INTERNAL REPORT SDL-88-02

NUMERICAL SIMULATION OF WAVE PROPAGATION IN CdS/BAMO:THF COMPOSITE

Z.P. TANG

November 30, 1987

Shock Dynamics Laboratory

Department of Physics

Washington State University

Pullman, WA 99164

Table of Contents

NUMERICAL SIMULATION OF WAVE PROPAGATION IN CdS/BAMO:THF COMPOSITE

Z.P. TANG

Shock Dynamics Laboratory

Department of Physics

Washington State University

November 30, 1987

1. INTRODUCTION

In our recent work,[1] we measured the longitudinal particle velocity-time wave profiles at different locations in impact experiments using the IMPS technique described by Gupta et al.[2] The measured profiles were analyzed to produce the loading and unloading paths of the experiments by means of Lagrangian analysis.[3] The results show that the behavior of this composite under shock loading is directly linked to the properties of the components, i.e. the phase transition of CdS and the viscoelasticity of BAMO:THF. The purpose of the numerical simulation is to present a simple phenomenological mixture model for this composite, which perhaps could be used to predict the material response under impact situations.

The emphasis of this report is put on the description of the mixture model and the use of the computer code. The detail results of the numerical simulation and their comparison with the experimental data will be presented in Ref. 4. The contents of this report are arranged as follows: The next section will present the calculation model. Section 3 will discuss numerical method using simple mixture model. Section 4 will describe the programs and their usage. The source programs related to this calculation will be shown in the Appendix C.

2. MATHEMATICAL MODEL OF CALCULATION

2.1 Simple Mixture Model

The good agreement between the experimental peak data and the Hugoniot based on the simple mixture model of CdS/BAMO:THF composite suggested to use this model in the wave propagation calculation. This model assumed that the stress between particles and matrix is always balanced at any time and no relative movement or interaction between local particles and matrix. We then have:

$$V = \sum_{j=1}^{2} f_j \, V_j \tag{1}$$

where V is the specific volume of the composite, V_j and f_j are the specific volume and mass fraction of the components respectively, j=1 means matrix BAMO:THF and j=2, CdS particles, and $\sum_{j=1}^{2} f_j = 1$.

Since both components are rate dependent materials,[5,6] a generalized Maxwell model is applied to both of them:

$$\dot{P}_j - F_j \, \dot{\epsilon}_j + G_j = 0 \tag{2}$$

where j=1 represents BAMO:THF matrix and j=2, CdS particles. Other symbols in Eq. (2) are, P - pressure, we use pressure instead of σ_z in longitudinal impact experiments since the fluidlike property of the matrix,[5] ϵ - strain, here $\epsilon=(V_o - V)/V_o$, F - instantaneous modulus and G - a term of relaxation function. For a single relaxation time process, we have:

$$G_j = (P_j - P_{ej}) \, / \, \tau_{ij} \tag{3}$$

where P_{ej} is the corresponding equilibrium pressure of the component j for the strain at time t, τ_{ij}

is the relaxation time constant of the same component. Munson et al[7] pointed out that relaxation times are different during loading and unloading procedures for Al_2O_3/epoxy mixture they studied. We also tried to use different relaxation times in the generalized model. In Eq. (3), the subscript $i=l$ means loading relaxation time, and $i=u$ means unloading relaxation time. The numerator $P_j - P_{ej}$ in Eq. (3) is the overstress which depends on the rate of the process.

2.2. Model of the Matrix BAMO:THF

The dynamic response of the elastomer BAMO:THF used in the present work under shock condition was studied by Majewski and Gupta.[8] Its Hugoniot response they fitted to the experimental data is:

$$P = 31.9\mu - 73.7\mu^2 + 1262\mu^3 \quad (kbar) \tag{4}$$

where the compression μ is defined as:

$$\mu = \rho/\rho_o - 1 = V_o/V - 1 \tag{5}$$

Assuming that the Hugoniot described by Eq. (4) is the equilibrium response of the BAMO:THF matrix, and the instantaneous response P_{i1} and equilibrium response P_{e1} have the following relation:

$$P_{i1} = \lambda\, P_{e1} \tag{6}$$

where λ is a constant ratio. From Eqs. (4) and (6), we can obtain the instantaneous modulus for the matrix as:

$$F_1 = \lambda(a + 2b\mu + 3c\mu^2)(\mu+1)^2 \quad (kbar) \tag{7}$$

where $a=31.9$, $b=-73.7$ and $c=1262$.

During unloading, it should be along isentropic curve, but we still use the Hugoniot instead for simplification. The material parameters λ, τ_{l1} and τ_{u1} for matrix BAMO:THF will be determined during the numerical simulation.

2.3. Model for CdS Particles

The basic idea of CdS model used in the calculation is shown in Fig. 1. The equilibrium response of CdS is based on the results of statical experiments of Samara and Giardini.[9] We use

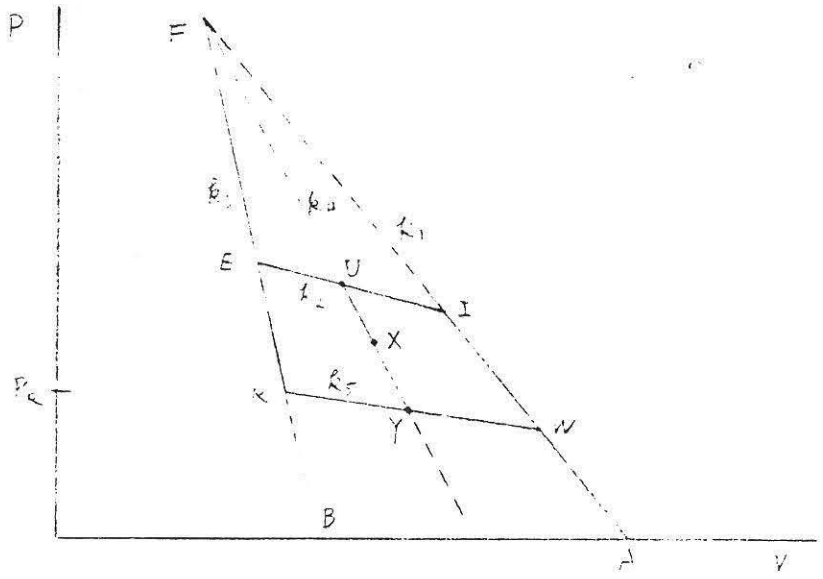


Fig. 1 Calculation model for CdS particles.

linear fitting for the experimental data of phase I (wurtzite), II (rock salt) and mixed phase region for simplification. F in Fig. 1 is the intersection point of the phase I and II fitting lines. The Hugoniot response or the equilibrium response during loading is the solid line AIEF, where I is the initial phase transition point, E is the phase transition finish point. The equilibrium response during unloading from the pressure history above P_E, is assumed along the solid line ERWA, where R

is the phase reverse starting point, and W is the phase reverse finish point. Actually, CdS probably returns to phase I' (zenc blend) during phase reverse. We still use phase I instead of phase I' for simplification, since the volume change is only 3% between phase I and I'. IERW is the hysteresis loop when the peak pressure is above P_E. From the metastable phenemena of c-axis single crystal CdS experiments,[6] we assume the instantaneous response of CdS particles is along its elastic P-V cureve, i.e., along dashed IF during loading and along dashed RB during unloading. For an arbitrary state X inside of the loop, its equilibrium response during loading will be along XUEF, where U is located on line IE, and U is the new phase transition point for state X. During unloading its equilibrium response will be along XYWA, where point Y is the new phase reverse point for state X and is located on line RW. The instantaneous response of the state X is along line XF (slope k_4), no matter loading or unloading. This slope corresponds to the mass fractions and the modulus of the components of phase I and II at the state X in the mixed region. The slope of instanatneous response, k_4 can be calculated by:

$$k_4 = (P_F - P_X) \; / \; (V_F - V_X) \tag{8}$$

The slope k_5 can be calculated from Eq. (9):

$$k_5 = (P_R - P_W) \; / \; (V_R - V_W) \tag{9}$$

Other parameters are as follows:

$$k_1 = -2429.1 \;\; kbar.gm/cc$$
$$k_2 = -221.1 \;\; kbar.gm/cc$$
$$k_3 = -4609.6 \;\; kbar.gm/cc$$
$$P_I = 23.1 \;\; kbar$$
$$P_E = 32.7 \;\; kbar$$
$$P_F = 235.6 \;\; kbar$$
$$P_W = 7.0 \;\; kbar$$
$$V_A = 0.2073 \;\; cc/gm$$

$$V_I = 0.1978 \quad cc/gm$$
$$V_E = 0. \quad cc/gm$$
$$V_F = 0.1103 \quad cc/gm$$
$$V_W = 0.2044 \quad cc/gm$$

All the data above are from the static results of Samara and Giardini[9] or the linear fitting of their results. The phase reverse pressure P_R and the loading and unloading relaxation times τ_{l2} and τ_{u2} will be determined in the numerical simulation.

3. NUMERICAL CALCULATION METHOD USING SIMPLE MIXTURE MODEL OF CdS/BAMO:THF COMPOSITE

The basic equations of wave propagation using Lagrangian coordinates are as following:

$$\frac{\partial x}{\partial X} = \frac{v}{v_o} \tag{10a}$$
$$\frac{\partial x}{\partial t} = u \tag{10b}$$
$$\frac{\partial u}{\partial t} = -v_o \frac{\partial P}{\partial X} \tag{10c}$$
$$\frac{\partial e}{\partial t} = -P \frac{\partial v}{\partial t} \tag{10d}$$

where x - space coordinate, X - Lagrangian coordinate, and v, u, P and e are volume, particle velocity, pressure and internal energy, respectively. Introducing centered finite difference network and artificial viscosity force, the equations above can be writen as:[10]

$$\frac{(u_j^{n+0.5} - u_j^{n-0.5})}{\Delta t^n} = -v_o \frac{(P_{j+0.5}^n + q_{j+0.5}^n - P_{j-0.5}^n - q_{j-0.5}^n)}{\Delta X} \tag{11a}$$
$$\frac{(x_j^{n+1} - x_j^n)}{\Delta t^{n+0.5}} = u_j^{n+0.5} \tag{11b}$$
$$\frac{(x_j^{n+1} - x_{j-1}^{n+1})}{\Delta X} = \frac{v_{j-0.5}^{n+1}}{v_o} \tag{11c}$$
$$e_{j+0.5}^{n+1} = e_{j+0.5}^n - (P_{j+0.5}^{n+0.5} + q_{j+0.5}^{n+0.5})(v_{j+0.5}^{n+1} - v_{j+0.5}^n) \tag{11d}$$

where j means the cell number along Lagrangian coordinate, n represents the layer number along

time coordinate, 0.5 means half step, and q is the artifitial viscosity force which can be calculated by the following equation:

$$q = c1\left(\frac{\partial u}{\partial X}\right)^2 + c2\left(\frac{\partial u}{\partial X}\right) \tag{12}$$

where c1 and c2 are quadratic and linear viscosity coefficients, respectively. The proper values should be chosen through the code examination. We chose c1=2.5 and c2=0.1 in our calculation.

When the quantities x, v, u, P, and e at time layer n and below are known, the quantities at n+1 layer can be calculated using Eq. (11). The computing consequences are as following: first $u_j^{n+0.5}$ from momentum equation, then x_j^{n+1} from the definition of the particle velocity, $v_{j-0.5}^{n+1}$ from mass conservation equation. The pressure at n+1 layer has to be calculated by means of the constitutive equation of the material involved. At last the internal energy from energy equation.

