

UNCLASSIFIED

AD NUMBER
AD843058
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 30 NOV 1967. Other requests shall be referred to Ballistic Research Lab., Aberdeen Proving Ground, MD.
AUTHORITY
ARDC D/A ltr 23 Jan 1970

THIS PAGE IS UNCLASSIFIED

978

Gulf General Atomic
Incorporated

Val Kuchera

[Handwritten signature]

GAMD-8073

Addendum

Category A

AD 843058

DDC
RECEIVED
NOV 14 1968
RECEIVED

TOIL

(A Two-material Version of the OIL Code)

[Handwritten signature] B

Work done by:

W. E. Johnson

Report written by:

W. E. Johnson

This document, which was prepared primarily for internal use at Gulf General Atomic, may contain preliminary or incomplete data. It is informal and is subject to revision or correction; it does not, therefore, represent a final report.

Advanced Research Projects Agency

ARPA Order No. 71-62

Ballistic Research Laboratories

Contract No. DA-04-495-AMC-1481(X)

GA Project 6003

November 30, 1967

THIS DOCUMENT IS SUBJECT TO SPECIAL EXPORT CONTROLS AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR FOREIGN AGENCIES MAY BE MADE ONLY WITH PRIOR APPROVAL OF COMMANDING OFFICER, U. S. ARMY ABERDEEN RESEARCH & DEVELOPMENT CENTER, ABERDEEN PROVING GROUND, MARYLAND 21005.

123

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

An addendum to GAMD-8073 by W. E. Johnson. This section is broken down into 5 parts as described below.

- (A) -TCLAM--the generator code for TOIL, presents sample problem, and instructions for usage.
- (B) TOIL Equations--discusses the difference equations used in TOIL,
- (C) Input for TOIL--describes the necessary data for using the TOIL code.
- (D) Definition of Variables--defines the variables used in the TOIL code,
- (E) TCLAM and TOIL--Fortran listings, consists of the actual Fortran listings with an abundant sprinkling of comments.

A. TCLAM CODE

General Description

TCLAM is a numerical code that provides the initial configuration and starting conditions for the TOIL code. This involves specifying the dimensions for each cell and the density, the two velocity components and the specific internal energy.

Below, Fig. 1, is a sketch of a typical two-dimensional grid. We display only a plane view, keeping in mind that each cell is really a solid of revolution, symmetric about the Z axis. In the discussion to follow, both X and R are referred to as the radial direction, and both Y and Z refer to the axial direction.

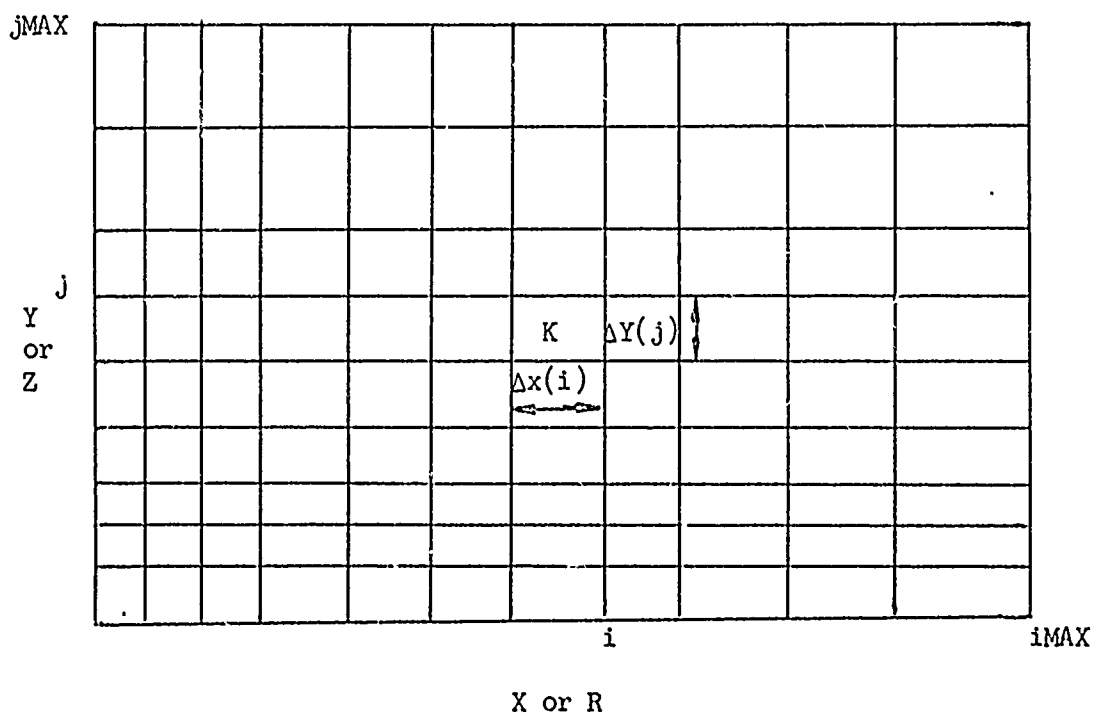


Fig. 1

i is the right boundary and j is the top boundary of cell K.

$$X(i) = \sum_{i=1}^i \Delta X(i)$$

$$Y(j) = \sum_{j=1}^j \Delta Y(j)$$

The area of cell (i,j) in the i direction = $2\pi X(i) \Delta Y(j)$.

The area of cell (i,j) in the j direction = $\pi(X_{(i)}^2 - X_{(i-1)}^2)$.

K is defined as (j-1) iMAX + i + 1, and is the index for all cell centered quantities.

First, we specify the total number of cells in the X direction (iMAX) and the total number of cells in the Y direction (jMAX). In addition, we specify the ΔX and ΔY for each cell. TCLAM then calculates the $X(i)$ and $Y(j)$ of each cell and the axial area ($TAU(i) = \pi(X_{(i)}^2 - X_{(i-1)}^2)$).

Next, specify the data for the various packages. This consists first of specifying the number of particles per cell for this package, and the type of material, and the origin of the radius vector for the density, energy and velocity functions. We also specify one of the 6 possible fits for the density, energy and velocity functions.

Next, specify the type of geometry (the configuration), of which there are four possible: circle, ellipse, rectangle and triangle. Thus, the configuration can be broken up into combinations of the four possible geometries (note, these geometries are really solids of revolution for the cylindrical geometry). The data for these geometries are stored in the array (TAB). A counter (IWS) is calculated from the initial data. This counter is stored in the first word of the TAB block for each package. Its values are as follows:

iWS = 1	delete this triangle
iWS = 2	generate this triangle
iWS = 3	delete this rectangle
iWS = 4	generate this rectangle
iWS = 5	delete this circle or ellipse
iWS = 6	generate this circle or ellipse

The next 6 words of this TAB block contain the coordinates of the desired geometry. The next card contains the data for the density fit and we store this data in the array (TABR). Next, read in the data for the internal energy and store it in the TABI array. The last data card of each package contains the input for the velocity fit, and this data is stored in the TABUV array.

We then compute the boundaries of the specified geometry, and the minimum and maximum i and j values for this geometry.

Next we subdivide all the cells in this package into N (the number of particles per cell) equal area cells. The particles are placed at the center of each sub-cell, where the volume of the sub-cell = $2\pi(XL)(DY/WS)(DX/WS)$ where $WS = (N)^{\frac{1}{2}}$ and XL = the X value of the center of the sub-cell. Some of the particles (N) may not be generated, however, for if the boundary of the geometry passes through the cell, those particles that fall outside of the boundary are deleted.

We assign a density, two velocity components and a specific internal energy to each package. These may be any function of XL , YL or R where XL = X coordinate of the particle N , YL = Y coordinate of particle N , and $R = (TTX^2 + TTY^2)^{\frac{1}{2}}$ where $TTX = XL - XC$ and $TTY = YL - YC$; XC and YC are the coordinates of the origin of the radius vector R , they are inputted on the first card of each package. The mass of each particle is the density times the volume of the subdivision cell of cell K .

After processing all N particles for cell K , we calculate the total mass of cell K as

$$\sum_{n=1}^N m_n,$$

the axial momenta as $\sum_{n=1}^N V_n m_n$, the radial momenta as $\sum_{n=1}^N U_n m_n$ and the internal energy as $\sum_{n=1}^N I_n m_n$. In addition, the total energy and mass of all cells are summed up for the entire package.

The normal units for TCLAM are as follows:

m = particle mass in grams
 AMX = mass of cell K in grams for the (x) material
 AMD = mass of cell K in grams for the (.) material
 U = radial velocity in cm/shake (1 shake = 10^{-8} sec)
 V = axial velocity in cm/shake
 AiX = specific internal energy in jerks/gram for (x) material (1 jerk = 10^{16} ergs)
 AiD = specific internal energy in jerks/gram for (.) material.

After all cells in this package have been processed, we read in another package and proceed as before.

After all packages have been processed, we then convert the axial and radial momenta of each cell K to an axial and radial velocity component. The internal energy of cell K is converted to specific internal energy.

The output from TCLAM (a binary tape that can be read by the TOIL code) consists of the cell dimensions, total number of cells in both directions, the mass and two velocity components, the specific internal energy of each cell K and other information required for the TOIL program.

The Fortran symbols and units are listed below:

AMD = total mass in cell K (grams) for the (.) material
 AMX = total mass in cell K (grams) for the (x) material
 AiX = specific internal energy in cell K (jerks/gram) for (x) material
 AiD = specific internal energy in cell K (jerks/gram) for (.) material
 U = radial velocity of cell K (cm/shake)
 V = axial velocity of cell K (cm/shake)
 X = dimension in cm of the right boundary of the cell
 Y = dimension in cm of the top boundary of the cell
 $iMAX$ = total number of cells in the X direction
 $jMAX$ = total number of cells in the Y direction
 $kMAX = (iMAX)(jMAX) + 1.$

For clarification, in generating data for the TOIL code, the creation of particles is only a computational technique to give the proper density, velocity and internal energy as specified. These particles are not saved after they have been summed for the cell in question.

Below we list a complete description of the required input and format for the TCLAM code.

INPUT DESCRIPTION FOR TCLAM

An asterisk before the work signifies that the data is floating point; otherwise it is fixed point data.

<u>Card No.</u>		<u>Column No.</u>	<u>Description</u>
1		2 - 72	Header card, any BCD information.
2	*	1 - 10	Contain the problem number.
	*	11 - 20	iMAX, the number of cells in the X-direction (maximum of 100).
	*	21 - 30	jMAX, the number of cells in the Y-direction (maximum of 100). $(iMAX)(jMAX) \leq 4499$
	*	31 - 40	= 0.
	*	41 - 50	2.
	*	51 - 60	Blank.
	*	61 - 70	Blank.
		71 - 72	N7 = binary tape number.
3	(2 number 3 cards is the minimum)		
	1		A (1) indicates that this is the last DX or DY card to be read in. A (0) indicates that there will be more DX or DY cards.
	2		A (0) indicates DX data. A (1) indicates DY data.

<u>Card No.</u>	<u>Column No.</u>	<u>Description</u>
	3 - 4	Indicates the number of zones that will have the same DX or DY values that appear in Columns 11 - 20.
	5 - 6	Indicates the number of zones that will have the same DX or DY values that appear in Columns 21 - 30.
	7 - 8	Indicates the number of zones that will have the same DX or DY values that appear in Columns 31 - 40.
	9 - 10	Indicates the number of zones that will have the same DX or DY values that appear in Columns 41 - 50.
	* 11 - 20	The value of DX or DY.
	* 21 - 30	The value of DX or DY.
	* 31 - 40	The value of DX or DY.
	* 41 - 50	The value of DX or DY.
4	* 1 - 10	Blank.
	* 11 - 20	Blank.
	* 21 - 30	80.
	* 31 - 40	Blank.

Now we begin leading the data to generate a package. The maximum number of geometries that may be generated is 72; to increase the maximum requires the changing of dimensions.

1	1	Load a 1 here.
	2	A (1) implies that X material will be generated in this package. A (0) implies that dot material will be generated.
	5 - 7	(N^2), the number of particles per cell to be generated, where $1 \leq N \leq 20$. Note, the unit digit in Column 7 the 10 digit in Column 6, the 100 digit in Column 5.

<u>Card No.</u>	<u>Column No.</u>	<u>Description</u>
*	11 - 20	YC = Y coordinate for the origin of the radius vector used in the density, energy and velocity fits.
*	21 - 30	XC = X coordinate for the origin of the radius vector used in the density, energy and velocity fits.
*	31 - 40	A number (1 through 6) that specifies the fit number or subroutine to use for this package to calculate the density, velocities and specific internal energy of the N particles.
*	41 - 70	Blank.

Following the first card of each package are five other types of cards.

1. Generate geometry (see options below).
2. Delete geometry (see options below).
3. A density card (only one per package).
4. An energy card (only one per package).
5. A velocity card (only one per package).

For cards (1) and (2), TCLAM has the following geometric options for generating or deleting:

1. A rectangle
 - A (4) in Column 1.
 - Columns 2 - 6 are blank.
 - A (1) in Column 7 means to generate this rectangle.
 - A (0) in Column 7 means to delete this rectangle.
- | | | |
|---|---------|--|
| * | 11 - 20 | X1 = the left X coordinate of this rectangle. |
| * | 21 - 30 | X2 = the right X coordinate of this rectangle. |
| * | 31 - 40 | Y1 = the lower Y coordinate of this rectangle. |
| * | 41 - 50 | Y2 = the upper Y coordinate of this rectangle. |

<u>Card No.</u>	<u>Column No.</u>	<u>Description</u>
2. A triangle		A (6) in Column 1, Columns 2 - 6 are blank. A (1) in Column 7 means to generate this triangle. A (0) in Column 7 means to delete this triangle.
	* 11 - 20	X1
	* 21 - 30	Y1
	* 31 - 40	X2
	* 41 - 50	Y2
	* 51 - 60	X3
	* 61 - 70	Y3
		NOTE: Vertices (1-3) can be in any order.
3. An ellipse or circle		A (41) in Columns (1-2), Columns 3 - 6 are blank. A (1) in Column 7 means to generate this ellipse or circle. A (0) in Column 7 means to delete this ellipse or circle.
	* 11 - 20	The semi-axis in the X-direction if an ellipse or the radius if for a circle.
	* 21 - 30	The semi-axis in the Y-direction if an ellipse or zero for a circle.
	* 31 - 40	The X-coordinate of the center of ellipse or circle.
	* 41 - 50	The Y-coordinate of the center of ellipse or circle.

Following the geometry cards are the following data cards that refer to all cells within this package:

Density card - a 51 in Columns (1-2).

Energy card - a 52 in Columns (1-2).

Velocity card - a 53 in Columns (1-2).

NOTE: If in this package, the density or internal energy or velocities will remain the same as the previous package, then a 51, 52, or 53 card is not required.

<u>Card No.</u>	<u>Column No.</u>	<u>Description</u>
*	11 - 20	Contains the values to be used in the analytical expressions for the density, energy, and velocities.
*	21 - 30	
*	31 - 40	
*	41 - 50	
*	51 - 60	
*	61 - 70	

This data is loaded into the following Fortran arrays:

TABR (1-6) The 6 constants available for the density fits.
 TABI (1-6) The 6 constants available for the internal energy fits.
 TABUV (1-6) The 6 constants available for the two velocity components.

Finally, the last card will have a 2 in Column 1, this signifies the completion of loading all input cards into the TCLAM code.

SPECIAL SUBROUTINES

There are six subroutines labeled FIT 1 - FIT 6, used to compute the density, internal energy and velocities.

The standard input to these subroutines is as follows:

TY = Y coordinate of particle N.
 TX = X coordinate of particle N.

The modified coordinates TTY and TTX are computed as follows:

TTY = Y coordinate = TY - YC (relative to YC)
 TTX = X coordinate = TX - XC (relative to XC)

Note: YC and XC are the Y and X coordinate for the origin of the radius vector used in the density, energy and velocity fits.

The standard output from these subroutines is as follows:

WSR - contains the density of particle N.
 WSI - contains the specific internal energy of particle N.
 WSU - contains the radial velocity component for particle N.
 WSV - contains the axial velocity component for particle N.

1. FIT 1

$$\begin{bmatrix} R = (X^2 + Y^2)^{\frac{1}{2}} \\ WS = (TTX^2 + TTY^2)^{\frac{1}{2}} \end{bmatrix}$$

$$\begin{bmatrix} \rho = A + B(Y - C) \\ WSR = TABR(1) + TABR(2) [TTY - TABR(3)] \end{bmatrix}$$

$$\begin{bmatrix} I = A + B(Y - C) \\ WSI = TABI(1) + TABI(2) [TTY - TABI(3)] \end{bmatrix}$$

$$\begin{bmatrix} U = 0. \\ WSU = 0. \end{bmatrix}$$

$$\begin{bmatrix} V = A + B(Y - C) \\ WSV = TABUV(1) + TABUV(2) [TTY - TABUV(3)] \end{bmatrix}$$

2. FIT 2

$$\begin{bmatrix} R = (X^2 + Y^2)^{\frac{1}{2}} \\ WS = (TTX^2 + TTY^2)^{\frac{1}{2}} \end{bmatrix}$$

$$\begin{bmatrix} \rho = \left(\frac{X-A}{B} \right)^2 + \left(\frac{Y-C}{D} \right)^2 \\ WSR = \left(\frac{TTX - TABR(1)}{TABR(2)} \right)^2 + \left(\frac{TTY - TABR(3)}{TABR(4)} \right)^2 \end{bmatrix}$$

$$\begin{bmatrix} I = A + BX + CX^2 + DY + EY^2 \\ WSI = TABI(1) + TABI(2)(TTX) + TABI(3)(TTX)^2 \\ + TABI(4)(TTY) + TABI(5)(TTY)^2 \end{bmatrix}$$

$$\begin{bmatrix} U = C + DY \\ WSU = TABUV(3) + TABUV(4) * TTY \end{bmatrix}$$

$$\begin{bmatrix} V = A + BY \\ WSV = TABUV(1) + TABUV(2) * TTY \end{bmatrix}$$

3. FIT 3

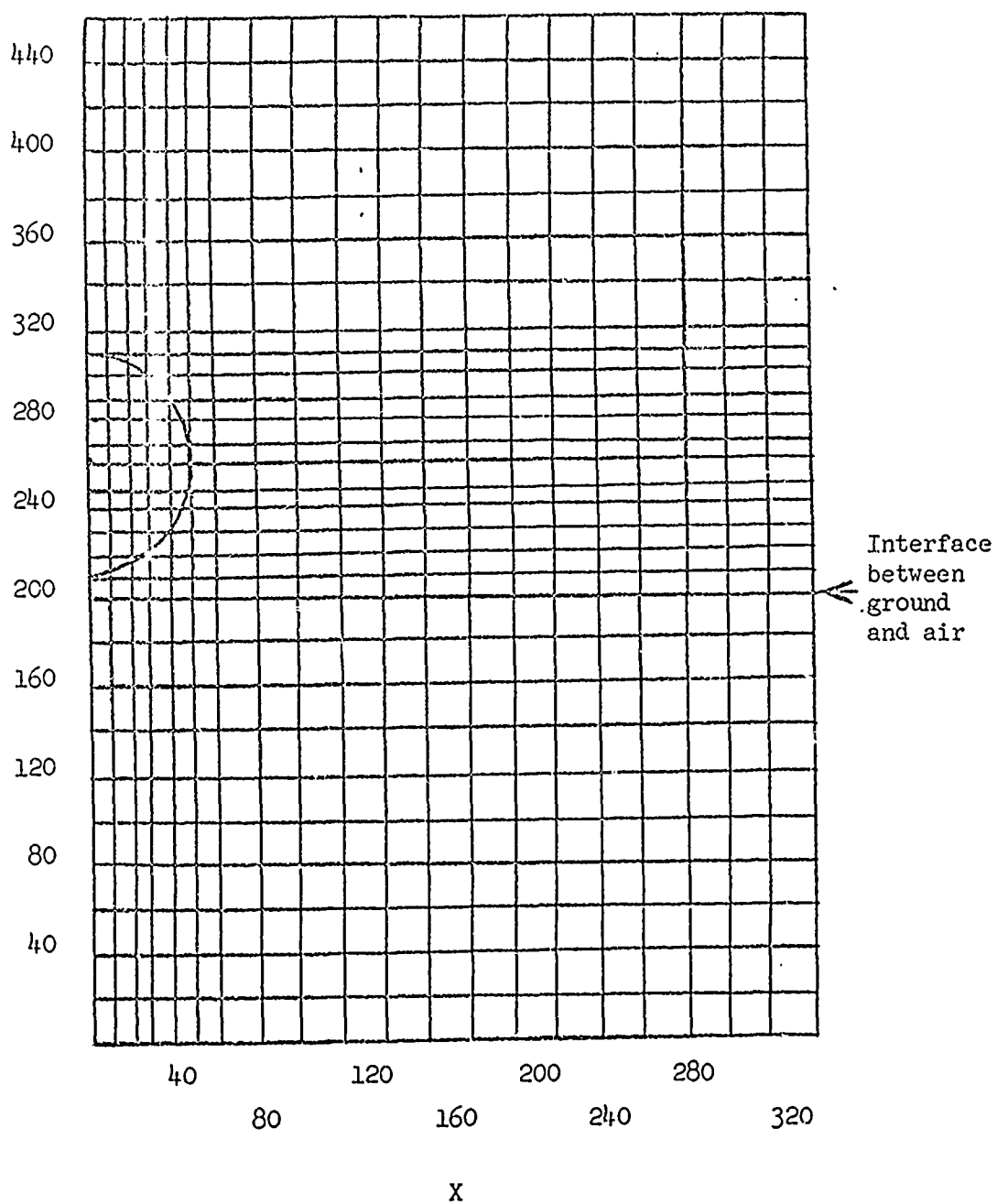
$$\begin{aligned}
 & \left[\begin{aligned} R &= (X^2 + Y^2)^{\frac{1}{2}} \\ WS &= (TTX^2 + TTY^2)^{\frac{1}{2}} \end{aligned} \right] \\
 & \left[\begin{aligned} \rho &= A + B (Y - C) \\ WSR &= TABR(1) + TABR(2) [TTY - TABR(3)] \end{aligned} \right] \\
 & \left[\begin{aligned} I &= A \sin \left(\frac{2\pi Y}{B} \right) \\ WSI &= TABi(1) \sin \left(\frac{TTY}{TABI(2)} 2\pi \right) \end{aligned} \right] \\
 & \left[\begin{aligned} U &= 0. \\ WSU &= 0. \end{aligned} \right] \\
 & \left[\begin{aligned} V &= A + B (Y - C) \\ WSV &= TABUV(1) + TABUV(2) [TTY - TABUV(3)] \end{aligned} \right]
 \end{aligned}$$

4. FIT 4 - FIT 6

These are dummy subroutines. Any analytical expression for density, energy and velocity may be programmed into these remaining subroutines by following the prescribed formats.

EXAMPLE OF A CONFIGURATION

Example of the input required to use the TCLAM code to generate starting conditions for TOLL code.



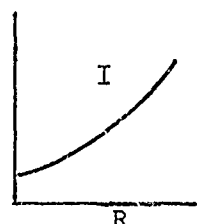
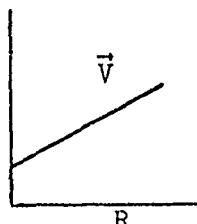
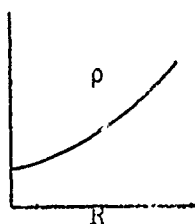
iMAX = 20

jMAX = 29

A hot source 60 cm from the ground

We will assume the source to be of the same material as the ground, and that the air be the other material.

The distribution of the hot source might be as follows:



- 6 ΔX 's of 10 cm
- 14 ΔX 's of 20 cm
- 10 ΔY 's of 20 cm
- 12 ΔY 's of 10 cm
- 7 ΔY 's of 20 cm

Package (1) generates a rectangle (the ground) from $X_1 = 0$ to $X_2 = 340$ cm and from $Y_1 = 0$ to $Y_2 = 200$ cm

1 particle/cell and $YC = 0 = XC$ (X material)

$$\rho = 1.97$$

$$I = 0$$

$$U = V = 0$$

Use FIT 1

Package (2) generates a circle of radius 50 cm at $X = 0$ and $Y = 260$ cm. Generate 16 particles/cell and $YC = 260$ cm and $XC = 0$ (X material). The analytical expressions for the ρ , $\vec{V}$ and I are as follows:

$$\rho = A + BR + CR^2$$

$$I = G + HR + KR^2$$

$$\vec{V} = E + FR$$

To achieve the radial distribution for the 2 velocity components, we will program a new FIT subroutine, say FIT 4:

$$WS = (TTX^2 + TTY^2)^{\frac{1}{2}} \equiv R$$

$$WSR = TABR(1) + TABR(2) WS + TABR(3) WS^2 \equiv A + BR + CR^2 = \rho \text{ (density)}$$

$$WSi = TABi(1) + TABi(2) WS + TABi(3) WS^2 \equiv G + HR + KR^2 = I \text{ (energy)}$$

$$WSA = TABUV(1) + TABUV(2) WS \equiv E + FR$$

$$WSU = \frac{TTX}{WS} [WSA] \equiv \frac{X}{R} [E + FR] = U \text{ (radial velocity component)}$$

$$WSV = \frac{TTY}{WS} [WSA] \equiv \frac{Y}{R} [E + FR] = V \text{ (axial velocity component)}$$

Package (3) generates a rectangle from $X1 = 0$ to $X2 = 340$ cm and for $Y1 = 200$ to $Y2 = 440$ cm.

16 particles/cell and $YC = XC = 0$ (dot material)

$$\rho = 1 \times 10^{-3}$$

$$I = 0$$

$$U = V = 0$$

Use FIT 1

The values of the TABI, TABR and TABUV arrays are read in on the 51, 52, and 53 cards.

A card with a 2 in Column 1 completes the data for the TCLAM code.

A subroutine SETUP is available to generate the initial grid (bypasses this generator code TCLAM) if both the target and projectile are of the same density. In addition, the projectile must be a right cylinder, with all the ΔX 's as constant, and all ΔY 's constant.

OUTPUT FROM TCLAM

The output, to be written on a binary tape, from the TCLAM code is the entire Z block (defined below), all the cell quantities (the two velocity components, the mass and internal energy), and the cell dimensions and areas. In the case where it is a particle run, the particles (their two coordinates and mass) and the i and j of the cell where the particle is located) are also put onto the binary tape.

The normal system of units are the cm-g-shake, where the units of energy are jerks/g and the pressure in units of jerks/cm³ (1 jerk = 10^{16} ergs and 1 shake = 10^{-8} sec).

The Z block or array contains the number of cells, the number of zones in both directions, and other necessary information to start the TOIL code. Below is a complete list of the Z block generated, which is then written on the binary output tape.

<u>Z</u>	<u>Equiv.</u>	<u>Units</u>	<u>Description</u>
1	PROB	-	Equals problem number, input to TCLAM.
2	CYCLE	-	Equals cycle number = 0.
3	DT	shake	Set to 0 by TCLAM.
4	PRINTS	-	Set to 0 by TCLAM.
5	PRINTL	-	Set to 0 by TCLAM.
6	DUMPT7	-	Set to 0 by TCLAM.
7	CSTOP	-	Set to 0 by TCLAM.
8	PIDY	-	Equals $\pi = 3.1415927$.
9	TMZ	grams	Total mass (X + .) generated by TCLAM.
10	GAM	-	If = 0. a cylindrical problem.
11	GAMD	-	Set to 0 by TCLAM.
12	GAMX	-	Set to 0 by TCLAM.
13	ETH	jerk	Total energy in system.
14	FFA	-	Set to 0 by TCLAM.
15	FFB	-	Set to 0 by TCLAM.
16	TMDZ	grams	Total mass (.) generated by TCLAM.
17	TMXZ	grams	Total mass (X) generated by TCLAM.
18	XMAX	cm	= X(iMAX).
19	TXMAX	cm	= 2.XMAX.
20	TYMAX	cm	= 2.YMAX (NOTE: YMAX is not in Z block).
21	AMDM	grams	= minimum mass/2. of the dot particles.
22	AMXM	grams	= minimum mass/2. of the X particles.
23	DNN	-	Set to 0 by TCLAM.
24	DMIN	-	Set to 0 by TCLAM.
25	FEF	-	Set to 0 by TCLAM.
26	DTNA	-	Set to 0 by TCLAM.

<u>Z</u>	<u>Equiv.</u>	<u>Units</u>	<u>Description</u>
27	CVIS	-	Set to 0 by TCLAM.
28	NPR	-	Set equal to 6 in TCLAM.
29	NPRi	-	TCLAM sets NPRi = N_4^1 (check definition of $N_4^1(Z(54))$).
30	NC	-	Fixed value of cycle number, set to 0 by TCLAM.
31	NPC	-	Used as indices in TCLAM.
32	NRC	-	Used as indices in TCLAM.
33	iMAX	-	Input to TCLAM = maximum number of zones in X direction for this run.
34	iMAXA	-	Equal iMAX + 1.
35	jMAX	-	Input to TCLAM = maximum number of zones in Y direction for this run.
36	jMAXA	-	= jMAX + 1.
37	KMAX	-	= (iMAX)(jMAX) + 1.
38	KMAXA	-	= KMAX + 1.
39	NMAX	-	= total number of particles + 1 that TCLAM has generated.
40	ND	-	= total number of dot particles + 1 that TCLAM has generated.
41	KDT	-	Set to 0 by TCLAM.
42	iXMAX	-	= iMAXA + 1.
43	NOD	-	Used as index.
44	NOPR	-	Set equal to N_3 (Note definition of $N_3(Z(53))$).
45	NiMAX	-	Set to 0 by TCLAM.
46	NjMAX	-	Set to 0 by TCLAM.
47	i1	-	Set to 0 by TCLAM.
48	i2	-	Set to 0 by TCLAM.
49	i3	-	Set to 0 by TCLAM.
50	i4	-	Set to 0 by TCLAM.
51	N1	-	= scratch tape number.
52	N2	-	= scratch tape number.
53	N3	-	= number of particle records of length $N_4^1 - 1$ that TCLAM has generated.
54	N_4^1	-	= number of particle records + 1 to be stored on each particle tape record.
55	N5	-	Set to 0 by TCLAM.

