left boundary of Region I and the right boundary of Region IV are always zero. Particle velocity u and pressure p are continuous at each interface.

*"POT" stands for Projectile on Target

Initial Conditions

Flyer plate velocity is u_o. The particle velocity produced by impact of flier plate on target is v_o.

$u = u_o$ in Region I.

$u = v_o$ at the interface between Regions I and II.

$u = 0$ in Regions II - IV.

$p = 0$ in all Regions.

$E = 0$ in all Regions.

$q = 0$ in all Regions.

$V = V_{ok}$ in Region k, k = I to IV.

Numerical computation is based on a discrete model of the continuum shown in Fig. 1. A representation of this model is shown in Fig. 2. Time increases upward with index n and x increases to the right with index j. Particle velocity is computed on the points marked "x"; specific volume, pressure, energy and other state variables are computed on the points marked "0". Artifical viscosity, q, is computed on the points marked "●".

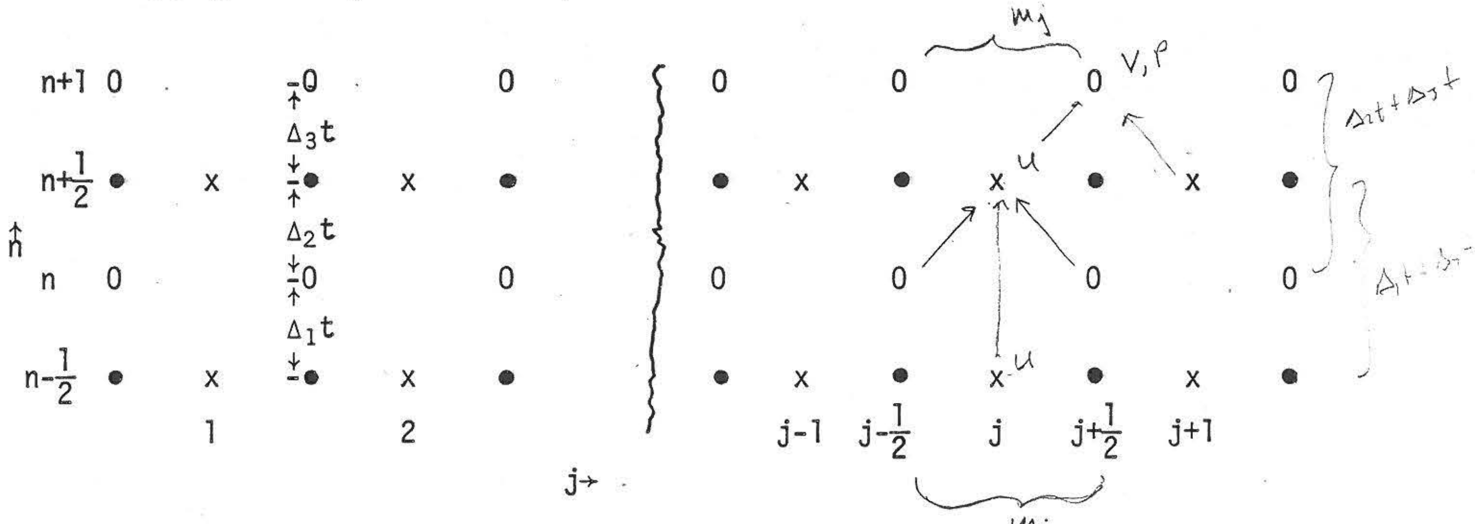


Figure 2. Mesh for numerical computation.

The differential equations can be converted to difference equations which are centered with respect to x and t, except for the viscosity term, q:

$$u_j^{n+1/2} = u_j^{n-1/2} - \frac{\Delta_1 t + \Delta_2 t}{m_j}\left(p^n_{j+1/2} + q^n_{j+1/2} - p^n_{j-1/2} - q^n_{\underset{j}{q}-1/2}\right) \quad (4)$$

$$V^{n+1}_{j+1/2} = V^n_{j+1/2} + \frac{\Delta_2 t + \Delta_3 t}{m_{j+1/2}}\left(u^{n+1/2}_{j+1} - u^{n+1/2}_j\right) \quad (5)$$

m_j = mass contained between planes labelled $j - 1/2$ and $j + 1/2$

= $(1/2)\ (m_{j-1/2} + m_{j + 1/2})$

$m_{j+1/2}$ = mass contained between planes labelled j and $j + 1$

Position x_j is defined at the planes labelled j.

The computational process is restricted by storage capability of the machine and by limitations of the programming language. For example, a typical problem might involve 300 space lattice points and 300 time steps or a total of about 100,000 points. At each point it is necessary to calculate u, p, V, and q, and usually other variables are desired as well. Then upwards of 500,000 words of memory would be required to store all the interesting numbers at each lattice point. While storage of this magnitude is available on modern large machines, its use in this manner is not usually necessary. Examination of Fig. 2 and Eqs. (4) and (5) shows that computation with the time frame shown can be accomplished for all j, given the boundary conditions: if values of u are known at $n - 1/2$, p is known at n, and q at n is assumed to be the same as q at $n - 1/2$, then $u_j^{n+1/2}$ can be computed from Eq. (4) for all j. Following this, $V^{n+1}_{j+1/2}$ can be calculated for all j. From this value, the constitutive relations and the equations for conservation of energy, $p^{n+1}_{j+1/2}$ and $q^{n+1/2}_{j+1/2}$ are calculated. This set of computations is called a cycle.

Now the sets of values at $n + 1/2$ and $n + 1$ are known and their new values at $n + 3/2$ and $n + 2$ can be calculated by repeating the procedure. Values at $n - 1/2$ and n are not needed again in the calculation, so they can be printed and removed from storage at the end of each cycle.

This means that the time indices, $n - 1/2, n, n + 1/2, n + 1 . . .$ can be replaced by only four indices which are used time after time, say 1, 2, 3, 4. Then the numerical mesh is as shown in Fig. 3. The computation proceeds as follows: $\overset{1}{u}_j$, $\overset{2}{V}_{j+1/2}$, $\overset{2}{p}_{j+1/2}$, $\overset{2}{q}_{j+1/2}$ are assumed known for all j. $m_{j+1/2}$ and m_j have been defined initially and do not change with time. Then