Because $\dot{\epsilon} = -\dot{v}/v_o$, the difference form of generized Maxwell model Eq. (2) can be written as:

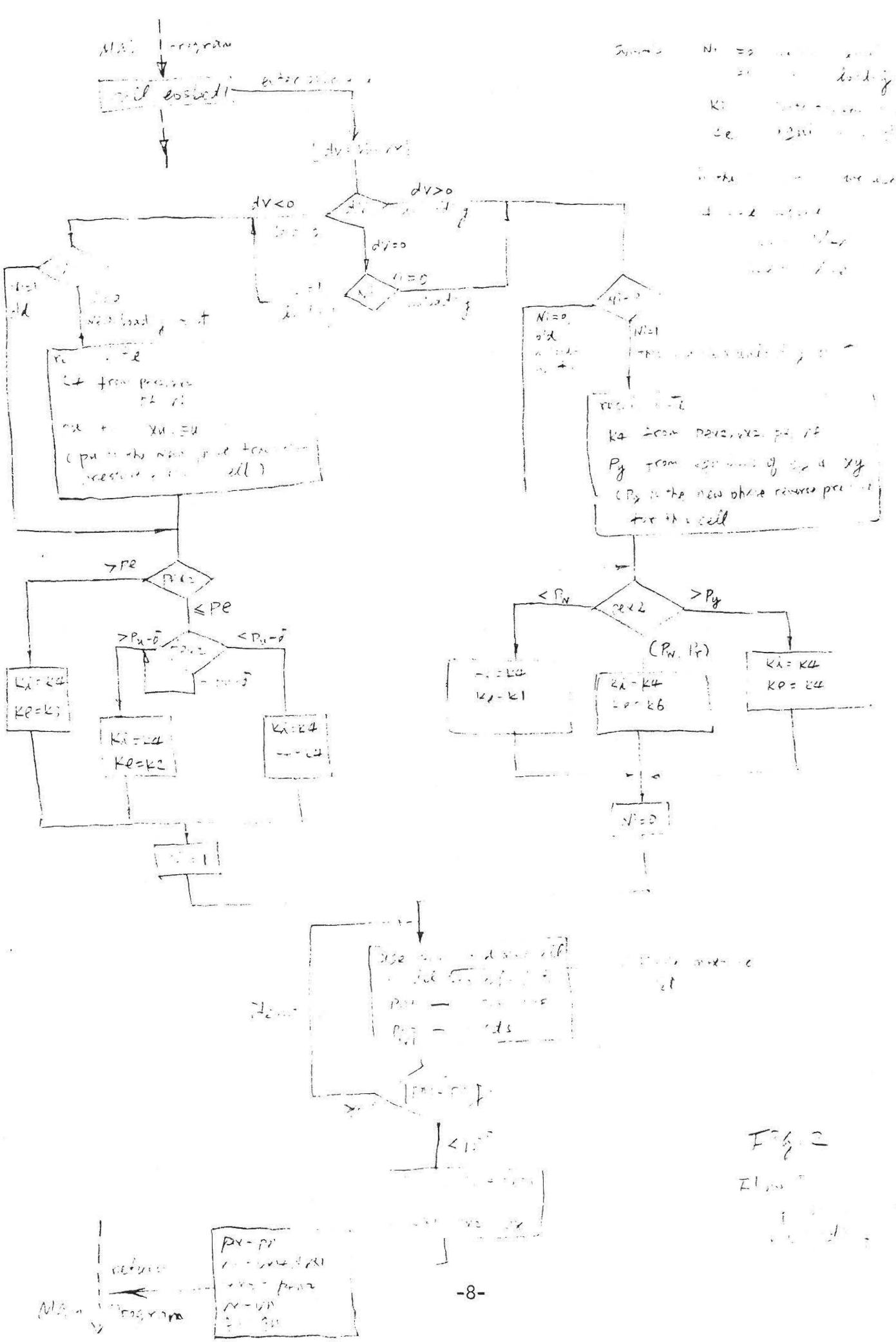
$$\frac{P_j^{n+1}-P_j^n}{\Delta t^n} + \frac{(F_j^{n+1}+F_j^n)(v_j^{n+1}-v_j^n)}{2\Delta t^n v_{oj}} + \frac{P_j^{n+1}+P_j^n-P_{ej}^{n+1}-P_{ej}^n}{2\tau_{ij}} = 0$$

After arrangement, the above equation can be rewritten as:

$$P_j^{n+1} = P_j^n\frac{1-r_j}{1+r_j} + (P_{ej}^{n+1}+P_{ej}^n)\frac{r_j}{1+r_j} - \frac{F_j^{n+1}+F_j^n}{2(1+r_j)}\frac{v_j^{n+1}-v_j^n}{v_{oj}} \tag{13}$$

where $r_j = \Delta t^n/2\tau_{ij}$.

In order to calculate the pressure we have written a subroutine called p4eosbcd1.f for the composite. Before it calls the subroutine of the constitutive equation of the composite in the main program, the quantities vx, vn, px, qx, qn and dt are already known, where x means the values at the previous layer of the cell, n means the values of the cell at current layer. Because CdS has the complicated path related behavior, and both of the components have the rate dependent


dV<0
dV>0
dV=0
>Pe
≤Pe
>PN
<PN
>Py
(PN, Py)

properties, it should save the pressure-strain history of the cell it experienced. In subroutine p4eosbcd1.f, we use variable ni to describe the loading or unloading history of the cell in the subroutine: ni=1 represents laoding and ni=0 unloading. Besides, we use pu to express the new phase transition pressure and py the new phase reverse pressure for a given cell, since pu and py are related to the current pressure-strain state and the history it experienced. vx1 is used to remember the specific volume of component 1 (BAMO:THF) at the previous time layer. The volume change of the composite between these two time layers are:

$$dv = vn - vx$$

According to the sign of dv, one can judge if it is loading or unloading for this step (dv<0, loading; dv>0, unloading). Then according to variable ni which keeps the laoding or unloading history of the previous step, one can judge whether it is new loading or unloading point or not. If it is a new loading or unloading point, it will recalculate pu or py. Then according to the previous equilibrium pressure of CdS pex2 to determine the instantaneous and equilibrium modulus in CdS. At last an iteration method is used to obtain the pressure at new time layer based on the pressure balance assumption between matrix and particles. The related sentences please see the source program of p4eosbcd1.f in Appendix C. The flow diagram of the subroutine is shown in Fig. 2.

4. NEW CODE POT AND ITS USAGE

4.1 Description of the New Code POT

New code POT used for wave propagation calculation in composite CdS/BAMO:THF is based on the POT code developed by Duvall[11] in our laboratory. The old code was originally used for numerical simulation of the impact experiments with four layers and limited varieties of materials, such as sapphire, CS_2 etc. New code is extended to the materials which are rate-dependent and have history effects such as viscoelastic material, composite material with viscoelastic and phase transition properties. The new code includes 10 programs: one main program p4main.f, one

printout subroutine twrite.f and 8 subroutines of constitutive equations: p4eoscs2.f for CS_2, p4eossap.f for sapphire, p4eosqtz.f for fused silica, p4eossvs.f for standard viscoelastiv solids, p4eosbm.f for elastomer BAMO:THF or other elastomer material, p4eoscds1.f for CdS with phase transition or other particle components in composite, and p4eoscmp2.f for other composite materials. All the source programs are written in Fortran language. In this report we only discuss the situation of wave propagation in CdS/BAMO:THF composite impacted by sapphire or fused silica impactors. Hence, we only describe the main program and the output subroutine twrite.f, since the subroutine p4eosbcd1.f has already been discussed in the previous section and will not discuss here. In the Appendix C, only the source programs p4main.f, p4eossap.f, p4eosqtz.f, p4eosbcd1.f and twrite.f, which are directly related to present calculation, are presented. The input data file is shown in the Appendix A and will discuss in the next subsection.

Since the old code POT was originally written for computer SUN operation system, we change all the programs in code POT to meet the requests of the new computer HP 9000 operation system in our laboratory. Besides these common changes, the main changes in main program are listed as below:

a) Add arrays vx1, pu, py, pex2, ski, and ni in the main program, which are used for memorizing the pressure-strain history that every cell itself has experienced. Array ski is the instantaneous modulus of the component CdS at the current pressure-strain state. Other arrays are described in the previous section. The purpose to add these arrays is to keep the history memory for every cell in the main program.

b) Change the calculaton methods of time step length delt and cell number nc(k).

c) Call 8 subroutines of constitutive equations. They are: 1 - CS_2, 2 - sapphire, 3 - fused silica, 4 - standard viscoelastic solid, 5 - elastomer or matrix, 6 - CdS or particles, 7 - composite 1 (CdS/BAMO:THF), and 8 - composite 2.

d) Input all initial parameters in the input file which will be discussed later in this report.

We changed the subroutine twrite.f to output u-t,p-t and p-v results at interfaces 1, 2, and 3 only. The output files of these results are automatically named by the program. They are ut1.*, ut2.* ut3.* for particle velocity - time profiles; pt1.*, pt2.* and pt3.* for pressure - time profiles and pv1.*, pv2.* and pv3.* for pressure - v/v_o curves at three interfaces, where symbol * represents the input data file name. Hence, all the output file ut1.* ... pv3.* are corresponding to the input data file *. Say, if the input file name is st86557.d2, the output files will be ut1.st86557.d2, ... pv3.st86557.d2. Different input file produces automatically different output file names of results. The units in the output files are kbar for pressure, $mm/\mu s$ for particle velocity and μs for time. The format in this files fits the format the graphic software "igraph" requests and the results in these files can be convienently plotted by means of "igraph" in the computer.

The original code POT printed out two big files of results. Since we are mainly interested in the P, v, u and t results, only these results are printed out for convienence. If the output of the results of the other variables are necessary, one can revise the main program and subroutine twrite.f a little bit to meet his own needs.

4.2 Description of the Input Data File

A sample of the input file is listed in the Appendix A. There are 13 data lines to be input to the program in the input file. The meaning of every line is explained in the file (see Appendix A). Only a few lines need to mention here. The first line is the second part of the output file name, normally, it is same as the name of the input file. The main reason we put it on the first line is to present the file name, which links the input data file to the output files of the calculated results. One thing should mention is that the number of the input file name is 10 letters or less, because the whole length of a file name the computer can display on the screen is only 14 letters and the output file name will automatically be added 4 letters to the front of the input file name. Lines 10-13 are the fitting parameters of the materials. If only one line remains non-zero, it represents a single material, say standard viscoelastic solid, elastomer or CdS crystal. If lines 11-13 all remain

non-zero, it represents composite material, say CdS/BAMO:THF. Line 14 is the date the file created or executed. Line 14 is not input into the program and is only for memory.

4.3 Usage of the New Code POT

a) Edite an input data file.

b) Type command "potnew", there will show "Input data file name1=" on the screen. type in the input file name.

c) All the data the computer reads from the input file will be shown on the terminal, check them.

d) After running, the files of calculation results named ut1.* ... pv3.* will exist in the current directiry. You can use command "igraph" to plot the results if you want.