<u>Z</u>	<u>Equiv.</u>	<u>Units</u>	<u>Description</u>
56	N6	-	= number of particles on the last particle tape record.
57	N7	-	= binary tape designation number.
58	N8	-	Set to 0 by TCLAM.
59	N9	-	Set to 0 by TCLAM.
60	N10	-	Set to 0 by TCLAM.
61	N11	-	Set to 0 by TCLAM.
62	NRM	-	Set to 0 by TCLAM.
63	TRAD	-	Set to 0 by TCLAM.
64	XNRG	-	Set to 0 by TCLAM.
65	SN	-	Set to 0 by TCLAM.
66	DXN	-	Set to 0 by TCLAM.
67	RADER	-	Set to 0 by TCLAM.
68	RADET	-	Set to 0 by TCLAM.
69	RADEB	-	Set to 0 by TCLAM.
70	DTRAD	-	Set to 0 by TCLAM.
71	REZFCT	-	Set to 0 by TCLAM.
72	RSTOP	-	Set to 0 by TCLAM.
73	SHELL	-	A counter that may be used to distinguish between codes.
74	BBOUND	-	Set to 0 by TCLAM.
75	TOZONE	-	Set to 0 by TCLAM.
76	EDK	-	Set to 0 by TCLAM.
77	SBOUND	-	Set to 0 by TCLAM.
78	X1	-	Set to 0 by TCLAM.
79	X2	-	Set to 0 by TCLAM.
80	Y1	-	Set to 0 by TCLAM.
81	Y2	-	Set to 0 by TCLAM.
82	CABLN	-	Set to 0 by TCLAM.
83	VISC	-	Set to 0 by TCLAM.
84	T	-	Set to 0 by TCLAM.
85	GMAX	-	Set to 0 by TCLAM.
86	WSGD	-	Set to 0 by TCLAM.
87	WSGX	-	Set to 0 by TCLAM.

<u>Z</u>	<u>Equiv.</u>	<u>Units</u>	<u>Description</u>
88	GMADR	-	Set to 0 by TCLAM.
89	GMAXR	-	Set to 0 by TCLAM.
90	S1	-	Set to 0 by TCLAM.
91	S2	-	Set to 0 by TCLAM.
92	S3	-	Set to 0 by TCLAM.
93	S4	-	Set to 0 by TCLAM.
94	S5	-	Set to 0 by TCLAM.
95	S6	-	Set to 0 by TCLAM.
96	S7	-	Set to 0 by TCLAM.
97	S8	-	Used for storage of FIT number for each package in TCLAM.
98	S9	-	Set to 0 in TCLAM.
99	S10	-	Set to 0 in TCLAM.

Z(100) through Z(150) is set to 0 by TCLAM.

The printed output from TCLAM is as follows:

1. The problem number, iMAX and jMAX.
2. A table of the values of X(i) from i = 1 to iMAX.
3. A table of the values of Y(j) from j = 1 to jMAX.
4. A table of the values of DX(i) from i = 1 to iMAX.
5. A table of the values of DY(j) from j = 1 to jMAX.
6. A table of the area's (in the axial direction) from i = 1 to iMAX.
7. Following this preliminary printout of the grid quantities we have the following information printed out per package:
 - (a) The package number and the number of particles per cell.
 - (b) The 6 constants for the density, energy and velocity fits.
 - (c) The type of geometry, generate or delete, followed by the coordinates of the geometry.
 - (d) The minimum and maximum i and j values of the package in question.
 - (e) The total number of particles, the type, and the total energy and mass in this package.

8. After all package information is edited, a statement will appear as follows: "There are no more packages." The total energy of the system is edited, followed by the total mass and particles. A statement "Tape dump at time 0" appears next. This indicates that the binary tape was written successfully.
9. An edit of each column of the occupied grid appears next. This contains the X, DX for the column, and the Y, DY, U, V, AiD, AiX, AMD and AMX as a function of j. This completes the printed output from the TCLAM code.

B. TOIL

BASIC EQUATIONS

The Eulerian equations we wish to solve are the following:

$$(A) \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0.$$

$$(B) \quad \frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = - \nabla P$$

$$(C) \quad \frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E \vec{u}) = - \nabla \cdot (P \vec{u})$$

Equation (A) is the conservation of mass equation (B) is the conservation of momentum, and (C) is the conservation of energy equation.

The second terms on the left side of Eqs. (B) and (C) are temporarily dropped. Their contributions are later approximated when we move mass across cell boundaries.

Rewriting Eqs. (A), (B), and (C) in cylindrical coordinates with axis of symmetry results in Eqs. (1), (2), (3), and (4).

$$\frac{\partial \rho}{\partial t} = - \frac{\partial r \rho u}{r \partial r} - \frac{\partial \rho v}{\partial z} \quad (1)$$

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial P}{\partial r} \quad (2)$$

$$\rho \frac{\partial v}{\partial t} = - \frac{\partial P}{\partial z} \quad (3)$$

$$\rho \frac{\partial E}{\partial t} = - \frac{\partial r P u}{r \partial r} - \frac{\partial P v}{\partial z} \quad (4)$$

$$P = f(\rho, I) \text{ Equation of State} \quad (5)$$

ρ = density of cell (K) in g/cm³,

r = r coordinate in cm,

z = z coordinate in cm,

u = radial component of velocity in cm/shake,

v = axial component of velocity in cm/shake,

P = material pressure in jerks/cm³,

E = total specific energy in jerks/g,

I = specific internal energy in jerks/g (1 jerk = 10^{16} ergs), and

t = time in shakes (1 shake = 10^{-8} sec).

The five variables listed are all located at the center of the cell.

Rewriting Eq. (4):

$$\rho \frac{\partial}{\partial t} \left[I + \frac{1}{2} (u^2 + v^2) \right] = - \frac{\partial r P u}{r \partial r} - \frac{\partial P v}{\partial z}$$

or

$$\rho \frac{\partial I}{\partial t} + \rho u \frac{\partial u}{\partial t} + \rho v \frac{\partial v}{\partial t} = - \frac{P}{r} \frac{\partial u r}{\partial r} - u \frac{r \partial P}{r \partial r} - v \frac{\partial P}{\partial z} - P \frac{\partial v}{\partial z}$$

but

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial P}{\partial r} \quad \text{and} \quad \rho \frac{\partial v}{\partial t} = - \frac{\partial P}{\partial z}$$

thus

$$\rho \frac{\partial I}{\partial t} = - P \left(\frac{\partial v}{\partial z} + \frac{1}{r} \frac{\partial u r}{\partial r} \right).$$

This is then, the internal energy equation that we will integrate in the first phase of our calculations. As mentioned previously, the solution to the three equations is completed in two steps. The first step (called PH1 in TOIL), the momentum and energy equation as a function of the pressure forces only, are solved. Then in the second step (called PH2) we approximate those transport terms (convective terms) that we omitted in the first phase by transporting mass, momenta and energy across the cell boundaries.

In the discussion to follow, we approximate the partial differential equations by difference equations.

The radial momentum equation (2) becomes in difference form

$$\rho \frac{\partial u}{\partial t} = \frac{P_{i-3/2, j-1/2} - P_{i+1/2, j-1/2}}{2 \Delta r_i}$$

and the axial momentum equation (3) becomes

$$\rho \frac{\partial v}{\partial t} = \frac{P_{j-3/2, i-1/2} - P_{j+1/2, i-1/2}}{2\Delta z_j}$$

where the acceleration of a cell is only a function of its two neighbor cells (not of itself).

Defining

$$P_L^n = \frac{P_{i-3/2}^n + P_{i-1/2}^n}{2}$$

$$P_{RR}^n = \frac{P_{i-1/2}^n + P_{i+1/2}^n}{2}$$

$$P_{BLO}^n = \frac{P_{j-3/2}^n + P_{j-1/2}^n}{2}$$

$$P_{ABOVE}^n = \frac{P_{j-1/2}^n + P_{j+1/2}^n}{2}$$

and substituting these interface pressures into the 2 momentum equations results in

$$\tilde{u}_{(i-1/2, j-1/2)} - u_{(i-1/2, j-1/2)}^n = \frac{\Delta t}{\rho_k} \left[\frac{P_L^n - P_{RR}^n}{\Delta r_i} \right]$$

where

$$k = i-1/2, j-1/2$$

or

$$\Delta u = \frac{2\pi \Delta t r_{i-1/2} D y_j}{AMX_k} [P_L^n - P_{RR}^n]$$

and

$$\tilde{v}_k - v_k^n = \frac{\Delta t}{\rho_k} \left(\frac{P_{BLO}^n - P_{ABOVE}^n}{\Delta z_j} \right)$$

Note ($\sim$) not $n+1$ is designated for the new velocities, since these changes are due to the pressure forces only.

$$\Delta v_k = \frac{\pi \Delta t (r_i^2 - r_{i-1}^2) (P_{BLO}^n - P_{ABOVE}^n)}{AMX_k}$$

and the change in specific internal energy becomes

$$\rho_k \frac{\partial \tilde{I}_k}{\partial t} = - P_k^n \left[\frac{v_{i-1/2, j+1/2}^{n+1/2} - v_{i-1/2, j-3/2}^{n+1/2}}{2\Delta Z_j} + \frac{r_{i+1/2} u_{i+1/2, j+1/2}^{n+1/2} - r_{i-3/2} u_{i-3/2, j-1/2}^{n+1/2}}{2r_{i-1/2} \Delta r_i} \right]$$

The reason for the velocities at time $n+1/2$ is apparent for energy conservation.

Defining

$$u_{i-1/2, j-1/2}^{n+1/2} = \frac{\tilde{u}_{i-1/2, j-1/2} + u_{i-1/2, j-1/2}^n}{2}$$

and

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

and

$$VBLO = \frac{v_{j-1/2} + v_{j-3/2}}{2}$$

$$VABOVE = \frac{v_{j-1/2} + v_{j+1/2}}{2}$$

where $VBLO$, $VABOVE$, UL and URR are calculated at time n and $\sim$, and

$$UL = \frac{u_{i-1/2} r_{i-1/2} + u_{i-3/2} r_{i-3/2}}{2}$$

and

$$URR = \frac{u_{i-1/2} r_{i-1/2} + u_{i+1/2} r_{i+1/2}}{2.}$$

then

$$\rho \frac{\partial \tilde{I}}{\partial t} = P^n \left[\frac{VBLO + \tilde{VBLO}}{2\Delta Z_j} - \frac{VABOVE + \tilde{VABOVE}}{2.\Delta Z_j} - \frac{URR^n + URR^{\sim} - UL^n - UL^{\sim}}{2r_{i-1/2} \Delta r_i} \right]$$

or

$$\begin{aligned} \tilde{I}_{i-1/2,j-1/2} &= I_{i-1/2,j-1/2}^n + \frac{\Delta t P_{i-1/2,j-1/2}^n}{AMX_{i-1/2,j-1/2}} \\ &\left[\left(\frac{VBLO + \tilde{VBLO}}{2.} - \frac{VABOVE + \tilde{VABOVE}}{2.} \right) \pi (r_i^2 - r_{i-1}^2) \right. \\ &\left. + \left(-URR - URR + UL_j^n + UL_j^{\sim} \right) \pi \Delta Z_j \right] \end{aligned}$$

where again

$$k = i-1/2, j-1/2 .$$

The change in internal energy for the entire cell is

$$\Delta Q = \Delta t (\text{Vol}) P \left[\frac{\partial v}{\partial z} + \frac{1}{r} \frac{\partial ru}{\partial r} \right] .$$

The change in specific internal energy for each material is proportional to the density of each material, or

$$\Delta I_x = \frac{\Delta Q}{f} \frac{1}{M_x + \frac{M}{1-f}}$$

and

$$\Delta I_{\cdot} = \frac{\Delta Q}{1-f} \frac{1}{M_x + \frac{M}{1-f}}$$

where (x) and (·) refer to the two different materials and f is the factor to multiply times the volume of the total cell to calculate the volume occupied by (x) material. The factor f is calculated from the equation of state, where we iterate on the densities until the pressures of each material are the same.

The solution of the momentum equations provide no difficulties, however, the solution to the energy equation requires the velocities at two different time steps.

We have chosen to make two passes through this routine, the first pass to integrate the momentum equations, and formulate the interface velocities (using the old velocities for their contributions to the work term) and the second pass to bypass the momentum equations, and just compute the new interface velocities for their contribution to the work term.

Another choice might be to solve the equations in one pass through, looking ahead two cells above and two cells to the right.

As an example, we will look at the energy conservation, say, in the axial direction. The radial direction would be very similar.

Since we have dropped the transport terms, our integration of the momentum and energy equations have not been advanced to time $(n+1)$. As customary, we designate the PHASE 1 velocities and energy as $\tilde{u}$, $\tilde{v}$ and $\tilde{I}$.

$$\tilde{v}_{j-1/2} = v_{j-1/2}^n + \frac{\Delta t}{\rho_{j-1/2}^n} \left[\frac{P_{j-3/2}^n - P_{j+1/2}^n}{2 \Delta y_j} \right]$$

and

$$\tilde{I}_{j-1/2} = I_{j-1/2}^n + \frac{\Delta t P_{j-1/2}^n}{\rho_{j-1/2}^n} \left[\frac{\bar{v}_{j-3/2} - \bar{v}_{j+1/2}}{2 \Delta y_j} \right]$$

where

$$\bar{v}_{j-3/2} = \frac{\tilde{v}_{j-3/2} + v_{j-3/2}^n}{2}$$

$$\bar{v}_{j+1/2} = \frac{\tilde{v}_{j+1/2} + v_{j+1/2}^n}{2}$$

Before entering PH1, where the quantities are at time n , the total energy of the system (again, we are referring to the axial direction only) is

$$E^n = \sum_{j=1}^{jMAX} \text{MASS}_{j-1/2} \left[I_{j-1/2}^n + \frac{1}{2} \left(v_{j-1/2}^n \right)^2 \right]$$

and the total energy at the end of Phase 1 is then

$$\tilde{E} = \sum_{j=1}^{jMAX} \text{MASS}_{j-1/2} \left[\tilde{I}_{j-1/2} + \frac{1}{2} \left(\tilde{v}_{j-1/2} \right)^2 \right]$$

the total change being $\Delta E = E^n - \tilde{E}$ should be equal to 0. for energy conservation.

$$\Delta E = \sum_{j=1}^{jMAX} \text{MASS}_{j-1/2} \left[I_{j-1/2}^n - \tilde{I}_{j-1/2} + \frac{1}{2} \left(v_{j-1/2}^n \right)^2 - \frac{1}{2} \left(\tilde{v}_{j-1/2} \right)^2 \right]$$

the Δ kinetic terms can be represented by

$$\left[\frac{v_{j-1/2}^n + \tilde{v}_{j-1/2}}{2} \right] \left[v_{j-1/2}^n - \tilde{v}_{j-1/2} \right]$$

or

$$\bar{v}_{j-1/2} \left(v_{j-1/2}^n - \tilde{v}_{j-1/2} \right)$$

or

$$\begin{aligned} \Delta E &= \sum_{j=1}^{jMAX} \text{MAXX}_{j-1/2} \left[I_{j-1/2}^n - \tilde{I}_{j-1/2} + \bar{v}_{j-1/2} \left(v_{j-1/2}^n - \tilde{v}_{j-1/2} \right) \right] \\ &= \sum_{j=1}^{jMAX} \text{MASS}_{j-1/2} \left[- \frac{\Delta t P_{j-1/2}^n}{\rho_{j-1/2}^n} \left(\frac{\bar{v}_{j-3/2} - \bar{v}_{j+1/2}}{2\Delta y_j} \right) \right. \\ &\quad \left. - \bar{v}_{j-1/2} \left(\frac{\Delta t}{\rho_{j-1/2}^n} \left(\frac{P_{j-3/2}^n - P_{j+1/2}^n}{2\Delta y_j} \right) \right) \right] \\ &= \Delta t \sum_{j=1}^{jMAX} \frac{\text{MASS}_{j-1/2}}{\rho_{j-1/2}^n \Delta y_j} \left[- P_{j-1/2}^n \bar{v}_{j-3/2} + P_{j-1/2}^n \bar{v}_{j+1/2} \right] \end{aligned}$$

$$\begin{aligned}
 & - P_{j-1/2}^n \bar{v}_{j-1/2} + P_{j+1/2}^n \bar{v}_{j-1/2} \Big] \\
 (A) \quad & = - \frac{\Delta t}{2} \sum_{j=1}^{j_{MAX}} \pi (r_i^2 - r_{i-1}^2) \Big[P_{j-1/2}^n \bar{v}_{j-3/2} + P_{j-3/2}^n \bar{v}_{j-1/2} \\
 & - P_{j+1/2}^n \bar{v}_{j-1/2} - P_{j-1/2}^n \bar{v}_{j+1/2} \Big]
 \end{aligned}$$

Thus the last two terms in j being cancelled by the first two terms in $j+1$. Now by prescribing the proper boundary conditions, we will have exact energy conservation for the entire grid.

EXAMPLE:

For $j = 1$

$$\begin{array}{l}
 j = 1 \quad P_{1/2} v_{-1/2} + P_{-1/2} v_{1/2} - P_{3/2} v_{1/2} - P_{1/2} v_{3/2} \\
 j = 2 \quad P_{3/2} v_{1/2} + P_{1/2} v_{3/2} - P_{5/2} v_{3/2} - P_{3/2} v_{5/2} \\
 j = 3 \quad P_{5/2} v_{3/2} + P_{3/2} v_{5/2} - P_{7/2} v_{5/2} - P_{5/2} v_{7/2}
 \end{array}$$

Thus we have non-cancellation of the first two and last two terms.

For our first example, assume the bottom boundary is reflective. Referring to Eq. (A) we have two terms that will not be cancelled as j increases, these terms are

$$P_{1/2}^n \bar{v}_{-1/2} + P_{-1/2}^n \bar{v}_{1/2}.$$

We set the pressure of the mirror cell

$$(P_{-1/2}^n) = P_{1/2}^n$$

(which does not imply that $\bar{v}_{1/2} = 0$.) The other condition which does lead to these two terms cancelling is that $\bar{v}_{-1/2} = -\bar{v}_{1/2}$. A similar treatment would be applied for the top boundary to be reflective.

Now, however, if we assume that the bottom boundary is transmissive, our boundary conditions are then that $\dot{v}_{1/2} = 0$. which means that

$$P_{-1/2}^n = P_{3/2}^n .$$

The condition on the velocity is that $v_{-1/2} = v_{1/2}$. Now this leaves us with the first two terms $P_{1/2} v_{1/2} + P_{3/2} v_{1/2}$. This term then is adding or subtracting energy to the system (depends on sign of velocity). To compensate, or a better word to use might be to keep the books straight, we also add this term to the quantity called Eth.

Eth is defined as the total energy at time = 0., less the energy lost by mass leaving the grid + the energy added if negative interval energies appear in the transport phase + the energy loss or gain at the transmissive boundary conditions in PHL.

A similar prescription would apply for the top boundary being transmissive. The conservation of energy in the radial direction follows the same logic and will not be repeated.

The term subtracted from Eth for the boundary at the right is

$$\frac{P_{(k)} + P_{(\text{cell to the left})}}{2} u_{(k)} r_{i-\frac{1}{2}} \pi \Delta t D Y_{(j)}$$

and the top is

$$\frac{P_{(k)} + P_{(\text{cell below})}}{2} v_{(k)} \pi (r_i^2 - r_{i-1}^2) \Delta t (.5)$$

and the bottom, if transmissive, is

$$\frac{P_{(k)} + P_{(\text{cell above})}}{2} v_{(k)} \pi (r_i^2 - r_{i-1}^2) \Delta t (.5)$$

and is added to ETH. K (in the above equations) refers to the border cell.

The left boundary (axis of symmetry) is always reflective, the bottom may be reflective or transmissive and the top and right are always transmissive.

Rewriting Eq. (1), the mass transport equation in finite difference form results in

$$\frac{\rho_{(k)}^{n+1} - \rho_{(k)}^n}{\Delta t} = \frac{r_{i-1} \rho_{i-1} u_{i-1}}{r_{i-\frac{1}{2}} \Delta r_i} - \frac{r_i \rho_i u_i}{r_{i-\frac{1}{2}} \Delta r_i} + \frac{\rho_{j-1} v_{j-1} - \rho_j v_j}{\Delta z(j)} \quad (6)$$

where

$$\Delta z(j) = \frac{V_{(k)}}{A_j^z} = \frac{V_{(k)}}{A_{j-1}^z}$$

where A^z for all $j = \pi(r_{(i)}^2 - r_{(i-1)}^2)$, and

$$V_{(k)} = \text{volume of cell } k = 2\pi r_{i-\frac{1}{2}} \Delta r_i \Delta z(j) \quad (7)$$

multiply both sides of Eq. (7) by r_i results in

$$V_{(k)} r_i = 2\pi r_i \Delta z_j r_{i-\frac{1}{2}} \Delta r_i$$

or

$$V_{(k)} r_i = A_i^r r_{i-\frac{1}{2}} \Delta r_i \quad (8)$$

where A^r = area in the direction perpendicular to the Z axis. And similarly, multiplying Eq. (7) by r_{i-1} results in

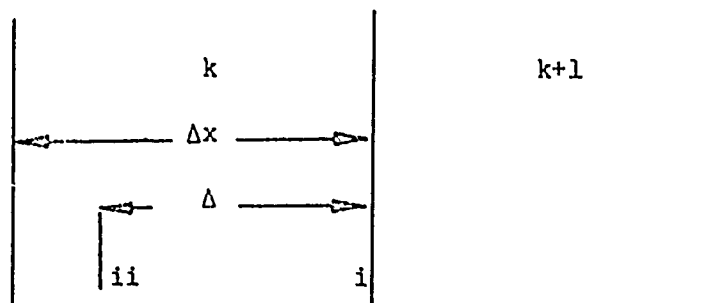
$$V_{(k)} r_{i-1} = 2\pi r_{i-1} \Delta z_j r_{i-\frac{1}{2}} \Delta r_i$$

or

$$V_{(k)} r_{i-1} = A_{i-1}^r r_{i-\frac{1}{2}} \Delta r_i \quad (9)$$

Solving Eqs. (8) and (9) for $r_{i-\frac{1}{2}} \Delta r_i$ and substituting their values into Eq. (6) results in

$$\frac{\rho_{(k)}^{n+1} - \rho_{(k)}^n}{\Delta t} = \frac{1}{V_{(k)}} (A_{j-1}^z \rho_{j-1} v_{j-1} - A_j^z \rho_j v_j + A_{i-1}^r \rho_{i-1} u_{i-1} - A_i^r \rho_i u_i)$$



The mass to move across i is between i and i_i where $\Delta = i - i_i$; thus $\Delta = \tilde{u} \Delta t$ where $\tilde{u}$ is the weighted velocity at Δ . Using the first two terms of the Taylor series at a distance of $-\Delta$ from i , we expand

$$u_{(i)} = \frac{u_{(k)} + u_{(k+1)}}{2.}$$

or

$$\tilde{u} = \frac{u_{(k)} + u_{(k+1)}}{2.} + (-\Delta) \frac{(u_{(k+1)} - u_{(k)})}{\Delta x}$$

or

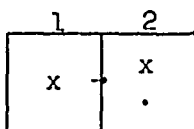
$$\frac{\Delta}{\Delta t} = \tilde{u} = \frac{\frac{u_{(k)} + u_{(k+1)}}{2.}}{\frac{(u_{(k+1)} - u_{(k)})}{\Delta x} \Delta t + 1}$$

If $u_{(k)} + u_{(k+1)}/2. > 0$ use $\rho_{(k)}$; otherwise use $\rho_{(k+1)}$ in the calculation of the mass flux. The density (ρ) is the total mass (x and dot) over the volume of the cell.

Mass, both components of momentum, and the energy across all four sides of the cell for both materials are calculated. By conserving both axial and radial momentum and the total energy, the new velocities are calculated and the new internal energy is then the difference between the total and the kinetic. Thus, up to this point, we have calculated the mass fluxes, now we must determine (for a mixed cell) how much of each material to move. Three possible situations concerned with two materials arise.

1. Material moving from a non-mixed cell to a mixed cell. This presents no difficulty or modification.

Example (non-mixed to mixed)



Mass flow is from cell 1 to cell 2.

$$\Delta M = \rho^1 \bar{U} A \Delta t$$

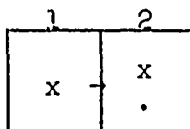
$$\Delta M_x = \Delta M$$

$$\Delta M = 0$$

where ρ^1 = density of x material in cell 1.

2. Material moving from a mixed cell to a non-mixed cell is calculated as follows: The acceptor material from the donor cell is moved to the acceptor cell. If the flux is such that this will more than empty the acceptor material from the donor cell, the excess is removed by assigning it to the other material.

Example (mixed to non-mixed)



Mass flow is from cell 1 to cell 2.

$\Delta M = \rho^1 \bar{U} A \Delta t$ where ρ^1 is the total density of both materials in cell 1 if $\Delta M > M^1$.

$$\Delta M = M^1$$

and

$$\Delta M_x = \Delta M - M^1$$

if $\Delta M \leq M^1$

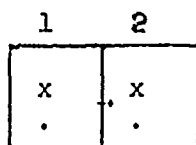
$$\Delta M = \Delta M$$

and

$$\Delta M_x = 0$$

3. Material moving from a mixed cell to a mixed cell requires some modification in order to keep the material interface defined in a single mixed cell. The prescription of recipe if you like, is that each material flux is weighted by the fraction of its mass to total in the acceptor cell, rather than the donor cell.

Example (mixed to mixed)



Mass flow is from cell 1 to cell 2.

$\Delta M = \rho^1 \bar{U} A \Delta t$, where ρ^1 is the total density of both materials in cell 1. Then

$$\Delta M_x = \frac{M_x^2}{M_x^2 + M^2} \Delta M$$

and

$$\Delta M_{\cdot} = \frac{M^2}{M_x^2 + M^2} \Delta M$$

Note: The superscripts refer to zone number of the subscript to material number.

Several techniques are available for calculating the specific internal energy of a mixed cell.

The scheme as reported in the listings of TOIL is as follows. The specific internal energy for each material is proportional to the specific total energy of that material (the total specific energy of the cell plus that which is transported in less the amount transported out).

$$\Delta E_x = \Sigma M_x \left[\tilde{I}_x + \frac{\tilde{U}^2 + \tilde{V}^2}{2} \right]$$

$$\Delta E_{\cdot} = \Sigma M_{\cdot} \left[\tilde{I}_{\cdot} + \frac{\tilde{U}^2 + \tilde{V}^2}{2} \right]$$

Where the ($\sim$) tilda refers to the velocities and specific internal energy after (PH1).

$M_x^{n+1} = \Delta M_x$ = total mass of cell plus that which is transported in less the amount transported out for the x material.

$M_{\cdot}^{n+1} = \Delta M_{\cdot}$ = similar term for the dot material.

Then the new internal energy for the mixed cell is

$$Q = \Delta E_{\cdot} + \Delta E_x - \frac{1}{2} \left(U_{n+1}^2 + V_{n+1}^2 \right) (\Delta M_x + \Delta M_{\cdot})$$

Thus each material then has the following specific internal energy

$$I_x^{n+1} = \frac{\Delta E_x}{\Delta M_x} \frac{Q}{\Delta E_x + \Delta E}$$

$$I^{n+1} = \frac{\Delta E}{\Delta M} \frac{Q}{\Delta E_x + \Delta E}$$

Several other techniques are available.

- (1) The change in specific internal energies being proportional to the mass, or

$$\frac{\Delta I_x}{\Delta I} = \frac{M_x}{M}$$

where ΔQ = change in internal energy either due to the work terms in PH1 or the transport terms in PH2.

$$\text{Then } M_x \Delta I_x + M \Delta I = \Delta Q$$

or

$$\Delta I_x = \frac{M_x \Delta Q}{M_x^2 + M^2}$$

and

$$\Delta I = \frac{M \Delta Q}{M_x^2 + M^2}$$

- (2) The change in internal energy for each material to be the same

or

$$M_x \Delta I_x = M \Delta I$$

and

$$M_x \Delta I_x + M \Delta I = \Delta Q$$

or

$$\Delta I_x = \frac{\Delta Q}{2M_x}$$

and

$$\Delta I = \frac{\Delta Q}{2M}$$

- (3) The change in specific internal energy for each material to be the same

$$\Delta I_x = \Delta I_.$$

and

$$M_x \Delta I_x + M_ . \Delta I_ . = \Delta Q$$

or

$$\Delta I_x = \frac{\Delta Q}{M_x + M_ .} = \Delta I_ .$$

C. NORMAL INPUT FOR THE TOIL CODE

An (*) designates that the work is fixed point (a 2 in Column 1) although all words are loaded via card routine in floating format. (A **, double asterisk signifies this is last card of the set.)