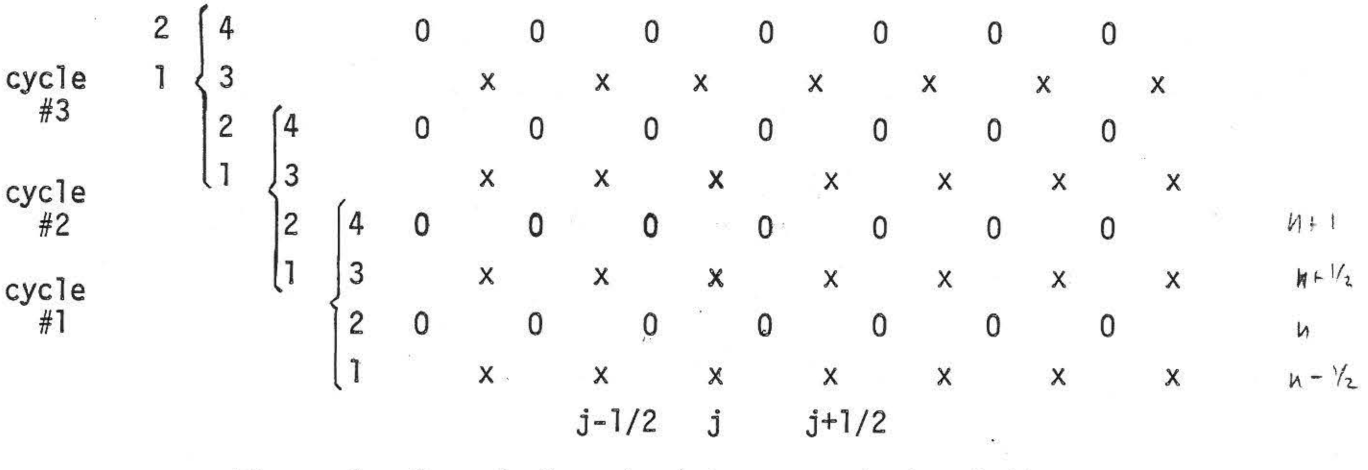


Figure 3. Numerical mesh with repeated time indices

u_j^3 are calculated from Eq. (4) with $u_j^3 \equiv u_j^{n+1/2}$, $u_j^1 \equiv u_j^{n-1/2}$, $p_{j+1/2}^2 \equiv p_{j+1/2}^n$ $q_{j+1/2}^2 \equiv q_{j+1/2}^n$. Then $V_{j+1/2}^4 \equiv V_{j+1/2}^{n+1}$ is calculated from Eq. (4) and values of $p_{j+1/2}^4$, $q_{j+1/2}^4$, $E_{j+1/2}^4$ are obtained from the constitutive relations and the First Law. All values with time indices 3 and 4 are then printed and cycle number 1 is complete. The following substitutions are made and cycle no. 2 proceeds:

$$u_j^1 \leftarrow u_j^3$$

$$V_j^2 \leftarrow V_j^4$$

$$p_j^2 \leftarrow p_j^4$$

$$q_j^2 \leftarrow q_j^4$$

$$E_j^2 \leftarrow E_j^4$$

where "←" is read "is replaced by." The time is recorded at each cycle and the process continues until computation is complete for the desired time interval.

A moment's reflection will show that we can, in fact, dispense with time indices entirely. Eqs. (4) and (5) can be written

$$u_j \leftarrow u_j - \frac{\Delta_1 t + \Delta_2 t}{m_j} (p_{j+1/2} + q_{j+1/2} - p_{j-1/2} - q_{j-1/2}) \qquad (6)$$

$$V_{j+1/2} \leftarrow V_{j+1/2} + \frac{\Delta_{\underset{2}{①}} t + \Delta_{\underset{3}{②}} t}{m_{j+1/2}} (u_{j+1} - u_j) \qquad (7)$$

where it is understood that u_j on the left in Eq. (6) is the "new" value, u_j^3, and u_j, $p_{j\pm1/2}$, $q_{j\pm1/2}$ on the right are "old" values, u_j^1, $p_{j\pm1/2}^2$, $q_{j\pm1/2}^2$. In Eq. (7), $V_{j+1/2}$ on the left is the "new" value, $V_{j+1/2}^4$, while on the right it is the "old" value, $V_{j+1/2}^2$. And u_{j+1}, u_j on the right of Eq. (7) are the "new" values, u_j^3, u_{j+1}^3. There is no ambiguity in this procedure and the new values can be permanently recorded as soon as they are computed. In this way the storage requirements are minimized. In practice it turns out th~~e~~at new values are stored temporarily for use in equation of state calculations.

Limitations of the FORTRAN language impose changes in the space indices which are apt to be confusing. They are described in detail in the following figures and text. They arise from the FORTRAN requirement that indices must be integer and non-zero. The simplest solution is to replace j by 2j, j+1/2 by 2j+1, etc., and to define the left boundary as j=1. This leads to waste of storage space, inasmuch as $U(J) \equiv u_j$ is then calculated only for even values of J, but storage must be reserved for all J. Similarly for V,p, etc. The solution chosen is illustrated in Figs. 4, 5, and 6.

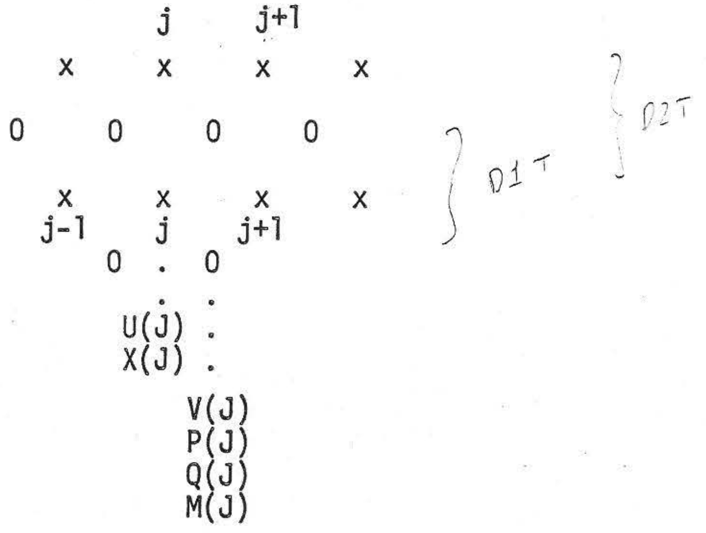


Figure 4. Interior

U(J) and X(J) are identified with the original indices, j. V(J), P(J), Q(J), M(J) and other state variables are identified with original indices j + 1/2. M(J) is the mass contained between J and J + 1. Then Eqs. (6) and (7) become

ll 1

$$U(J) \leftarrow U(J) - \frac{\Delta_1 t + \Delta_2 t}{\frac{1}{2}[M(J) + M(J-1)]} [P(J) + Q(J) - P(J-1) - Q(J-1)] \quad (8)$$

$$V(J) \leftarrow V(J) + \frac{\Delta_2 t + \Delta_3 t}{M(J)} [U(J + 1) - U(J)] \quad (9)$$

The left boundary is pressure-free and, according to the conventions of Fig. 4 and Fig. 5, P(J) = P(0) = 0 at the left boundary. P(0) cannot be defined in FORTRAN, so at the start of each cycle, J = 1 and Eq. (8) is replaced by one written explicitly for the left boundary:

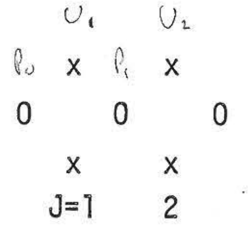


Figure 5. Left Boundary

$$U(1) \leftarrow U(1) - \frac{\Delta_1 t + \Delta_2 t}{M(1)} [P(1) + Q(1)] \quad (10)$$

At the right boundary Eqs. (8) and (9) can be used without modification because $P(J_B)$, $V(J_B)$, etc. are defined as indicated in Fig. 6.