4.4 Sample

Appendix A is the sample of input file of shot86557, where a 3.175 mm thick fused silica impactor was used to impact a CdS/BAMO:THF composite target. Two particle velocity gauges are embedded at 0 mm and 1.9787 mm in the target. The impact velocity measured was 0.903 $mm/\mu s$. Line 4 in the input file shown in Appendix A lists the thicknesses of four pieces. It indicates that three interfaces are located at 0. mm, 1.9787 mm and 3.9787 mm, respectively. The locations of the first two interfaces are same as the locations of the two gauges. So, the simulation results of first two interfeces can be compared with the measured data of the two gauges. Lines 11 - 13 in the input file are the fitting parameters for the composite. Appendix B shows the simulation results u - t, p - t, and p - v/vo at three interfaces. The simulated results are also shown in Figs. 3 - 5 (dotted lines in Figs. 3, solid lines in Fig. 4, dashed line in Fig. 5). The experimental results are also shown in Figs. 3 and 5 (the solid lines). In Fig. 3, the the solid lines are experimental measured profiles, and in Fig. 5, the solid line are calculated from experimental measurements based on

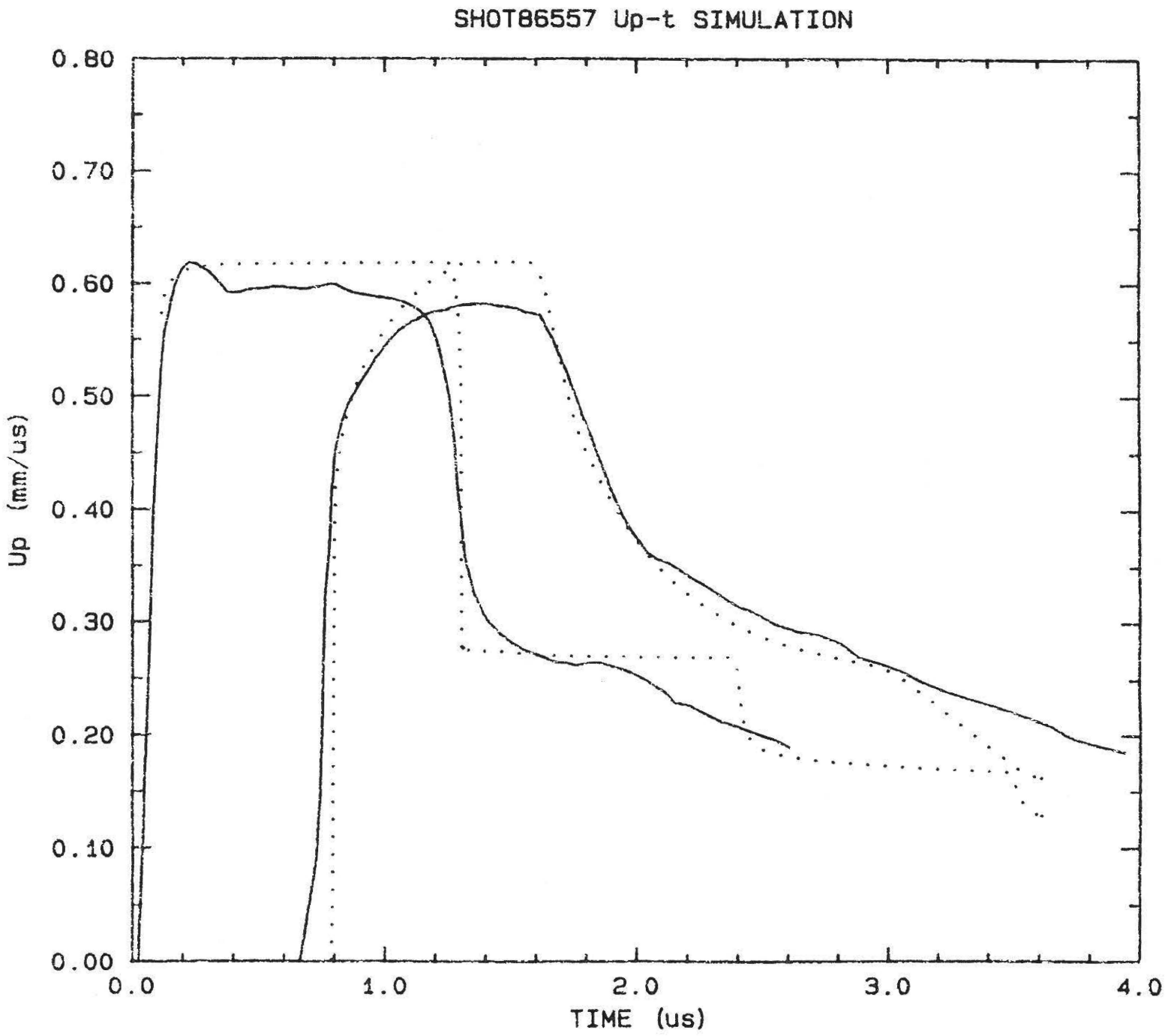

SHOT86557 Up-t SIMULATION
Up (mm/us)
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.0
1.0
2.0
3.0
4.0
TIME (us)

Fig 3.

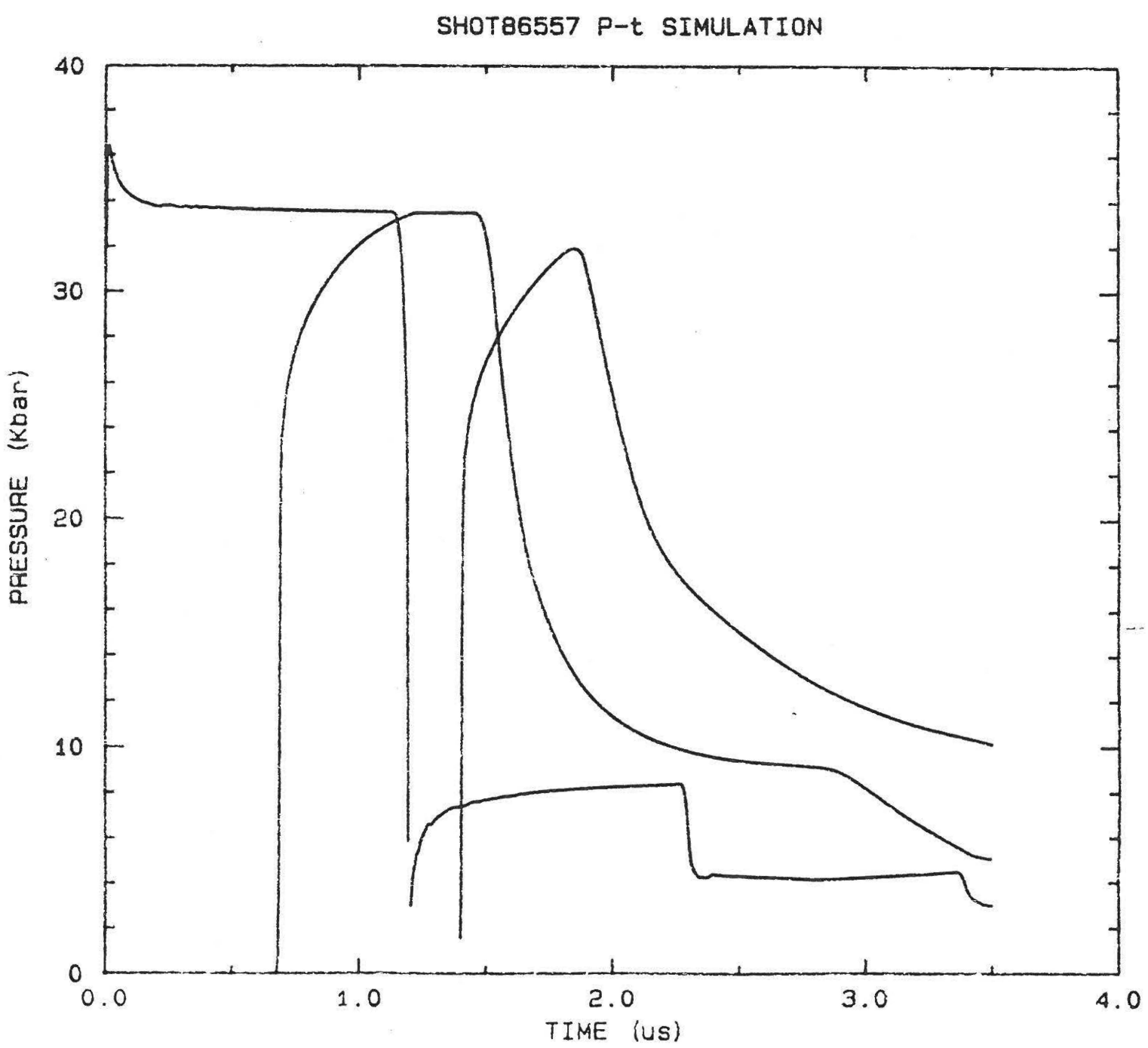
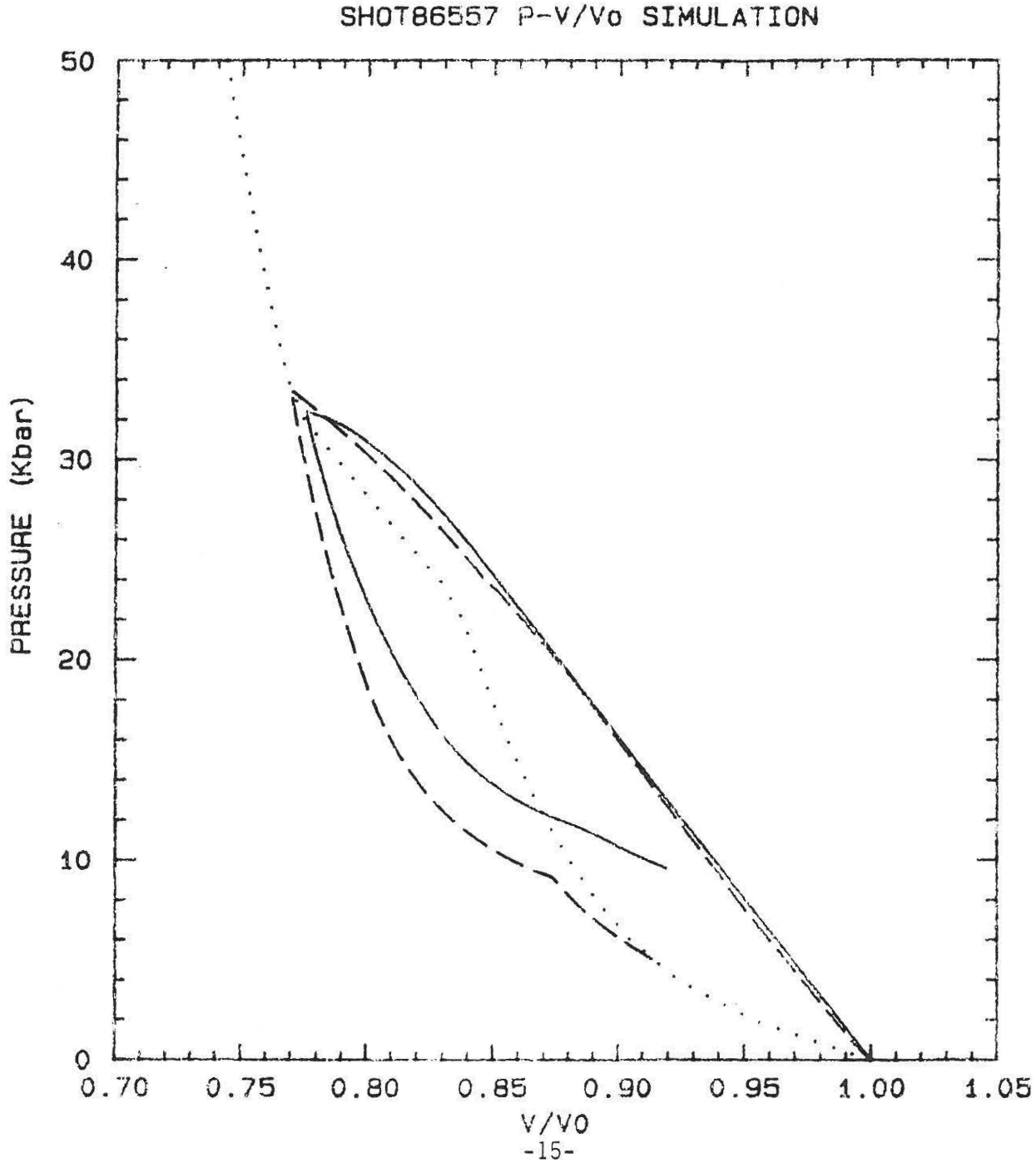

SHOT86557 P-t SIMULATION
PRESSURE (Kbar)
TIME (us)
0
10
20
30
40
0.0
1.0
2.0
3.0
4.0

Fig. 5

---- calculated at [illegible]

—— experimental

..... Hugoniot [illegible]



Lagrangian analysis.[3] The dotted line in Fig. 5 is the Hugoniot of the CdS/BAMO:THF composite using the simple mixture model described in Section 2. Other simulation samples using new code POT and the comparison with the experiments are described in Ref. 4.