1st Set--Column 1-3--contains N (the number of BCD cards to follow (format I3) N BCD cards.

2nd Set

<u>LOC</u>	<u>Name</u>	<u>Description</u>
*57	N7	Binary tape dump number.
**282	PK array	PK(1) = problem number PK(2) = cycle number to start problem PK(3) = -1, for restart or starting from the TCLAM tape PK(3) = 0, starting problem at $t = 0$, via subroutine SETUP.

3rd Set

<u>LOC</u>	<u>Name</u>	<u>Description</u>
14	FFA	See usage as described in OIL report.
15	FFB	See usage as described in OIL report.
24	DMIN	$\sim 10^{-6}$ energy check $\Delta E/E/\text{cycle}$.
25	FeF	$\Delta .03$, used in iteration routine for calculating pressures in a mixed cell.
27	CVIS	Bottom boundary condition. If < 0 transmittive otherwise reflective.
*47	i1	Active grid counter in the i direction (= the i value of the right most cell(that has internal or kinetic energy)+ 2).
*48	i2	Similar to i1 but for the j direction.

<u>LOC</u>	<u>Name</u>	<u>Description</u>
*51	N1	~ 20, the maximum number of iterations allowed for calculating the pressure of a mixed zone.
77	SBOUND	Same value as for OIL ~ 1.0.
71	REZFCT	PH2 routine <u>always</u> will check for the amount of material (TOZONE) leaving the top or right boundaries. It may trigger REZONE (sets REZ = 1.). At the completion of PH2, if REZ = 1. a further check is done, using REZFCT (if = 1.) code will call the subroutine REZONE, otherwise it will ignore the flag set in PH2.
75	TOZONE	The mass flux at a interior free surface is set to 0. unless the flux produces a density > TOZONE, if so mass is allowed to move.
82	CARLN	See usage as described in the OIL report.
86	GAMMA	(For dot material.)
87	GAMMA	(For x material.)
105	Z(105)	Fraction of stability for early times if the initial energy is primarily internal, rather than kinetic (~ .05).
106	Z(106)	Factor to increase Z(105)/cycle to build up its value to the normal value located in Z(139).
107	Z(107)	ρ_{min}^x Minimum density of x material allowed in any cell.
108	Z(108)	$\rho_{min}^{\cdot}$ Minimum density of . material allowed in any cell.
111	Z(111)	Density of (.) material to add on to target for REZONE (hypervelocity type calculations).
112	Z(112)	Initial (Z component) projectile velocity in cm/shake.
113	Z(113)	ϵ (epsilon) for emptying the bottom cells of the projectile up until the reflected shock reaches the bottom surface of the projectile.

<u>LOC</u>	<u>Name</u>	<u>Description</u>
115	ρ_0^x	The superscripts refer to the material. The definitions and numerical values for different material are the same as in the OIL report.
116	$\rho_0^\cdot$	
117	a^x	
118	$a^\cdot$	
119	E_0^x	
120	$E_0^\cdot$	
121	b^x	The superscripts refer to the material, the definitions and numerical values for the different materials are the same as in the OIL report.
122	$b^\cdot$	
123	A^x	
124	$A^\cdot$	
125	V_s^x	
126	$V_s^\cdot$	
127	E_s^x	
128	$E_s^\cdot$	
129	α^x	
130	$\alpha^\cdot$	
131	β^x	
132	$\beta^\cdot$	
133	B^x	
134	$B^\cdot$	
138	Z(138)	Minimum density that a cell must have such that a stability check will be performed on it.
139	Z(139)	Fraction of stability ($\sim .5$).
143	Z(143)	= minimum density of the dot (.) material allowable to transport across a cell or to remain in a cell ($\sim 10^{-3} \rho_0^\cdot$).

<u>LOC</u>	<u>Name</u>	<u>Description</u>
144	Z(144)	= similar term for the x material.
145	Z(145)	= ϵ (epsilon) on the specific internal energy. If A_{ix} or $A_{iD} < Z(145)$, A_{ix} or A_{iD} is set to 0. and the books are balanced.
146	Z(146)	= ϵ (epsilon) on the 2 velocity component. If $ u $ or $ v < Z(146)$, u or v are set to 0. and the books are balanced.
148	C_0	$C = C_0 + AP^\epsilon$ Where the units of C_0 and A are 10^5 cm/sec and P is in megabars. CDT routine converts (C) speed of sound to cm/shake.
149	A	
150	ϵ	
**3	DT	See usage and definitions as described in the OIL report.
4	PRINTS	
5	PRINTL	
6	DUMPT7	
7	CSTOP	

If one uses the subroutine SETUP for generating the initial configuration, the following data cards must be added to the 3rd set, and a 4th set must be added, usually duplicate the last card of the 3rd set. Subroutine SETUP implies certain restrictions as stated:

1. Constant DX and constant DY,
2. The projectile is a right circular cylinder,
3. The projectile has kinetic (Z component only) energy only.

<u>LOC</u>	<u>Name</u>	<u>Description</u>
285	PK(4)	Set = 1.
286	PK(5)	Right boundary of projectile (i).
287	PK(6)	Bottom (j) + 1 of projectile.
288	PK(7)	Top (j) of projectile.

<u>LOC</u>	<u>Name</u>	<u>Description</u>
289	PK(8)	= 1.
290	PK(9)	Right (i) boundary of target.
291	PL(10)	Bottom (j) + 1 of target.
292	PK(11)	Top (j) of target.
23389	DX(1)	DX, to be used for all (i).
23489	DY(1)	DY, to be used for all (j).
1	PROB	The problem number.
*33	iMAX	The maximum number of zones in the i direction.
*35	jMAX	The maximum number of zones in the j direction.
		NOTE: (iMAX)(jMAX) must be ≤ 4499 .
*57	N7	The binary tape number.

D. TOIL (TWO MATERIAL)

Symbol	Loc	No. of Words	Units	Description
AiD	5253	4500	jerk/g	Specific internal energy in jerk/g of cell K (for . material).
AiX	9753	4500	jerk/g	Specific internal energy in jerk/g of cell K (for X material).
AM	14253	130	many	Equivalenced to DMASL in PH2 and used for partial editing in EDIT.
AMD	14383	4500	grams	Mass in grams of cell K for (.) material.
AMDM	23	1	grams	Not used in TOIL. Set = minimum (.) particle mass/2. in TCLAM.
AMK	267	15	many	Used in EDIT, also equivalenced to UR(16).
AMX	18883	4500	grams	Mass in grams of cell K for (X) material.
AMXM	22	1	grams	Not used in TOIL. Set = minimum (X) particle mass/2. in TCLAM.
AREA	23383	1	-	Not used.
BBOUND	74	1	-	Not used.
BIG	23384	1	-	Not used.
BOUNCE	23385	1	-	Not used.
CABL	82	1	-	Same definition as in OIL code.
CSTOP	7	1	-	Cycle number at which problem will stop, edits and writes dump tape.
CVIS	27	1	-	If -, bottom boundary is transmittive, otherwise reflective.

Symbol	Loc	No. of Words	Units	Description
CYCLE	2	1	1	Cycle number.
DDVK	23387	1	-	Not used.
DDXN	23386	1	-	Not used.
DENRG	37539	130	jerks	Equivalenced to iW1, used in PH2 transport array for (.) fluxes.
DKE	652	4500	-	Partial volume of (X) material in a mixed cell, fraction of.
DMASL	14253	130	grams	Equivalenced to AM, used in PH2 transport array for (.) left flux.
DMIN	24	1	-	For energy check $\sim 10^{-6}$.
DNN	23	1	-	$= (ETH - E)/ETH$ at the last energy check cycle.
DT	3	1	shakes	Δt^{n+1} (hydro time step).
DTNA	26	1	shakes	Δt^n (old hydro time step).
DTRAD	70	1	shakes	Not used in TOIL, reserved for radiation, although calculated in CDT.
DUMPT7	6	1	-	Frequency in cycles at which code will make a binary tape dump.
DVK	23388	1	-	Not used.
DX	23389	100	cm	$DX(i) = X(i) - X(i - 1)$.
DXML	37264	130	g-cm-shake	Equivalenced to XL in PH2, contains the (.) (X) component of momentum at the left.
DXN	66	1	-	Not used.
DY	23489	100	cm	$DY(j) = Y(j) - Y(j - 1)$.
DYML	36885	130	g-cm-shake	Equivalenced to YL array in PH2. Contains Y component of momentum at left for (.) material.
E	23589	1	jerks	Used in REZONE as total energy of new cell.

Symbol	Loc	No. of Words	Units	Description
ECK	76	1	-	Energy check = $\frac{ETH^n - E^n}{ETH^n} - \frac{ETH^{n-m} - ETH^{n-m}}{ETH^{n-m}}$ M
ETH	13	1	jerks	Total energy of system less any that leaves or adjustments at transmittive grid boundaries.
FD	23590	1	-	Not used.
FEF	25	1	-	e factor in ES.
FFA	14	1	-	Upper limit for stability and to calculate Δt only if CABLN = 0.
FFB	15	1	-	Lower limit for stability and to calculate Δt only if CABLN = 0.
FLEFT	252	100	grams	Used in PH2 as X mass at left boundary.
FS	23591	1	-	Used in PH1, PH2, independent of each.
FX	23592	1	-	Not used.
GAM	10	1	-	Not used.
GAMC	452	1	many	Equivalenced to PL of PR.
GAMD	11	1	-	$1./(\gamma.-1.)$ Used for gamma law equation of state.
GAMX	12	1	-	$1./(\gamma X-1.)$ Used for gamma law equation of state.
GMADR	88	1	-	$\gamma./(\gamma.-1)$ Calculated but not used.
GMAX	85	1	-	Maximum gamma.
GMAXR	89	1	-	$\gamma X/(\gamma X-1.)$ Calculated, but not used.
I	37531	1	-	Index in X direction.
II	37532	1	-	Working index, used in INPUT.

Symbol	Loc	No. of Words	Units	Description
IMAX	38	1	-	Maximum number of zones in the X direction.
IMAXA	34	1	-	= iMAX + 1, never used.
IN	37533	1	-	Not used.
IR	37534	1	-	Not used.
IWS	37535	1	-	Used as working storage in INPUT and CDT.
IWSA	37536	1	-	Not used.
IWSB	37537	1	-	Not used.
IWSC	37538	1	-	Not used.
IW1	37539	130	many	Equivalenced to DENRG.
IXMAX	42	1	-	Calculated as iMAX = 2 in TCLAM, never used. SETUP does not calculate it.
IZ	7	150	many	Fixed point. Block equivalenced to Z.
I1	47	1	-	Active grid counter in X direction, maximum value = iMAX.
I2	48	1	-	Active grid counter in Y direction, maximum value = jMAX.
I3	49	1	-	Not used.
I4	50	1	-	Not used.
J	37669	1	-	Index in Y direction (temporary).
JMAX	35	1	-	Maximum number of zones in the Y direction.
JMAXA	36	1	-	= jMAX + 1, never used.
JN	37670	1	-	Not used.
JP	37671	1	-	Not used.
JR	37672	1	-	Not used.

Symbol	Loc	No. of Words	Units	Description
K	37673	1	-	Index of center of cell, defined as $= (j-i) \text{ iMAX} + i + 1$.
KDT	41	1	-	Flag in CDT to signal print rework DT has changed.
kMAX	37	1	-	$= (\text{iMAX})(\text{jMAX}) + 1$.
kMAXA	38	1	-	$= \text{kMAX} + 1$.
KN	37674	1	-	Not used.
KP	37675	1	-	Index in PH2 (temporary).
KR	37676	1	-	Not used.
KRM	37677	1	-	Not used.
L	37678	1	-	Index (temporary).
M	37679	1	-	Index (temporary).
MA	37680	1	-	Index (used in SETUP).
MB	37681	1	-	Index (used in SETUP).
MC	37682	1	-	Index (used in SETUP).
MD	37683	1	-	Index (used in SETUP).
ME	37684	1	-	Index (used in SETUP).
MZ	37685	1	-	Index (used in SETUP).
N	37686	1	-	Index (temporary).
NC	30	1	-	Fixed point value of cycle number
ND	40	1	-	Used temporarily in PH2.
NIMAX	45	1	-	$= \text{iMAX}/2$ calculated in REZONE.
NJMAX	46	1	-	$= \text{jMAX}/2$ calculated in REZONE.
NK	37687	1	-	Index in EDIT.
NKMAX	37688	1	-	Not used.
NK1	37689	1	-	Index in EDIT.

Symbol	Loc	No. of Words	Units	Description
NMAX	39	1	-	Flag for stability check in radial direction (PIC or OIL).
NO	37690	1	-	Not used.
NOD	43	1	-	Not used.
NOPR	44	1	-	Not used.
NPC	31	1	-	Number of cycles between energy checks.
NPR	28	1	-	Not used.
NPRi	29	1	-	Not used.
NR	37691	1	-	Identification of routine when a dump is called.
NRC	32	1	-	Used as flag for advancing active grid counters in PH1 and PH2.
NRM	62	1	-	For radiation option, is the maximum number of radiation cycles per hydro.
N1	51	1	-	Maximum number of iterations for mixed cell pressure.
N10	60	1	-	= i of zone controlling time step.
N11	61	1	-	= j of zone controlling time step.
N2	52	1	-	Not used.
N3	53	1	-	Set = 0 in TCLAM.
N4	54	1	-	Not used.
N5	55	1	-	Not used.
N6	56	1	-	Not used.
N7	57	1	-	Binary tape number.
N8	58	1	-	Not used.
N9	59	1	-	Not used.

Symbol	Loc	No. of Words	Units	Description
OUT	23593	1	-	Not used.
P	23594	4500	jerks/ cm ³	Pressure of cell K.
PABOVE	28094	1	jerks/ cm ³	= [P(K) + P(cell above)]/2. PH1.
PBLO	28095	1	jerks/ cm ³	= [P(K) + P(cell below)]/2. PH1.
PIDTS	28096	1	many	= 1./ $\pi\Delta t\Delta y$ in PH1; 1./ $\pi\Delta t$ in PH2.
PIDY	8	1	-	π .
PK	282	15	many	Used for inputting starting data.
PL	452	200	many	Used in PH1 and PH2.
PPABOV	28097	1	-	Not used.
PR	452	200	many	Used in PH1 and PH2.
PRINTL	5	1	cycles	Number of cycles between long prints.
PRINTS	4	1	cycles	Number of cycles between short prints.
PROB	1	1	-	Problem number.
PRR	28098	1	jerks/ cm ³	= [P(K) + P(cell to right)]/2. PH1.
PUL	28099	1	-	Not used.
QDT	28100	1	-	Not used.
QK	297	15	g-cm-shake	Axial momentum in selected angles.
QOOOFL	28107	1	-	Not used.
RADEB	69	1	-	Not used.
RADER	67	1	g-cm-shake	Total positive radial momenta for X material.
RADET	68	1	g-cm-shake	Total positive axial momenta for X material.
RC	28101	1	cm	= [X(i) + X(i-1)]/2. in PH1.

Symbol	Loc	No. of Words	Units	Description
REZ	28102	1	-	If mass leaves top, right or bottom, REZ set > 0.
REZFCT	71	1	-	If REZ (trigger in PH2) > 0 and REZFCT > 0 PH2 calls REZONE.
RHO	28103	1	-	Not used.
RL	28104	1	-	Not used.
RR	28105	1	cm	$= [X(i) + X(i+1)]/2$. in PH1.
RSTOP	72	1	-	Not used.
SBOUND	77	1	-	Factor in velocity weighting PH2.
SHELL	73	1	-	Not used.
SIG	28106	1	cm	Minimum ΔX or ΔY in CDT routine.
SIGC	551	100	many	Used in PH1 and PH2.
SN	65	1	-	Not used.
SWITCH	28108	1	-	Not used.
S1	90	1	-	Not used.
S10	99	1	-	Not used.
S2	91	1	-	Not used.
S3	92	1	-	Not used.
S4	93	1	-	Not used.
S5	94	1	-	Not used.
S6	95	1	-	Not used.
S7	96	1	-	Not used.
S8	97	1	-	Not used.
S9	98	1	-	Not used.
T	84	1	shake	$t^n = t^{n-1} + \Delta t$.

Symbol	Loc	No. of Words	Units	Description
TAB	252	15	-	Tangent of 12 selected angles (EDIT).
TABLM	28109	1	-	Not used.
TAU	28110	100	cm ²	$TAU(i) = \pi(X_i^2 - X_{i-1}^2)$.
TAUDTS	28210	1	cm ² -shake	= TAU(i) * DT. (PH1)
TAUDTX	28211	1	-	Not used.
THETA	652	4500	-	Equivalenced to DKE array.
TMDZ	16	1	grams	= total (.) mass. (If TCLAM generates the data.)
TMXZ	17	1	grams	= total (X) mass. (If TCLAM generates the data.)
TMZ	9	1	grams	Total mass (X + .).
TOZONE	75	1	g/cm ³	If mass flux (across free surface) produces $\rho < TOZONE$, flux set to 0. PH2.
TRAD	63	1	shake	Δt radiation (not used in this version).
TXMAX	19	1	cm	2. * X(iMAX) never used.
TYMAX	20	1	cm	2. * Y(jMAX) never used.
U	28212	4500	cm-shake	Radial velocity component of cell K.
UK	32712	1	-	Not used.
UL	252	200	many	Arrays in PH1, PH2 of twice (jMAX).
UR	252	200	many	Arrays in PH1, PH2 of twice (jMAX).
URR	32713	1	cm ² -shake	$[U(K) X_{i-1/2} + U(K+L) X_{i+1/2}]/2$.
UT	32714	1	-	Not used.
UTEF	32717	1	-	Not used.
UU	32715	1	-	Not used.

Symbol	Loc	No. of Words	Units	Description
UUU	32716	1	-	Not used.
UVMAX	32718	1	1/shake	Minimum $ U $ or $ V /DX$ or DY . CDT.
V	32719	4500	cm/shake	Axial velocity component of cell K.
VABOVE	37219	1	cm/shake	$= [V(K) + V(\text{cell above})]/2$. PH1.
VBLO	37220	1	cm/shake	$= [V(K) + V(\text{cell below})]/2$. PH1.
VEL	37221	1	-	Flag PH1, PH2.
VISC	83	1	-	Not used.
VK	37222	1	-	Not used.
VT	37223	1	-	Not used.
VTEF	37224	1	cm/shake	Initial pellet velocity (Z(112)) if generating via subroutine SETUP.
VV	37225	1	-	Not used.
VVABOV	37226	1	-	Not used.
VVBLO	37227	1	-	Not used.
WPS	37259	1	-	Working storage.
WS	37260	1	-	Working storage.
WSA	37261	1	-	Working storage.
WSB	37262	1	-	Working storage.
WSC	37263	1	-	Working storage.
WSGD	86	1	-	γ .
WSGX	87	1	-	γX (input) ES atores .5 into it.
W2	37228	1	-	Not used.
W3	37258	1	-	Not used.

Symbol	Loc	No. of Words	Units	Description
X	152	100	cm	$X(i)$ = right boundary of cell i .
XL	37264	130	-	Used temporarily in EDIT and PH2.
XLf	32394	1	-	Not used.
XMAX	18	1	cm	$X(iMAX)$.
XN	37395	1	-	Not used.
XNRG	64	1	-	Not used.
XR	37396	1	-	Not used.
XX	151	101	cm	$XX(i) = X(i-1)$ not used.
X1	78	1	-	Not used.
X2	79	1	-	Not used.
Y	5153	100	cm	$Y(j)$, the top dimension of cell K .
YAMC	351	100	many	Used in PH1 and PH2.
YL	37397	130	many	Used temporarily in PH2 and EDIT.
YLW	37527	1	-	Not used.
YN	37528	1	-	Not used.
YU	37529	1	-	Not used.
YY	5152	101	cm	$YY(2) = Y(1)$ not used.
Y1	80	1	-	Not used.
Y2	81	1	-	Not used.
Z	1	150	many	Definitions have been made.
ZMAX	37530	1	-	Not used.

Z BLOCK

Location	Symbol	Units	Description
Z(1)	PROB	-	Problem number (if positive, this is an OIL run; if negative, this is a PIC run).
Z(2)	CYCLE	-	Cycle number (floating point value).
Z(3)	DT	shake	$\Delta t_{\text{hydro}} = t^n - t^{n-1}$.
Z(4)	PRINTS	-	Cycle frequency for short print.
Z(5)	PRINTL	-	Cycle frequency for long print.
Z(6)	DUMPT7	-	Cycle frequency for binary tape dumps.
Z(7)	CSTOP	-	Cycle number at which problem stops.
Z(8)	PIDY	-	$= \pi = 3.1415927$.
Z(9)	TMZ	grams	Total (X + .) mass at $t = 0$ (calculated in TCLAM).
Z(10)	GAM	-	Not used.
Z(11)	GAMD	-	$1./(\gamma.-1)$ $1./(\gamma X-1)$ } Computed in INPUT.
Z(12)	GAMX	-	
Z(13)	ETH	jerk	Total energy (computed in TCLAM for $t = 0$). Changed in PH1 at transmissive boundaries and in PH2 if mass leaves the system, and by radiation flow out of the system.
Z(14)	FFA	-	Upper limit for stability and to calculate Δt , only if CABLN = 0.
Z(15)	FFB	-	Lower limit for stability and to calculate Δt , only if CABLN = 0.
Z(16)	TMDZ	grams	Total (.) mass ($t = 0$) calculated in TCLAM.
Z(17)	TMXZ	grams	Total (X) mass ($t = 0$) calculated in TCLAM.
Z(18)	XMAX	cm	$= X(iMAX)$.

Location	Symbol	Units	Description
Z(19)	TXMAX	cm	2 (XMAX) $t = 0$. calculated in TCLAM.
Z(20)	TYMAX	cm	2 (YMAX) $t = 0$. calculated in TCLAM.
Z(21)	AMDM	grams	Minimum (.) particle mass/2.; calculated in TCLAM.
Z(22)	AMXM	grams	Minimum (X) particle mass/2.; calculated in TCLAM.
Z(23)	DNN	-	$(ETH - E)^{n-NPC}/ETH$.
Z(24)	DMIN	-	IE (ECK). NOTE: Z(76) > DMIN, problem will stop and the EDIT routine will call dump.
Z(25)	FEF	-	$\sim .03$ used in iteration routine for calculating pressures for partial volume.
Z(26)	DTNA	shake	Δt^{n-1} .
Z(27)	CVIS	-	If < 0, bottom boundary is transmittive; otherwise reflective boundary.
Z(28)	NPR	-	Index (working storage).
Z(29)	NPRi	-	Index (working storage).
Z(30)	NC	-	Cycle number in fixed point.
Z(31)	NPC	-	Number of cycles between short prints.
Z(32)	NRC	-	Index.
Z(33)	iMAX	-	Maximum number of zones in R direction.
Z(34)	iMAXA	-	iMAX + 1.
Z(35)	jMAX	-	Maximum number of zones in Z direction.
Z(36)	jMAXA	-	jMAX + 1.
Z(37)	kMAX	-	(iMAX)(jMAX) + 1.
Z(38)	kMAXA	-	kMAX + 1.
Z(39)	NMAX	-	Total number of particles + 1, generated in TCLAM for PIC problem only.

Location	Symbol	Units	Description
Z(40)	ND	-	Total number of (.) particles + 1 generated in TCLAM.
Z(41)	KDT	-	Defined previously.
Z(42)	IXMAX	-	Not used.
Z(43)	NOD	-	Index.
Z(44)	NOPR	-	Index.
Z(45)	NIMAX	-	New iMAX before adding new zones.
Z(46)	NJMAX	-	New jMAX before adding new zones.
Z(47)	I1	-	Defined previously.
Z(48)	I2	-	Defined previously.
Z(49)	I3	-	Not used.
Z(50)	I4	-	Not used.
Z(51)	N1	-	Maximum number of iterations allowed in ES calculation of a mixed cell.
Z(52)	N2	-	Not used.
Z(53)	N3	-	Not used.
Z(54)	N4	-	Not used.
Z(55)	N5	-	Not used.
Z(56)	N6	-	Not used.
Z(57)	N7	-	Not used.
Z(58)	N8	-	Not used.
Z(59)	N9	-	Not used.
Z(60)	N10	-	= i value of zone that is controlling Δt .
Z(61)	N11	-	= j value of zone that is controlling Δt .
Z(62)	NRM	-	= maximum number of radiation cycles/hydro (input number).

Location	Symbol	Units	Description
Z(63)	TRAD	shake	NR · Δt radiation = Δt̄ hydro; not used in this version.
Z(64)	XNRG	-	Not used.
Z(65)	SN	-	Not used.
Z(66)	DXN	-	Not used.
Z(67)	RADER	g-cm-shake	Total positive radial momentum (X only).
Z(68)	RADET	g-cm-shake	Total positive axial momentum (X only).
Z(69)	RADEB	-	Not used.
Z(70)	DTRAD	-	Not used.
Z(71)	REZFCT	-	If = 0, PH2 will not trigger REZONE.
Z(72)	RSTOP	-	Not used in continuous version.
Z(73)	SHELL	-	Not used.
Z(74)	BBOUND	-	Not used in this version.
Z(75)	TOZONE	g/cm ³	Minimum density for mass flow at free surface.
Z(76)	ECK	energy check	$\left[\left(\frac{ETH - E^n}{ETH} - \frac{ETH - E}{ETH} \right)^{n-NPC} \right] / NPC.$
Z(77)	SBOUND	-	Fraction of Δ in mass weighting velocity EUL PH2 ~ 1.0.
Z(78)	X1	-	Not used.
Z(79)	X2	-	Not used.
Z(80)	Y1	-	Not used.
Z(81)	Y2	-	Not used.
Z(82)	CABLN	-	Already defined.
Z(83)	VISC	-	Not used.
Z(84)	T	shake	Total time up to cycle N, $t^n = t^{n-1} + \Delta t.$
Z(85)	GMAX	-	Maximum of γX or γ.

Location	Symbol	Units	Description
Z(86)	WSGD	-	γ .
Z(87)	WSGX	-	γX and $(\gamma_{MAX} - 1)$ in CDT.
Z(88)	GMADR	-	$\gamma / (\gamma - 1)$.
Z(89)	GMAXR	-	$\gamma X / (\gamma X - 1)$.
Z(90)	S1	-	Not used.
Z(91)	S2	-	Not used.
Z(92)	S3	-	Not used.
Z(93)	S4	-	Not used.
Z(94)	S5	-	Not used.
Z(95)	S6	-	Not used.
Z(96)	S7	-	Not used.
Z(97)	S8	-	Used in TCLAM only.
Z(98)	S9	-	Not used.
Z(99)	S10	-	Not used.
Z(100)		grams	Mass thrown away (PH2) continuous transport.
Z(101)		jerks	Total energy thrown away.
Z(102)		-	Not used.
Z(103)		-	Not used.
Z(104)		jerks	Energy (internal) added to system when internal is set to 0 if $I < 0$.
Z(105)		yes	Fraction of stability at early times.
Z(106)		yes	$(1. + \%)$ increase/cycle until $Z(106) = Z(139)$.
Z(107)		yes	X mass cut off in PH2.
Z(108)		yes	. mass cut off in PH2.
Z(109)		-	Not used.