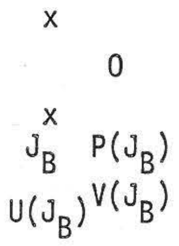


Figure 6. Right Boundary

Representation of the (x,t) continuum by a staggered lattice as in Figs. 2 ff is required if the difference equations are to represent the differential equations to second order in the mesh parameters. This is achieved at the cost of uncertainty in location of the boundaries, as indicated in Figs. 5 and 6. No difficulty occurs if one keeps in mind that the numerical solutions are approximations to the solutions of the differential equations and that variations in the solution which occur on a scale the order of the mesh size and which depend on the mesh size are features of the difference equations, not of the differential equations.

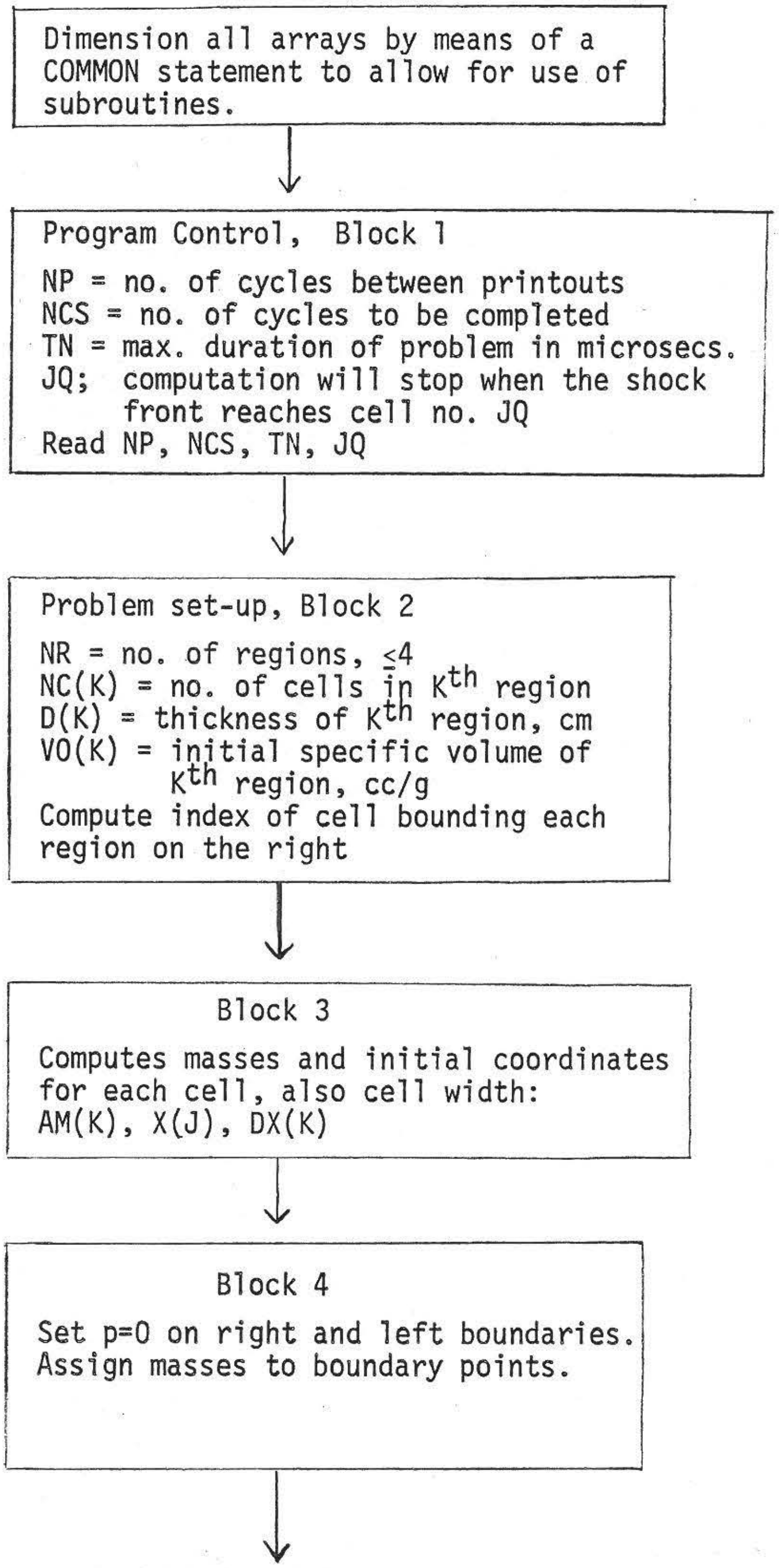
Artificial Viscosity

Artificial viscosity introduced by von Neumann and Richtmyer was quadratic in velocity gradient. It is sometimes useful to use a linear term in addition to the quadratic. The recipes used here have been taken from earlier codes and they appear to function satisfactorily. Coefficients can be changed at will. The essential requirement is that the artificial viscous stress be small when velocity gradient is small and large otherwise, i.e., in the shock front.

Program

Much of the complexity of a 1D program involves program control, initializing procedures, advancing computational parameters in an orderly way, etc. The organization can best be seen from a general flow chart.

FLOW CHART - 1



Dimension all arrays by means of a COMMON statement to allow for use of subroutines.