ACKNOWLEDGEMENTS

Dr. Y.M. Gupta is sincerely thanked for his helpful advice and suggestion. Dr. G.E. Duvall is also thanked for his help in code POT.

REFERENCES

1. Z.P. Tang and Y.M. Gupta, "Shock response of CdS/BAMO:THF composite," the FiFth APS Topical Conference on SHOCK Waves in Condensed Matter, Monterey, California, July 20 - 23, 1987.
2. Y.M. Gupta et al, Rev. Sci. Instr., 51 (1980) 183.
3. R. Fowles and R.F. Williams, J. Appl. Phys., 41 (1970) 360.
4. Z.P. Tang and Gupta, to be published.
5. Y.M. Gupta, "Compression and shear wave propagation in plain and filled elastomers," Final Report, Shock Dynamics Laboratory, Dept. of Physics, Washington State Univ. September 30, 1986, unpublished.
6. Z.P. Tang and Y.M. Gupta, "Shock induced phase transition in CdS single crystals," to be published.
7. D.E. Muson, R.R. Boade and K.W. Schuler, J. Appl. Phys., 49 (1978) 4797.
8. P.P. Majewski and Y.M. Gupta, "Shock wave response of an energetic elastomer", to be published.
9. G.A. Samara and A.A. Giardini, Phys. Rev. 140 (1965) A388.
10. R.T. Walsh, "Finite-difference method," in "Dynamic Response ofMaterials to Intense Impulsive Loading," Edited by P.C. Chou and A.K. Hopkins, (1973) 363.
11. G.E. Duvall, "Code POT," Shock Dynamics Laboratory, Dept. of Physics, Washington State University.

APPENDIX A:

EX SAMPLE OF INPUT DATA FILE

FILE NAME: st86557.d2

```
st86557.d2                              nam2,output file, normally same as input.
5,10, 3.5,                              nw,np,tn, (2i5,f8.3)
300,200,200,500,                        nc(i), (4i5), cell number
.31750, .19787, .20000, .50000,         d(i), (4f10.5), thickness (cm)
.59000, .13800, .13800, .13800,         co(i), (4f10.5), sound speed (cm/us)
.45372, .49569, .49569, .49569,         vo(i), (4f10.5), volume (cc/g)
.09030, .10000, 2.5000,                 uo,conq,cq, (3f10.5) impact vel. & vis. coef
00, 1.0e-06,    .0000,                  nleftp,zz,pappl, (i5,e10.3,f10.4)
3, 7, 7, 7,                             neos(i),  equation of state
 .0000, .0000, .0000,                   e1,e2,yita, (3e14.4), standard vis. solid
 .0200, .0200,1.5000,                   t11,tu1,rmda (3e14.4), elastomer(BAMO:THF)
 .0100, .0150, .0180,                   t12,tu2,pr (3e14.4), CdS crystal
 .8333,-0.411673,                       vol,sk2 (2e14.6), composite (CdS/BAMO:THF)
DATE: 11/26/1987.
explanation:
This is the input file for new p4main.f which is same as t.f, except that
input file and program have been changed so that nc(k),k=1,2,4, are computed,
ncs and jq are computed, jcol is set to jr(2), jq is set to jr(4)+10, nave
is set to nc(3), delt is computed.  The above numbers can be interpre-
ted with the aid of the following table:

nam2=output name, 10 letters or less.
nw=print control parameter.
np=no. of cycles between printouts
tn=length of run in microsec
nc(3)=no of cells in sample, region 3
d(k)=thickness of kth region in cm
co(k)=initial sound velocity in region k,cm/microsec
vo(k)=specific vol of region k, cc/g
cona=linear viscosity coeff
cq=quadratic viscosity coeff
uo=projectile velocity, cm/microsec
nleftp=control parameter for pressure on left boundary.  set to zero
zz 'noise' control parameter set to  e-6
pappl pressure on left bdry. set to zero.
neos(k)=identification index for eos of region k. 1=cs2,2=sapphire,
        3=qtz, 4=standard viscoelastic solid, 5=elastomer(BAMO:THF),
        6=CdS crystal, 7=composite 1(bcd), 8=composite 2.
e1,e2,yita=elastic modulas, and viscosity coefficient of standard viscoelastic
           solid.
t11,tu1,rmda=relaxation times (us) of loading and unloading, ratio of the
             instantaneous and equilibrium functions of elastomer
             (say BAMO:THF).
t12,tu2,pr=relaxation times (us) of CdS crystal of loading and unloading and
           phase reverse pressure (Mbar).
vol,sk2= for composite, specific volume (cc/g) of component 1 and instantaneous
         slope of the second component(kbar/cm/g).
```

APPENDIX B

SAMPLE OF OUTPUT FILES

FILE NAME: utl.st86557.d2*

t(us)	u(mm/us)
.11739623E+00	.57365000E+00
.12176683E+00	.58561599E+00
.12619339E+00	.58757102E+00
.13069260E+00	.59013999E+00
.13520490E+00	.59232301E+00
.13979320E+00	.59496498E+00
.14438280E+00	.59737200E+00
.14904399E+00	.59915501E+00
.15370570E+00	.60075200E+00
.15844010E+00	.60231698E+00
.16316789E+00	.60377300E+00
.16795270E+00	.60502499E+00
.17273270E+00	.60622197E+00
.17754790E+00	.60721898E+00
.18235940E+00	.60792899E+00
.18718921E+00	.60859799E+00
.19201970E+00	.60941601E+00
.19685490E+00	.61012799E+00
.20168810E+00	.61065501E+00
.20651489E+00	.61111897E+00
.21133479E+00	.61156797E+00
.21614900E+00	.61202902E+00
.22095899E+00	.61246097E+00
.22576401E+00	.61283898E+00
.23056599E+00	.61316198E+00
.23536400E+00	.61344099E+00
.24015801E+00	.61374497E+00
.24494900E+00	.61407900E+00
.24973701E+00	.61437500E+00
.25452200E+00	.61463100E+00
.25930598E+00	.61486101E+00
.26408699E+00	.61505002E+00
.26886800E+00	.61520600E+00
.27364600E+00	.61534899E+00
.27842298E+00	.61549300E+00
.28319898E+00	.61563402E+00
.28797299E+00	.61579102E+00
.29274499E+00	.61598599E+00
.29751599E+00	.61618501E+00
.30228499E+00	.61637002E+00

* only first 40 lines are listed.

FILE NAME: ut2.st86557.d2*

t(us)	u(mm/us)
.79018998E+00	.18767701E-02
.79468501E+00	.41398801E-01
.79918098E+00	.20312899E+00
.80367601E+00	.38000500E+00
.80817097E+00	.42172399E+00
.81266701E+00	.43064100E+00
.81716198E+00	.43937200E+00
.82165802E+00	.44759700E+00
.82615399E+00	.45452201E+00
.83064997E+00	.46056399E+00
.83514601E+00	.46613401E+00
.83964199E+00	.47125599E+00
.84413797E+00	.47592500E+00
.84863400E+00	.48023000E+00
.85312998E+00	.48425499E+00
.85762596E+00	.48803401E+00
.86212200E+00	.49158999E+00
.86661798E+00	.49495199E+00
.87111402E+00	.49814901E+00
.87561101E+00	.50120002E+00
.88010800E+00	.50412101E+00
.88460398E+00	.50692099E+00
.88910097E+00	.50959998E+00
.89359701E+00	.51218200E+00
.89809400E+00	.51468599E+00
.90258998E+00	.51711202E+00
.90708697E+00	.51946700E+00
.91158396E+00	.52175802E+00
.91608000E+00	.52398801E+00
.92057699E+00	.52616298E+00
.92507297E+00	.52828598E+00
.92956996E+00	.53035998E+00
.93406796E+00	.53238797E+00
.93856502E+00	.53437299E+00
.94306201E+00	.53631699E+00
.94755900E+00	.53822100E+00
.95205599E+00	.54008800E+00
.95655298E+00	.54191798E+00
.96105099E+00	.54371399E+00
.96554798E+00	.54547799E+00

* only first 40 lines are listed.

FILE NAME: ut3.st86557.d2*

t(us)	u(mm/us)
.15140200E+01	.64670600E-01
.15186000E+01	.24276200E+00
.15231900E+01	.38531199E+00
.15277700E+01	.40817299E+00
.15323499E+01	.41807899E+00
.15369300E+01	.42558101E+00
.15415200E+01	.43275800E+00
.15461000E+01	.43910700E+00
.15506800E+01	.44455299E+00
.15552599E+01	.44943699E+00
.15598400E+01	.45393300E+00
.15644300E+01	.45805201E+00
.15690100E+01	.46182999E+00
.15735900E+01	.46533599E+00
.15781800E+01	.46862701E+00
.15827700E+01	.47172800E+00
.15873700E+01	.47466099E+00
.15919600E+01	.47745100E+00
.15965700E+01	.48011699E+00
.16011800E+01	.48267600E+00
.16058000E+01	.48514301E+00
.16104300E+01	.48752800E+00
.16150600E+01	.48983699E+00
.16196899E+01	.49207801E+00
.16243200E+01	.49425900E+00
.16289600E+01	.49638301E+00
.16335900E+01	.49845800E+00
.16382200E+01	.50048900E+00
.16428500E+01	.50247997E+00
.16474800E+01	.50443798E+00
.16521100E+01	.50637299E+00
.16567400E+01	.50827801E+00
.16613799E+01	.51013798E+00
.16660100E+01	.51195598E+00
.16706400E+01	.51374602E+00
.16752700E+01	.51551300E+00
.16798999E+01	.51725698E+00
.16845300E+01	.51897699E+00
.16891600E+01	.52067602E+00
.16937900E+01	.52235401E+00

* only first 40 lines are listed.