Location	Symbol	Units	Description
Z(110)		-	Not used.
Z(111)		g/cm^3	Initial density of material.
Z(112)		cm-shake	Initial velocity of pellet.
Z(113)		-	Epsilonics for emptying pellet $\approx .01$.
Z(114)		-	Not used.
Z(115)	ρ_X	}	For equation of state constants. Are the same as in OIL report GAMD-5580.
Z(116)	ρ		
Z(117)	a_X		
Z(118)	a		
Z(119)	E^X		
Z(120)	E^*		
Z(121)	b_X		
Z(122)	b		
Z(123)	A_X		
Z(124)	A		
Z(125)	V_S^X		
Z(126)	V_S^*		
Z(127)	E_S^X		
Z(128)	E_S^*		
Z(129)	α^X		
Z(130)	α^*		
Z(131)	β^X		
Z(132)	β^*		
Z(133)	B^X		
Z(134)	B^*		

Location	Symbol	Units	Description
Z(135)		-	Not used.
Z(136)		-	Not used.
Z(137)		-	Not used.
Z(138)		g/cm ³	Density check if $\rho(K) < Z(138)$ stability check for cell (K) is bypassed.
Z(139)		-	Percent of instability, used in CDT if CABLN < 0 $\approx .5$.
Z(140)		-	Not used.
Z(141)		-	Not used.
Z(142)		-	Not used.
Z(143)		g/cm ³	Minimum (.) density in PH2 $\sim 10^{-3} \rho'_0$.
Z(144)		g/cm ³	Minimum (X) density in PH2 $\sim 10^{-3} \rho'_x$.
Z(145)		jerk/g	ϵ on I in PH1, PH2 $\sim 10^{-9}$.
Z(146)		cm-shake	ϵ on U or V in PH1, PH2 $\sim 10^{-6}$.
Z(147)		-	j (of pellet-target interface) at $t = 0$.
Z(148)	A	10 ⁵ cm-sec	C (speed of sound = $A + BP^\epsilon$ where $A = C_0$ and P is in megabars.
Z(149)	B	-	
Z(150)	ϵ	-	

E. TCLAM AND TOIL LISTINGS

NOTE: THE FOLLOWING SET OF DIMENSION,
COMMON AND EQUIVALENCE ARE TO BE USED FOR ALL SUBROUTINES

*** TCLAM ***

D I M E N S I O N

INPU0020
INPU0030

DIMENSION AIX(5000), AID(5000), AM(130),
1AMX(5000), AMD(5000), DX(100), ENDD(2), ITAB(502),
2IZ(100), RONE(2), TAB(502), TABI(20), TABIY(21), TABR(20),
3TABUV(20), TABX(21), TABY(21), TAU(100), TEMP(13),
4U(5000), V(5000), X(100), XL(130),
5YL(130), Z(150), Y(100), DY(100)

INPU0100

DIMENSION IW1(130), IW2(130)

COMMON Z, X, TAB, Y

COMMON	AID	, AIMAX	, AIX	, AJMAX	, AM	, AMD	INPU0120
COMMON	AMX	, DX	, ENDD	, FMX	, GYN	, GXX	INPU0130
COMMON	GYN	, GYX	, I	, IA	, IB	, IBA	INPU0140
COMMON	IBB	, ID	, IG	, II	, IIC	, IJ	INPU0150
COMMON	IR	, IRC	, IUV	, IUVC	, IWS	, IWSA	INPU0160
COMMON	IWSB	, IX	, IXN	, IXX	, IYN	, IYX	INPU0170
COMMON	J	, JA	, JT	, JTM	, K	, KE	INPU0180
COMMON	KF	, KK	, L	, LA	, LB	, LD	INPU0190
COMMON	LE	, LI	, LX	, M	, MI	, MIJ	INPU0200
COMMON	MJ	, MN	, MNP	, MX	, MXA	, MXS	INPU0210
COMMON	MZ	, NPKG	, NPP	, NT	, NX	, NY	INPU0220
COMMON	Q000FL, RHO	, RONE, SLA	, SLB	, TABI			INPU0230
COMMON	TABIY	, TABR	, TABUV	, TABX	, TABY	, TAM	INPU0240
COMMON	TAU	, TEMP	, TFMX	, TPIDY	, TX	, TY	INPU0250
COMMON	U	, V	, WPIDY	, WS	, WSA	, WSB	INPU0260
COMMON	WSC	, WSD	, WSE	, WSF	, WSG	, WSI	INPU0270
COMMON	WSL	, WSU	, WSV	, WSX	, WSY	, WS5	INPU0280
COMMON	XC	, XL	, YC	, YL	, YMAX	, WSR	INPU0290
COMMON	PE	, PM	, TTX	, TTY	, LF	, E	INPU0300
COMMON	PEE	, NPRR	, NYI	, DY	, NK	, SWITCH	INPU0310
COMMON	IW1	, IW2					INPU0320

E Q U I V A L E N C E

OEQUIVALENCE	(Z, IZ, PROB),	(Z(2), CYCLE),	(Z(3), DT),	INPU0430
1(Z(4), PRINTS),	(Z(5), PRINTL),	(Z(6), DUMPT7),	(Z(7), CSTOP),	INPU0440
2(Z(8), PIDY),	(Z(9), TMZ),	(Z(10), GAM),	(Z(11), GAMD),	INPU0450
3(Z(12), GAMX),	(Z(13), ETH),	(Z(14), FFA),	(Z(15), FFB),	INPU0460
4(Z(16), TMDZ),	(Z(17), TMXZ),	(Z(18), XMAX),	(Z(19), TXMAX),	INPU0470
5(Z(20), TYMAX),	(Z(21), AMDM),	(Z(22), AMXM),	(Z(23), DNN),	INPU0480
6(Z(24), DMIN),	(Z(25), FEF),	(Z(26), DTNA),	(Z(27), CVIS),	INPU0490
7(Z(28), NPR),	(Z(29), NPRI),	(Z(30), NC),	(Z(31), NPC),	INPU0500
8(Z(32), NRC),	(Z(33), IMAX),	(Z(34), IMAXA),	(Z(35), JMAX),	INPU0510
9(Z(36), JMAXA),	(Z(37), KMAX),	(Z(38), KMAXA),	(Z(39), NMAX),	INPU0520
OEQUIVALENCE	(Z(40), ND),	(Z(41), KDT),	(Z(42), IXMAX),	INPU0530
1(Z(43), NOD),	(Z(44), NOPR),	(Z(45), NIMAX),	(Z(46), NJMAX),	INPU0540
2(Z(47), I1),	(Z(48), I2),	(Z(49), I3),	(Z(50), I4),	INPU0550
3(Z(51), N1),	(Z(52), N2),	(Z(53), N3),	(Z(54), N4),	INPU0560
4(Z(55), N5),	(Z(56), N6),	(Z(57), N7),	(Z(58), N8),	INPU0570
5(Z(59), N9),	(Z(60), N10),	(Z(61), N11),	(Z(62), NRM),	INPU0580
6(Z(63), TRAD),	(Z(64), XNRG),	(Z(65), SN),	(Z(66), DXH),	INPU0590
7(Z(67), RADER),	(Z(68), RADET),	(Z(69), RADEB),	(Z(70), DTRAD),	INPU0600
8(Z(71), REZFCT),	(Z(72), RSTOP),	(Z(73), SHELL),	(Z(74), BBOUND),	INPU0610
9(Z(75), TOZONE),	(Z(76), ECK),	(Z(77), SBOUND),	(Z(78), X1),	INPU0620
OEQUIVALENCE	(Z(79), X2),	(Z(80), Y1),	(Z(81), Y2),	INPU0630

```

1(Z(82),CABLN), (Z(83),VISC), (Z(84),T), (Z(85),GMAX), INPU0640
2(Z(86),WSGD), (Z(87),WSGX), (Z(88),GMADR), (Z(89),GMAXR), INPU0650
3(Z(90),S1), (Z(91),S2), (Z(92),S3), (Z(93),S4), INPU0660
4(Z(94),S5), (Z(95),S6), (Z(96),S7), (Z(97),S8), INPU0670
5(Z(98),S9), (Z(99),S10) INPU0680
EQUIVALENCE (Z,IZ),(TAB,ITAB)
DIMENSION PLOT(10)
DATA PLOT/3H X ,3HDOT,3HGEN,3HDEL/

CMAIN
C L A M ***** M A I N *****
C
C
C CALL SLITE (0)
C INPUT ROUTINE CALCULATES THE ACTUAL GRID,
C DIMENSIONS AND INDICES.
10 CALL INPUT
C PH1, READS IN DATA CARDS FOR THE
C PACKAGES, PH2 CALCULATES THE GEOMETRICS,
C PH3 THE PARTICLES, PH4 CALLS THE
C 6 POSSIBLE FITS THAT CALCULATE THE
C DENSITY, VELOCITIES AND INTERNAL ENERGY
C OF THE PARTICLES.
20 CALL PH1
C OUTPUT CALCULATES THE VELOCITY (BOTH
C RADIAL AND AXIAL) AND SPECIFIC INTERNAL
C ENERGY OF EACH CELL FROM THE
C TOTAL MOMENTA AND INTERNAL
C ENERGY AND MASS OF EACH CELL.
C OUTPUT ALSO PREPARES A DUMP TAPE
C WHICH IS USED THEN TO START TOIL
30 CALL OUTPUT
CALL EXIT
END
SUBROUTINE INPUT
C
C
C ***** A 2 MATERIAL CLAM FOR THE TOIL CODE *****
C
C
C
C
C MZ=150
C CLEAR Z BLOCK.
C DO 30 I=1,MZ
30 Z(I)=0.0
C READ IN HEADING CARD
READ (5,8012)IWS
IWS=1
WRITE (6,8012)(IWS)
WRITE (6,8100)
C READ IN PROBLEM CONSTANTS
C PROB=PROBLEM NO. AIMAX=IMAX,
C AJMAX=JMAX, Q00UFL IS NOT USED-SET
C TO ZERO, SHELL SET=2.,S8,S9 ARE
C ZERO, SET N7 TO=TAPE NO.
READ (5,8004)PROB,AIMAX,AJMAX,Q00UFL,SHELL,S8,S9,N7
IF(N7)40,40,50
40 N7=9

```

MAIN0010

MAIN0020

MAIN0030

MAIN0050

MAIN0060

MAIN0070

MAIN0080

MAIN0090

MAIN0100

MAIN0110

INPU0010

INPU0710

INPU0730

INPU0940

INPU0960

INPU0970

INPU0980

INPU0990

INPU1000

INPU1010

INPU1020

INPU1030

INPU1040

INPU1050

INPU1060

50	CONTINUE	INPU1080
C	MAX. NUMBER OF ZONES IN R DIRECTION. MI=100	
C	MAX. NUMBER OF ZONES IN Z DIRECTION. MJ=100	INPU1100
C	MAX. NUMBER OF PARTICLES/CELL. MNP=400	INPU1110
C	SIZE OF TABLE (TAB)	INPU1120
	JTM=500	INPU1130
C	MAXIMUM I*J	INPU1140
C	MAX. NUMBER OF CELLS.	
60	MIJ=4999	
C	CALCULATE ADDITIONAL INDICES FOR TCLAM	
C	AND TOIL.	INPU1160
70	IMAX=AIMAX	INPU1170
	JMAX=AJMAX	INPU1180
	IMAXA=IMAX+1	INPU1190
	IXMAX=IMAXA+1	INPU1200
	JMAXA=JMAX+1	INPU1210
	KMAX=(IMAX*JMAX)+1	INPU1220
	KMAXA=KMAX+1	INPU1230
	WRITE (6,8048)(PRIB,IMAX,JMAX)	
C	CHECK INPUT NOS. CONCERNED WITH GRID SIZE.	INPU1240
101	IF(IMAX-MI)102,103,9901	INPU1250
102	IF(JMAX-MJ)104,105,9902	INPU1260
104	IF(KMAX-MIJ-1)106,106,9903	INPU1270
106	NOD=1	INPU1280
	NPC=1	INPU1290
	NRC=0	INPU1300
C	READ IN DY AND DX	INPU1310
	I=0	INPU1320
	J=0	INPU1330
	X(I)=0.0	INPU1340
	Y(J)=0.0	INPU1350
2000	READ (5,8102)IWS,IWSB,N1,N2,N3,N4,(TEMP(K),K=1,4)	INPU1360
	L=1	
C	COUNT NO. OF DIFFERENT DX OR DY.	INPU1370
	IF(N4)2003,2001,2003	INPU1380
2001	IF(N3)2004,2002,2004	INPU1390
2002	IF(N2)2006,2008,2006	INPU1400
2003	L=L+1	INPU1410
2004	L=L+1	INPU1420
2006	L=L+1	INPU1430
2008	IF(IWSB)2010,2010,2030	
C	PROCESS THE DX AND DY VALUES.	INPU1440
2010	DO 2014 N=1,L	INPU1450
	NK=IZ(N+50)	INPU1460
	DO 2012 K=1,NK	INPU1470
	I=I+1	INPU1480
	DX(I)=TEMP(N)	INPU1490
	X(I)=X(I-1)+DX(I)	INPU1500
2012	CONTINUE	INPU1510
2014	CONTINUE	INPU1520
	GO TO 2050	INPU1530
C	CALC THE Y AND DY VALUES	INPU1540
2030	DO 2034 N=1,L	INPU1550
	NK=IZ(N+50)	

DO 2032 K=1,NK	INPU1560
J=J+1	INPU1570
DY(J)=TEMP(N)	INPU1580
Y(J)=Y(J-1)+DY(J)	INPU1590
2032 CONTINUE	INPU1600
2034 CONTINUE	INPU1610
2050 IF(IWSA)2052,2000,2052	INPU1620
C IF(=) READ MORE DX OR DY DATA CARDS.	
2052 IF(J-JMAX)9905,2053,9905	INPU1630
C CHECK INPUT NUMBERS.	
2053 IF(I-IMAX)9906,2054,9906	INPU1640
2054 CONTINUE	INPU1650
READ (5,8004)WS,WSA,WSB,SWITCH	INPU1660
C N4=MAX. NUMBER OF PARTICLES-1 PER RECORD.	
N4=WSB	INPU1710
NPRI=N4	INPU1720
NPRR=N4	INPU1730
WRITE (6,8064)IMAX,(X(I),I=0,IMAX)	INPU1740
WRITE (6,8065)JMAX,(Y(J),J=0,JMAX)	INPU1750
WS=3.1415927	INPU1760
WSA=0.0	INPU1770
C CALCULATE THE AREA-S(TAU)=PI(R(I)**2-	
C R(I-1)**2).	
DO 1008 I=1,IMAX	INPU1780
WSB=WSA	INPU1790
WSA=X(I)**2	INPU1800
1008 TAU(I)=WS*(WSA-WSB)	INPU1810
C WRITE OUT X,Y,DX,DY, AND TAU VALUES.	
WRITE (6,8066)IMAX,(DX(I),I=1,IMAX)	INPU1820
WRITE (6,8067)JMAX,(DY(I),I=1,JMAX)	INPU1830
WRITE (6,8092)(IMAX,(TAU(I),I=1,IMAX)).	INPU1840
1010 XMAX=X(IMAX)	INPU1850
TXMAX=XMAX*2.0	INPU1860
YMAX=Y(JMAX)	INPU1870
TYMAX=YMAX*2.0	INPU1880
C PIDY IS REALLY PI(3.1415927).	
PIDY=WS	INPU1890
C SET VELOCITIES, INTERNAL ENERGIES AND MASSES	
C TO 0.	
DO 1014 I=1,KMAXA	INPU1910
U(I)=0.0	INPU1920
V(I)=0.0	INPU1930
AIX(I)=0.0	INPU1940
AMX(I)=0.0	INPU1950
AID(I)=0.	
AMD(I)=0.	
1014 CONTINUE	INPU1960
C SET TOTAL ENERGY TO ZERO.	
ETH=0.0	INPU1970
C INITIALIZE MIN. MASS PARTICLE TO A LARGE NO.	
AMDM=1.E+28	INPU1980
AMXM=AMDM	INPU1990
GO TO 2016	INPU2000
C ERROR	INPU2010
C YC H' C IMAX GREATER THAN 100	
9901 NK=101	INPU2020
GO TO 9999	INPU2030

C	YOU HAVE JMAX GREATER THAN 100	
9902	NK=102	INPU2040
	GO TO 9999	INPU2050
C	YOU HAVE TRIED TO GENERATE MORE THAN	
C	4999 CELLS.	
9903	NK=104	INPU2060
	GO TO 9999	INPU2070
C	JMAX DOES NOT EQUAL THE SUM OF THE INPUT J	INPU2080
9905	NK=2052	INPU2090
	GO TO 9999	INPU2100
C	IMAX DOES NOT EQUAL THE SUM OF THE INPUT I	INPU2110
9906	NK=2053	INPU2120
9999	WRITE (6,8888)NK,I,J,K,L,M,N	INPU2130
	PRINT 8888,NK,I,J,K,L,M,N	INPU2140
	CALL DUMP	INPU2150
2016	RETURN	INPU2160
C	FORMATS	INPU2170
8004	FORMAT(7E10.5,I2)	INPU2180
80120	FORMAT (I1,71HTHIS IS THE CLAM PROGRAM AND THERE IS AN ERROR.	INPU2190
1	)	INPU2200
8048	FORMAT(1H /9H PROB NO.F9.3,12X,2HI=I2,26X,2HJ=I2)	INPU2210
8064	FORMAT(1H /10H X(I) I=0,I2/(5F16.6))	INPU2220
8065	FORMAT(1H /10H Y(J) J=0,I2/(5F16.6))	INPU2230
8066	FORMAT(1H /11H DX(I) I=1,I2/(5F16.6))	INPU2240
8067	FORMAT(1H /11H DY(J) J=1,I2/(5F16.6))	INPU2250
8092	FORMAT(1H /13H AREA(I) I=1,I2/(5F16.6))	INPU2260
8100	FORMAT(1H /14H (TOIL INPUT))	
8102	FORMAT(2I1,4I2,4E10.4)	INPU2280
8888	FORMAT(1H+/26H1 INPUT ERROR IN STATEMENT15,12X,12H INDICES ARE6I7)	INPU2290
	END	INPU2300
	SUBROUTINE PH1	PH1 0010
C		INPU0710
C		
C	***** A 2 MATERIAL CLAM FOR THE TOIL CODE *****	
C		
C	READ IN GEOMETRY ETC.	PH1 0960
	NPP=7	PH1 0980
	NPR=NPP-1	PH1 0990
	TPIDY=PIDY*2.0	PH1 1000
	ND=0	PH1 1010
	NX=0	PH1 1020
	NT=1	PH1 1030
	NY=1	PH1 1040
C	FIRST CARD OF EACH PACKAGE.	
	READ (5,8008)IX,LX,MX,TEMP(1),TEMP(2),TEMP(3)	PH1 1050
C	INITIALIZE THE NUMBER OF PACKAGES TO 0.	
	NPKG=0	PH1 1060
2015	IF(IX-1)9901,2018,2018	PH1 1070
2016	IX=I	PH1 1080
	LX=L	PH1 1090
	MX=M	PH1 1100
C	IF THERE ARE NO MORE PACKAGES GO COMPUTE TOTAL VALUES	PH1 1110
C	THE LAST CARD HAS A 2 PUNCH IN COL 1.	
2017	IF(IX-2)2018,7000,9902	PH1 1120
2018	J=0	PH1 1130
	NPKG=NPKG+1	PH1 1140

C	SET PACKAGE MASS AND ENERGY TO 0.	
	PE=0.0	PH1 1150
	PM=0.0	PH1 1160
C	ORIGIN FOR THE RADIUS VECTORS TO BE USED	
C	FOR THE FIT ROUTINES(1 THRU 6).	
	YC=TEMP(1)	PH1 1170
	XC=TEMP(2)	PH1 1180
C	S8 CONTAINS THE FIT NUMBER FOR THE	
C	PACKAGE IN QUESTION.	
	S8=TEMP(3)	PH1 1190
	WRITE (6,8100)(NPKG,MX)	PH1 1200
C	NOW READ IN THE GEOMETRY AND DENSITY,	
C	ENERGY AND VELOCITY CARDS.	
	2020 READ (5,8008)I,L,M,(TEMP(N),N=1,6)	PH1 1210
	IWS=1	PH1 1220
	IF(I-5)2021,2040,2022	PH1 1230
C	IF=, THIS IS A RHO, VELOCITY OR ENERGY CARD.	
C	IF LESS, YOU HAVE READ ALL CARDS FOR THIS	
C	PACKAGE IN, PLUS THE FIRST CARD FROM THE	
C	NEXT PACKAGE.	
	2021 IF(I-3)2060,9903,2026	PH1 1240.
C	IF GREATER, EITHER A TRIANGLE OR PERTURBED ELLIPSE.	
	2022 IF(L)9904,2030,2024	PH1 1250
C	A PERTURBED ELLIPSE.	
	2024 IWS=7	PH1 1260
	GO TO 2030	PH1 1270
	2026 IWS=3	PH1 1280
	2027 IF(L)9905,2030,2028	PH1 1290
	2028 IWS=5	PH1 1300
C	A TRIANGLE.	
	2030 IF(M)9906,2034,2032	PH1 1310
C	IF=, DELETE THIS GEOMETRY.	
	2032 IWS=IWS+1	PH1 1320
	2034 J=J+1	PH1 1330
C	TAB STORAGE CONTAINS THE COORDINATES OF	
C	GEOMETRY.	
	ITAB(J)=IWS	PH1 1340
	DO 2036 N=1,NPR	PH1 1350
	J=J+1	PH1 1360
	2036 TAB(J)=TEMP(N)	PH1 1370
	GO TO 2020	PH1 1380
C	ONE ONLY RHO,I,U OR V ALLOWED PER PACKAGE	PH1 1390
C	IF=, THIS IS A DENSITY CARD.	
	2040 IF(L-1)9907,2046,2042	PH1 1400
C	IF GREATER, EITHER A VELOCITY OR ENERGY CARD.	
	2042 IF(L-3)2052,2058,9908	PH1 1410
C	IF=, THIS IS A VELOCITY CARD, IF LESS, THIS IS A	
C	ENERGY CARD.	
C	DENSITY	PH1 1420
	2046 DO 2048 N=1,6	PH1 1430
	2048 TABR(N)=TEMP(N)	PH1 1440
	GO TO 2020	PH1 1450
C	ENERGY	PH1 1460
	2052 DO 2054 N=1,6	PH1 1470
	2054 TABI(N)=TEMP(N)	PH1 1480
	GO TO 2020	PH1 1490
C	VELOCITY (U AND V)	PH1 1500

2058	DO 2059 N=1,6	PH1 1510
2059	TABUV(N)=TEMP(N)	PH1 1520
	GO TO 2020	PH1 1530
C	OUTPUT DENSITY, ENERGY, AND VELOCITY PARAMETERS	PH1 1540
C	ALL CARDS FOR THIS PACKAGE HAVE	
C	BEEN READ IN.	
2060	IF(J-JTM)2070,2070,9915	PH1 1550
2070	WRITE (6,8036)(TABR(II),II=1,6)	PH1 1560
	WRITE (6,8038)(TABI(II),II=1,6)	PH1 1570
	WRITE (6,8040)(TABUV(II),II=1,6)	PH1 1580
C	COMPUTE BOUNDARIES OF GEOMETRIES FOR EFFICIENCY IN	PH1 1590
C	GENERATING OR DELETING PARTICLES	PH1 1600
3000	CALL PH2	PH1 1610
C	COMPUTE I(0),I(N),J(0) AND J(N),FROM PREVIOUSLY	PH1 1620
C	COMPUTED VALUES,FOR UPPER AND LOWER LIMITS IN	PH1 1630
C	THE CELL MESH SCAN	PH1 1640
C	IXN=MINIMUM (I) OF GEOMETRY OF PACKAGE	
C	IYN=MINIMUM (J) OF GEOMETRY OF PACKAGE	
C	IXX=MAXIMUM (I) OF GEOMETRY OF PACKAGE	
C	IYX=MAXIMUM (J) OF GEOMETRY OF PACKAGE	
3001	IXN=1	PH1 1650
	IXX=1	PH1 1660
	IWS=IMAX-1	PH1 1670
3800	IF(IWS)9929,3820,3801	PH1 1680
3801	DO 3808 N=1,IWS	PH1 1690
	IF(X(N)-GXN)3802,3804,3804	PH1 1700
3802	IXN=IXN+1	PH1 1710
3804	IF(X(N)-GXX)3806,3806,3808	PH1 1720
3806	IXX=IXX+1	PH1 1730
3808	CONTINUE	PH1 1740
	IF(IXN)3812,3812,3814	PH1 1750
3812	IXN=1	PH1 1760
3814	IF(IMAX-IXX)3816,3818,3818	PH1 1770
3816	IXX=IMAX	PH1 1780
3818	IF(IXN-IXX)3820,3820,9930	PH1 1790
3820	IYN=1	PH1 1800
	IYX=1	PH1 1810
	IWS=JMAX-1	PH1 1820
3821	IF(IWS)9929,3834,3822	PH1 1830
3822	DO 3813 N=1,IWS	PH1 1840
3823	IF(Y(N)-GYN)3819,3817,3817	PH1 1850
3819	IYN=IYN+1	PH1 1860
3817	IF(Y(N)-GYX)3815,3815,3813	PH1 1870
3815	IYX=IYX+1	PH1 1880
3813	CONTINUE	PH1 1890
	IF(IYN)3824,3824,3826	PH1 1900
3824	IYN=1	PH1 1910
3826	IF(JMAX-IYX)3828,3830,3830	PH1 1920
3828	IYX=JMAX	PH1 1930
3830	IF(IYN-IYX)3834,3834,9931	PH1 1940
3834	WRITE (6,8044)IXN,IYN,IXX,IYX	PH1 1950
C	SCAN CELL MESH TO DETERMINE IF PARTICLES ARE TO BE	PH1 1960
C	GENERATED OR DELETED	PH1 1970
C	GENERATE PARTICLES	PH1 1980
4000	CALL PH3	PH1 1990
C	REARRANGE X,Y AND M FOR PARTICLES IF NECESSARY	PH1 2000
6011	LA=NY-NT	PH1 2010

IF(LX)9947,6020,6022	PH1 2020
6020 ND=ND+LA	PH1 2030
GO TO 6024	PH1 2040
6022 NX=NX+LA	PH1 2050
6024 NT=NY	PH1 2060
ETH=ETH+PE	PH1 2070
WS=PLOT(1)	
6026 IF(LX)9933,6028,6030	PH1 2100
6028 WS=PLOT(2)	
6030 WRITE (6,8501)LA,WS,PE,PM	PH1 2130
C GO READ IN NEXT PACKAGE	PH1 2140
6050 GO TO 2016	PH1 2150
7000 NMAX=NT	PH1 2160
C NMAX=MAX. NUMBER OF PARTICLES+1.	
C YOU HAVE PROCESSED ALL PACKAGES, ALL	
C PARTICLES, NOW GO TO THE OUTPUT.	
IF(AM(2))4051,4050,4051	PH1 2170
4050 N3=NRC	PH1 2180
GO TO 4060	PH1 2190
4051 NRC=NRC+1	PH1 2200
N3=NRC	PH1 2210
C N3=NO. OF PARTICLE RECORDS OF	
C N4 WORDS.	
4060 N6=NMAX-(N4-1)*(N3-1)	PH1 2240
NOPR=N3	PH1 2250
GO TO 10000	PH1 2270
C ERROR	PH1 2280
9901 NK=2015	PH1 2290
GO TO 9999	PH1 2300
9902 NK=2017	PH1 2310
GO TO 9999	PH1 2320
9903 NK=2021	PH1 2330
GO TO 9999	PH1 2340
9904 NK=2022	PH1 2350
GO TO 9999	PH1 2360
9905 NK=2027	PH1 2370
GO TO 9999	PH1 2380
9906 NK=2030	PH1 2390
GO TO 9999	PH1 2400
9907 NK=2040	PH1 2410
GO TO 9999	PH1 2420
9908 NK=2042	PH1 2430
GO TO 9999	PH1 2440
9915 NK=2060	PH1 2450
GO TO 9999	PH1 2460
9929 NK=3800	PH1 2470
GO TO 9999	PH1 2480
9930 NK=3818	PH1 2490
GO TO 9999	PH1 2500
9931 NK=3830	PH1 2510
GO TO 9999	PH1 2520
9933 NK=6026	PH1 2530
GO TO 9999	PH1 2540
9947 NK=6011	PH1 2550
9999 WRITE (6,8888)NK	PH1 2560
PRINT 8888,NK	PH1 2570
CALL DUMP	PH1 2580

```

10000 RETURN
C          FORMATS
8008 FORMAT (2I1,I5,E13.5,5E10.5)
8036 FORMAT(1H07X,8HDENSITY 9X,1P6E16.6)
8038 FORMAT(1H07X,8HENERGY 9X,1P6E16.6)
8040 FORMAT(1H07X,8HVELOCITY9X,1P6E16.6/1H0/)
8044 FORMAT(1H /6H I(1)=I2,4X,5HJ(1)=I2,4X,5HI(N)=I2,4X,5HJ(N)=I2)
81000FORMAT(1H0///12H0PACKAGE NO.I3,I20,15H PARTICLES/CELL//33X,2HA114X
1,2HA214X,2HA314X,2HA414X,2HA514X,2HA6)
85010FORMAT(1H0/I28,2H (A3,11H) PARTICLES22X,4HPE =1PE12.6,16X,4HPM =E1
12.6)
8888 FORMAT(23H1PH1 ERROR IN STATEMENTI5)
END
SUBROUTINE PH2

C
C ***** A 2 MATERIAL CLAM FOR THE TOIL CODE *****
C
CALCULATE THE PACKAGE GEOMETRIES
C
C
C
C          GENERATING OR DELETING PARTICLES
C
C J=VALUE OF LAST COORDINATE READ IN.
C JT=J
C INITIALIZE OUTER BOUNDARIES.
C   GXN=XMAX
C   GYN=YMAX
C   GXX=0.0
C   GYX=0.0
C   NPP=7(SET IN PH1).
C   DO 3700 J=1,JT,NPP
C     IWS STORED IN ITAB ARRAY IN PH1.
C     IF IWS=2(A TRIANGLE),IF=4(A RECTANGLE),
C     IF=6,A ELLIPSE OR CIRCLE. IF IWS=8,A
C     PERTURBED ELLIPSE. IF IWS IS LESS THAN
C     THESE VALUES, THE DEFINITION STILL HOLDS, BUT
C     NOW DELETE THIS GEOMETRY.
C     KK=(ITAB(J)-1)/2
3007 IF(KK)9919,3010,3008
3008 IF(KK-2)3100,3200,3009
3009 IF(KK-4)3400,9920,9920
C           TRIANGLE
C     VERTICES CAN BE INPUTED IN ANY ORDER,
C     X COORDINATE FIRST.
C     SEARCH FOR THE LARGEST X(WSE) AND
C     SMALLEST X(WSD).
C     FIND MAXIMUM(WSE) AND MINIMUM(WSD) X COORDINATE
3010 IF(TAB(J+1)-TAB(J+3))3011,3012,3013
3011 WSE=TAB(J+3)
3012 WSD=TAB(J+1)
3013 GO TO 3014
3012 TAB(J+1)=TAB(J+1)*1.0000001+1.0E-8
3013 WSE=TAB(J+1)
3014 WSD=TAB(J+3)
3014 IF(TAB(J+5)-WSD)3020,3019,3016
3016 IF(TAB(J+5)-WSE)3024,3017,3018
3017 TAB(J+5)=TAB(J+5)*1.0000001+1.0E-8