Program Control, Block 1

NP = no. of cycles between printouts
NCS = no. of cycles to be completed
TN = max. duration of problem in microsecs.
JQ; computation will stop when the shock front reaches cell no. JQ
Read NP, NCS, TN, JQ

Problem set-up, Block 2

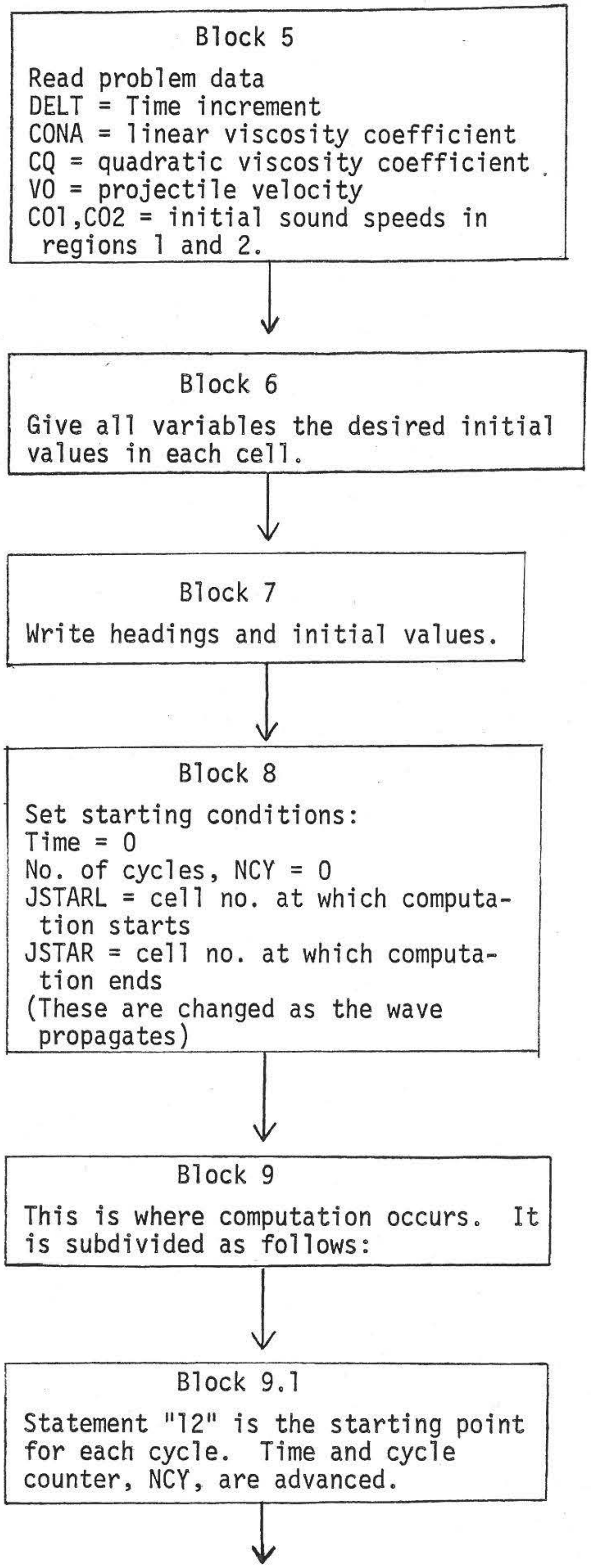
NR = no. of regions, ≤ 4
NC(K) = no. of cells in K^{th} region
D(K) = thickness of K^{th} region, cm
VO(K) = initial specific volume of K^{th} region, cc/g
Compute index of cell bounding each region on the right

Block 3

Computes masses and initial coordinates for each cell, also cell width: AM(K), X(J), DX(K)

Block 4

Set p=0 on right and left boundaries. Assign masses to boundary points.



Block 5

Read problem data
DELT = Time increment
CONA = linear viscosity coefficient
CQ = quadratic viscosity coefficient
VO = projectile velocity
CO1,CO2 = initial sound speeds in regions 1 and 2.

Block 6

Give all variables the desired initial values in each cell.

Block 7

Write headings and initial values.

Block 8

Set starting conditions:
Time = 0
No. of cycles, NCY = 0
JSTARL = cell no. at which computation starts
JSTAR = cell no. at which computation ends
(These are changed as the wave propagates)

Block 9

This is where computation occurs. It is subdivided as follows:

Block 9.1

Statement "12" is the starting point for each cycle. Time and cycle counter, NCY, are advanced.

Block 9.2

The left hand boundary condition is applied here. If JSTARL is greater than 1, the boundary condition is not needed. If it is equal to 1, particle velocity at the advanced time, UN, is calculated with pressure at the left boundary equal to zero.

↓

Block 9.3

This is the interior computation for J > 1. It includes the right hand boundary computation. New values of U,V,Q are computed here.

↓

Block 9.4

The equations of state or constitutive relations for the four regions are inserted here. They are solved simultaneously with the First Law for pressure, p_x, stress deviators and internal energy at the advanced time. Temperature is also calculated. Variable indices are reset here:

```
P(J) ← PN
Q(J) ← QN
V(J) ← VN
E(J) ← EN
T(J) ← TN

   etc.
```

↓

Block 9.5

The decision is made here whether to advance K and continue the computation through another region. If JSTAR has been exceeded, computation jumps out of the K-loop and continues in Block 9.6. If not, it returns to "DO 18 . . ."

↓

Block 9.6

If the wave front has reached JSTAR, JSTAR is advanced one cell. A similar test is made for JSTARL. Neither is changed after it reaches the appropriate boundary. A decision is made whether to print data at this cycle or not:

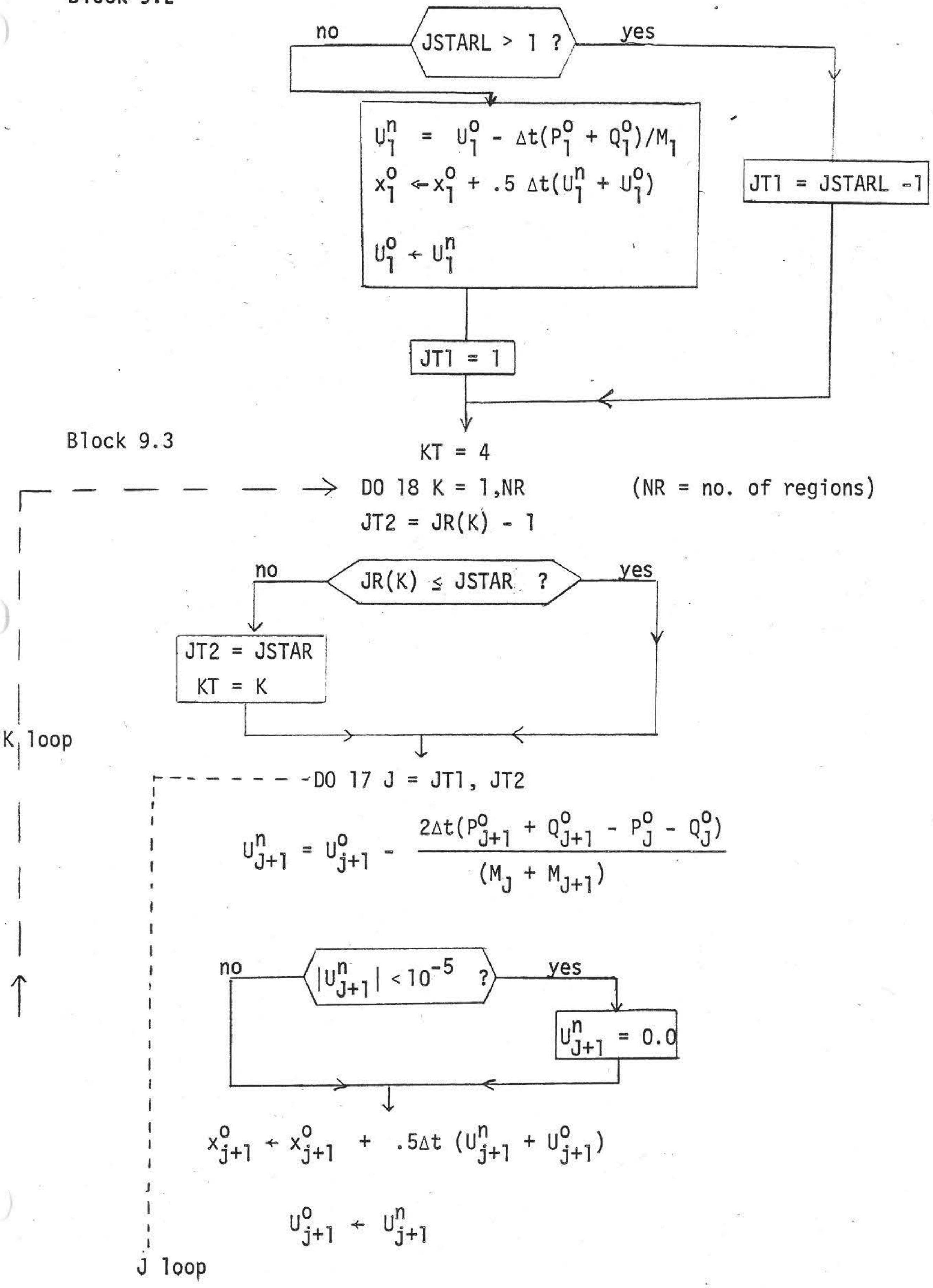
```
      IF(MOD(NCY,NP).NE.0) GO TO 22
```