FILE NAME: pt1.st86557.d2*

t(us)	P(kbar)
.539623E-02	.319375E+02
.976683E-02	.363770E+02
.141934E-01	.364562E+02
.186926E-01	.361419E+02
.232049E-01	.358675E+02
.277932E-01	.356897E+02
.323828E-01	.355074E+02
.370440E-01	.353266E+02
.417057E-01	.352001E+02
.464401E-01	.350761E+02
.511679E-01	.349298E+02
.559527E-01	.348091E+02
.607327E-01	.347144E+02
.655479E-01	.346309E+02
.703594E-01	.345597E+02
.751892E-01	.344958E+02
.800197E-01	.344247E+02
.848549E-01	.343595E+02
.896881E-01	.343083E+02
.945149E-01	.342707E+02
.993348E-01	.342316E+02
.104149E+00	.341831E+02
.108959E+00	.341426E+02
.113764E+00	.341122E+02
.118566E+00	.340833E+02
.123364E+00	.340544E+02
.128158E+00	.340240E+02
.132949E+00	.339950E+02
.137737E+00	.339660E+02
.142522E+00	.339403E+02
.147306E+00	.339216E+02
.152087E+00	.339055E+02
.156868E+00	.338903E+02
.161646E+00	.338769E+02
.166423E+00	.338651E+02
.171199E+00	.338513E+02
.175973E+00	.338337E+02
.180745E+00	.338150E+02
.185516E+00	.337964E+02
.190285E+00	.337772E+02

* only first 40 lines are listed.

FILE NAME: pt2.st86557.d2*

t(us)	P(kbar)
.678190E+00	.197724E-01
.682685E+00	.879600E+00
.687181E+00	.732079E+01
.691676E+00	.189672E+02
.696171E+00	.235422E+02
.700667E+00	.239190E+02
.705162E+00	.245208E+02
.709658E+00	.249866E+02
.714154E+00	.253485E+02
.718650E+00	.256903E+02
.723146E+00	.260078E+02
.727642E+00	.262866E+02
.732138E+00	.265366E+02
.736634E+00	.267687E+02
.741130E+00	.269848E+02
.745626E+00	.271848E+02
.750122E+00	.273710E+02
.754618E+00	.275462E+02
.759114E+00	.277116E+02
.763611E+00	.278684E+02
.768108E+00	.280174E+02
.772604E+00	.281603E+02
.777101E+00	.282990E+02
.781597E+00	.284317E+02
.786094E+00	.285589E+02
.790590E+00	.286816E+02
.795087E+00	.288000E+02
.799584E+00	.289146E+02
.804080E+00	.290256E+02
.808577E+00	.291333E+02
.813073E+00	.292381E+02
.817570E+00	.293400E+02
.822068E+00	.294392E+02
.826565E+00	.295360E+02
.831062E+00	.296303E+02
.835559E+00	.297224E+02
.840056E+00	.298123E+02
.844553E+00	.299001E+02
.849051E+00	.299859E+02
.853548E+00	.300699E+02

* only first 40 lines are listed.

FILE NAME: pt3.st86557.d2*

t(us)	P(kbar)
.140202E+01	.158279E+01
.140660E+01	.941978E+01
.141119E+01	.198082E+02
.141577E+01	.224904E+02
.142035E+01	.229124E+02
.142493E+01	.233716E+02
.142952E+01	.238155E+02
.143410E+01	.241604E+02
.143868E+01	.244589E+02
.144326E+01	.247393E+02
.144784E+01	.249937E+02
.145243E+01	.252202E+02
.145701E+01	.254277E+02
.146159E+01	.256212E+02
.146618E+01	.258018E+02
.147077E+01	.259704E+02
.147537E+01	.261291E+02
.147996E+01	.262795E+02
.148457E+01	.264228E+02
.148918E+01	.265599E+02
.149380E+01	.266915E+02
.149843E+01	.268184E+02
.150306E+01	.269410E+02
.150769E+01	.270596E+02
.151232E+01	.271746E+02
.151696E+01	.272865E+02
.152159E+01	.273954E+02
.152622E+01	.275015E+02
.153085E+01	.276052E+02
.153548E+01	.277058E+02
.154011E+01	.278033E+02
.154474E+01	.278995E+02
.154938E+01	.279959E+02
.155401E+01	.280914E+02
.155864E+01	.281852E+02
.156327E+01	.282773E+02
.156790E+01	.283679E+02
.157253E+01	.284570E+02
.157716E+01	.285448E+02
.158179E+01	.286312E+02

* only first 40 lines are listed.

FILE NAME: pv1.st86557.d2*

v/vo	P(kbar)
.854608E+00	.319375E+02
.822135E+00	.363770E+02
.816984E+00	.364562E+02
.813203E+00	.361419E+02
.809827E+00	.358675E+02
.806715E+00	.356897E+02
.804005E+00	.355074E+02
.801553E+00	.353266E+02
.799295E+00	.352001E+02
.797243E+00	.350761E+02
.795442E+00	.349298E+02
.793765E+00	.348091E+02
.792223E+00	.347144E+02
.790802E+00	.346309E+02
.789486E+00	.345597E+02
.788264E+00	.344958E+02
.787159E+00	.344247E+02
.786139E+00	.343595E+02
.785179E+00	.343083E+02
.784273E+00	.342707E+02
.783435E+00	.342316E+02
.782671E+00	.341831E+02
.781953E+00	.341426E+02
.781271E+00	.341122E+02
.780633E+00	.340833E+02
.780035E+00	.340544E+02
.779476E+00	.340240E+02
.778952E+00	.339950E+02
.778460E+00	.339660E+02
.777994E+00	.339403E+02
.777549E+00	.339216E+02
.777126E+00	.339055E+02
.776725E+00	.338903E+02
.776344E+00	.338769E+02
.775981E+00	.338651E+02
.775640E+00	.338513E+02
.775321E+00	.338337E+02
.775022E+00	.338150E+02
.774742E+00	.337964E+02
.774480E+00	.337772E+02

* only first 40 lines are listed.

FILE NAME: pv2.st86557.d2*

v/vo	P(kbar)
.999835E+00	.197724E-01
.992842E+00	.879600E+00
.951485E+00	.732079E+01
.882048E+00	.189672E+02
.850807E+00	.235422E+02
.846624E+00	.239190E+02
.842933E+00	.245208E+02
.839904E+00	.249866E+02
.837295E+00	.253485E+02
.834913E+00	.256903E+02
.832752E+00	.260078E+02
.830798E+00	.262866E+02
.829005E+00	.265366E+02
.827337E+00	.267687E+02
.825779E+00	.269848E+02
.824318E+00	.271848E+02
.822940E+00	.273710E+02
.821633E+00	.275462E+02
.820387E+00	.277116E+02
.819196E+00	.278684E+02
.818054E+00	.280174E+02
.816953E+00	.281603E+02
.815887E+00	.282990E+02
.814856E+00	.284317E+02
.813856E+00	.285589E+02
.812885E+00	.286816E+02
.811939E+00	.288000E+02
.811018E+00	.289146E+02
.810118E+00	.290256E+02
.809240E+00	.291333E+02
.808380E+00	.292381E+02
.807538E+00	.293400E+02
.806713E+00	.294392E+02
.805904E+00	.295360E+02
.805110E+00	.296303E+02
.804332E+00	.297224E+02
.803567E+00	.298123E+02
.802816E+00	.299001E+02
.802078E+00	.299859E+02
.801353E+00	.300699E+02

* only first 40 lines are listed.

FILE NAME: pv3.st86557.d2*

v/vo	P(kbar)
.987465E+00	.158279E+01
.936765E+00	.941978E+01
.872679E+00	.198082E+02
.852112E+00	.224904E+02
.848710E+00	.229124E+02
.845496E+00	.233716E+02
.842693E+00	.238155E+02
.840331E+00	.241604E+02
.838232E+00	.244589E+02
.836325E+00	.247393E+02
.834600E+00	.249937E+02
.833031E+00	.252202E+02
.831584E+00	.254277E+02
.830236E+00	.256212E+02
.828975E+00	.258018E+02
.827788E+00	.259704E+02
.826665E+00	.261291E+02
.825596E+00	.262795E+02
.824573E+00	.264228E+02
.823591E+00	.265599E+02
.822643E+00	.266915E+02
.821726E+00	.268184E+02
.820836E+00	.269410E+02
.819972E+00	.270596E+02
.819130E+00	.271746E+02
.818309E+00	.272865E+02
.817506E+00	.273954E+02
.816720E+00	.275015E+02
.815948E+00	.276052E+02
.815193E+00	.277058E+02
.814452E+00	.278033E+02
.813722E+00	.278995E+02
.813001E+00	.279959E+02
.812288E+00	.280914E+02
.811584E+00	.281852E+02
.810890E+00	.282773E+02
.810205E+00	.283679E+02
.809528E+00	.284570E+02
.808859E+00	.285448E+02
.808198E+00	.286312E+02

* only first 40 lines are listed.