```

```

PH1 2590
PH1 2600
PH1 2610
PH1 2620
PH1 2630
PH1 2640
PH1 2650
PH1 2660
PH1 2670
PH1 2680
PH1 2690
PH1 2700
PH1 2710
PH2 0010
PH2 0020
PH2 0740
PH2 0950
PH2 0960
PH2 0970
PH2 0980
PH2 0990
PH2 1000
PH2 1010
PH2 1020
PH2 1030
PH2 1040
PH2 1050
PH2 1060
PH2 1070
PH2 1080
PH2 1090
PH2 1100
PH2 1110
PH2 1120
PH2 1130
PH2 1140
PH2 1150
PH2 1160
PH2 1170
PH2 1180
PH2 1190

```

3018	WSE=TAB(J+5)	PH2 1200
	GO TO 3024	PH2 1210
3019	TAB(J+5)=TAB(J+5)*0.9999999-1.0E-8	PH2 1220
3020	WSD=TAB(J+5)	PH2 1230
C	ARRANGE VERTICES IN ASCENDING ORDER	PH2 1240
3024	IF(TAB(J+2)-TAB(J+4))3036,3034,3038	PH2 1250
3034	TAB(J+2)=TAB(J+2)*1.0000001+1.0E-8	PH2 1260
	GO TO 3038	PH2 1270
3036	WSA=TAB(J+1)	PH2 1280
	WSB=TAB(J+2)	PH2 1290
	TAB(J+1)=TAB(J+3)	PH2 1300
	TAB(J+2)=TAB(J+4)	PH2 1310
	TAB(J+3)=WSA	PH2 1320
	TAB(J+4)=WSB	PH2 1330
3038	IF(TAB(J+4)-TAB(J+6))3042,3040,3044	PH2 1340
3040	TAB(J+6)=TAB(J+6)*0.9999999-1.0E-8	PH2 1350
	GO TO 3044	PH2 1360
3042	WSA=TAB(J+3)	PH2 1370
	WSB=TAB(J+4)	PH2 1380
	TAB(J+3)=TAB(J+5)	PH2 1390
	TAB(J+4)=TAB(J+6)	PH2 1400
	TAB(J+5)=WSA	PH2 1410
	TAB(J+6)=WSB	PH2 1420
	GO TO 3024	PH2 1430
C	WSF=MINIMUM VALUE OF Y	
C	WSG=MAXIMUM VALUE OF Y	
3044	WSF=TAB(J+6)	PH2 1440
	WSG=TAB(J+2)	PH2 1450
C	COMPUTE SLOPES	PH2 1460
	SLA=(TAB(J+4)-TAB(J+2))/(TAB(J+3)-TAB(J+1))	PH2 1470
	SLB=(TAB(J+6)-TAB(J+2))/(TAB(J+5)-TAB(J+1))	PH2 1480
3053	IF(SLA-SLB)3054,9921,3058	PH2 1490
3054	IF(SLA)3056,9922,3064	PH2 1500
3056	IF(SLB)3064,9923,3062	PH2 1510
3058	IF(SLA)3062,9924,3056	PH2 1520
3062	WSA=TAB(J+3)	PH2 1530
	WSB=TAB(J+4)	PH2 1540
	WSC=SLA	PH2 1550
	TAB(J+3)=TAB(J+5)	PH2 1560
	TAB(J+4)=TAB(J+6)	PH2 1570
	SLA=SLB	PH2 1580
	TAB(J+5)=WSA	PH2 1590
	TAB(J+6)=WSB	PH2 1600
	SLB=WSC	PH2 1610
3064	IF(TAB(J+3)-TAB(J+5))3066,9925,3068	PH2 1620
3066	ITAB(J)=ITAB(J)+2	PH2 1630
	IWS=ITAB(J)-3	PH2 1640
	GO TO 3069	PH2 1650
3068	IWS=ITAB(J)-1	PH2 1660
3069	KE=J+1	PH2 1670
	KF=KE+5	PH2 1680
	WS=PLOT(3)	
	IF(IWS)3072,3070,3072	PH2 1710
3070	WS=PLOT(4)	
3072	WRITE (6,8016)WS,(TAB(N),N=KE,KF)	PH2 1740
	WS=TAB(J+2)-SLB*TAB(J+1)	PH2 1750
	TAB(J+1)=TAB(J+2)-SLA*TAB(J+1)	PH2 1760

TAB(J+6)=(TAB(J+6)-TAB(J+4))/(TAB(J+5)-TAB(J+3))	PH2 1770
TAB(J+5)=TAB(J+4)-TAB(J+6)*TAB(J+3)	PH2 1780
TAB(J+2)=SLA	PH2 1790
TAB(J+3)=WS	PH2 1800
TAB(J+4)=SLB	PH2 1810
GO TO 3600	PH2 1820
C RECTANGLE	PH2 1830
3100 ITAB(J)=ITAB(J)+2	PH2 1840
IWS=ITAB(J)-5	PH2 1850
WS=PLOT(3)	
IF(IWS)3110,3105,3110	PH2 1880
3105 WS=PLOT(4)	
3110 WRITE (6,8020)WS,TAB(J+1),TAB(J+2),TAB(J+3),TAB(J+4)	PH2 1910
WSD=TAB(J+1)	PH2 1920
WSE=TAB(J+2)	PH2 1930
WSF=TAB(J+3)	PH2 1940
WSG=TAB(J+4)	PH2 1950
GO TO 3600	PH2 1960
C ELLIPSE OR CIRCLE	PH2 1970
3200 IF(ABS(TAB(J+1)-TAB(J+2))-1.0E-8)3300,3300,3202	PH2 1980
3202 IF(TAB(J+2))9926,3300,3203	PH2 1990
C ELLIPSE WITH NO PERTURBATION	PH2 2000
3203 ITAB(J)=ITAB(J)+2	PH2 2010
IWS=ITAB(J)-7	PH2 2020
WS=PLOT(3)	
IF(IWS)3210,3205,3210	PH2 2050
3205 WS=PLOT(4)	
3210 WRITE (6,8024)WS,TAB(J+1),TAB(J+2),TAB(J+3),TAB(J+4)	PH2 2080
3215 WSD=TAB(J+3)-TAB(J+1)	PH2 2090
WSE=TAB(J+3)+TAB(J+1)	PH2 2100
WSF=TAB(J+4)-TAB(J+2)	PH2 2110
WSG=TAB(J+4)+TAB(J+2)	PH2 2120
TAB(J+1)=TAB(J+1)**2	PH2 2130
TAB(J+2)=TAB(J+2)**2	PH2 2140
GO TO 3600	PH2 2150
C CIRCLE	PH2 2160
3300 ITAB(J)=ITAB(J)+4	PH2 2170
IWS=ITAB(J)-9	PH2 2180
TAB(J+2)=TAB(J+1)	PH2 2190
WS=PLOT(3)	
IF(IWS)3310,3305,3310	PH2 2220
3305 WS=PLOT(4)	
3310 WRITE (6,8028)WS,TAB(J+1),TAB(J+3),TAB(J+4)	PH2 2250
GO TO 3215	PH2 2260
C ELLIPSE WITH PERTURBATION	PH2 2270
3400 ITAB(J)=ITAB(J)+4	PH2 2280
WS=1.0-(TAB(J+5)/TAB(J+1))**2	PH2 2290
IWSA=ITAB(J+7)	PH2 2300
OTAB(J+7)=(TAB(J+6)-TAB(J+4)-TAB(J+2)*SQRT(WS))/	PH2 2310
1 ((TAB(J+5)*(TAB(J+5)-TAB(J+1)))**2)	PH2 2320
IWS=ITAB(J)-11	PH2 2330
KE=J+1	PH2 2340
KF=KE+6	PH2 2350
WSA=PLOT(3)	
IF(IWS)3410,3405,3410	PH2 2380
3405 WSA=PLOT(4)	
3410 WRITE (6,8032)WSA,(TAB(N),N=KE,KF)	PH2 2410

3415 IF(WS)9927,9927,3420	PH2 2420
3420 IF(TAB(J+3))9928,3425,9928	PH2 2430
3425 TAB(J+3)=TAB(J+7)	PH2 2440
ITAB(J+7)=IWSA	PH2 2450
WSA=TAB(J+2)+TAB(J+2)/4.0	PH2 2460
WSD=0.0	PH2 2470
WSE=TAB(J+1)+TAB(J+1)/4.0	PH2 2480
WSF=TAB(J+4)-WSA	PH2 2490
WSG=TAB(J+4)+WSA	PH2 2500
C DETERMINE BOUNDARIES OF GEOMETRIES	PH2 2510
3600 IF(WSD-GXN)3602,3604,3604	PH2 2520
C MAXIMUM (X)	
3602 GXN=WSD	PH2 2530
3604 IF(WSE-GXX)3608,3608,3606	PH2 2540
C MINIMUM (X)	
3606 GXX=WSE	PH2 2550
3608 IF(WSF-GYN)3610,3612,3612	PH2 2560
C MAXIMUM (Y)	
3610 GYN=WSF	PH2 2570
3612 IF(WSG-GYX)3700,3700,3614	PH2 2580
C MINIMUM (Y)	
3614 GYX=WSG	PH2 2590
3700 CONTINUE	PH2 2600
J=JT	PH2 2610
GO TO 10000	PH2 2620
C E R R O R	PH2 2630
9919 NK=3007	PH2 2640
GO TO 9999	PH2 2650
9920 NK=3009	PH2 2660
GO TO 9999	PH2 2670
9921 NK=3053	PH2 2680
GO TO 9999	PH2 2690
9922 NK=3054	PH2 2700
GO TO 9999	PH2 2710
9923 NK=3056	PH2 2720
GO TO 9999	PH2 2730
9924 NK=3058	PH2 2740
GO TO 9999	PH2 2750
9925 NK=3064	PH2 2760
GO TO 9999	PH2 2770
9926 NK=3202	PH2 2780
GO TO 9999	PH2 2790
9927 NK=3415	PH2 2800
GO TO 9999	PH2 2810
9928 NK=3420	PH2 2820
9999 WRITE (6,8888)NK	PH2 2830
PRINT 8888,NK	PH2 2840
CALL DUMP	PH2 2850
10000 RETURN	PH2 2860
8016 FORMAT(15H0TRIANGLE ---- A3,7H -----1P6E16.6)	PH2 2870
8020 FORMAT(15H0RECTANGLE --- A3,7H -----1P6E16.6)	PH2 2880
8024 FORMAT(15H0ELLIPSE ----- A3,7H -----1P6E16.6)	PH2 2890
8028 FORMAT(15H0CIRCLE ----- A3,7H -----1PE16.6,16X,4E16.6)	PH2 2900
8032 FORMAT(15H0P ELLIPSE --- A3,7H -----1P6E16.6)	PH2 2910
8888 FORMAT(23H1PH2 ERROR IN STATEMENT15)	PH2 2920
END	PH2 2930
SUBROUTINE PH3	PH3 0010

INPU0710

***** A 2 MATERIAL CLAM FOR THE TOIL CODE *****

GENERATE (OR DELETE) THE PARTICLES

INPU0030

PH3 0020

PH3 0740

PH3 0950

PH3 0960

PH3 0970

PH3 0980

PH3 0990

PH3 1000

SCAN CELL MESH TO DETERMINE IF PARTICLES ARE TO BE
GENERATED OR DELETED

GENERATE PARTICLES

SAVE CURRENT VALUES OF COUNTERS.

4000 IA=I

PH3 1010

JA=J

PH3 1020

IJ=K

PH3 1030

JT=L

PH3 1040

IF(IX-1)9932,4010,9932

PH3 1050

4010 IF(MX-MNP)4012,4012,9935

PH3 1060

IF GREATER, YOU TRIED TO GENERATE MORE THAN

400 PARTICLES / CELL.

4012 WS=MX

PH3 1070

FMX=SQRT(WS)

PH3 1080

MXS=FMX+.5

PH3 1090

4011 IF(MXS*MXS-MX)9933,4013,9936

PH3 1100

IF(GREATER OR LESS) THE NO. OF PARTICLES / CELL

THAT YOU REQUESTED WAS NOT N SQ. WHERE

N IS FROM 1 TO 20.

4013 MXA=1-MX

PH3 1110

TFMX=.5/FMX

PH3 1120

WPIDY=TPIDY/FMX

PH3 1130

4015 IF(MXA)4018,4018,9937

PH3 1140

IF GREATER, YOU HAVE FAILED TO SPECIFY THE

NO. OF PARTICLES TO GENERATE.

4018 NY=NT

PH3 1150

DO 5700 I=IXN,IXX

PH3 1160

COMPUTE THE COORDINATE OF THE PARTICLE

UNDER CONSIDERATION

PH3 1170

WS5=DX(I)/FMX

PH3 1180

THE VOLUME OF THE SUBDIVIDED CELL =

PI(2.*XL(N)DY/N*DY/N).

PH3 1190

TABX(1)=X(I)-TFMX*DX(I)

PH3 1200

4019 IF(MXA)4020,4024,9938

PH3 1210

4020 DO 4022 K=2,MXS

PH3 1220

WE START AT THE RIGHT AND TOP OF CELL(K).

SET UP ARRAY FOR X COORDINATES OF THE

PARTICLES.

4022 TABX(K)=TABX(K-1)-WS5

PH3 1230

J LOOP, LIMITS OF Y FOR THIS PACKAGE.

4024 DO 5700 J=IYN,IYX

PH3 1240

TAM=WPIDY*WS5*DY(J)

PH3 1250

TAM= 2PI/N*DX/N*DY

E=0.0

PH3 1260

IIWS=0

PH3 1270

IWS=0

PH3 1280

IB=0	PH3 1290
WS=DY(J)/FMX	PH3 1300
TABY(1)=Y(J)-TFMX*DY(J)	PH3 1310
C MXS=N	
DO 4026 K=2,MXS	PH3 1320
C SET UP ARRAY FOR Y COORDINATES OF THE	
C PARTICLES.	
4026 TABY(K)=TABY(K-1)-WS	PH3 1330
C K USED FOR THE CELL QUANTITIES.	
K=(J-1)*IMAX+I+1	PH3 1340
4028 IBB=IB/MXS	PH3 1350
IB=IB+1	PH3 1360
IBA=MOD(IB,MXS)	PH3 1370
C TX=X COORDINATE OF PARTICLE IN QUESTION.	
TX=TABX(IBA+1)	PH3 1380
C TY=Y COORDINATE OF PARTICLE IN QUESTION.	
TY=TABY(IBB+1)	PH3 1390
C GENERATE OR DELETE THE PARTICLE	PH3 1400
ID=0	PH3 1410
IG=0	PH3 1420
L=1	
4202 CONTINUE	
KK=ITAB(L)	PH3 1440
IF(KK-5)4062,4073,4078	PH3 1450
C TRIANGLE	PH3 1460
4062 WSX=(TY-TAB(L+1))/TAB(L+2)	PH3 1470
IF(WSX-TX)4064,4064,4200	PH3 1480
4064 WSX=(TY-TAB(L+3))/TAB(L+4)	PH3 1490
IF(WSX-TX)4200,4066,4066	PH3 1500
4066 WSY=TAB(L+6)*TX+TAB(L+5)	PH3 1510
IF(KK-2)4068,4068,4072	PH3 1520
4068 IF(WSY-TY)4200,4070,4070	PH3 1530
4070 GO TO (4074,4076,4074,4076),KK	PH3 1540
4072 IF(WSY-TY)4070,4070,4200	PH3 1550
4074 ID=1	PH3 1560
GO TO 4200	PH3 1570
4076 IG=1	PH3 1580
GO TO 4200	PH3 1590
4078 KK=KK-4	PH3 1600
4077 IF(KK-8)4079,4094,9939	PH3 1610
4079 GO TO (4080,4080,4090,4090,4092,4092,4094),KK	PH3 1620
C RECTANGLE	PH3 1630
4080 IF(TAB(L+1)-TX)4082,4082,4200	PH3 1640
4082 IF(TAB(L+2)-TX)4200,4084,4084	PH3 1650
4084 IF(TAB(L+3)-TY)4086,4086,4200	PH3 1660
4086 IF(TAB(L+4)-TY)4200,4088,4088	PH3 1670
4088 GO TO (4074,4076),KK	PH3 1680
C ELLIPSE WITH NO PERTURBATION	PH3 1690
4090 KK=KK-2	PH3 1700
IF(((TX-TAB(L+3))*2/TAB(L+1)+(TY-TAB(L+4))*2	PH3 1710
1/TAB(L+2)-1.0)4088,4088,4200	PH3 1720
C CIRCLE	PH3 1730
4092 KK=KK-4	PH3 1740
IF(((TX-TAB(L+3))*2+(TY-TAB(L+4))*2-TAB(L+1))	PH3 1750
2 4088,4088,4200	PH3 1760
C ELLIPSE WITH PERTURBATION	PH3 1770
4094 KK=KK-6	PH3 1780

PH3 1790
PH3 1800

```

0IF((TX/TAB(L+1))*2+(TY-TAB(L+4)-TAB(L+3)*(TX*
1 (TX-TAB(L+1))*2)**2/TAB(L+2)-1.0)4088,4088,4200
4200 L=L+NPP
IF(L-JA)4202,4201,4201

```

PH3 1820
PH3 1830
PH3 1840
PH3 1850
PH3 1860
PH3 1870
PH3 1880
PH3 1890
PH3 1900
PH3 1910
PH3 1950

```

C
C      IF ID=1 DELETE
4201 IF(ID)9940,4310,4800
C      IF ID=0 AND IG=0 DELETE
4310 IF(IG)9941,4800,4312
C      GENERATE PARTICLE
4312 NY=NY+1
      IF(IIWS)23,22,23
      22 IIWS=1
      23 IWS=1
      NYY=NYY+1
      CALL PH4
C      RETURN FROM PH4 WITH THE FOLLOWING DATA,
C      WSR=PARTICLE DENSITY
C      WSI=PARTICLE SPECIFIC INTERNAL ENERGY
C      WSU=RADIAL VELOCITY COMPONENT OF PARTICLE
C      WSV=AXIAL VELOCITY COMPONENT OF PARTICLE
4332 N=NYY
      IF(IIWS)4335,4335,24
      24 IIWS=-1
4333 IF(AMX(K)+AMD(K))9951,4335,4334
C      CALCULATE PACKAGE ENERGY.
4334 E=((U(K)**2+V(K)**2)/(AMX(K)+AMD(K)))*.5+AIX(K)+AID(K)
C      THE FOLLOWING IS FOR PIC TRANSPORT ONLY
C      SET THE PARTICLE COORDINATES INTO THE
C      PROPER ARRAYS.
4335 XL(N)=TX
      YL(N)=TY
C      SET I AND J OF CELL K(LOCATION OF PARTICLE).
      IW1(N)=I
      IW2(N)=J
C      CALCULATE PARTICLE MASS AS
C      =2PI/N*DX/N*DY*XL(N)*RHO.
      AM(N)=TAM*TX*WSR
C      CHECK FOR TYPE OF MASS(X OR .)
4341 IF(LX)9945,4342,4344
4342 WS=AM(N)*WSI
      IF(AM(N)-AMDM)16,15,15
      16 AMDM=AM(N)
C      NOTE, AID HERE IS INTERNAL ENERGY,
C      NOT SPECIFIC INTRNAL ENERGY.
      15 AID(K)=AID(K)+WS
      PM=PM+AM(N)
      AMD(K)=AMD(K)+AM(N)
      AM(N)=-AM(N)
      GO TO 4346
      4344 WS=AM(N)*WSI
      IF(AM(N)-AMXM)18,17,17
      18 AMXM=AM(N)
C      NOTE, AIX HERE IS INTERNAL ENERGY,
C      NOT SPECIFIC INTERNAL ENERGY.
      17 AIX(K)=AIX(K)+WS
      PM=PM+AM(N)

```

PH3 1960
PH3 1970
PH3 1980

PH3 2010
PH3 2020

PH3 2030
PH3 2040

PH3 2050

PH3 2060
PH3 2070
PH3 2080
PH3 2090

PH3 2110

PH3 2120
PH3 2130
PH3 2140
PH3 2150
PH3 2160

PH3 2170
PH3 2180

C	SUM UP MASS, BOTH COMPONENTS OF MOMENTA	
C	AND TOTAL INTERNAL ENERGY IN CELL K.	
	AMX(K)=AMX(K)+AM(N)	PH3 2190
C	** NOTE, U AND V ARE NOT VELOCITY COMPONENTS	
C	HERE IN PH3, BUT ARE THE RESPECTIVE	
C	RADIAL AND AXIAL MOMENTAS.	
4346	U(K)=U(K)+ABS(AM(N))*WSU	PH3 2200
	V(K)=V(K)+ABS(AM(N))*WSV	PH3 2210
	IF(NY-NPRR)4800,14,9945	PH3 2220
14	NRC=NRC+1	PH3 2230
	NPRR=NPRR+NPRI-1	PH3 2240
5001	NYI=1	PH3 2270
3	DO 2 N=2,NPRI	PH3 2280
C	SET PARTICLE ARRAYS TO ZERO.	
	XL(N)=0.0	PH3 2290
	YL(N)=0.0	PH3 2300
	AM(N)=0.0	PH3 2310
	IW1(N)=0	PH3 2320
	IW2(N)=0	PH3 2330
2	CONTINUE	PH3 2340
4800	IF(MX-IB)9946,4830,4028	PH3 2350
C	C A L C U L A T E E N E R G Y F O R P K G	PH3 2360
4880	IF(IWS)4900,5700,4900	PH3 2370
4900	IF(AMX(K)+AMD(K))9951,5700,4910	
4910	PEE=(U(K)**2+V(K)**2)/(AMX(K)+AMD(K))*0.5+AIX(K)+AID(K)	
4930	IF(E)4950,4950,4940	PH3 2400
4940	PEE=PEE-E	PH3 2410
4950	PE=PE+PEE	PH3 2420
5700	CONTINUE	PH3 2430
	I=IA	PH3 2440
	J=JA	PH3 2450
	K=IJ	PH3 2460
	L=JT	PH3 2470
	GO TO 10000	PH3 2480
C	E R R O R	PH3 2490
9932	NK=4000	PH3 2500
	GO TO 9999	PH3 2510
9935	NK=4010	PH3 2520
	GO TO 9999	PH3 2530
9936	NK=4011	PH3 2540
	GO TO 9999	PH3 2550
9937	NK=4015	PH3 2560
	GO TO 9999	PH3 2570
9938	NK=4019	PH3 2580
	GO TO 9999	PH3 2590
9939	NK=4077	PH3 2600
	GO TO 9999	PH3 2610
9940	NK=4201	PH3 2620
	GO TO 9999	PH3 2630
9941	NK=4310	PH3 2640
	GO TO 9999	PH3 2650
9945	NK=4341	PH3 2660
	GO TO 9999	PH3 2670
9946	NK=4800	PH3 2680
	GO TO 9999	PH3 2690
9951	NK=4905	PH3 2700
9999	WRITE (6,8888)NK,I,J,K,L,M,N	PH3 2710

```

PRINT 8888,NK,I,J,K,L,M,N
CALL DUMP
10000 RETURN
8888 FORMAT(1H+/26H1 P H 3 ERROR IN STATEMENTIS,12X,12H INDICES ARE6I7)
END
SUBROUTINE PH4

C
C
C    ** NOTE, XC AND YC ARE COORDINATES FOR RELOCATING
C    THE ORIGIN FOT THE
C    RHO,INTERNAL ENERGY, AND VELOCITY FITS.
C
THE ACTUAL COORDINATES USED IN THE FIT
SUBROUTINES IS TTX=TX-XC,TTY=TY-YC.
TTX=TX-XC
TTY=TY-YC
LL=S8
GO TO(1,2,3,4,5,6),LL
1 CALL FIT1
GO TO 7
2 CALL FIT2
GO TO 7
3 CALL FIT3
GO TO 7
4 CALL FIT4
GO TO 7
5 CALL FIT5
GO TO 7
6 CALL FIT6
7 RETURN
END
SUBROUTINE FIT1

C
C
C    WS=SQRT(TTX**2+TTY**2)
C    DENSITY
C    WSR=TABR(1)+TABR(2)*(TTY-TABR(3))
C    ENERGY
C    WSI=TABI(1)+TABI(2)*(TTY-TABI(3))
C    VELOCITIES
C    WS=TABUV(1)+TABUV(2)*(TTY-TABUV(3))
C    WSU=0.0
C    WSV=WS
C    RETURN
C    END
SUBROUTINE FIT2

C
C
C    WS=SQRT(TTX**2+TTY**2)
C    DENSITY
C    WSR=((TTX-TABR(1))/TABR(2))**2+((TTY-TABR(3))/
1TABR(4))**2
ENERGY
WSI=TABI(1)+TABI(2)*TTX+TABI(3)*TTX**2

```

C	1+TABI(4)*TTY+TABI(5)*TTY**2.	FIT21010
C	VELOCITIES	FIT21020
	WSV=TABUV(1)+TABUV(2)*TTY	FIT21030
	WSU=TABUV(3)+TABUV(4)*TTY	FIT21040
	RETURN	FIT21050
	END	FIT21060
	SUBROUTINE FIT3	FIT30010
C		INPU0710
C		FIT30730
C		FIT30940
C	THIS FIT FOR SIN KZ/KZ *****	
C	WS=SQRT(TTX**2+TTY**2)	FIT30950
	DENSITY	FIT30960
	WSR=TABR(1)+TABR(2)*(TTY-TABR(3))	
	WSA=TTY/TABI(2)	
	WSB=WSA*PIDY*2.	
	WSC=SIN(WSB)	
	WSI=WSC/WSA*TABI(1)	
	WS=TABUV(1)+TABUV(2)*(TTY-TABUV(3))	
	WSU=0.	
	WSV=WS	
	WSI=WSI*TABI(3)	
C	TABI(3) US SCALE FACTOR FOR YIELD NORMALLY SET TO 1.	
	RETURN	
	END	
	SUBROUTINE FIT4	FIT40010
	RETURN	FIT40020
	END	FIT40030
	SUBROUTINE FIT5	
	RETURN	
	END	
	SUBROUTINE FIT6	FIT60010
	RETURN	FIT60020
	END	FIT60030
	SUBROUTINE OUTPUT	OUTP0010
C		INPU0710
C		
C	***** A 2 MATERIAL CLAM FOR THE TOIL CODE *****	
C		
C	L A M ***** O U T P U T *****	OUTP0020
C		OUTP0030
C		OUTP0750
C		OUTP0970
C	PACKAGES HAVE BEEN READ IN AND PROCESSED	OUTP0980
C	COMPUTE TOTAL ENERGIES AND TOTAL MASSES	OUTP0990
	E=ETH	OUTP1000
	WRITE (6,8104)	OUTP1010
7001	ND=ND+1	OUTP1020
	IF(E)6000,6000,6001	OUTP1030
6000	AMDM=0.0	OUTP1040
	AMXM=0.0	OUTP1050
	GO TO 7016	OUTP1060
6001	AMDM=AMDM/2.0	OUTP1070
	AMXM=AMXM/2.0	OUTP1080
7013	IF(AMDM)9901,9901,7014	OUTP1090
7014	IF(AMXM)9902,9902,7016	OUTP1100

7016	ETH=0.0	OUTP1110
	TMDZ=0.0	OUTP1120
	TMXZ=0.0	OUTP1130
	DO 7015 I=2,KMAX	
7017	IF(AMD(I))9904,7010,7004	
7010	WSI=0.	
	IF(AMX(I))9904,7012,7006	
7004	TMDZ=AMD(I)+TMDZ	
	WSI=AID(I)	
C	CALCULATE THE SPECIFIC INTERNAL ENERGY	
C	FOR(.) MATERIAL IN CELL K.	
	AID(I)=AID(I)/AMD(I)	
7005	IF(AMX(I))9904,7008,7006	
7006	WSI=WSI+AIX(I)	
C	SUM UP TOTAL (X) MASS IN GRID.	
	TMXZ=AMX(I)+TMXZ	OUTP1170
C	CALCULATE THE SPECIFIC INTERNAL ENERGY	
C	FOR(X) MATERIAL IN CELL K.	
	AIX(I)=AIX(I)/AMX(I)	OUTP1180
7008	WS=AMX(I)+AMD(I)	
C	CALCULATE RADIAL AND AXIAL VELOCITIES BY	
C	CONSERVING BOTH COMPONENTS OF MOMENTA.	
	U(I)=U(I)/WS	OUTP1200
	V(I)=V(I)/WS	OUTP1210
C	SUM UP TOTAL ENERGY IN SYSTEM.	
	ETH=((U(I)**2+V(I)**2)/2.)*WS+WSI+ETH	
	GO TO 7012	OUTP1230
C	SET FLAGS FOR TYPE OF MATERIAL IN CELL K.	
7012	IF(AMX(I)+AMD(I))2000,2000,2001	
2000	DKE(I)=0.	
	GO TO 7015	
2001	IF(AMX(I))2002,2002,2003	
2002	DKE(I)=-1.	
	GO TO 7015	
2003	IF(AMD(I))2004,2004,2005	
2004	DKE(I)=-2.0	
	GO TO 7015	
2005	DKE(I)=1.0	
7015	CONTINUE	
	TMZ=TMDZ+TMXZ	OUTP1250
	WRITE (6,8072)ETH,E,TMDZ,TMXZ,TMZ	OUTP1260
	IWS=ND-1	OUTP1270
	IWSA=NMAX-ND	OUTP1280
	IWSB=NMAX-1	OUTP1290
	WRITE (6,8073)(IWS,IWSA,IWSB)	OUTP1300
7113	REWIND N7	OUTP1320
C		OUTP1330
C	WRITE TAPE FOR THE TOIL CODE.	
C		OUTP1350
	IF(PROB)7162,7162,7163	OUTP1360
7163	N3=0	OUTP1370
7162	WS=555.0	OUTP1380
	WRITE (N7)WS,CYCLE,N3	OUTP1390
	WRITE (N7)(Z(I),I=1,MZ)	OUTP1400
7131	WRITE(N7)(U(K),V(K),AMD(K),AMX(K),AID(K),AIX(K),	
	1AIX(K),DKE(K),K=1,KMAX)	
	GO TO 7140	OUTP1420