i.e. if NCY/NP is an integer, print; otherwise not.
A test is made to determine whether or not the computation is to be stopped here.

FLOW CHART - 2

"n" = new, "o" = old

Block 9.2



no
JSTARL > 1 ?
yes
$U_1^n = U_1^o - \Delta t(P_1^o + Q_1^o)/M_1$
$x_1^o \leftarrow x_1^o + .5\ \Delta t(U_1^n + U_1^o)$
$U_1^o \leftarrow U_1^n$
JT1 = JSTARL -1
JT1 = 1
Block 9.3
KT = 4
DO 18 K = 1,NR
(NR = no. of regions)
JT2 = JR(K) - 1
no
JR(K) ≤ JSTAR ?
yes
JT2 = JSTAR
KT = K
K loop
DO 17 J = JT1, JT2
$U_{J+1}^n = U_{j+1}^o - \frac{2\Delta t(P_{J+1}^o + Q_{J+1}^o - P_J^o - Q_J^o)}{(M_J + M_{J+1})}$
no
$|U_{J+1}^n| < 10^{-5}$?
yes
$U_{J+1}^n = 0.0$
$x_{j+1}^o \leftarrow x_{j+1}^o + .5\Delta t\ (U_{j+1}^n + U_{j+1}^o)$
$U_{j+1}^o \leftarrow U_{j+1}^n$
J loop

$$V_j^n = V_j^o + (\Delta t/M_j)\,(U_{j+1}^o - U_j^o)$$

$$\Delta U = U_{j+1}^o - U_j^o$$

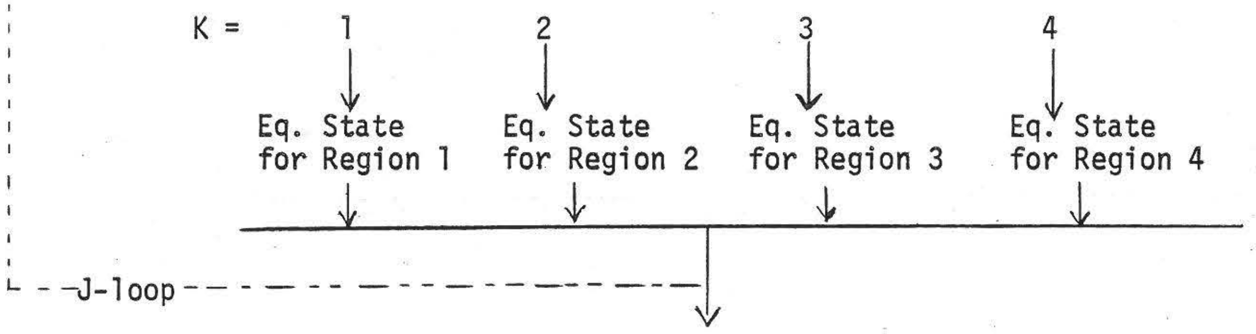
$$\Delta x = x_{j+1}^o - x_j^o$$

$$Q_j^n = -\Delta U \cdot C_q{}^2 \cdot |\Delta U| + c_j^o \cdot C_\ell / V_j^n$$

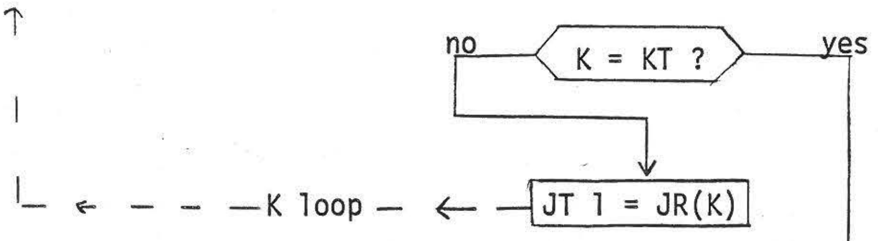
c_j = sound speed

C_ℓ = linear viscosity coefficient

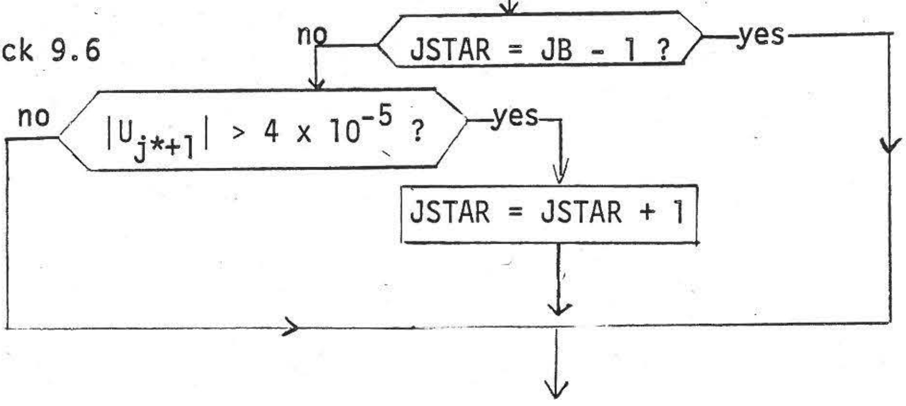
If $Q_j^n < 0, \quad Q_j^n \leftarrow 0$