APPENDIX C

SOURCE PROGRAMS OF CODE POT USED IN NUMERICAL SIMULATION IN CdS/BAMO:THF COMPOSITE

1. MAIN PROGRAM: p4main.f

```
c  units are megabars,cm/microsec,g/cc.
c
c name is     p4main.f
      common/region/nm(4),nc(4),d(4),vo(4),jr(4),dx(4)
      common/field/x(4000),am(4000),u(4000),v(4000),p(4000),q(4000),zz
      common/consti/e(4000),t(4000),csp(4000)
     *,tlima(4000),co(4)
      common/grun/csq(4),s(4),b(4),amu(4),a(4),c(4),cv(4)
      dimension vxl(4000),pu(4000),py(4000),
     *          pex2(4000),ski(4000),ni(4000)
c    *          pw(4000),pex2(4000),ski(4000),ni(4000)
      dimension ts(100),pmax(100),tmax(100),jmaxp(100),jmaxt(100)
      dimension neos(6),jcol(4)
      character*15 nam1,nam2*20
c
c     I CHANGE THAT INPUT pw AND sk6 CALCULATED USING pr AND pw. Nov. 20, 1987.
c
c     block 1
c
c     HISTRY MEMORIES:
c     vxl(j)--THE REAL SPECIFIC VOLUME OF COMPONENT I (BAMO).
c     ni(j)---THE LOADING HISTRY. 1=LOADING, 0=UNLOADING.
c       vxl(j) AND ni(j) ARE USED TO REMEMBER THE LOADING HISTRY OF j-CELL
c       IN TWO-COMPONENT COMPOSITE SUCH AS CdS/BAMO:THF.
c     pu(j),vu(j),py(j),vy(j),pw(j),ski(j)----UNLOADING PARAMETER MEMORIES.
c     pex2(j)----EQUILIBRIUM PRESSURE OF CdS.
c
c program control
c  np=no. of cycles between printout
c  ncs=length of run in cycles
c  jq=length of run in no. of cells reached by shock
c  tn=length of run in time, microsec
c  ncs or jq or tn can be used to stop a run
c
       write(6,1010)
 1010  format(1x,'Input data file nam1=')
       read(5,1020) nam1
 1020  format(a15)
       open(8,file=nam1, status='old')
c     write(6,626)
c 626 format(1h1)
       read(8,1020)nam1
       write(6,1021)nam1
 1021  format(' Output file nam2= ', a15)
      read(8,500)nw,np,tn
  500 format(2i5,f8.3)
      write(6,600)nw,np,tn
  600 format(2x,'nw= ',i4,'np=',i4,4x,'tn=',f8.4)
c     block 2
c
c regions
c   nr=no. of regions,=4
c  nc(k)=no of cells of kth region
c   d(k)=thickness of kth region,cm
c   vo(k)=initial specific volume of kth region, cc/gram
```

```
c   jr(k)=boundary of kth region
c
      nr=4
      read(8,502)(nc(k),k=1,nr)
  502 format(4i5)
      read(8,505) (d(k), k=1,nr)
  505 format(4f10.5)
      read(8,505)(co(k),k=1,4)
      read(8,505)(vo(k),k=1,nr)
c     read(8,702)(vo(k),k=1,nr)
c 702 format(4f10.5)
      jq=0
c     delt=.6*d(3)/(co(3)*nc(3))
c     do 954 k=1,4
c     nc(k)=d(k)*co(3)*nc(3)/(d(3)*co(k))
c 954 jq=jq+nc(k)
c
c I CHANGED THE CALCULATION METHODS OF delt AND nc(k).
c
        delt=tn
        do 954 k=1,nr
        dt=0.6*d(k)/(co(k)*nc(k))
        if(delt.lt.dt) goto 953
        delt=dt
 953    jq=jq+nc(k)
 954    continue
      jq=jq+10
      ncs=tn*2./delt
      write(6,602)(nc(k),k=1,nr),(d(k),k=1,nr),
     *(vo(k),k=1,nr)
  602 format(2x,'nc(k)=',4i4,6x,'d(k)=',4f10.6,6x/2x,
     *'vo(k)=',4f10.6)
      write(6,704)jq,ncs
  704 format(2x,'jq= ',i4,' ncs= ',i5)
c
c compute jr(k)
c
      mt=1
      do 2 k=1,nr
      jr(k)=mt+nc(k)
      mt=jr(k)
    2 continue
      write(6,603)(jr(k),k=1,nr)
  603 format(2x,'jr(k)=',4(i4,4x))
c
c     block 3
c
c assignment of coordinates and masses to each cell
c   dx(k)=initial width of  region,cm
c   am(j)=mass contained in cell j
c   x(j)=coordinate of jth cell. it varies with time. the initial value
c    is set here.
c
      x(1)=0.0
      jt1=1
      do 4 k=1,nr
      jt2=jr(k)-1
      dx(k)=d(k)/(float(nc(k)))
      do 3 j=jt1,jt2
      x(j+1)=x(j)+dx(k)
      am(j)=dx(k)/vo(k)
    3 continue
      jt1=jr(k)
    4 continue
c
c     block 4
```

```
c     boundary conditions
c     set masses at far right boundary
      jb=jr(nr)
      am(jb)=am(jb-1)
      am(jb+1)=am(jb)
c
      write(6,604)(dx(k),k=1,nr)
  604 format(2x,'dx=',2x,4f14.9/)
      write(6,605)
  605 format(3x,'j',9x,'x',14x,'am')
      do 5 j=1,2
      write(6,606)j,x(j),am(j)
  606 format(1x,i4,3x,2f14.9)
    5 continue
      do 6 k=1,nr
      jt1=jr(k)-1
      jt2=jr(k)+1
      do 6 j=jt1,jt2
      write(6,606)j,x(j),am(j)
    6 continue
c
c     block 5
c
c delt is time increment
      read (8,503)uo,cona,cq
  503 format(3f10.5)
      write(6,607)delt,cona,cq
  607 format(2x,'delt=',f10.7,4x,'cona=',f10.6,4x,'cq=',f10.6)
c
c quadratic and linear artificial viscosities are used to smooth the
c shock.  the quadratic coefficient is 'cq', the linear is 'cona'
c
      cqsq=cq*cq
c
c read initial values for flier plate velocity, 'uo', and for undis-
c turbed bulk sound velocities in regions 1 and 2, co1 and co2.
c
c     read (8,504)uo,co1,co2
c 504 format(3f8.5)
        co1=co(1)
        co2=co(2)
      write (6,608)uo,co1,co2
  608 format(2x,'uo=',f10.5,'  co1=',f10.5,'  co2=',f10.5/)
c
      write(6,601)(co(k),k=1,4)
  601 format(2x,'co(k)= ',4f8.4)
cblock 6
c
c set initial velocity for flyer plate.  velocity at impact boundary
c is set equal to uo/(1+vo1*co2/(vo2*co1)).    ******note***** initial
c oscillations may result from this choice.  if so a more exact cal-
c culation will be required.
c
c
c
c jcol is cell no. at which values of t and p at successive times
c are to be collected.  nave is the no. of cells, starting with
c jcol, for which history is to be preserved. zz is the'zero'
c value of abs u used to suppress noise ahead of the wave.
c     if nleftp=1, pf(t) is applied to left boundary.  no flyer.
c     if nleftp=2,const. press 'pappl' is applied to left boundary
c
      read(8,506)nleftp,zz,pappl
  506 format(i5,e10.3,f10.4)
      do 44 k=1,4
   44 jcol(k)=jr(k)
```

```
      nave=nc(3)
      navel=nave +jcol(2)-1
      write(6,617)nleftp,jcol(2),nave,zz,pappl
  617 format(2x,'nleftp= ',i2,' jcol(2)= ',i4,' nave= ',i4,
     *' zz= ',e10.2,' pappl=',f8.4)
      read(8,501)(neos(k),k=1,nr)
  501 format(4i5)
      write(6,613)(neos(k),k=1,nr)
  613 format(2x,'neos(k)= ',4i4)
c
c
c   INPUT e1,e2,yita FOR STANDARD VISCOELASTIC SOLID:
c
      read(8,1023)e1,e2,yita
 1023 format(3e14.4)
      write(6,1024)e1,e2,yita
 1024 format(/' S.V. SOLID:   e1=', f9.4,'    e2=',f9.4,
     *'   yita=',f9.4)
c
c   INPUT t11,tu1,rmda FOR ELASTOMER (BAMO:THF) OR MATRIX:
c
      read(8,1023)t11,tu1,rmda
      write(6,1026)t11,tu1,rmda
 1026 format(' ELASTOMER:   t11=', f9.4,
     *'   tu1=',f9.4,'   rmda=',f9.4)
      read(8,1023)t12,tu2,pr
      write(6,1027)t12,tu2,pr
 1027 format(' CdS CRYSTAL: t12=', f9.4,
     *'   tu2=',f9.4,'     pr=',f9.4)
c
c   INPUT vol, sk2 FOR COMPOSITE MATERIAL:
c     vol--SPECIFIC VOLUME OF COMPONENT 1 (SAY, BAMO:THF)
c     sk2--INITIAL INSTANTANEOUS MODULUS OF CdS.
c
c
      do 2010 i=1,nr
      if(neos(i).le.6) goto 2010
      read(8,2011)vol,sk2
 2011 format(2e14.6)
      write(6,2012)vol,sk2
 2012 format(' COMPOSITE:   vol=',f10.6,'  sk2=',f10.6/)
      goto 2015
 2010 continue
 2015 continue
      close(8)
c
c
     m1=1
     m2=1
     m3=1
     m4=1
     m5=1
     m6=1
       m7=1
       m8=1
c
c      open(4,file='4bcd.model', status='new')
       m7=2
 2030  continue
c
c
       nam2="pt1."//nam1
       open(1,file=nam2,status='new')
       nam2="pt2."//nam1
       open(2,file=nam2,status='new')
       nam2="pt3."//nam1
```

```
        open(3,file=nam2,status='new')
        nam2="ut1."//nam1
        open(7,file=nam2,status='new')
        nam2="ut2."//nam1
        open(8,file=nam2,status='new')
        nam2="ut3."//nam1
        open(9,file=nam2,status='new')
        nam2="pv1."//nam1
        open(10,file=nam2,status='new')
        nam2="pv2."//nam1
        open(11,file=nam2,status='new')
        nam2="pv3."//nam1
        open(12,file=nam2,status='new')
c
c
      jstare=jcol(2)
      jrt=navel
      write(6,857)jstare,jrt
  857 format(2x,'jstare= ',i4,'   jrt=  ',i4)
      to=293.
      jt1=1
      do 8 k=1,nr
      jt2=jr(k)
      do 7 j=jt1,jt2
      u(j)=0.0
      v(j)=vo(k)
      p(j)=0.0
      q(j)=0.0
      e(j)=0.0
      t(j)=to
      csp(j)=co(k)
        if(neos(k).le.6) goto 7
        vxl(j)=vol
        ni(j)=1
        pex2(j)=0.
        ski(j)=sk2
        pu(j)=0.0231016476
        py(j)=0.
        pw=0.007
c       pw(j)=0.
    7 continue
      jt1=(jr(k))
    8 continue
      p(jb)=0.0
      v(jb)=vo(nr)
      jt2=jr(1)-1
      do 9 j=1,jt2
      u(j)=uo
    9 continue
      u(jr(1))=uo/(1.+vo(1)*co2/(vo(2)*col))
c
c     block 7
c
      write (6,609)
  609 format(3x,'j',6x,'u(j)',9x,'p(j)',7x,'q(j)',7x,'v(j)',
     *7x,'t(j)',/)
      jt1=1
      nr1=nr+1
      do 11 k=1,nr1
      jt2=jt1+1
      do 10 j=jt1,jt2
      write(6,610)j,u(j),p(j),q(j),v(j),t(j)
  610 format(i4,5f12.6)
   10 continue
      if(k.eq.nr1)go to 11
      jt1=jr(k)-1
```

```
   11 continue
      dtm=.6*dx(3)/co(3)
c
c     start of time loop
c
c block 8
      al = 1.0 + 2.0*cona
      mt=1
      time = 0.0
      ncy=0
      jstar1=jr(1)-4
      jstar=jr(1)+4
      if(nleftp.ne.0)jstar=4
      if (nleftp.ne.0) jstar1=1
c
      write(6,804)jcol(2),nave
  804 format(2x,'jcol(2)= ',i4,' nave= ',i4//)
c block 9
      write(6,851)
  851 format(10x,'up,time,delt,ncy,jstar1,jstar')
c
c   block 9.1
   12 continue
      tlimb = 1.0
      time=time+delt
c       if(ncy.eq.700) pause 99999
      ncy=ncy+1
c     if(nleftp.eq.1)plft=pf(time)
      if(nleftp.eq.2)plft=pappl
c
c                            block9.2
      if(jstar1.gt.1)go to 13
      un=u(1)-delt*(p(1)+q(1)-plft)/am(1)
      x(1)=x(1)+.5*(un+u(1))*delt
      u(1)=un
      go to 14
   13 continue
      jt1=jstar1-1
      go to 15
   14 jt1=1
c                    block 9.3
   15 kt=4
      do 18 k=1,nr
      jt2=jr(k)-1
      if (jr(k).le.jstar) go to 16
c           branch 1
c          branch2
      jt2=jstar
      kt=k
   16 continue
      do 17 j=jt1,jt2
      un=u(j+1)-2.*delt*(p(j+1)+q(j+1)-p(j)-q(j))/(am(j+1)+am(j))
      if(abs(un).lt.4.00e-5)un=0.0
      x(j+1)=x(j+1)+.5*(un+u(j+1))*delt
      u(j+1)=un
      vn=v(j)+delt*(u(j+1)-u(j))/am(j)
      delu=u(j+1)-u(j)
c
c     check this when jstar1=1
c
      delx=x(j+1)-x(j)
      tlima(j)=delx/(al*csp(j) + 4.0*cqsq*abs(delu))
c     tlima(j)=delx/csp(j)
c
      qn=-delu*(cqsq*abs(delu)+csp(j)*cona)/vn
c       xxx=float(ncy)/float(np)
```

```
c       yyy=float(ncy/np)
c       if(abs(xxx-yyy).gt.1.e-6)goto 1011
c       write(6,1003) j,delu,csp(j),vn,qn
c       write(6,1013)q(j-1),q(j),q(j+1)
c1013   format(' q(j-1),q(j),q(j+1)',3e12.4)
c       write(1,1003) j,delu,csp(j),vn,qn
c1003   format(' *j,du,csp,vn,qn',i4,4e14.4)
 1011 if (qn.lt.0.0)qn=0.0
      jell=neos(k)
c
c  neos(k)= 1-cs2,                          2-sapphire,
c           3-fused silica,                 4-standard viscoelastic solid,
c           5-elastomer(BAMO) or matrix,    6-CdS crystal or particles,
c           7-composite 1 (CdS/BAMO:THF),   8-Composite 2.
c
c
      go to (101,201,301,401,510,511,512,513),jell
c
c     block 9.4
c
  101 call eoscs2(vn, v(j),p(j),qn,q(j),e(j),t(j),csp(j),ncy,j)
      go to 42
  201 call eossap (vn, v(j), p(j),qn,q(j),e(j),t(j),csp(j))
      go to 42
  301 call eosqtz (vn, v(j), p(j),qn,q(j),e(j),t(j),csp(j))
      go to 42
  401 call eossvs(vn,v(j),p(j),qn,q(j),e(j),delt,csp(j),vo(k),el,
     *            e2,yita)
      go to 42
  510 call eosbm(vn,v(j),p(j),qn,q(j),e(j),delt,csp(j),vo(k),
     *           tl1,tu1,rmda)
      go to 42
  511 call eoscds(vn,v(j),p(j),qn,q(j),e(j),t(j),delt,csp(j),vo(k),
     *           tl2,tu2,pr)
      go to 42
  512 call eosbcd1(vn,v(j),p(j),qn,q(j),e(j),t(j),delt,csp(j),vx1(j),
     *    pu(j),py(j),pw,pex2(j),ski(j),ni(j),ncy,j,vo(k),tl1,tu1,
     *    rmda,tl2,tu2,pr)
      go to 42
  513 call eoscmp2(vn,v(j),p(j),qn,q(j),e(j),t(j),delt,csp(j),vx1(j),
     *    pu(j),py(j),pw,pex2(j),ski(j),ni(j),ncy,j,vo(k),tl1,tu1,
     *    rmda,tl2,tu2,pr)
c
c     end of constitutive relations
c
 42   continue
c       if(j.ne.jcol(2)) goto 1115
c       write(4,1113)pex2(j),py(j),ski(j),vx1(j)
 1113   format(' pex2,py,ski,vx1 ', 4f10.6)
 1115   continue
c     if(tlima(j).ge.tlimb) go to 17
c4004   if (ncy.lt.599) goto 1004
c       if(j.le.148) goto 1004
c       if(j.gt.153) goto 1004
c     write(6,1002)ncy,j,v(j),p(j),q(j),delt
c     write(4,1002) ncy,j,v(j),p(j),q(j),delt
c1002 format(' *',2i5,4e14.6)
c       if(ncy.eq.700) pause 99999
 1004   continue
      if(tlima(j).ge.tlimb) go to 17
      tlimb=tlima(j)
  651 format(i4,7f12.6)
  652 format(i4,f12.6)
c     block 9.5
c
   17 continue
```

```
      if(k.eq.kt)go to 19
      jtl=jr(k)
   18 continue
c
c     block 9.6
   19 continue
      dtn=0.6*tlimb
      if(dtn/delt.gt.1.1) dtn=1.1*delt
      delt=(delt+dtn)/2.0
      if(jstar.eq.jb-1)go to 20
      if(abs(u(jstar+1)).gt.zz)jstar=jstar+1
   20 continue
      if(jstarl.eq.1)go to 21
      if(abs(u(jstarl)-uo).gt.1.0e-5)jstarl=jstarl-1
c
c     block 9.7
c
   21 continue
   23 if(mod(ncy,np).ne.0)go to 22
      jstop=jstar+2
      if(jstar.lt.20)go to 92
      jstart=jstop-20
      go to 93
   92 jstart=1
   93 continue
      call twrite(nw,jstar,jstart,jstop,jcol,navel,nave,time,
     *            ncy,np,delt,vo)
   51 continue
   22 if(jstar.gt.jq.or.time.gt.tn)go to 999
      if(jstar.gt.jcol(1).and.delt.gt.dtm)delt=dtm
      go to 12
  999 continue
   50 continue
        close(1)
        close(2)
        close(3)
        close(7)
        close(8)
        close(9)
        close(10)
        close(11)
        close(12)
      stop
      end
```