7140	CONTINUE	OUTP1430
	WRITE(N7)(X(K),TAU(K),K=1,IMAX)	
	WRITE(N7)(Y(K),K=1,JMAX)	OUTP1460
	WS=666.0	OUTP1470
C		
C	EDIT OUT THE VELOCITIES, MASS	
C	AND SPECIFIC INTERNAL ENERGIES AS A FUNCTION	
C	OF J FOR ALL I .	
7161	WRITE (N7)WS,WS,WS	OUTP1530
	REWIND N7	OUTP1540
	WRITE (6,8120)T,INC	OUTP1550
	IWS=IMAX*JMAX+1	OUTP1560
	CALL SLITE (0)	OUTP1570
	DO 7517 I=1,IMAX	OUTP1580
	CALL SLITE (1)	OUTP1590
	J=JMAXA	OUTP1600
	K=IWS+I	OUTP1610
	DO 7517 JP=1,JMAX	OUTP1620
	J=J-1	OUTP1630
	K=K-IMAX	OUTP1640
7170	IF(AMX(K)+AMD(K))9905,7517,7175	
7175	CALL SLITET(1,K000FX)	OUTP1660
	GO TO(7180,7185),K000FX	OUTP1670
C	PRINT OUT CELL QUANTITIES.	
7180	WRITE (6,8080)I,X(I),DX(I)	OUTP1680
7185	WRITE(6,8084)J,Y(J),DY(J),U(K),V(K),AID(K),AIX(K),AMD(K),AMX(K)	
7517	CONTINUE	OUTP1710
	IF(Q000FL)7520,7520,7616	OUTP1720
7616	CONTINUE	
	GO TO 7520	OUTP1820
C	ERROR	OUTP1830
9901	NK=7013	OUTP1840
	GO TO 9999	OUTP1850
9902	NK=7014	OUTP1860
	GO TO 9999	OUTP1870
9904	NK=7005	OUTP1880
	GO TO 9999	OUTP1890
9905	NK=7170	OUTP1900
9999	WRITE (6,8808)NK,I,J,K,L,M,N	OUTP1910
	PRINT 8888,NK,I,J,K,L,M,N	OUTP1920
	CALL DUMP	OUTP1930
7520	RETURN	OUTP1940
C	FORMATS	OUTP1950
80720	FORMAT(1H ///6H THE =1PE16.9,7X,3HE =E16.9///5H M. =E11.5,5X,4HMX	OUTP1960
	1 =E11.5,7X,7HM.+MX =E11.5)	OUTP1970
8073	FORMAT(1H0/17H0PARTICLES - - -I12,4H DOTI14,2H XI14,6H TOTAL)	OUTP1980
80800	FORMAT(1H0///3H0I=12,10X,2Hx=1PE13.7,10X,3HDX=E13.7/3H0 J10X,1HY130	OUTP1990
	1X,2HDY12X,1HU13X,1HV12X,3HAID11X,3HAIX11X,3HAMD11X,3HAMX)	OUTP2000
8084	FORMAT(I3,3X,1P8E14.7)	OUTP2010
8104	FORMAT(1H /31H THERE ARE NO MORE PACKAGES----	OUTP2060
8120	FORMAT(1H ///18H TAPE DUMP AT TIMEF10.1,7X,5HCYCLEI4)	OUTP2070
8888	FORMAT(1H+/26H10UTPUT ERROR IN STATEMENTI5,12X,12H INDICES ARE6I7)	OUTP2080
	END	OUTP2090

SUBROUTINE PH1

PH1 003

A TWO MATERIAL OIL

NOTE,, THE FOLLOWING SET OF DIMENSIONS,
COMMON, AND EQUIVALENCE ARE TO BE USED
FOR ALL SUBROUTINES EXCEPT
THE CARDS ROUTINE

D I M E N S I O N

 EDIT0060
 EDIT0070
 EDIT0080

DIMENSION AM(130), XL(130), YL(130),
 1U(4500),V(4500),AMD(4500),AMX(4500),AID(4500),AIX(4500),
 2P(4500),DKE(4500),THETA(4500),
 3IW1(130),W2(30),
 4DX(100),X(100),XX(101),DY(100),Y(100),YY(101),
 5TAB(15), AMK(15), PK(15), QK(15), Z(150), IZ(150),
 6TAU(100),PL(200),PR(200),UL(200),UR(200),
 7FLEFT(100),YAMC(100), SIGC(100), GAMC(100)

EDIT0130

EDIT0150

DIMENSION DMASL(130),DXML(130),DYML(130),DENRG(130)

COMMON Z ,XX ,UR ,PR ,THETA ,YY
 COMMON AID ,AIX ,AM ,AMD ,AMX ,AREA
 COMMON BIG ,BOUNCE ,DDXN ,DDVK ,DVK ,DX
 COMMON DY ,E ,FD ,FS ,FX ,OUT
 COMMON P ,PABOVE ,PBLO ,PIDTS ,PPABOV ,PRR
 COMMON PUL ,QDT ,RC ,REZ ,RHO ,RL
 COMMON RR,SIG,BOOFL,SWITCH ,TABLM,TAU
 COMMON TAUOTS ,TAUDTX ,U ,UK ,URR ,UT
 COMMON UU ,UUU ,UTEF ,UVMAX ,V ,VABOVE
 COMMON VBLO ,VEL ,VK ,VT ,VTEF ,VV
 COMMON VVABOV ,VVBL0 ,W2 ,W3 ,WPS ,WS
 COMMON WSA ,WSB ,WSC ,XL ,XLF ,XN
 COMMON XR ,YL ,YLW ,YN ,YU ,ZMAX
 COMMON I ,II ,IN ,IR ,IWS ,IWSA
 COMMON IWSB ,IWSC ,IW1 ,J ,JN ,JP
 COMMON JR ,K ,KN ,KP ,KR ,KRM
 COMMON L ,M ,MA ,MB ,MC ,MD
 COMMON ME ,MZ ,N ,NK ,NKMAX ,NK1
 COMMON NO ,NR

 EDIT0160
 EDIT0170
 EDIT0180
 EDIT0190
 EDIT0200
 EDIT0210
 EDIT0220
 EDIT0230
 EDIT0240
 EDIT0250
 EDIT0260
 EDIT0270
 EDIT0280
 EDIT0290
 EDIT0300
 EDIT0310
 EDIT0320
 EDIT0330
 EDIT0340

E Q U I V A L E N C E

 EDIT0440
 EDIT0450
 EDIT0460

OEQUIVALENCE (Z,IZ,PROB), (Z(2),CYCLE), (Z(3),DT),
 1(Z(4),PRINTS), (Z(5),PRINTL), (Z(6),DUMPT7), (Z(7),CSTOP),
 2(Z(8),PIDY), (Z(9),TMZ), (Z(10),GAM), (Z(11),GAMD),
 3(Z(12),GAMX), (Z(13),ETH), (Z(14),FFA), (Z(15),FFB),
 4(Z(16),TMDZ), (Z(17),TMXZ), (Z(18),XMAX), (Z(19),TXMAX),
 5(Z(20),TYMAX), (Z(21),AMDM), (Z(22),AMXM), (Z(23),DNN),
 6(Z(24),DMIN), (Z(25),FEF), (Z(26),DTNA), (Z(27),CVIS),
 7(Z(28),NPR), (Z(29),NPRI), (Z(30),NC), (Z(31),NPC),
 8(Z(32),NRC), (Z(33),IMAX), (Z(34),IMAXA), (Z(35),JMAX),
 9(Z(36),JMAXA), (Z(37),KMAX), (Z(38),KMAXA), (Z(39),NMAX),
 OEQUIVALENCE (Z(40),ND), (Z(41),KDT), (Z(42),IXMAX),
 1(Z(43),NOD), (Z(44),NOPR), (Z(45),NT AX), (Z(46),NJMAX),
 2(Z(47),I1), (Z(48),I2), (Z(49),I3), (Z(50),I4),
 3(Z(51),N1), (Z(52),N2), (Z(53),N3), (Z(54),N4),
 4(Z(55),N5), (Z(56),N6), (Z(57),N7), (Z(58),N8),
 5(Z(59),N9), (Z(60),N10), (Z(61),N11), (Z(62),NRM),

 EDIT0470
 EDIT0480
 EDIT0490
 EDIT0500
 EDIT0510
 EDIT0520
 EDIT0530
 EDIT0540
 EDIT0550
 EDIT0560
 EDIT0570
 EDIT0580
 EDIT0590
 EDIT0600
 EDIT0610
 EDIT0620

```

6(Z(63),TRAD), (Z(64),XNRG), (Z(65),SN), (Z(66),DXN), EDIT0630
7(Z(67),RADER), (Z(68),RADET), (Z(69),RADEB), (Z(70),DTRAD), EDIT0640
8(Z(71),REZECT), (Z(72),RSTOP), (Z(73),SHELL), (Z(74),BBOUND), EDIT0650
9(Z(75),TOZONE), (Z(76),ECK), (Z(77),SBOUND), (Z(78),X1) EDIT0660
0EQUIVALENCE (Z(79),X2), (Z(80),Y1), (Z(81),Y2), EDIT0670
1(Z(82),CABL), (Z(83),VISC), (Z(84),T), (Z(85),GMAX), EDIT0680
2(Z(86),WSGD), (Z(87),WSGX), (Z(88),GMADR), (Z(89),GMAXR), EDIT0690
3(Z(90),S1), (Z(91),S2), (Z(92),S3), (Z(93),S4), EDIT0700
4(Z(94),S5), (Z(95),S6), (Z(96),S7), (Z(97),S8), EDIT0710
5(Z(98),S9), (Z(99),S10) EDIT0720

```

```

0EQUIVALENCE (XX(2),X(1)), (UR,UL,FLEFT), (UR(100),YAMC), EDIT0740
1(PR(100),SIGC), (PR,PL,GAMC), (DKE,THETA), (UR,TAB), EDIT0750
2(UR(16),AMK), (UR(31),PK), (UR(46),QK), (YY(2),Y(1)) EDIT0760
EQUIVALENCE (AM,DHSL),(XL,DXML),
1(YL,DYML),(IW1,DENRG)
DIMENSION PLOT(10)
DATA PLOT/1H,1H,1H,1H,1H,1H,1H,1H,1H,1H-/

```

00000030

```

C INPUT READS THE TOIL DUMP TAPE OR
C WILL CALL SUBROUTINE SET'UP WHICH
C WILL MAKE A DUMP TAPE FOR CERTAIN TYPES OF PROBLEM
C CALL INPUT

```

```

C CDT ROUTINE CALCULATES DT(HYDRO TIME STEP)
C AND PRESSURES, ADVANCE CYCLE NO. ETC.

```

10 CALL CDT

```

C IN EDIT, DETERMINE WHETHER TO EXECUTE A LONG
C PRINT, A SHORT PRINT, A TAPE DUMP, ETC. AND
C CALCULATE TOTAL ENERGY IN SYSTEM(COMPARE
C WITH ETH) TOTAL MASS, INTEGRATE TOTAL
C COMPONENTS OF MOMENTA.

```

CALL EDIT

CALL SLITET(1,K000FX)

SENSE LITE 1 SIGNIFIES THIS

IS THE LAST CYCLE OF THIS RUN \$\$\$\$\$\$\$\$\$\$\$\$\$\$

LITE TURNED ON IN THE EDIT ROUTINE *****

GO TO(30,20),K000FX

```

C PH1, INTEGRATE THE MOMENTA EQS. INTEGRATE
C ENERGY EQUATION(ONLY CHANGES DUE TO WORK
C TERMS). NO MOVEMENT OF MASS HERE

```

20 CALL PH1

```

C TRANSPORT MASS ACROSS BOUNDARIES (SOLVE
C MASS TRANSPORT EQ.) TRANSPORT TERMS IN
C THE MOMENTA AND ENERGY EQS. LEFT OUT OF
C PH1, HERE APPROXIMATED BY MASS MOVEMENT. CONSERVE
C MASS, MOMENTA AND TOTAL ENERGY.

```

CALL PH2

GO TO 10

30 CALL EXIT

END

SUBROUTINE INPUT

***** A 2 MATERIAL OIL CODE *****

CALL SLITE (3)

INPU0049

INPU1050

INPU1060

```

C      READ IN COUNTER, FOR THE NO. OF HEADER CARDS.
      READ(5,8009)II
C009  FORMAT(6I3)
C      READ IN THE HEADER CARDS.
      DO 8010 I=1,II
      READ  (5,8004)IWS
      WRITE (6,8004)IWS
      8010 CONTINUE
      6  CALL CARDS
C      NOTE, OPTION FOR CALLING SETUP.
      IF(PK(3)) 8887,8888,8888
      8888 CALL CARDS
      CALL  SETUP
      8887 CONTINUE
C      READ TAPE
      GO TO 1000
      10 CONTINUE
      CALL CARDS
C      EXECUTE R E S
      GO TO 2000
C
C      40 CONTINUE
C      DECREASE T BY DT, SINCE CDT ROUTINE
C      INTEGRATES THE TIME.
      T=T-DTNA
C      ALSO CYCLE NO.
      NC=NC+1
      CYCLE=NC
C      AND NO. OF CYCLES BETWEEN ENERGY CHECKS.
      NPC=NPC+1
      UVMAX=0.0
C      GENERATE DX AND DY FOR ALL I AND J
C      SINCE THEY ARE NOT ON THE DUMP TAPE.
      DX(1)=X(1)
      DO 50 I=2,IMAX
      50 DX(I)=X(I)-X(I-1)
      DY(1)=Y(1)
      DO 55 J=2,JMAX
      55 DY(J)=Y(J)-Y(J-1)
C      EDIT OUT THE Z BLOCK.
      K=1
      DO 80 I=1,3
      L=K+8
      WRITE(6,8005)K,(Z(N),N=K,L)
      80 K=L+1
      K=28
      DO 81 I=1,4
      L=K+8
      WRITE(6,8006)K,(IZ(N),N=K,L)
      81 K=L+1
      K=62
      DO 82 I=1,10
      L=K+8
      WRITE(6,8005)K,(Z(N),N=K,L)
      82 K=L+1
      8005 FORMAT(I4,1X,1P9E12.5)

```

INPU1080
INPU1090

INPU1120

INPU1270
INPU1280
INPU1340
INPU1350
INPU1360
INPU1370
INPU1380
INPU1390

INPU1400

INPU1410
INPU1420INPU1430
INPU1440

INPU1470

INPU1490

```

8006 FORMAT(I4,1X,9I7)
GO TO 10000
C
C
C
C
C**** READ TAPE *****
1000 MZ=150
IWS=0
1003 REWIND N7
1004 READ(N7)PR(1),PR(2),N3
C          NR = NUMBER OF RECORDS
C NOTE *****
NR=N3+5
1006 IF(PR(1)-555.0)1010,1016,1010
1010 IWS=IWS+1
1011 IF(MOD(IWS,3))9902,9902,1003
1016 IF(PR(2))1010,1016,1018
C CHECK FOR CORRECT CYCLE NO.
1018 IF(PK(2)-PR(2))1023,1023,1020
1020 DO 1022 L=2,NR
1022 READ(N7)DUM
GO TO 1004
:1023 READ(N7)(Z(I),I=1,MZ)
C CHECK FOR CORRECT PROBLEM NO.
IF(ABS(PROB-PK(1))-0.01)1024,1024,9901
:1024 READ(N7)(U(I),V(I),AMD(I),AMX(I),AID(I),AIX(I),
1P(I),DKE(I),I=1,K:AXA)
READ(N7)(X(K),TAU(K),K=1,IMAX)
READ(N7)(Y(K),K=1,JMAX)
1034 READ(N7)PR(1),PR(2),PR(3)
1036 IF(PR(1)-555.0)9904,1040,1038
1038 IF(PR(2)-666.0)9905,1040,9905
1040 GO TO 10
C**** END OF READ TAPE *****
C
C
C
C
C CALCULATE MAXIMUM GAMMA AND
GAMMA/(GAMMA-1.) FOR EACH MATERIAL.
2000 IF(WSGX)9906,2010,2005
2005 GAMX=1.0/(WSGX-1.0)
2010 WSGX=(GAMX+1.0)/GAMX
GMAXR=GAMX*WSGX
2012 IF(WSGD)9907,2020,2015
2015 GAMD=1.0/(WSGD-1.0)
2020 WSGD=(GAMD+1.0)/GAMD
GMAXR=GAMD*WSGD
GMAX=WSGD
IF(WSGD-WSGX)2025,2030,2030
2025 GMAX=WSGX
2030 GO TO 40
C**** END OF RES *****
C
C
C
C          ERROR
9901 NK=1023
GO TO 9999

```

INPU1600

INPU1610

INPU1620

INPU1630

INPU1640

INPU1650

INPU1660

INPU1690

INPU1700

INPU1720

INPU1730

INPU1740

INPU1750

INPU1760

INPU1770

INPU1810

INPU1830

INPU1960

INPU1970

INPU1980

INPU1990

INPU2000

INPU2010

INPU2020

INPU2040

INPU2050

INPU2060

INPU2070

INPU2080

INPU2090

INPU2100

INPU2110

INPU2120

INPU2130

INPU2140

INPU2150

INPU2160

INPU2170

INPU2180

INPU2190

INPU2200

INPU2210

```

9902 NK=1011
      GO TO 9999
9904 NK=1036
      GO TO 9999
9905 NK=1038
      GO TO 9999
9906 NK=2000
      GO TO 9999
9907 NK=2012
9999 NR=1
      CALL DUMP

```

```

C
10000 RETURN

```

```

C
C          FORMATS
8000 FORMAT(7E11.3,I2)
80040FORMAT(I1,7I1H
1          )

```

```

C
      END

```

```

C
C -----
C

```

```

SUBROUTINE SETUP

```

```

      D I M E N S I O N

```

```

EDIT0060
EDIT0070

```

```

C
C      TWO MATERIAL SETUP
C      PACKAGES MUST BE RECTANGLES.
C      ASSUMPTION OF = DX AND = DY
C      LOAD PK(4)=1.
      M=PK(4)
C      LOAD PK(5)=RIGHT BOUNDARY OF PELLET(I).
      MA=PK(5)
C      LOAD PK(6)=BOTTOM(J)+1 OF PELLET.
      MB=PK(6)
C      LOAD PK(7)=TOP(J) OF PELLET.
      MC=PK(7)
C      LOAD PK(8)=1.
      MD=PK(8)
C      LOAD PK(9)=RIGHT(I) BOUNDARY OF TARGET.
      ME=PK(9)
C      LOAD PK(10)=BOTTOM(J)+1 OF TARGET.
      MZ=PK(10)
C      LOAD PK(11)=TOP(J) OF TARGET.
      N=PK(11)
C      LOAD INITIAL DENSITY INTO Z(115) FOR (X) MATERIAL
C      FOR THE PROJECTILE.
C      AND Z(116) FOR THE DOT MATERIAL(TARGET))
C      LOAD INITIAL PELLET VELOCITY INTO Z(112).
      VTEF=Z(112)
      KMAX=IMAX*JMAX+1
      KMAXA=KMAX+1
      JMAXA=JMAX+1
      IMAXA=IMAX+1
C      CLEAR ALL CELL ARRAYS.
      DO 1 K=1,KMAX
      U(K)=0.0

```

```

INPU2220
INPU2230
INPU2240
INPU2250
INPU2260
INPU2270
INPU2280
INPU2290
INPU2300
INPU2310
INPU2340
INPU2350
INPU2360
INPU2370
INPU2380
INPU2390
INPU2400
INPU2410
INPU2430
INPU2440

```

```

V(K)=0.0
P(K)=0.0
AMX(K)=0.0
AIX(K)=0.0
AID(K)=0.
AMD(K)=0.
DKE(K)=0.
1 CONTINUE
DX(1)=DX(1)
X(1)=DX(1)
WS=X(1)**2
PIDY=3.1415927
TAU(1)=WS*PIDY
C CALCULATE DX,X,TAU.
DO 10 I=2,IMAX
X(I)=X(I-1)+DX(1)
DX(I)=DX(1)
WSA=X(I)**2
TAU(I)=PIDY*(WSA-1'S)
WS=WSA
10 CONTINUE
Y(1)=DY(1)
C CALCULATE DY AND Y.
DO 20 J=2,JMAX
Y(J)=Y(J-1)+DY(1)
DY(J)=DY(1)
20 CONTINUE
ETH=0.0
DO 30 I=M,MA
K=(MB-1)*IMAX+I+1
C CALCULATE MASS, AND VELOCITY OF PELLET.
DO 40 J=MB,MC
AMX(K)=Z(115)*DY(J)*TAU(I)
V(K)=VTEF
C CALCULATE TOTAL ENERGY (ETH.)
ETH=ETH+AMX(K)*(V(K)**2)/2.0
DKE(K)=-2.
40 K=K+IMAX
30 CONTINUE
C CALCULATE MASS OF TARGET.
DO 50 I=MD,ME
K=(MZ-1)*IMAX+I+1
DO 60 J=MZ,N
AMD(K)=Z(116)*DY(J)*TAU(I)
DKE(K)=-1.
60 K=K+IMAX
50 CONTINUE
IMAX=IMAX
JMAX=JMAX
SHELL=2.0
CYCLE=0.0
DT=0.0
NMAX=0
XMAX=X(IMAX)
TXMAX=XMAX*2.0
YMAX=Y(JMAX)
TYMAX=YMAX*2.0

```

C DUMP ON TAPE N7

C
C WRITE STARTING CONDITIONS FOR TOIL
C

```

: REWIND N7
  WS=555.0
  WRITE(N7)WS,CYCLE,N3
  WRITE(N7)(Z(I),I=1,150)
  WRITE(N7)(U(I),V(I),AMD(I),AMX(I),AID(I),AIX(I),
1P(I),DKE(I),I=1,KMAXA)
  WRITE(N7)(X(I),TAU(I),I=1,IMAX)
  WRITE(N7)(Y(I),I=1,JMAX)
  WS=666.0
  WRITE(N7)WS,WS,WS
  REWIND N7
  RETURN
  END

```

C	SUBROUTINE CARDS	CARD0030
C	DIMENSION TABLE(1),CARD(7),LABLE(1)	CARD0040
	COMMON TABLE	CARD0050
	EQUIVALENCE(TABLE(1),LABLE(1))	CARD0070
	WRITE (6,10)	CARD0090
	1 READ (5,11)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)	CARD0100
	WRITE (6,12)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)	CARD0110
	DO 4 I=1,NUMWPC	CARD0120
	J=LOC+I-1	CARD0130
	IF(IEND-2)2,5,2	CARD0140
	5 LABLE(J)=IFIX(CARD(I))	CARD0150
	GO TO 4	CARD0160
	2 TABLE(J)=CARD(I)	CARD0170
	4 CONTINUE	CARD0180
	IF(IEND-1)1,3,1	CARD0190
	3 RETURN	CARD0200
C	FORMATS	CARD0210
	10 FORMAT(20H1TOIL INPUT CARDS///)	
	11 FORMAT(I1,I5,I1,0P7E9.4)	CARD0230
	12 FORMAT(1H I4,I7,I3,1P7E14.6)	CARD0240
	END	CARD0250

C	SUBROUTINE CDT	CDT 0060
C		CDT 0780
C	*****	CDT 0010
C	***** A 2 MATERIAL OIL CODE *****	
C		
C	Z(138) FOR DENSITY CHECK,IF CELL K	CDT 1030
C	HAS RHO LESS THAN Z(138) ,NO STABILITY CHECK	CDT 1040
C	DONE IN CELL K	CDT 1050
C	IF (CABL N) GREATER THAN 0, THE DT LOADED WILL REMAIN	CDT 1080
C	CONSTANT.	CDT 1090
C	IF(CABL N) =0. CODE CONTROLS TIME STEP BETWEEN FFA,FFB	CDT 1100
C	IF (CABL N) LESS THAN 0. CODE CONTROLS,BUT AT	CDT 1110
C	Z(139) OF STABILITY.	CDT 1120
C	\$\$\$\$\$ Z(139) IS A INPUT NO. *****	CDT 1130

3000	VEL=0.0	
3005	DO 3050 I=1,I1	CDT 1140
3010	K=I+1	
3015	DO 3050 J=1,I2	CDT 1
3020	IF(AMX(K)+AMD(K))9901,3050,3025	CDT 1200
C		CDT 1320
C	CALL EQUATION OF STATE	
3025	CALL ES	
3030	IF(ABS(P(K))-1.0E-10)3035,3035,3040	
3035	P(K)=0.0	CDT 1340
3040	IF(WSGX-VEL)3050,3050,3045	CDT 1350
3045	VEL=WSGX	CDT 1360
3050	K=K+IMAX	CDT 1370
3055	KDT=1	CDT 1380
	UVMAX=-1.0	CDT 1390
3070	DO 3255 I=1,I1	
3075	K=I+1	CDT 1410
3095	DO 3255 J=1,I2	
3100	KP=K+IMAX	CDT 1430
	IF(AMX(K)+AMD(K))9901,3255,4	CDT 1440
C	IF THE DENSITY IS LESS THAN Z(138) WHICH IS A	
C	INPUT NO, THIS CELL WILL BE BYPASSED	
C	FOR STABILITY CONSIDERATIONS.	
:	4 IF((AMX(K)+AMD(K))/(TAU(I)*DY(J))-Z(138))3255,3255,3115	CDT 1450
3115	SIG=DX(I)	CDT 1460
3120	IF (DY(J)-SIG)3125,3130,3130	CDT 1470
3125	SIG=DY(J)	CDT 1480
3130	IF(Z(148))4000,4000,4001	
C	SPEED OF SOUND FOR POLYTROPIC GAS.	
4000	WS=SQRT(GMAX*TAU(1)*DY(J)*ABS(P(K))/(AMX(K)+AMD(K)))	
	GO TO 3205	
C	SPEED OF SOUND FOR METALS.	
4001	WSA=ABS(P(K))*1.E+4	
	WS=Z(148)+Z(149)*(WSA**Z(150))	
	WS=WS*1.E-3	
3205	WS=WS/SIG	CDT 1500
3210	IF(UVMAX-WS)3215,3220,3220	CDT 1510
3215	N10=I	CDT 1520
	N11=J	CDT 1530
	UVMAX=WS	CDT 1540
3220	IF(NMAX)1,1,2	CDT 1550
1	CONTINUE	CDT 1560
3	WS=ABS(U(K))/TAU(I)*X(I)/.5*PIDY	CDT 1570
	GO TO 3225	CDT 1580
2	WS=ABS(U(K))/DX(I)	CDT 1590
3225	IF(UVMAX-WS)3230,3235,3235	CDT 1600
3230	UVMAX=WS	CDT 1610
	N10=I	CDT 1620
	N11=J	CDT 1630
3235	WS=ABS(V(K))/DY(J)	CDT 1640
3240	IF(UVMAX-WS)3245,3250,3250	CDT 1650
3245	N10=I	CDT 1660
	N11=J	CDT 1670
	UVMAX=WS	CDT 1680
3250	CONTINUE	CDT 1690
3255	K=K+IMAX	CDT 1700
C	SAME TIME CONTROL OPTIONS THAT	

C EXIST IN OIL
 C N10 AND N11 CONTAIN THE COLUMN
 C AND ROW NO. OF THE CELL THAT IS
 C CONTROLLING DT.