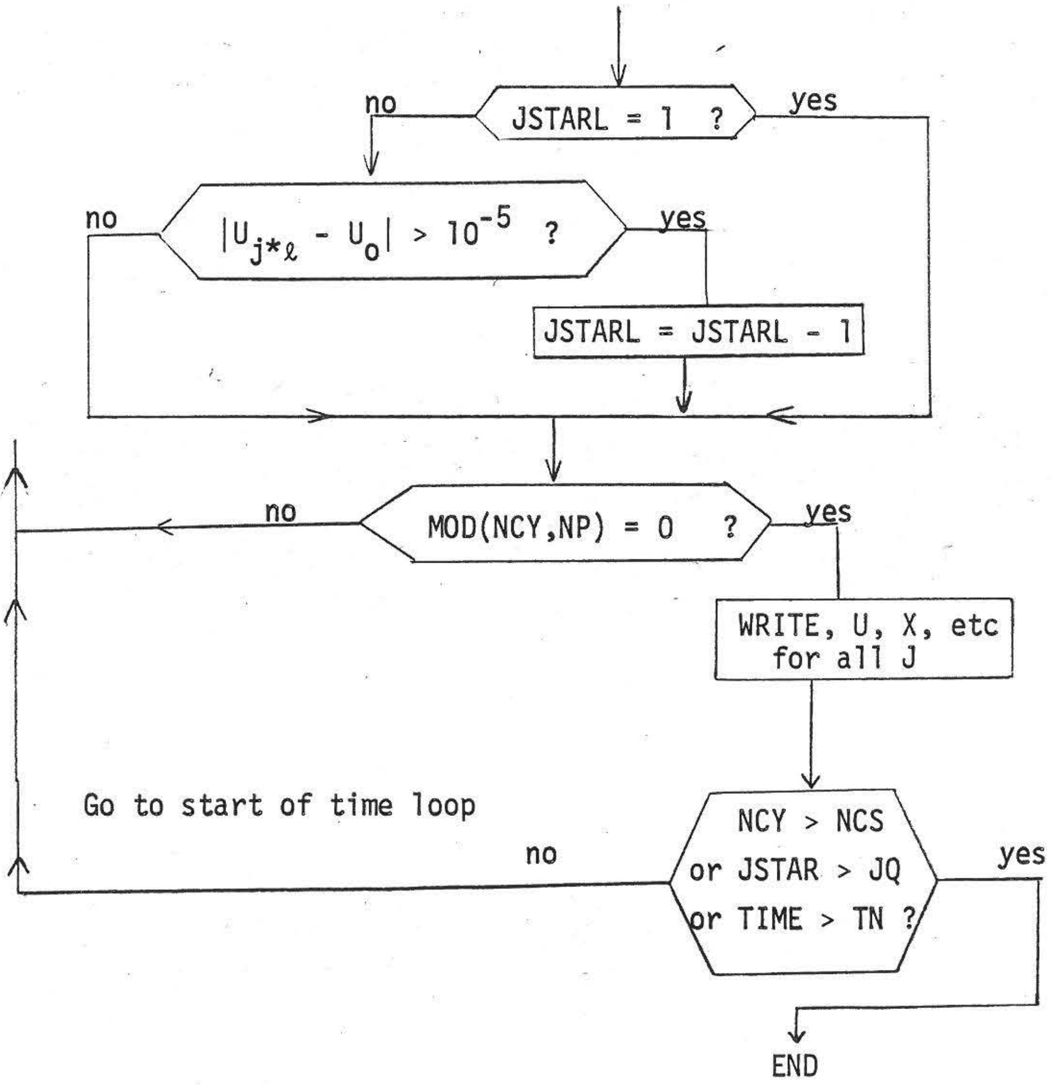
Block 9.5



Block 9.6



JB = No. of cell on right hand boundary


JSTARL = 1 ?
no
yes
$|U_{j*\ell} - U_o| > 10^{-5}$?
no
yes
JSTARL = JSTARL - 1
MOD(NCY,NP) = 0 ?
no
yes
WRITE, U, X, etc for all J
Go to start of time loop
NCY > NCS or JSTAR > JQ or TIME > TN ?
no
yes
END

Equations of State Used in "POT"

The difference equation for Newton's 2nd Law involves the x-component of compressive stress, p_x. (It is denoted by P(J) in the program.) p_x can be resolved into hydrostatic pressure, p, and the x-component of the stress deviator, S_x:

$$p_x = p + S_x \tag{1}$$

The hydrostatic pressure is assumed to be given by a Mie-Grüneisen equation of state in which the Hugoniot (p,V) curve centered at room temperature, atmospheric pressure is a reference:

$$p(E,V) = p_H(V) + (\Gamma/V)(E - E_H(V)) \tag{2}$$

where

$p_H(V)$ is the Hugoniot curve

$$E_H(V) = p_H(V_o - V)/2 \tag{3}$$

V_o = specific volume at room temperature and atmospheric pressure

Γ is the Grüneisen parameter

E is internal energy per unit volume

$\Gamma/V \equiv b$ is assumed to be constant. ?

The stress deviator is assumed to be constant. The yield stress, Y, is assumed to be a linear function of p,

$$Y = a + Cp\ . \tag{4}$$

The increment in internal energy is obtained from the first law, assuming there is no heat transfer from the mass element:

$$dE = -(p_x + Q)dV = -pdV - S_x dV - QdV \tag{5}$$

where Q is artifical viscous stress. Equations (1) to (5) turn out to be interdependent and they can be solved simultaneously, approximately to second order, in the following way.

Values of p,V,Q,S_x,E,T obtained at the previous time, t, are assumed to have been stored as P_j, V_j, Q_j, $S_{x,j}$, E_j, T_j and we are to compute new values for time $t + \Delta t$. Denote the new values by P^n, V^n, Q^n, $S_x{}^n$, E^n, T^n, and the old values by P^o, V^o, Q^o, $S_x{}^o$, E^o, T^o. V^n and Q^n have already been computed, so we enter the equation of state routine with V^n, V^o, P^o, $S_x{}^o$, E^o, T^o, Q^n, Q^o in storage. Define

$$\Delta V = v^n - V^o.$$

Then from Eq. (5)

$$E_n = E^o - (1/2)(P^n + P^o + Q^n + Q^o + S_x{}^n + S_x{}^o)\ \Delta V \tag{6}$$