2. SUBROUTINE: p4eosbcd1.f

```
      subroutine eosbcd1(vn, vx, px, qn, qx, ex, tx, dt,cx,vx1,
     *                   pu,py,pw,pex2,ski,ni,ncy,j,vo,tl1,tu1,
     *                   rmda,tl2,tu2,pr)
c     real k1,k2,k3,k4,k5,k6,ki
c     write(6,1000)vn,vx,px,pn,dt,vx1
 1000 format(' vn,vx,px,pn,dt,vx1',6e10.3)
c
c     UNITS ARE:
c          p-Mbar, v-cm**3/g, gv: g/cm**3
c          e-Mbar*cm**3/g,    cv-Mbar*cm**3/g/c
c          t-c,    cx-cm/us,  tao-us
c          rmda-instantaneous coeficient
c
c     THIS SUBROUTINE IS USED TO CALCULATE THE PRESSURE pn IN CdS/BAMO:THF
c     COMPOSITE. THE MODEL OF THIS COMPOSITE IS GENERIZED MAXWELL MODEL.
c
c     THE EQUILIBRIUM RESPONSE OF BAMO:THF IS ASSUMED AS HUGONIOT.
c     THE INSTANTANEOUS RESPONSE IS ASSUMED THAT RMDA TIMES HUGONIOT.
c
c     THE MODEL OF CdS USED HERE IS MODEL 3, I.E. py=pr, sk6 NOT EQUAL TO sk2.
c     THE EQUILIBRIUM RESPONSE OF CdS IS ASSUMED TO BE SECTIONAL STRAIGHT
c     LINES: sk1,sk2 AND sk3 DURING LOADING, AND k3,k6 and k1 (WHEN pu>pe), OR
c     k4,k6,k1 (WHEN pe>pu>pi), OR k1 (WHEN pu<pi) DURING UNLOADING.
c     THE INSTANTANEOUS RESPONSE OF CdS IS ASSUMED TO BE LINEAR: k1 DURING
c     LOADING AND k3 (WHEN pu>pe), OR k4 (WHEN pe>pu>pi), OR k1 (WHEN pu<pi)
c     DURING UNLOADING.
c
c     SIGNS NOTE: 1--BAMO:THF; 2--CdS
c     RELAXATION TIMES:
c     LOADING:    tl1-BAMO; tl2-CdS
c     UNLOADING:  tu1-BAMO; tu2-CdS
c     ni--JUDGE THAT THE PREVIOUS STEP WAS LOADING OR UNLOADING.
c         ni=1, LOADING; ni=0, UNLOADING.
c     pen1 pen2--EQUILIBRIUM PRESSURES OF BAMO AND CdS.
c     fn1, fn2---INSTANTANEOUS MODULAS OF BAMO AND CdS.
c     pn1,pn2----PRESSURES OF BAMO AND CdS, WHICH ARE ITERATED TO BE EQUAL.
c     cx---------SOUND SPEED.
c     OTHER SIGNS SEE THE INSTRUCTION OF THE PROGRAM.
c
c
c
c     if (m8.eq.2) go to 1
c
c eo=0,po=0,to=293.
c       gv = 5.6
c       cv = 3.3858e-5
        vo1= 0.8333
        vo2=0.2073
c       f1=0.46
c       f2=0.54
c       vo=f1*vo1+f2*vo2
        f1=(vo-vo2)/(vo1-vo2)
        f2=1.-f1
c       write(6,20)f1,f2
c       m8 = 2
        a=0.0319
        b=-0.0737
        c=1.262
        sk1=-0.411673
        sk2=-4.52357
```

```
        sk3=-0.216939
c       sk5=4.33032
c       sk6=-8.093758
        vi=0.197789675
        pi=0.0231016476
        ve=0.154319663
        pe=0.03270112008
        pf=0.235635219
        vf=0.110295342
        vr=ve-(pe-pr)*sk3
        sk6=(vr-vo2-pw*sk1)/(pr-pw)
        delta=1.e-6
c
 1      continue
c       write(6,1001)ni
 1001   format('### 1, ni=',i4)
        vx2=(vx-f1*vx1)/f2
        dv=vn-vx
c       if(abs(dv).lt.1.e-6) goto 550
c       if(dv.gt.0.) goto 300
        if(dv) 200,100,300
 100    if(ni.ne.1) goto 310
c
c    LOADING PART
c
        goto 202
 200    if(ni.eq.1) goto 202
c    NEW LOADING POINT, NEED TO RECALCULATE pu & ski (i.e. sk4).
c    WHERE ski IS THE SLOPE OF NEW LOADING CURVE, AND pu IS THE NEW
c    PHASE TRANSITION PRESSURE FROM THIS NEW LOADING POINT.
c    ske MEANS EQUILIBRIUM SLOPE.
c
c       write(6,303)vx2,pex2,vf,pf
        ski=(vf-vx2)/(pf-pex2)
        pu=(vi-vf-sk2*pi+ski*pf)/(ski-sk2)
 202    t1=t11
        t2=t12
c       write(6,303)pu,ski,t1,t2
 303    format('????', 4e12.4)
        dmax=0.
        dmin=dv
        if(pex2.lt.pe) goto 203
        ske=sk3
        goto 230
 203    if(pex2.ge.pu) goto 204
        ske=ski
        goto 230
 204    ske=sk2
c
 230    ni=1
        goto 400
c
c    UNLOADING PART
c
 300    if(ni.ne.1) goto 310
c
c    NEW UNLOADING POINT, NEED TO RECALCULATE py, pw, ski(i.e. sk4).
c    WHERE ski IS THE NEW UNLOADING SLOPE OF CdS (i.e. INSTANTANEOUS
c    RESPONSE). py IS THE PHASE REVERSE PRESSURE, AND pw IS THE PHASE
c    REVERSE END POINT.
c
        ski=(vf-vx2)/(pf-pex2)
c       vr=ve-(pe-pr)*sk3
        py=(vx2-vr-ski*pex2+sk6*pr)/(sk6-ski)
c       write(6,113) pex2,py,ski,sk6,vx2,vr
c       write(4,113) pex2,py,ski,sk6,vx2,vr
```

```
 113    format(' pex2,py,ski,sk6,vx2,vr', 6f9.5)
c       vy=vr-sk6*(pr-py)
c       pw=(vo2-vr+sk6*pr)/(sk6-sk1)
c
 310    t1=tu1
        t2=tu2
        dmax=dv
        dmin=0.
        if(pex2.lt.py) goto 320
        ske=ski
        goto 390
 320    if(pex2.lt.pw) goto 330
        ske=sk6
        goto 390
 330    ske=sk1
c
 390    ni=0
c
c
c    CALCULATE THE PRESSURE pn
c
 400    r1=0.5*dt/t1
        r2=0.5*dt/t2
        fn2=-1./ski
c       vx2=(vx-f1*vx1)/f2
c
c     dv,dv1,dv2 ARE REAL VOLUME CHANGES IN CdS/BAMO, BAMO, AND CdS RESPECTIVELY
c     dvx1,dvx2 ARE CORESPONDING SPECIFIC VOLUME CHANGES IN BAMO AND CdS.
c       write(6,1003)
 1003   format('### 3')
c
        do 420 i=1,50
        dv1=0.5*(dmax+dmin)
        dv2=dv-dv1
        dvx1=dv1/f1
        dvx2=dv2/f2
        wx=vol/vx1-1.
        wn=vol/(vx1+dvx1)-1.
        pex1=wx*(a+b*wx+c*wx*wx)
        pen1=wn*(a+b*wn+c*wn*wn)
        pen2=pex2+dvx2/ske
        fx1=rmda*(wx+1.)**2*(a+2.*b*wx+3.*c*wx*wx)
        fn1=rmda*(wn+1.)**2*(a+2.*b*wn+3.*c*wn*wn)
        pn1=(px*(1.-r1)-0.5*(fn1+fx1)*dvx1/vol+r1*(pen1+pex1))/(1.+r1)
        pn2=(px*(1.-r2)-fn2*dvx2+r2*(pen2+pex2))/(1.+r2)
c       write(6,500)i,dv1,dv2,dmax,dmin,pn1,pn2
c500    format(' dv12,m12,pn12',i3,6e10.3)
c       if(abs((pn1-pn2)/pn1).le.1.e-6) goto 450
        if(abs(pn1-pn2).le.1.e-8) goto 450
        if(pn2.gt.pn1) goto 410
        dmin=dv1
        goto 420
 410    dmax=dv1
 420    continue
 450    pn=0.5*(pn1+pn2)

c       write(6,65)ski,ske,vx1,fn1,fn2,pn
 65     format(' ski,sk,vx1,fn1,fn2,pn',6e10.3)
c       write(6,360)fn,fx,phn,phx
 360     format(' fn,fx,phn,phx',4e12.4)
c       en=ex-0.5*(pn+px+qn+qx)*dv
c     de = en - ex
c     p  = (pn + px)/2.
c     tn = tx
c     do 2 m = 1,4
c     t  = (tn + tx)/2.
```

```
c  2  tn = tx + (de + (p - t*cv*gv)*dv)/cv
c     tx = tn
c     kh = (mu + 1.)*(a + 2.0*b*mu + 3.0*c*mu**2 + 4.0 *d*mu**3)
c     at = 0.5*vn*gv*(2.0*pn - ph - px)
c     ks = at + kh*(1.0 - gv*(vo - vn)/2.0)
c     cx2 = vn*ks
c       cx=sqrt(abs(vo*fn))
        cx1=sqrt(abs(fn1*vo1))
        cx2=sqrt(abs(fn2*vo2))
        alpha=f1*(vx1+dvx1)/vn
        cx=cx1*cx2/(alpha*cx2+(1.-alpha)*cx1)
        if(ncy.le.1000) goto 80
        if(ncy.gt.1200) goto 80
        if(j.lt.151) goto 80
        if(j.gt.152) goto 80
c       if(dv.le.0) goto 80
        write(4,50)pn,px,pex2,dv1,dv2
c       write(6,50)pn,px,pex2,dv1,dv2
 50     format(/' pn,px,pex2,dv1,dv2',5e12.4)
        write(4,60)ncy,pn1,pn2,fn1,fn2
c       write(6,60)ncy,pn1,pn2,fn1,fn2
 60     format(' ncy,pn1,pn2,fn1,fn2,',i5,4e12.4)
        write(4,70)j,pen1,pen2,r1,r2
c       write(6,70)j,pen1,pen2,r1,r2
 70     format(' j,pen1,pen2,r1,r2,',i5,4e12.4)
        write(4,75) ni,pu,vu,vx1,vx2
c       write(6,75) ni,pu,vu,vx1,vx2
 75     format(' ni,pu,vu,vx1,vx2',i3,4e12.4)
        write(4,76)vn,vx,dv,vy,pw
c       write(6,76)vn,vx,dv,vy,pw
 76     format(' vn,vx,dv,vy,pw',5e12.4)
 80     px = pn
        vx1=vx1+dvx1
c       vx2=vx2+dvx2
c       vx1=(vx-f2*vx2)/f1
c       pex1=pen1
        pex2=pen2
c       fx1=fn1
c       fx2=fn2
c
 550    vx=vn
        qx=qn
c       write(6,1002)
 1002   format('### 2, ni=',i4)
        return
        end
```