3260 IF(CABL N)90,91,3300
 : 90 IF(Z(105)-Z(139))7001,7000,7000
 7000 Z(105)=1.

GO TO 7002

7001 Z(105)=Z(105)*Z(106)

7002 DT=.5/VEL/UVMAX*Z(139)*Z(105)

GO TO 3295

91 WS=UVMAX*DT

WSA=0.5/VEL

3265 IF(FFA-WSA)3276,3276,3270

3270 FFA=WSA

3276 IF(WS-FFA)3285,3300,3280

3280 DT=DT/WS*FFB/0.9

GO TO 3295

3285 IF(WS-FFB)3290,3290,3300

3290 DT=DT*FFA/WS*0.9

3295 KDT=0

3300 T=T+DTNA

85 IF(DTRAD)9911,80,81

: 80 NR=NRM

84 WS=NR

TRAD=DT/WS

GO TO 82

81 IWS=DT/DTRAD

NR=IWS+1

83 IF(NR-NRM)84,84,80

82 NC=NC+1

CYCLE=NC

NPC=NPC+1

3305 IF(T)9909,3320,3310

3310 IF(KDT)9910,3315,3320

3315 WRITE (6,8000)T,DTNA,DT

3320 DTNA=DT

GO TO 3325

9901 NK=3020

GO TO 9999

9909 NK=3305

GO TO 9999

9910 NK=3310

GO TO 9999

9911 NK=85

9999 NR=2

C *****

CALL DUMP

3325 RETURN

80000FORMAT (17H)CHANGE DT ... T=1PE9.3,11H DT(N)=1PE9.3,13H

1(N+1)=1PE9.3)

END

C

C

C

SUBROUTINE PH1

C

***** A 2 MATERIAL OIL CODE *****

CDT 1730
 CDT 1740
 CDT 1750
 CDT 1760
 CDT 1770
 CDT 1780
 CDT 1790
 CDT 1800
 CDT 1810
 CDT 1820
 CDT 1830
 CDT 1840
 CDT 1850
 CDT 1860
 CDT 1870
 CDT 1880
 CDT 1890
 CDT 1900
 CDT 1910
 CDT 1920
 CDT 1930
 CDT 1940
 CDT 1950
 CDT 1960
 CDT 1970
 CDT 1980
 CDT 1990
 CDT 2000
 CDT 2020
 CDT 2030
 CDT 2040
 CDT 2050
 CDT 2060
 CDT 2070
 CDT 2080
 CDT 2090
 CDT 2100
 CDT 2120
 CDT 2130
 DTCDT 2140
 CDT 2150
 CDT 2160

PH1 0030

C		PH1 0990
C	SINCE WE INITIALIZE THE BOUNDARY CONDITION	
C	AT THE LEFT FOR THE FIRST COLUMN, AND THE	
C	BOTTOM BOUNDARY CONDITION OF THE FIRST CELL,	
C	WE NEED ONLY CONCERN OURSELVES WITH	
C	CALCULATING QUANTITIES AT THE TOP AND RIGHT	
C	OF EACH CELL, SINCE THE LEFT AND BOTTOM	
C	HAVE ALREADY BEEN CALCULATED.	
C	***** STANDARD PH.1 VELOCITIES AT CENTER OF CELL *****	PH1 1010
C	INTEGRATION OF VELOCITIES AND INTERNAL ENERGIES	PH1 1020
C	REQUIRING 2 PASSES	PH1 1030
	ETH1=0.	
	NRT=0	
	NRC=0	
8000	VEL=1.0	PH1 1090
C	INITIALIZE MID-POINTS OF FIRST AND SECOND CELL	
C	IN THE R DIRECTION.	
3301	RC=DX(1)/2.0	PH1 1100
	RR=(X(1)+X(2))/2.0	PH1 1110
C	AXIS OF SYMMETRY BOUNDARY CONDITIONS	
C		
3304	K=2	PH1 1120
	DO 3302 J=1,JMAX	PH1 1130
	PL(J)=P(K)	PH1 1140
	UL(J)=0.0	PH1 1150
3302	K=K+IMAX	PH1 1160
C	FIRST PASS, CALCULATE U AND V TILDA, AND	
C	THE WORK TERMS USING PRE-PHI VELOCITIES.	
C	SECOND PASS, CALCULATE ONLY THE WORK TERMS	
C	USING THE NEW VELOCITIES (U AND V TILDA)	
	DO 3360 I=1,I1	
8	K=I+1	
	IF(CVIS)7002,7003,7003	PH1 1240
C	BOTTOM BOUNDARY IS TRANSMITTIVE.	
7002	VBL0=V(K)	PH1 1250
	PBL0=0.0	PH1 1260
	GO TO 7004	PH1 1270
C	BOTTOM BOUNDARY IS REFLECTIVE.	
7003	VBL0=0.0	PH1 1280
	PBL0=P(K)	PH1 1290
7004	TAUDTS=TAU(I)*DT	PH1 1300
4	DO 3348 J=1,I2	
	PIDTS=1.0/(PIDY*DT*DY(J))	PH1 1360
C	K IS INDEX OF CELL IN QUESTION.	
C	N IS INDEX OF CELL ABOVE.	
	N=K+IMAX	PH1 1370
	IF(VEL)3305,3305,3303	PH1 1380
3303	CONTINUE	
3305	IF(AMD(K)+AMX(K))9902,3340,3306	PH1 1400
3306	IF(IMAX-I)9903,3311,3310	PH1 1410
3310	IF(AMD(K+1)+AMX(K+1))9904,3312,3314	PH1 1420
C	WE ARE AT THE RIGHT BOUNDARY OF THE GRID,	
C	SET PRESSURE GRADIENT TO. ZERO, AND MODIFY	
C	ETH.	
3311	PRR=PL(J)	PH1 1430
3307	ETH=ETH-PRR*U(K)/PIDTS*RC	PH1 1440
	GO TO 3313	PH1 1450

```

C   THE BOUNDARY CONDITION FOR A EMPTY CELL ON
C   THE RIGHT, THE PRESSURE AT THE RIGHT
C   INTERFACE=0. AND THE VELOCITY IS THAT
C   OF THE CELL CENTER.
3312 PRR=0.0                                PH1 1460
:3313 URR=RC*U(K)                            PH1 1470
      GO TO 3316                             PH1 1480
C   CALCULATE PRESSURE AND RU AT INTERFACE I
3314 PRR=(P(K)+P(K+1))/2.0                  PH1 1490
3315 URR=(U(K)*RC+U(K+1)*RR)/2.0           PH1 1500
3316 IF(JMAX-J)9905,3318,3320               PH1 1510
C   SET PRESSURE GRADIENT TO. ZERO, FOR TOP OF
C   GRID, AND MODIFY ETH.
3318 PABOVE=PBLO                            PH1 1520
3319 ETH=ETH-PABOVE*V(K)/2.0*TAUDTS        PH1 1530
      GO TO 3323                             PH1 1540
3320 IF(AMD(N)+AMX(N))9906,3322,3324        PH1 1550
C   CELL ABOVE IS EMPTY, SET TOP BOUNDARY
C   CONDITIONS, PRESSURE AT TOP SURFACE=0.
C   AND VELOCITY = THAT OF CELL.
3322 PABOVE=0.0                             PH1 1560
3323 VABOVE=V(K)                            PH1 1570
      GO TO 3328                             PH1 1580
C   CALCULATE PRESSURE AT INTERFACE (J)
3324 PABOVE=(P(K)+P(N))/2.0                 PH1 1590
      IF(CVIS)7001,3325,3325                 PH1 1600
7001 IF(1-J)3325,7000,9905
C   BOTTOM BOUNDARY CONDITION OF GRID IS REFLECTIVE,
C   AND WE HAVE ALREADY SET THE CONDITIONS.
C
C   BOTTOM BOUNDARY OF GRID IS TRANSMITTIVE, SET
C   PRESSURE GRADIENT TO ZERO AND MODIFY ETH.
7000 PBLO=PABOVE                           PH1 1630
      ETH=ETH+PBLO*V(K)/2.0*TAUDTS          PH1 1640
C   CALCULATE VELOCITY AT INTERFACE (J)
3325 VABOVE=(V(K)+V(N))/2.0                 PH1 1650
3328 IF(VEL)9907,3404,3400                  PH1 1660
C   CALCULATE THE U AND V TILDA QUANTITIES
3400 V(K)=V(K)+(PBLO-PABOVE)*TAUDTS/(AMD(K)+AMX(K)) PH1 1670
3402 U(K)=U(K)+(PL(J)-PRR)/(AMX(K)+AMD(K))*RC/PIDTS*2.0 PH1 1710
C   CHECK FOR ADVANCING ACTIVE GRID COUNTERS IN
C   THE R DIRECTION.
3404 IF(I-I1)6016,6005,6005
6005 IF(U(K))6605,6606,6605
6605 NRC=1
6606 IF(V(K))6607,6004,6607
6607 NRC=1
6004 IF(AIX(K)+AID(K))6015,6016,6015
6015 NRC=1
6016 CONTINUE
6044 WS=(VBLO-VABOVE)*TAUDTS/2.0*P(K)
C   CALCULATE THE CHANGE IN INTERNAL ENERGY
C   DUE TO PRESSURE FORCES ONLY.
      DE=WS+(UL(J)-URR)/PIDTS*P(K)
3405 CONTINUE
3331 IF(AMD(K))9908,3332,3334
3332 AIX(K)=AIX(K)+DE/AMX(K)

```

PH1 1770

GO TO 3342	
3334 IF (AMX(K)) 9909, 3336, 3338	PH1 1800
3336 AID(K)=AID(K)+DE/AMD(K)	
GO TO 3342	
C CONVERT TO SPECIFIC INTERNAL	
C ENERGY FOR EACH MATERIAL.	
3338 FS=AMX(K)/DKE(K)+AMD(K)/(1.-DKE(K))	
WSD=DE/(1.-DKE(K))/FS+AID(K)	
WSX=DE/DKE(K)/FS+AIX(K)	
1022 AIX(K)=WSX	PH1 2220
AID(K)=WSD	PH1 2230
GO TO 3342	PH1 2240
C CAME HERE BECAUSE CELL IN QUESTION IS EMPTY.	
C SET INTERFACE QUANTITIES ASSUMING CELL ABOVE	
C AND TO THE RIGHT ARE NOT VOID.	
3340 PRR=0.0	PH1 2250
URR=U(K+1)*RR	PH1 2260
PABOVE=0.0	PH1 2270
VABOVE=V(N)	PH1 2280
C SET RIGHT QUANTITIES TO THE LEFT (FOR	
C NEXT COLUMN SWEEP) AND ABOVE QUANTITIES	
C TO BELOW FOR NEXT CELL ABOVE.	
3342 VBLO=VABOVE	PH1 2290
PL(J)=PRR	PH1 2300
UL(J)=URR	PH1 2310
K=N	PH1 2320
3348 PBLO=PABOVE	PH1 2330
C CHECK FOR ADVANCING ACTIVE GRID COUNTERS	
C IN (J) DIRECTION.	
LL=K-IMAX	
IF(U(LL)) 6000, 6001, 6000	
6000 NRT=1	
6001 IF(V(LL)) 6002, 6003, 6002	
6002 NRT=1	
6003 IF(AIX(LL)+AID(LL)) 6017, 6018, 6017	
6017 NRT=1	
6018 CONTINUE	
3355 RC=RR	PH1 2340
RR=(X(I+1)+X(I+2))/2.0	PH1 2350
3360 CONTINUE	PH1 2360
3361 IF(VEL) 9911, 7030, 3363	
3363 VEL=0.0	PH1 2380
C RECYCLE FOR SECOND PASS.	
GO TO 3301	PH1 2390
C ERROR	PH1 2400
9902 NK=3305	PH1 2410
GO TO 9999	PH1 2420
9903 NK=3306	PH1 2430
GO TO 9999	PH1 2440
9904 NK=3310	PH1 2450
GO TO 9999	PH1 2460
9905 NK=3316	PH1 2470
GO TO 9999	PH1 2480
9906 NK=3320	PH1 2490
GO TO 9999	PH1 2500
9907 NK=3328	PH1 2510
GO TO 9999	PH1 2520

```

9908 NK=3331
GO TO 9999
9909 NK=3334
GO TO 9999
9911 NK=3361
9999 NR=3
C *****
CALL DUMP
C SET VELOCITIES AND INTERNAL ENERGY
C (FOR SMALL VALUES) TO ZERO.
C ALSO NEGATIVE INTERNAL ENERGIES
7030 DO 10 K=2,KMAX
IF (ABS(V(K))-Z(146)) 3401,3401,11
3401 ETH1=ETH1+(AMX(K)+AMD(K))/2.*(V(K)**2)
V(K)=0.
11 IF (ABS(U(K))-Z(146)) 3403,3403,12
3403 ETH1=ETH1+(AMX(K)+AMD(K))/2.*(U(K)**2)
U(K)=0.
12 CONTINUE
IF (ABS(AIX(K))-Z(145)) 8002,8002,8003
8002 ETH1=ETH1+AMX(K)*AIX(K)
AIX(K)=0.
8003 IF (ABS(AID(K))-Z(145)) 8004,8004,8005
8004 ETH1=ETH1+AMD(K)*AID(K)
AID(K)=0.
8005 CONTINUE
10 CONTINUE
C INCREASE ACTIVE GRID COUNTERS.
I1=I1+NRC
I2=I2+NRT
ETH=ETH-ETH1
IF (I1-IMAX) 6100,6100,6200
6200 I1=IMAX
6100 IF (I2-JMAX) 6201,6201,6202
6202 I2=JMAX
6201 RETURN
END

```

PH1 2530
PH1 2540
PH1 2550
PH1 2560
PH1 2570
PH1 2580
PH1 2590
PH1 2610

PH1 2770

```

C
C -----
C
SUBROUTINE PH2
***** A 2 MATERIAL OIL CODE *****
C
C FOR X MATERIAL
C AMPY = MASS AT TOP
C AMUT = RADIAL MOMENTUM
C AMVT = AXIAL MOMENTUM
C DELET = SPECIFIC ENERGY
C
C FOR DOT MATERIAL
C TOM = MASS AT TOP
C TOXM = RADIAL MOMENTUM
C TOYM = AXIAL MOMENTUM
C TOTE = SPECIFIC ENERGY
C
C FOR X MATERIAL
C AMXP = MASS AT RIGHT

```

PH2 0740

```

C      AMUR = RADIAL MOMENTUM
C      AMVR = AXIAL MOMENTUM
C      DELER = SPECIFIC ENERGY
C
C      FOR DOT MATERIAL
C      RDM = MASS AT RIGHT
C      RDXM = RADIAL MOMENTUM
C      RDYM = AXIAL MOMENTUM
C      ROTE = SPECIFIC ENERGY
C
C      FOR X MATERIAL
C      AMMY = MASS AT BOTTOM
C      AMMU = RADIAL MOMENTUM
C      AMMV = AXIAL MOMENTUM
C      DELEB = SPECIFIC ENERGY
C
C      FOR DOT MATERIAL
C      BDM = MASS AT BOTTOM
C      BOXM = RADIAL MOMENTUM
C      BDYM = AXIAL MOMENTUM
C      BOTE = SPECIFIC ENERGY
C
C      FOR X MATERIAL
C      GAMC = MASS AT THE LEFT
C      FLEFT = RADIAL MOMENTUM
C      YAMC = AXIAL MOMENTUM
C      SIGC = SPECIFIC ENERGY
C
C      FOR DOT MATERIAL
C      DMASL = MASS AT THE LEFT
C      DXML = RADIAL MOMENTUM
C      DYML = AXIAL MOMENTUM
C      DENRG = SPECIFIC ENERGY
C      ETH1=0.
C      RECYC=0.
C      NRT=0
C      NRC=0
C      REZ=0.0
C      CALL SLITE (0)
C      PIDTS=1.0/(PIDY*DT)
C      SET BOUNDARY CONDITIONS FOR THE AXIS OF SYMMETRY
101 DO 103 J=1,JMAX
102 GAMC(J)=0.0
    FLEFT(J)=0.0
    YAMC(J)=0.0
    SIGC(J)=0.0
    DMASL(J)=0.
    DXML(J)=0.
    DYML(J)=0.
    DENRG(J)=0.
103 CONTINUE
C      BEGIN DO LOOP ON I
104 DO 547 I=1,I1
    J=1
105 K=I+1
    80 IF(AMX(K)+AMD(K))9900,97,81
    81 IF(V(K))82,97,97

```

PH2 1020
PH2 1030
PH2 1040
PH2 1050
PH2 1060

PH2 1070
PH2 1080
PH2 1090
PH2 1100
PH2 1110

PH2 1120

PH2 1130
PH2 1140
PH2 1150

PH2 1170

PH2 1180

```

C      NO MASS FLUX
97  AMMV=0.0
    BDYM=0.
    GO TO 98
C      CHECK BOTTOM BOUNDARY OF GRID
82  IF(AMX(K))9900,2,3
C      DOT ONLY
    2  ND=1
    GO TO 6
    3  IF(AMD(K))9900,4,5
C      X ONLY
    4  ND=0
    GO TO 6
C      MIXED CELL
    5  ND=-1
C      MASS OUT OF THE BOTTOM OF THE GRID
    6  WS=(AMX(K)+AMD(K))*V(K)/DY(J)*DT
C      CHECK FOR MORE THAN EMPTYING THE CELL
    7  IF(WS+AMX(K)+AMD(K))8,9,9
    8  AMMY=-AMX(K)
    BDM=-AMD(K)
    GO TO 85
C      THE RESPECTIVE FLUXES ARE PROPORTIONATE TO THE MASSES
    9  WSA=AMX(K)+AMD(K)
    AMMY=WS*AMX(K)/WSA
    BDM=WS*AMD(K)/WSA
85  IF(CVIS)106,99,99
C      BOTTOM BOUNDARY IS TRANSMITTIVE.
106 WS=(U(K)**2+V(K)**2)/2.
    IF(ND)11,10,11
C      X MATERIAL ONLY
    10 AMMU=AMMY*U(K)
    AMMV=AMMY*V(K)
    DELEB=AIX(K)+WS
    ETH=ETH+AMMY*DELEB
    GO TO 107
C      DOT FOR SURE AND PERHAPS X ALSO
    11 BDXM=BDM*U(K)
    BDYM=BDM*V(K)
    BDTE=AID(K)+WS
    ETH=ETH+BDM*BDTE
    IF(ND)10,107,107
C      BOTTOM BOUNDARY IS REFLECTIVE
    99 AMMV=2.*AMMY*V(K)
    BDYM=2.*BDM*V(K)
    98 AMMY=0.
    BDM=0.
    AMMU=0.
    BDXM=0.
    DELEB=0.
    BDTE=0.
C      BEGIN DO LOOP IN THE J DIRECTION
107 DO 546 J=1,I2
108 L=K+1,MAX
    VEL=0.0
    FS=0.0
C      BEGIN CALCULATION OF VABOVE.

```

PH2 1340

PH2 1350

PH2 1390

PH2 1400

210 IF(JMAX-J)211,211,207	PH2 1410
C AT TOP BOUNDARY OF GRID.	
211 VEL=1.0	PH2 1420
GO TO 208	PH2 1430
207 IF(AMX(L)+AMD(L))215,215,214	
214 IF(AMX(K)+AMD(K))216,216,209	
216 VABOVE=V(L)	PH2 1460
GO TO 212	PH2 1470
215 IF(AMX(K)+AMD(K))205,205,208	
205 VABOVE=0.0	PH2 1490
GO TO 212	PH2 1500
208 VABOVE=V(K)	PH2 1510
GO TO 212	PH2 1520
209 VABOVE=(V(K)+V(L))/2.0	PH2 1530
212 CONTINUE	PH2 1540
C NOW WE HAVE VABOVE	
C BEGIN CALCULATION OF URIGHT	
404 IF(IMAX-I)412,412,405	PH2 1580
405 IF(AMX(K+1)+AMD(K+1))411,411,409	
409 IF(AMX(K)+AMD(K))410,410,407	
410 URR=U(K+1)	PH2 1610
GO TO 408	PH2 1620
411 IF(AMX(K)+AMD(K))403,403,406	
403 URR=0.0	PH2 1640
GO TO 408	PH2 1650
412 FS=1.0	PH2 1660
406 URR=U(K)	PH2 1670
GO TO 408	PH2 1680
407 URR=(U(K)+U(K+1))/2.0	PH2 1690
408 CONTINUE	PH2 1700
109 CONTINUE	
C CHECK HERE FOR EMPTYING THE PROJECTILE FOR	
C IMPACT PROBLEMS	
C Z(112) = INITIAL AXIAL VELOCITY.	
C Z(113) = EPSILONICS, LIKE .05	
301 IF(VABOVE)300,304,302	PH2 1720
302 IF(AMX(K)+AMD(K))9900,304,8800	
8800 IF(J-1)9900,303,8801	PH2 1740
8801 KP=K-IMAX	PH2 1750
IF(AMX(KP)+AMD(KP))9900,8803,303	
8803 IF(ABS(VABOVE-Z(112))/Z(112)-Z(113))306,303,303	PH2 1770
303 M=K	PH2 1780
JJ=J	PH2 1790
GO TO 307	PH2 1800
304 AMPY=0.0	PH2 1810
TOM=0.	
308 AMUT=0.0	PH2 1820
TDXM=0.	
AMVT=0.0	PH2 1830
TDYM=0.	
DELET=0.0	PH2 1840
TDTL=0.	
GO TO 501	PH2 1850
300 IF(VEL)9901,305,304	PH2 1860
305 IF(AMX(L)+AMD(L))9903,304,306	
306 M=L	PH2 1880
JJ=J+1	PH2 1890

```

307 IF(VEL)6130,6130,6140
6130 WSA=(V(K)+V(L))/2.0
      WSB=1.0+(V(L)-V(K))/(DY(JJ)*SBOUND)*DT
      VABOVE=WSA/WSB
C     HERE WE HAVE CALCULATED THE MASS FLUX AT THE TOP,
C     THIS MAY CONSIST OF BOTH X AND DOT
C     MATERIAL, NO DISTINCTION YET.
6140 AMPY=(AMX(M)+AMD(M))*VABOVE/DY(JJ)*DT
501 IF(URR)500,504,502
502 IF(AMX(K)+AMD(K))9900,504,503
503 M=K
      N=I
      GO TO 508
504 AMMP=0.0
      AMUR=0.0
      AMVR=0.0
      DELER=0.0
      RDM=0.
      RDXM=0.
      RDYM=0.
      RDTE=0.
      GO TO 9500
500 IF(FS)9905,506,504
506 IF(AMX(K+1)+AMD(K+1))9904,504,507
507 M=K+1
      N=I+1
508 IF(FS)6100,6100,6110
6100 WSA=(U(K)+U(K+1))/2.
      WSB=1.0+(U(K+1)-U(K))/(DX(N)*SBOUND)*DT
      URR=WSA/WSB
6110 DEN=(AMX(M)+AMD(M))/TAU(N)
C     HERE WE HAVE CALCULATED THE MASS FLUX AT THE
C     RIGHT, THIS MAY CONSIST OF BOTH X AND DOT
C     MATERIAL, NO DISTINCTION MADE YET
      AMMP=DEN/PIOTS*X(I)/.5*URR
C
C
C     BEGIN HERE TO CALCULATE THE FLUX FOR
C     EACH MATERIAL, THE FIRST PASS THROUGH,
C     WE CALCULATE THE FLUX FOR BOTH
C     MATERIALS (IF NECESSARY) AT THE TOP, THE
C     SECOND PASS IS FOR THE RIGHT SIDE.
C
C     NOTE, THE RULES TO FOLLOW FOR
C     TRANSPORTING OF BOTH MATERIALS ARE
C     EXPLAINED IN DETAIL IN THIS REPORT
C
9500 IF(AMPY)13,26,4600
4600 IF(AMX(L)+AMD(L))9900,170,14
170 IF(AMX(K))171,171,172
172 IF(AMD(K))173,173,722
171 CONTINUE
174 TDM=AMPY
      GO TO 733
173 GO TO 26
13 IF(AMX(K)+AMD(K))9900,190,14
190 IF(AMX(L))191,191,192

```

PH2 1900

PH2 1920

PH2 1930

PH2 1950

PH2 1970

PH2 1980

PH2 1990

PH2 2000

PH2 2010

PH2 2020

PH2 2030

PH2 2050

PH2 2070

PH2 2080

PH2 2090

PH2 2110

PH2 2120

PH2 2140

```

192 IF(AMD(L))193,193,23
191 IF(AMD(L)+AMPY)195,194,194
. 194 TDM=AMPY
    GO TO 733
195 GO TO 733
: 193 IF(AMX(L)+AMPY)197,26,26
197 GO TO 26
    14 IF(AMX(L))9900,16,18
    16 ND=1
        GO TO 17
    18 IF(AMD(L))9900,19,20
    19 ND=0
        GO TO 17
    20 ND=-1
    17 IF(AMX(K))9900,73,75
    73 NX=1
        GO TO 720
    75 IF(AMD(K))9900,76,51
    76 NX=0
        GO TO 720
    51 NX=-1
720 IF(AMPY)22,26,726
    22 IF(NX)24,25,25
: 25 IF(ND)15,193,28
    26 TDM=0.
        GO TO 27
. 24 IF(ND)23,193,28
    23 IF(AMX(K)+AMD(K)) 975,975,976
975 KK=L
    GO TO 977
976 KK=K
977 WS=AMX(KK)+AMD(KK)
    WSA=AMPY
    AMPY=WSA*AMX(KK)/WS
    TDM=WSA*AMD(KK)/WS
252 CONTINUE
    GO TO 27
    28 TDM=AMPY
        IF(TDM+AMD(L)) 732,732,733
732     TDM = -AMD(L)
733 AMPY=0.
    GO TO 27
    15 IF(NX)9900,29,31
    29 IF(AMX(L)+AMPY)32,26,26
    32 WS=AMX(L)+AMPY
        AMPY=-AMX(L)
        IF(AMX(L))9401,9401,33
9401 AMPY=0.
    33 IF(WS+AMD(L))35,34,34
    34 TDM=WS
        GO TO 27
. 35 TDM=-AMD(L)
    GO TO 27
    31 IF(AMD(L)+AMPY)721,9940,9940
-9940 TDM=AMPY
    AMPY=0.
    GO TO 27

```

```

721 TDM=-AMD(L)
36 WS=AMD(L)+AMPY
37 IF(WS+AMX(L)) 39,38,38
38 AMPY=WS
GO TO 27
39 AMPY=-AMX(L)
GO TO 27
722 IF(AMX(L)+AMD(L)) 972,972,970
972 KK=K
GO TO 971
970 KK=L
971 WS=AMX(KK)+AMD(KK)
WSA=AMPY
AMPY=WSA*AMX(KK)/WS
TDM=WSA*AMD(KK)/WS
182 CONTINUE
9971 CONTINUE
GO TO 27
726 IF(ND) 41,40,40
40 IF(NX) 42,730,171
730 CONTINUE
GO TO 26
41 IF(NX) 722,173,171
42 IF(ND) 9900,60,724
60 IF(AMX(K)-AMPY) 723,26,26
723 WS=AMX(K)-AMPY
AMPY=AMX(K)
43 IF(WS+AMD(K)) 46,45,45
45 TDM=-WS
GO TO 27
46 TDM=AMD(K)
GO TO 27
724 IF(AMD(K)-AMPY) 70,128,128
128 TDM=AMPY
AMPY=0.
GO TO 27
70 TDM=AMD(K)
WS=AMD(K)-AMPY
47 IF(WS+AMX(K)) 49,48,48
48 AMPY=-WS
GO TO 27
49 AMPY=AMX(K)
GO TO 27
27 CONTINUE
IF(RECYC) 152,150,152
150 RECYC=1.
SAVE1=AMPY
SAVE2=TDM
ISAVE1=K
ISAVE2=L
9403 L=K+1
9601 AMPY=AMMP
TDM=0.
GO TO 9500
152 AMMP=AMPY
ROM=TDM
AMPY=SAVE1

```

TDM=SAVE2
K=ISAVE1
L=ISAVE2
RECYC=0.