We write $S_x^n = S_x^o + \Delta S_x$, though ΔS_x has not yet been computed. Then Eq. (6) becomes

$$E^n = E^o - (1/2)\ (P^o + Q^o + 2S_x^o)\Delta V - P^n\Delta V/2 - \Delta S_x \Delta V/2$$

$$= E' - P^n\Delta V/2 - \Delta S_x\Delta V/2 \qquad (7)$$

where $E' \equiv E^o - (P^o + Q^o + Q^n + 2S_x{}^o)\Delta V/2. \qquad (8)$

Equation (2) can be written explicitly for P^n:

$$P^n = P_H^n + b\ E^n - bE_H^n \qquad (9)$$

where

$$P_H = P_H(V^n) \qquad (10)$$

and $E_H^n(V^n)$ is obtained from Eq. (3). Substituting Eq. (7) into (9) yields

$$P^n = \frac{P_H^n + b(E' - E_H^n)}{1 + .5b\Delta V} - \frac{b\Delta S_x\Delta V}{2+b\Delta V} \qquad (11)$$

$$= P^{n_1} - b\Delta S_x\Delta V/(2 + b\Delta V) \qquad (12)$$

where P^{n_1} is the first term on the r.h.s. of Eq. (11). P^{n_1} can be computed with the information already available, but P^n depends on ΔS_x. If the material is in the plastic regime, ΔS_x depends on the change in yield stress between t and t + Δt. Since the yield stress is expected to depend on p and to change only slightly, ΔS_x should be small, $P^n \simeq P^{n_1}$, and $Y(P^{n_1})$ be a good approximation to the correct value of Y. We therefore proceed as follows: First of all assume that the total increment of strain between t and t + Δt is elastic,

$$\Delta\varepsilon_x^e \equiv \Delta\varepsilon_x = -2\Delta V/(V^n + V^o).$$

The x-component of the increment in strain deviator is $2\Delta\varepsilon_x/3$.

Define a temporary quantity

$$\Delta S_x' = (4\mu/3)\ \Delta\varepsilon_x$$

where μ is the rigidity modulus. Then a tentative value of S_x^n is

$$S_x^{n_1} = S_x^o + \Delta S_x'$$

Now calculate Y^n

$$Y^n = Y(P^{n_1}) \tag{13}$$

If $|S_x{}^{n_1}|$ is less than $(2/3)Y^n$, then $\Delta\varepsilon_x$ was indeed elastic and $S_x^n = S_x^{n_1}$. Other·wise $S_x^n = \pm(2/3)Y^n$, since it cannot exceed Y in a rate-dependent process:

$$S_x^n = (2/3)Y^n \text{ if } S_x^{n_1} > 0$$

$$= -(2/3)Y^n \text{ if } S_x^{n_1} < 0$$

Then

$$\Delta S_x = S_x^n - S_x^o \tag{14}$$

$$P^n = P^{n_1} - b\Delta S_x \Delta V/(2 + b\Delta V) \tag{15}$$

This sequence from (13) to (15) can be iterated so as to insure that there is no accumulation of error in S_x.

It is easy at this point to calculate the plastic work. The increment in plastic strain is

$$\Delta\varepsilon_x^p = \Delta\varepsilon_x - \Delta\varepsilon_x^e$$

and

$$\Delta\varepsilon_x^e = (\Delta S_x/2\mu) + \Delta\varepsilon_x/3 .$$

Then

$$\Delta W_p = 3(V^n + V^o)\ (S_x^n + S_x^o)\ \Delta\varepsilon_x^p/8$$

The new internal energy is

$$E^n = E' - p^n\Delta V/2 - \Delta S_x \Delta V/2$$

The increment in temperature is obtained from the expression

$$dT = dE/C_V - (bT - P/C_V)dV$$

or

$$T^n = \{ T^o + \frac{E^n - E^o}{C_V} - \frac{1}{2}[T^o b - (P^n + P^o)/2C_V]\ \Delta V \}/(1 + .5b\Delta V)$$

where C_V is specific heat at constant volume.

Putting this together we have the following program:

$V^n, V^o, P^o, Q^n, Q^o, S_x^o, E^o, T^o, P_x^o$

$$V = V^n - V^o$$

$$E' = E^o - (P^o + Q^n + Q^o + 2S_x^o)\ \Delta V/2$$

$$P_H^n = P_H(V^n)$$

$$E_H^n = P_H^n(V^o - V^n)/2$$

$$P^{n_1} = [P_H^n + b(E' - E_H^n)] \ / \ (1 + .5b\Delta V)$$

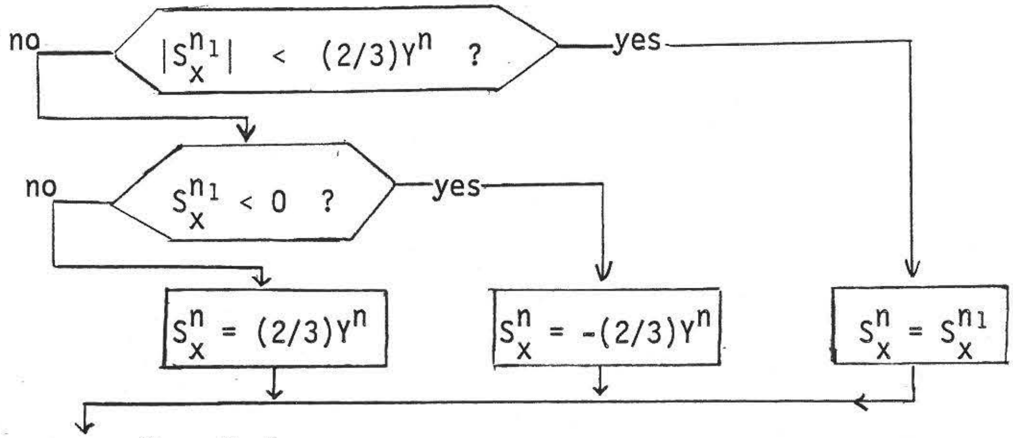
$$\ldots \rightarrow Y^n = Y(P_n')$$

$$\Delta\varepsilon_x = -2\Delta V/(V^n + V^o)$$

$$\Delta S_x' = (4\mu/3)\Delta\varepsilon_x$$

$$S_x^n = S_x + \Delta S_x'$$

Iterate



$$\Delta S_x = S_x^n - S_x^o$$

$$\ldots\ pn = P^{n_1} - b\Delta S_x\Delta V/\ (2 + b\Delta V)$$

$$\Delta\varepsilon_x^e = \Delta S_x/2\mu + \Delta\varepsilon_x/3$$

$$\Delta\varepsilon_x^p = \Delta\varepsilon_x - \Delta\varepsilon_x^e$$

$$P^n_x = P^n + S^n_x$$

$$\Delta W_p = 3(V^n + V^o)\ (S^n_x + S^o_x)\ \Delta\varepsilon^p_x/8$$

$$W^n_p = W^o_p + \Delta W_p$$

$$p^n_x = P^n + S^n_x$$

$$E^n = E' - P^n \Delta V/2 - \Delta S_x\ \Delta V/2$$

$$T^n = \{T^o + E^n - E^o)/C_V + [(P^n + P^o)/\ C_V - bT^o]\Delta V/2\}\ /(1 + b\Delta V/2)$$

In translating this to FORTRAN, we assume the following

$$P_H(V) = c^2\omega/[V_o(1-\omega S)^2];\quad \omega = 1 - V/V_o$$

$$Y(P) = A + CP$$

C_V = constant

μ = constant

b = constant

V_o, c^2, A, C, C_V, μ, b must be read into storage before computation starts. Since material may vary from region to region, each depends on K.