3. SUBROUTINE: p4eossap.f

```
      subroutine eossap (vn, vx, px, qn, qx, ex, tx, cx)
      real mu,kh,ks
c     if (ml.eq.2) go to 1
      a =  4.99
      b =  4.637
      c = 17.57
      d =  0.0
c eo=0,po=0,to=293.
      gv = 2.42
      cv = .724e-5
      vo = 0.250940
c     ml = 2
c
    1 dv = vn - vx
      mu = vo/vn - 1.0
      ph = mu*(4.99 + mu*(4.637 + mu*17.57))
      eh = 0.5*ph * (vo - vn)
      et = ex - 0.5*(px + qx + qn)*dv
      pn = (ph + gv*(et - eh))/(1. + gv*dv/2.)
      en = et - pn*dv/2.
      de = en - ex
      p  = (pn + px)/2.
      tn = tx
      do 2 j = 1,4
      t  = (tn + tx)/2.
   2  tn = tx + (de + (p - t*cv*gv)*dv)/cv
      tx = tn
      kh = (mu + 1.)*(a + 2.0*b*mu + 3.0*c*mu**2 + 4.0 *d*mu**3)
      at = 0.5*vn*gv*(2.0*pn - ph - px)
      ks = at + kh*(1.0 - gv*(vo - vn)/2.0)
      cx2 = vn*ks
      vx = vn
      qx = qn
      ex = en
      px = pn
      cx=sqrt(cx2)
c
      return
      end
```

4. SUBROUTINE: p4eosqtz.f

```
      subroutine eosqtz (vn, vx, px, qn, qx, ex, tx, cx)
      real mu,kh,ks
c     if (m2.eq.2) go to 1
      a= .7760
      b=-4.159
      c= 30.340
      d= -69.260
      gv = 0.076
      cv = .7858e-5
      alpha = 0.77e-6
      vo = 1./2.204
c     m2 = 2
c
    1 dv = vn - vx
c this statement is adequate for compression
c     if(vn.gt.0.42424731)qn=0.
c for this subroutine, the definition of mu is changed
c to what i normally call eta.
      mu = 1.0-vn/vo
      ps = mu*(a+mu*(b+mu*(c+mu*d)))
c eqn for ps is from barker and hollenbach
      es =vo*mu**2*(a/2.+mu*(b/3.+mu*(c/4.+mu*d/5.)))
      et = ex - 0.5*(px + qx + qn)*dv
      pn = (ps + gv*(et - es))/(1. + gv*dv/2.)
c       write(6,20) vn,vx,px,ps,pn
c       write(1,20) vn,vx,px,ps,pn
c20     format(' **',5e13.4)
      en = et - pn*dv/2.
      de = en - ex
      p  = (pn + px)/2.
      tn = tx
      do 2 j = 1,4
      t  = (tn + tx)/2.
   2  tn = tx + (de + (p - t*cv*gv)*dv)/cv
      tx = tn
      ks = (1 - mu)*(a + mu*(2.*b+mu*(3.*c+mu*4.*d)))
      cx2 = vn*ks
      vx = vn
      qx = qn
      ex = en
      px = pn
      cx=sqrt(cx2)
c
      return
      end
```

5. SUBROUTINE: twrite.f

```
      subroutine twrite(nw,jstar,jstart,jstop,jcol,navel,nave,time,
     *                  ncy,np,delt,vo)
      common/field/x(4000),am(4000),u(4000),v(4000),p(4000),q(4000),zz
      common/consti/e(4000),t(4000),csp(4000)
     *,tlima(4000),co(4)
        dimension jcol(4),vo(4)
      jcoll=jcol(2)
      if(nw.eq.2)go to 20
c
c nw=5 FOR PRINTING THE RESULTS AT 3 INMATERIAL GAUGES.
c
c
      if(nw.eq.5)go to 50
      if(nw .eq. 4) go to 4
c     if nw = 4 ==> print out cells in material 3 only.
      go to 5
    4 if(jstar.lt.jcoll) go to 99
    5 continue
      if(jstar.lt.jcoll)go to 20
      ts=0.
      pa=0.
      pmax=0.
      tmax=0.
      do 30 ji=jcoll,navel
      pa=pa+p(ji)
      ts=ts+t(ji)
      if(p(ji).gt.pmax)go to 27
      go to 26
   27 pmax=p(ji)
      jmaxp=ji
   26 if(t(ji).gt.tmax) go to 29
      go to 30
   29 tmax=t(ji)
      jmaxt=ji
   30 continue
      pa=pa/float(nave)
      ts=ts/float(nave)
      jrt=jcoll+nave-1
      write(1,101)time,pa,p(jcoll),p(jrt),pmax,jmaxp,ncy
  101 format(2x,5f9.5,2i4)
      write(2,102)time,ts,t(jcoll),t(jrt),tmax,jmaxt,ncy
  102 format(2x,5f9.3,2i4)
      if(nw.eq.1)go to 99
   20 continue
        npl=np*10
        if(mod(ncy,npl).ne.0) goto 99
      write(6,601)time,delt,ncy,jstar
  601 format(2x,'time=',f9.5,'delt=',f9.5,'ncy=',i4,'jstar= ',i4)
      write (6,609)
  609 format(3x,'j',6x,'u(j)',9x,'p(j)',7x,'q(j)',7x,'v(j)',
     *7x,'t(j)',/)
      if (nw .eq. 4) go to 1
      go to 2
    1 jstart = jcoll - 1
      jstop  = jrt + 1
    2 continue
      do 10 j=jstart,jstop
c     write(6,610)j,u(j),p(j),q(j),v(j),t(j)
      write(6,610)j,u(j),p(j),q(j),v(j),time
  610 format(i4,2f12.6,e12.4,2f12.6)
```

```
   10 continue
      goto 99
c
c THIS BLOCK IS FOR PRINTING OUT THE RESULTS AT INTERFACES 1,2,3
c THE UNITS ARE TRANSFORMED TO pa-Kbar, ua-mm/us and va-V/Vo.


  50  do 55 k=1,3
      if(jstar.lt.jcol(k))go to 99
        j10=jcol(k)
c       pa=(p(j10-2)+p(j10-1)+p(j10)+p(j10+1))/4.
c       pa=p(j10+1)
        if(k.eq.3) goto 65
c       pa=(p(j10+9)+p(j10+10)+q(j10+9)+q(j10+10))/2.
c       va=(v(j10+9)+v(j10+10))/2.
c       pa=p(j10)+q(j10)
c       va=v(j10)
        pa=500.*(p(j10+1)+p(j10)+q(j10+1)+q(j10))
        va=(v(j10+1)+v(j10))/2./vo(k+1)
        ua=5.*(u(j10+1)+u(j10))
        goto 66
c65     pa=(p(j10-1)+p(j10))/2.
 65     pa=500.*(p(j10-1)+p(j10-2)+q(j10-1)+q(j10-2))
        ua=5.*(u(j10-1)+u(j10-2))
        va=(v(j10-1)+v(j10-2))/2./vo(k+1)
 66     k5=k+9
        k6=k+6
        write(k5,105)va,pa
      write(k6,105)time,ua
  55  write(k,105)time,pa
  105 format(2e17.6)
      go to 20
  99  continue
      return
      end
```