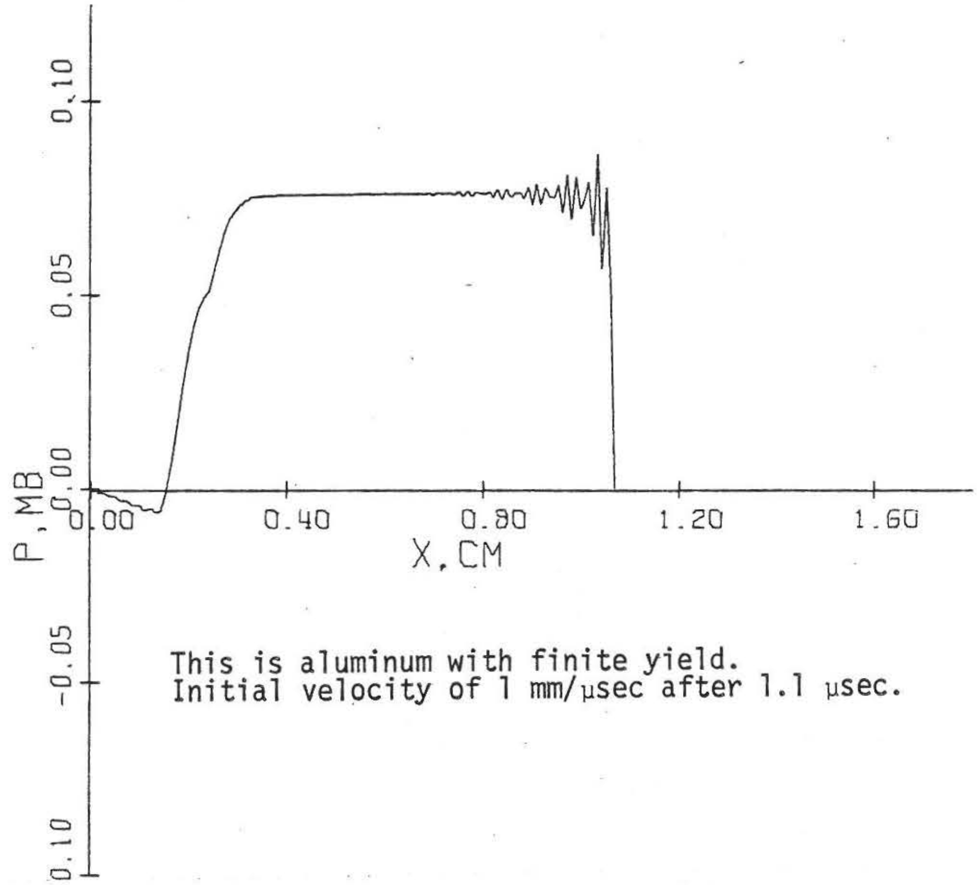
<u>Notation</u>

ΔV	=	DV
V^n	=	VN
V^o	=	V(J)
E'	=	EP
E^o	=	E(J)
P^o	=	Y1(J)
Q^n	=	QN
Q^o	=	Q(J)
S^o_x	=	Y2(J)
c^2	=	CSQ(K)
S	=	S(K)
ω	=	OMEG
P^n_H	=	PH
E^n_H	=	EH
P^{n_1}	=	PN1
V_o	=	VO(K)
b	=	B(K)
Y^n	=	YN
$\Delta\varepsilon_x$	=	DELEP
$\Delta S_x{}'$	=	DELS1
A	=	A(K)
C	=	C(K)
μ	=	MU(K)
$S^{n_1}_x$	=	SN1
S^n_x	=	SN
ΔS_x	=	DELS
$\Delta\varepsilon^e_x$	=	DEPE
$\Delta\varepsilon^p_x$	=	DEPP
ΔW_p	=	DELWP
P^n_x	=	PXN
P^n	=	PN
E^n	=	EN
T^n	=	TN
C_V	=	CV
W^n_p	=	Y3(J)

The program follows. (VO(K) has been read earlier)

```
      READ (5,505)(CSQ(K),S(K),B(K),AMU(K),A(K),C(K),CV(K),K=1,4)
505   FORMAT (7F11.6)
      .
      .
      .
      .
      DV = VN-V(J)
      OMEG = 1. - VN/VO(K)
      EP = E(J) - .5 * (Y1(J) + QN+Q(J) + 2. * Y2(J)) * DV
      PH = (CSQ(K) * OMEG/VO(K)) / ((1. - S(K) * OMEG) ** 2)
      EH = .5 * PH * (VO(K) - VN)
      PN1 = (PH + B(K) * (EP - EH)) / (1. + .5 * B(K) * DV)
      DELEP = -2. * DV/(VN + V(J))
      DELS1 = (4. * MU(K)/3.) * DELEP
      SN1 = Y2(J) + DELS1
      PN = PN1
      SN = SN1
      DO 102 M = 1,3
      YN = A(K) + C(K) * PN
      YT = 2. * YN/3.
      IF(ABS(SN).LT.YT) GO TO 104
      IF(SN.LT.0.0) GO TO 103
      SN = YT
      GO TO 105
103   SN = -YT
      GO TO 105
104   SN = SN1
105   CONTINUE
      DELS = SN - Y2(J)
      PN = PN1 - B(K) * DELS * DV/(2. + B(K) * DV)
102   CONTINUE
      DEPE = DELS/(2. * MU(K)) + DELEP/3.
      DEPP = DELEP - DEPE
      DELWP = .375 * (VN + V(J)) * (SN + Y2(J)) * DEPP
      EN = EP - PN * DV/2. - .5 * DELS * DV
      TN = (T(J) + (EN - E(J))/CV(K) + ((PN + Y1(J))/CV(K)
         -B(K) * T(J)) * DV/2.)/(1. + .5B(K) * DV)
```

```
PXN = PN + SN
P(J) = PXN
T(J) = TN
V(J) = VN
Q(J) = QN
Y1(J) = PN
Y2(J) = SN
Y3(J) = Y3(J) + DELWP
IF(ABS(P(J)).LE.1.0E-5)P(J)=0.0
GO TO 17
```



This is aluminum with finite yield.
Initial velocity of 1 mm/μsec after 1.1 μsec.