C
C NOW, WE HAVE THE 4 POSSIBLE FLUXES,
C NOW BEGIN CHECKING FOR PREFERENTIAL
C MASS MOVEMENT BECAUSE OF CHOICE OF
C INDEXING.
C THE LOGIC OF CHECKING INVOLVES LOOKING
C AHEAD IN THE J AND I DIRECTIONS, THE
C PROGRAM CONTINUES FOR THE NEXT 5
C PAGES OR SO UP TO STATEMENT NO. 5500
C

3006 IF(GAMC(J)+DMASL(J))3007,3002,3002
3007 WS=AMX(K)+GAMC(J)
WSA=AMD(K)+DMASL(J)
GO TO 3008
3002 WS= AMX(K)
WSA=AMD(K)
3008 IF(AMMY+BDM)3009,3010,3010
3009 WS=WS+AMMY
WSA=WSA+BDM
3010 IF(AMPY+TDM)3012,3013,3011
3013 TF=0.
GO TO 3014
3012 TF=-1.
GO TO 3014
3011 TF=1.
3014 IF(AMMP+RDM)3017,3016,3015
3016 TR=0.
GO TO 3018
3017 TR=-1.
GO TO 3018
3015 TR=1.0
3018 IF(TF)3030,3019,3019
3019 IF(TR)3025,3025,3020
3020 IF(WS-AMPY-AMMP)3021,3022,3022
3021 WSS=AMPY+AMMP
AMPY=AMPY/WSS*WS
AMMP=AMMP/WSS*WS
3022 IF(WSA-TDM-RDM)3023,3024,3024
3023 WSS=TDM+RDM
TDM=TDM/WSS*WSA
RDM=RDM/WSS*WSA
GO TO 3024
3025 IF(WS-AMPY)3026,3027,3027
3026 AMPY=WS
3027 IF(WSA-TDM)3028,3100,3100
3028 TDM=WSA
GO TO 3100
3030 IF(TR)3100,3100,3040
3040 IF(WS-AMMP)3050,3051,3051
3050 AMMP=WS
3051 IF(WSA-RDM)3052,3100,3100
3052 RDM=WSA
3100 IF(TF)3159,3200,3200

```

3159 IF(VEL)3101,3101,3200
3101 IF(FS)3103,3103,3102
3102 FRA=U(L)
      GO TO 3104
3103 FRA=(U(L)+U(L+1))/2.
3104 IF(FRA)3158,3158,3105
3158 FRA=0.
      GO TO 3150
3105 FRA=FRA*(AMX(L)+AMD(L))/(TAU(I)*DY(J))*2.*PIDY
      1*X(I)*DY(J+1)*DT
3150 IF(J+1-JMAX)3152,3151,3152
3151 FTA=V(L)
      KA=L
      GO TO 3154
3152 KA=L+IMAX
3153 FTA=(V(L)+V(KA))*0.5
3154 IF(FTA)3155,3155,3157
3155 FTA=0.
      GO TO 3156
3157 FTA=FTA*(AMX(L)+AMD(L))*DT/DY(J)
3156 IF(DKE(L))3166,3200,3166
3166 GO TO 3500
3500 IF(GAMC(J+1)+DMASL(J+1))3502,3501,3501
3501 WS1=AMX(L)
      WS2=AMD(L)
      GO TO 3503
3502 WS1=AMX(L)-GAMC(J+1)
      WS2=AMD(L)-DMASL(J+1)
3503 IF(AMPY+TDM)3504,3162,3162
3162 WS=0.
      WSA=0.
      GO TO 3505
3504 WS=AMPY
      WSA=TDM
3505 IF(FTA)3700,3700,3506
3506 IF(DKE(L))3900,3508,3508
3900 IF(DKE(L)+1.)3507,3509,3509
3507 WS=WS-FTA
      GO TO 3700
3509 WSA=WSA-FTA
      GO TO 3700
3508 IF(DKE(KA))3901,3901,3510
3901 IF(DKE(L)+1.0)3515,3511,3511
3510 FTAX=AMX(KA)/(AMX(KA)+AMD(KA))*FTA
      FTAD=AMD(KA)/(AMX(KA)+AMD(KA))*FTA
      GO TO 3520
3511 FTAD=FTA
3512 IF(AMD(L)-FTAD)3514,3513,3513
3513 FTAX=0.
      GO TO 3520
3514 FTAX=FTAD-AMD(L)
      FTAD=AMD(L)
      GO TO 3520
3515 FTAX=FTA
3516 IF(AMX(L)-FTAX)3517,3518,3518
3518 FTAD=0.
      GO TO 3520

```

```

FTAD=FTAX-AMX(L)
FTAX=AMX(L)
GO TO 3520
WS=WS-FTAX
WSA=WSA-FTAD
GO TO 3700
IF(FRA)3800,3800,3701
IF(DKE(L))3902,3902,3708
IF(DKE(L)+1.)3702,3703,3703
WSA=WSA-FRA
GO TO 3800
WS=WS-FRA
GO TO 3800
IF(DKE(L+1))3903,3903,3710
IF(DKE(L+1)+1.)3715,3711,3711
FTAX=AMX(L+1)/(AMX(L+1)+AMD(L+1))*FRA
FTAD=AMD(L+1)/(AMX(L+1)+AMD(L+1))*FRA
WS=WS-FTAX
WSA=WSA-FTAD
GO TO 3800
FTAD=FRA
IF(AMD(L+1)-FTAD)3714,3713,3713
FTAX=0.
GO TO 3750
FTAX=FTAD-AMD(L+1)
FTAD=AMD(L+1)
GO TO 3750
FTAX=FRA
IF(AMX(L+1)-FTAX)3719,3718,3718
FTAD=0.
GO TO 3750
FTAD=FTAX-AMX(L+1)
FTAX=AMX(L+1)
GO TO 3750
IF(-WS-WS1)3802,3802,3801
AMPY=-AMPY/WS*WS1
IF(-WSA-WS2)3200,3200,3803
TDM=-TDM/WSA*WS2
IF(TR)4010,3024,3024
WS1=AMX(K+1)
WS2=AMD(K+1)
IF(J-1)3999,4001,3999
IF(I-IMAX)4002,3024,4002
FB=V(K+1)
IF(CVIS)4004,4005,4005
IF(I-IMAX)4003,3024,4003
KB=K+1-IMAX
FB=(V(K+1)+V(KB))*0.5
IF(FB)4006,4005,4005
WS=0.
WSA=0.
GO TO 4100
IF(DKE(K+1))4030,3024,4007
KB=K+1-IMAX
FB=FB*(AMX(K+1)+AMD(K+1))/DY(J)*DT
IF(DKE(KB))4014,4013,4011
KB=K+1

```

```

      GO TO 4011
4011 FBAX=AMX(KB)/(AMX(KB)+AMD(KB))*FB
      FBAD=AMD(KB)/(AMX(KB)+AMD(KB))*FB
4012 WS=WS+FBAX
      WSA=WSA+FBAD
      GO TO 4100
4014 IF(DKE(KB)+1.0)4015,4019,4019
4015 FBAX=FB
4016 IF(AMX(K+1)+FB)4018,4017,4017
4017 FBAU=0.
      GO TO 4012
4018 FBAX=-AMX(K+1)
      FBAD=AMX(K+1)+FB
      GO TO 4012
4019 FBAD=FB
4020 IF(AMD(K+1)+FB)4022,4021,4021
4021 FBAX=0.
      GO TO 4012
4022 FBAD=-AMD(K+1)
      FBAX=AMD(K+1)+FB
      GO TO 4012
4030 FB=FB*(AMX(K+1)+AMD(K+1))/DY(J)*DT
      IF(DKE(K+1)+1.0)4032,4031,4031
4032 WS=WS+FB
      GO TO 4100
4031 WSA=WSA+FB
4100 IF(I+1-IMAX)4102,4101,4102
4101 FRR=U(K+1)
      GO TO 4103
4102 FRR=(U(K+1)+U(K+2))*0.5
4103 IF(FRR)4200,4200,4104
4104 IF(DKE(K+1))4130,4200,4105
4130 FRR=FRR*(AMX(K+1)+AMD(K+1))/(TAU(I)*DY(J))*2.0*PIDY*
      1*X(I+1)*DT
4140 IF(DKE(K+1)+1.0)4141,4142,4142
4141 WS=WS-FRR
      GO TO 4200
4142 WSA=WSA-FRR
      GO TO 4200
4105 KR=K+2
4106 FRR=FRR*(AMX(K+1)+AMD(K+1))/(TAU(I)*DY(J))*2.0*PIDY*
      1X(I+1)*DT
4107 IF(DKE(KR))4110,4109,4108
4109 KR=K+1
      GO TO 4108
4108 FBAX=AMX(KR)/(AMX(KR)+AMD(KR))*FRR
      FBAD=AMD(KR)/(AMX(KR)+AMD(KR))*FRR
4150 WS=WS-FBAX
      WSA=WSA-FBAD
      GO TO 4200
4110 IF(DKE(KR)+1.0)4112,4111,4111
4111 FBAD=FRR
4116 IF(AMD(K+1)-FRR)4118,4117,4117
4117 FBAX=0.
      GO TO 4150
4118 FBAD=AMD(K+1)
      FBAX=FRR-AMD(K+1)

```

```

      GO TO 4150
4112 FBAX=FRR
4113 IF (AMX(K+1)-FRR) 4115,4114,4114
4114 FBAD=0.
      GO TO 4150
4115 FBAX=AMX(K+1)
      FBAD=FRR-AMX(K+1)
      GO TO 4150
4200 IF (VEL) 4203,4203,4201
4201 IF (FS) 4202,4202,4300
4202 FAB=V(K+1)
      GO TO 4206
4203 IF (FS) 4204,4204,4300
4204 KA=K+1+IMAX
4205 FAB=(V(KA)+V(K+1))*0.5
4206 IF (FAB) 4300,4300,4207
4207 FAB=FAB*(AMX(K+1)+AMD(K+1))/DY(J)*DT
4208 IF (DKE(K+1)) 4209,4300,4212
4212 KA=K+1+IMAX
      GO TO 4220
4209 IF (DKE(K+1)+1.0) 4210,4211,4211
4210 WS=WS-FAB
      GO TO 4300
4211 WSA=WSA-FAB
      GO TO 4300
4220 IF (DKE(KA)) 4230,4221,4222
4221 KA=K+1
      GO TO 4222
4222 FBAX=AMX(KA)/(AMX(KA)+AMD(KA))*FAB
      FBAD=AMD(KA)/(AMX(KA)+AMD(KA))*FAB
4250 WS=WS-FBAX
      WSA=WSA-FBAD
      GO TO 4300
4230 IF (DKE(KA)+1.0) 4224,4223,4223
4224 FBAX=FAB
4225 IF (AMX(K+1)-FAB) 4227,4226,4226
4226 FBAD=0.
      GO TO 4250
4227 FBAX=AMX(K+1)
      FBAD=FAB-AMX(K+1)
      GO TO 4250
4223 FBAD=FAB
4228 IF (AMD(K+1)-FAB) 4231,4229,4229
4229 FBAX=0.
      GO TO 4250
4231 FBAD=AMD(K+1)
      FBAX=FAB-AMD(K+1)
      GO TO 4250
4300 IF (-WS-WS1) 4302,4302,4301
4301 AMMP=-AMMP/WS*WS1
4302 IF (-WSA-WS2) 4304,4303,4303
4303 RDM=-RDM/WSA*WS2
3024 CONTINUE

```

```

C
C      FINIS OF ELABORATE LOOK AHEAD.
C      CHECK POSSIBLE OPTIONS TO LIMIT THE
C      MAGNITUDE OF THE FLUXES

```

```

5500 IF(AMPY) 5504,5600,5501
5501 IF(VEL) 5502,5502,5600
5502 WS=TAU(I)*DY(J+1)
      IF(AMPY/WS - Z(144)) 5503,5503,5600
5503 AMPY=0.
      GO TO 5600
5504 WS=TAU(I)*DY(J)
      IF(-AMPY/WS - Z(144)) 5503,5503,5600
5600 IF(TDM) 5604,5700,5601
5601 IF(VEL) 5602,5602,5700
5602 WS=TAU(I)*DY(J+1)
      IF(TDM/WS - Z(143)) 5603,5603,5700
5603 TDM=0.
      GO TO 5700
5604 WS=TAU(I)*DY(J)
      IF(-TDM/WS - Z(143)) 5603,5603,5700
5700 IF(AMMP) 5704,5800,5701
5701 IF(FS) 5702,5702,5800
5702 WS=TAU(I+1)*DY(J)
      IF(AMMP/WS-Z(144)) 5703,5703,5800
5703 AMMP=0.
      GO TO 5800
5704 WS=TAU(I)*DY(J)
      IF(-AMMP/WS-Z(144)) 5703,5703,5800
5800 IF(RDM) 5804,5900,5801
5801 IF(FS) 5802,5802,5900
5802 WS=TAU(I)*DY(J)
      IF(RDM/WS-Z(143)) 5803,5803,5900
5803 RDM=0.
      GO TO 5900
5804 WS=TAU(I)*DY(J)
      IF(-RDM/WS - Z(143)) 5803,5803,5900
5900 CONTINUE
900 IF(AMPY) 901,920,920
901 IF(GAMC(J+1)) 903,902,902
903 WS=AMX(L)+GAMC(J+1)
904 IF(WS + AMPY) 905,920,920
905 AMPY =-WS
      GO TO 920
902 WS=AMX(L)
      GO TO 904
920 IF(TDM) 921,930,930
921 IF(DMASL(J+1)) 923,922,922
923 WS=AMD(L)+DMASL(J+1)
924 IF(WS+TDM) 925,930,930
925 TDM=-WS
      GO TO 930
922 WS=AMD(L)
      GO TO 924
930 IF(AMMP) 931,940,940
931 IF(AMX(K+1) + AMMP) 932,940,940
932 AMMP=-AMX(K+1)
940 IF(RDM) 941,954,954
941 IF(AMD(K+1) + RDM) 942,954,954
942 RDM=-AMD(K+1)
954 CONTINUE
74 JTAG=0

```

```

309 IF(AMPY+TDM)8834,8831,8833
8833 IF(JMAX-J)9911,318,8835
8835 KP=K+IMAX
8836 IF(AMX(KP)+AMD(KP))9900,8837,318
C RULES FOR TOP FREE SURFACE WITHIN THE GRID
8837 IF((AMPY+TDM)/(TAU(I)*DY(J))-TOZONE)8838,318,318
8838 AMPY=0.0
TDM=0.
GO TO 8831
8834 IF(J-1)9911,325,8839
8839 IF(AMX(K)+AMD(K))9900,8840,325
C RULES FOR BOTTOM FREE SURFACE WITHIN THE GRID
8840 IF((-AMPY-TDM)/(TAU(I)*DY(J+1))-TOZONE)8841,325,325
8841 AMPY=0.0
TDM=0.
GO TO 8831
318 DELM=GAMC(J)+AMMY-AMPY
DELMO=DMASL(J)+BDM-TDM
322 IF(VEL)9901,324,323
323 WS=U(K)**2+V(K)**2
ETH=ETH-AMPY*(AIX(K)+WS/2.0)
ETH=ETH-TDM*(AID(K)+WS/2.0)
C A TRANSMITTIVE SURFACE AT TOP GRID BOUNDARY,
C CHECK FOR SUFFICIENT MASS TO TRIGGER REZONE.
IF((AMPY+TDM)/(TAU(I)*DY(J))-TOZONE)324,324,6900
6900 REZ=1.0
C CALCULATE THE MOMENTUM OF THESE TOP FLUXES
324 AMUT=AMPY*U(K)
AMVT=AMPY*V(K)
TDXM=TDM*U(K)
TDYM=TDM*V(K)
GO TO 326
325 CONTINUE
C CALCULATE THE MOMENTUM OF THESE TOP FLUXES
8831 AMUT=AMPY*U(L)
AMVT=AMPY*V(L)
TDXM=TDM*U(L)
TDYM=TDM*V(L)
C DELM = MASS AT LEFT + BOTTOM - TOP FOR X MATERIAL
DELM=GAMC(J)-AMPY+AMMY
C DELMO = SIMILAR FUNCTION FOR DOT MATERIAL
DELMO=DMASL(J)+BDM-TDM
326 IF(AMPY)327,328,328
327 DELET=AIX(L)+(U(L)**2+V(L)**2)/2.0
GO TO 333
328 IF(AMMY)329,330,330
329 DELET=DELEB
GO TO 333
330 IF(GAMC(J))331,332,332
331 DELET=SIGC(J)
GO TO 333
C NOW WE HAVE SPECIFIC ENERGY CARRIED BY
C THE X FLUX.
332 DELET=AIX(K)+(U(K)**2+V(K)**2)/2.0
333 IF(TDM)8810,8811,8811
8810 TOTE=AID(L)+(U(L)**2+V(L)**2)/2.
GO TO 8817

```

PH2 3400

PH2 3410

PH2 3440

PH2 3450

PH2 3460

PH2 3490

PH2 3500

PH2 3510

PH2 3520

PH2 3530

PH2 3540

PH2 3560

PH2 3570

PH2 3580

PH2 3590

PH2 3600

PH2 3610

PH2 3620

PH2 3630

PH2 3640

PH2 3650

PH2 3660

PH2 3670

PH2 3680

PH2 3690

PH2 3700

PH2 3710

PH2 3720

PH2 3730

```

8811 IF(BDM)8812,8813,8813
8812 TOTE=BOTE
      GO TO 8817
8813 IF(DMASL(J))8814,8815,8815
8814 TOTE=DENRG(J)
      GO TO 8817
C     NOW WE HAVE SPECIFIC ENERGY CARRIED BY
C     THE DOT FLUX
8815 TOTE=AID(K)+(U(K)**2+V(K)**2)/2.
C     SUM UP EACH COMPONENT OF MOMENTUM
C     FOR EACH MATERIAL, EXCEPT THE RIGHT FLUX
C     AND MOMENTA OF CELL IN QUESTION.
8817 SIGMU=FLEFT(J)+AMMU-AMUT
      SIGMUD=DXML(J)+BDM-TOXM
      SIGMV=YAMC(J)+AMMV-AMVT
      SIGMVD=DYML(J)+BDM-TDYM
C     SUM UP THE CHANGE IN ENERGY (BOTH X
C     AND .) FOR THE CELL IN QUESTION EXCEPT
C     FOR ENERGY AT THE RIGHT AND ENERGY OF THE
C     CELL IN QUESTION.
      DELEK=GAMC(J)*SIGC(J)+AMMY*DELEB-AMPY*DELET
      DELED=DMASL(J)*DENRG(J)+BDM*BOTE
      1-TDM*TOTE
509 IF(AMMP+RDM)8843,518,8844
8844 IF(IMAX-I)9911,518,8845
8845 IF(AMX(K+1)+AMD(K+1))9900,8846,518
C     RULES FOR FREE SURFACE AT THE RIGHT WITHIN
C     THE GRID.
8846 IF((AMMP+RDM)/(TAU(I)*DY(J))-TOZONE)8847,518,518
8847 AMMP=0.0
      RDM=0.
      GO TO 518
8843 IF(I-1)9911,512,8848
8848 IF(AMX(K)+AMD(K))9900,8849,512
C     RULES FOR FREE SURFACE AT THE LEFT WITHIN
C     THE GRID.
8849 IF((-AMMP-RDM)/(TAU(I+1)*DY(J))-TOZONE)8850,512,512
8850 AMMP=0.0
      RDM=0.
      GO TO 518
C     NOW DELM = CHANGE IN X MASS FOR CELL K
512 DELM=DELM-AMMP+AMX(K)
C     NOW DELMD = CHANGE IN . MASS FOR CELL K
      DELMD=DELMD-RDM+AMD(K)
513 CONTINUE
514 CONTINUE
C     CALCULATE THE MOMENTUM OF THE RIGHT FLUXES
8828 AMUR=AMMP*U(K+1)
      RDXM=RDM*U(K+1)
      AMVR=AMMP*V(K+1)
      RDYM=RDM*V(K+1)
      GO TO 525
C     NOW DELM = CHANGE IN X MASS FOR CELL K
518 DELM=DELM-AMMP+AMX(K)
C     NOW DELMD = CHANGE IN . MASS FOR CELL K
      DELMD=DELMD-RDM+AMD(K)
521 CONTINUE

```

PH2 3750

PH2 3760

PH2 3780

PH2 3810

PH2 3820

PH2 3830

PH2 3860

PH2 3870

PH2 3880

PH2 3890

PH2 3900

PH2 3910

PH2 3920

PH2 3930

PH2 3940

PH2 3950

```

522 IF(FS)9905,524,523
523 WS=U(K)**2+V(K)**2
      ETH=ETH-AMMP*(AIX(K)+WS/2.0)
      ETH=ETH-RDM*(AID(K)+WS/2.0)
C    A TRANSMITTIVE SURFACE AT RIGHT GRID BOUNDARY,
C    CHECK FOR SUFFICIENT MASS TO TRIGGER REZONE.
      IF((AMMP+RDM)/(TAU(I)*DY(J))-TOZONE)524,524,6901
6901 REZ=1.0
524 AMUR=AMMP*U(K)
      RDXM=RDM*U(K)
      AMVR=AMMP*V(K)
      RDYM=RDM*V(K)
C    NOW SUM THE NET MOMENTA CHANGES
C    BY THESE FLUXES
525 SIGMU=SIGMU-AMUR
      SIGMUD=SIGMUD-RDXM
      SIGMV=SIGMV-AMVR
      SIGMVD=SIGMVD-RDYM
526 TIC=0.0
527 IF(AMMP)528,529,529
528 DELER=AIX(K+1)+(U(K+1)**2+V(K+1)**2)/2.0
      GO TO 537
529 IF(AMMY)530,531,531
530 DELER=DELEB
      GO TO 536
531 IF(GAMC(J))532,533,533
532 DELER=SIGC(J)
      GO TO 536
533 IF(AMPY)535,535,534
534 DELER=DELET
      GO TO 536
C    NOW WE HAVE THE SPECIFIC ENERGY FOR FLUX
C    AT THE RIGHT FOR X MATERIAL
535 DELER=AIX(K)+(U(K)**2+V(K)**2)/2.0
536 TIC=1.0
C    NOW WE HAVE TOTAL CHANGE IN ENERGY BY
C    THE 4 FLUXES FOR X MATERIAL
537 DELEK=DELEK-AMMP*DELER
999 IF(RDM)700,701,701
700 RTE=AID(K+1)+(U(K+1)**2+V(K+1)**2)/2.0
      GO TO 710
701 IF(BDM)702,703,703
702 RTE=BDTE
      GO TO 710
703 IF(DMASL(J))704,705,705
704 RTE=DENRG(J)
      GO TO 710
705 IF(TDM)706,706,707
707 RTE=TDTE
      GO TO 710
C    NOW, THE SPECIFIC ENERGY FOR THE FLUX
C    AT THE RIGHT FOR DOT MATERIAL.
706 RTE=(U(K)**2+V(K)**2)/2.+AID(K)
C    NOW WE HAVE TOTAL CHANGE IN ENERGY
C    BY THE 4 FLUXES FOR DOT MATERIAL.
710 DELED=DELED-RDM*RTE

```

PH2 3960
PH2 3970
PH2 3980

PH2 4000
PH2 4010

PH2 4020

PH2 4030

PH2 4040

PH2 4050

PH2 4060

PH2 4070

PH2 4080

PH2 4090

PH2 4100

PH2 4110

PH2 4120

PH2 4130

PH2 4140

PH2 4150

PH2 4160

PH2 4170

PH2 4180

PH2 4190

PH2 4200

```

539 WS=(U(K)**2+V(K)**2)/2.0
    IF(DELM)998,712,712
998 IF(AMX(K)*1.E-6+DELM)9906,997,997
997 DELM=0.
712 IF(DELMD)713,714,714
713 IF(AMD(K)*1.E-6+DELMD)9906,715,715
715 DELMD=0.
714 CONTINUE
540 ENK=AMX(K)*(WS+AIX(K))+DELEK
    DENK=AMD(K)*(WS+AID(K))+DELED
    WSA=DELM+DELMD
    IF(WSA)543,543,541
C
C   CONSERVE MOMENTUM TO CALCULATE THE
C   RADIAL VELOCITY COMPONENT
C
541 U(K)=(SIGMU+SIGMUD+(AMX(K)+AMD(K))*U(K))/WSA
    IF(ABS(U(K))-Z(146)) 9951,9951,601
9951 ETH1=ETH1+(DELM+DELMD)/2.*(U(K)**2)
    U(K)=0.
C
C   CONSERVE MOMENTUM TO CALCULATE THE
C   AXIAL VELOCITY COMPONENT.
C
601 V(K)=(SIGMV+SIGMVD+(AMX(K)+AMD(K))*V(K))/WSA
    IF(ABS(V(K))-Z(146)) 9952,9952,9953
9952 ETH1=ETH1+(DELM+DELMD)/2.*(V(K)**2)
    V(K)=0.
C   CHECK FOR ADVANCING ACTIVE GRID COUNTER
C   IN THE RADIAL DIRECTION.
9953 IF(I-I1)603,6604,6604
6604 IF(U(K))6605,6606,6605
6605 NRC=1
6606 IF(V(K))6607,6608,6607
6607 NRC=1
6608 IF(AIX(K)+AID(K))6609,6610,6609
6609 NRC=1
6610 CONTINUE
603 WS=U(K)**2+V(K)**2
542 CONTINUE
    IF(DELM+DELMD)543,543,750
750 IF(DELM)751,751,752
751 AID(K)=DENK/DELMD-WS/2.
    DKE(K)=-1.
    GO TO 543
752 IF(DELMD)753,753,754
753 AIX(K)=ENK/DELM-WS/2.
    DKE(K)=-2.
    GO TO 543
754 CONTINUE
C
C   THE NEW INTERNAL ENERGY IS THE TOTAL
C   LESS THE KINETIC
C
DQ=ENK+DENK-.5*WS*(DELM+DELMD)
WS= ENK +DENK
C

```

PH2 4330

PH2 4340

PH2 4350

PH2 4360

PH2 4380

PH2 4390

PH2 4400

C CALCULATE THE NEW SPECIFIC INTERNAL ENERGIES
C FOR EACH MATERIAL
-C

AIX(K)=ENK/DELM/WS*DQ
AID(K)=DENK/DELM/WS*DQ
DKE(K)=1.

543 AMX(K)=DELM
AMD(K)=DELM
IF (AMX(K)+AMD(K))9900,2007,725

PH2 4420

2007 AIX(K)=0.

AID(K)=0.

DKE(K)=0.

U(K)=0.

V(K)=0.

P(K)=0.

GO TO 544

725 IF (AMX(K))9900,716,717

716 AIX(K)=0.

DKE(K)=-1.

GO TO 544

717 IF (AMD(K))9900,718,719

718 AID(K)=0.

DKE(K)=-2.

GO TO 544

719 CONTINUE

C
-C SET THE LEFT QUANTITIES WITH THOSE FROM THE
C RIGHT FOR THE NEXT COLUMN SWEEP.
C

544 GAMC(J)=AMMP
FLEFT(J)=AMUR
YAMC(J)=AMVR
SIGC(J)=DELER
DMASL(J)=RDM
DXML(J)=RDXM
DYML(J)=RDYM
DENRG(J)=RDTE

PH2 4480

PH2 4490

PH2 4500

PH2 4510

C
C SET THE BOTTOM QUANTITIES WITH THOSE
C FROM THE TOP FOR THE NEXT CELL ABOVE
C

545 AMMY=AMPY
AMMU=AMUT
AMMV=AMVT
DELEB=DELET
BDM=TDM
BDXM=TDXM
BOYM=TDYM
BDTE=TDTE

PH2 4520

PH2 4530

PH2 4540

PH2 4550

546 K=K+IMAX
LL=K-IMAX

PH2 4560

PH2 4570

C
C CHECK FOR ADVANCING THE ACTIVE GRID IN THE
C AXIAL DIRECTION.

IF (U(LL))6500,6600,6500

PH2 4580

6500 NRT=1

PH2 4590

6600 IF (V(LL))6601,6602,6601

PH2 4600

6601 NRT=1

PH2 4610

6602 IF(AIX(LL)+AID(LL))6611,547,6611	
6611 NRT=1	PH2 4630
547 CONTINUE	PH2 4640
C ADVANCE ACTIVE GRID COUNTERS	
I1=I1+NRC	PH2 4650
I2=I2+NRT	PH2 4660
IF(IMAX-I1)6700,6701,6702	PH2 4670
6700 I1=IMAX	PH2 4680
6701 CONTINUE	PH2 4690
6702 IF(JMAX-I2)6800,6801,6802	PH2 4700
6800 I2=JMAX	PH2 4710
6801 CONTINUE	PH2 4720
6802 GO TO 548	PH2 4730
9901 NK=300	PH2 4740
GO TO 9999	PH2 4750
9900 NK=302	PH2 4760
GO TO 9999	PH2 4770
9903 NK=305	PH2 4780
GO TO 9999	PH2 4790
9904 NK=506	PH2 4800
GO TO 9999	PH2 4810
9905 NK=500	PH2 4820
GO TO 9999	PH2 4830
: 9906 NK=998	
C AND STATEMENT NO. 713 ALSO	
GO TO 9999	PH2 4850
. 9911 NK=8833	PH2 4860
GO TO 9999	PH2 4870
9908 NK= 17	PH2 4880
GO TO 9999	PH2 4890
9909 NK= 22	PH2 4900
GO TO 9999	PH2 4910
9910 NK= 47	PH2 4920
GO TO 9999	PH2 4930
9907 NK=538	PH2 4940
9999 NR=4	PH2 4950
WRITE(6,9939)I,J,K,L,N,I1,I2,NK,NR	
WRITE(6,9938)AMPY,AMUT,AMVT,DELET,AMMP,AMUR,AMVR,DELER	
WRITE(6,9938)AMMY,AMMU,AMMV,DELEB,GAMC(J),FLEFT(J),YAMC(J),SIGC(J)	
WRITE(6,9938)TDM,TDXM,TDYM,TOTE,RDM,RDXM,RDYM,ROTE	
WRITE(6,9938)BDM,BDXM,BDYM,BOTE,DMASL(J),DXML(J),DYML(J),DENRG(J)	
WRITE(6,9939)NX,ND	
WRITE(6,9938)DELM,DELM,D,SIGMU,SIGMUD,SIGMV,SIGMVD,DELEK,DELED	
WRITE(6,9938)AMX(K),AIX(K),U(K),V(K),AMD(K),AID(K),P(K)	
WRITE(6,9938)AMX(L),AIX(L),U(L),V(L),AMD(L),AID(L),P(L)	
9939 FORMAT(9I6)	
9938 FORMAT(1P8E12.5)	
CALL DUMP	PH2 4960
548 SUM=0.0	PH2 4970
2005 DO 2001 I=1,I1	PH2 4980
K=I+1	PH2 4990
DO 2013 J=1,I2	
WSA=0.	
WS=TAU(I)*DY(J)	
IF(AMX(K))5952,5952,5950	
C OPTION FOR REMOVING LOW DENSITY X MASS	
5950 IF(AMX(K)/WS-Z(107))5951,5951,5952	

```

5951 WSA=(U(K)**2+V(K)**2)/2.
      Z(100)=Z(100)+AMX(K)
      WSA=AMX(K)*(AIX(K)+WSA)
      SUM=SUM+WSA
      Z(101)=Z(101)-WSA
      AMX(K)=0.
      AIX(K)=0.
5952 IF(AMD(K))5960,5960,5963
C  OPTION FOR REMOVING LOW DENSITY . MASS
5963 IF(AMD(K)/WS-Z(108))5962,5962,5960
5962 WSA=(U(K)**2+V(K)**2)/2.
      Z(100)=Z(100)+AMD(K)
      WSA=AMD(K)*(AID(K)+WSA)
      SUM=SUM+WSA
      Z(101)=Z(101)-WSA
      AMD(K)=0.
      AID(K)=0.
5960 IF(AMD(K)+AMX(K))5961,5961,2008
5961 U(K)=0.
      V(K)=0.
      P(K)=0.
      DKE(K)=0.
      GO TO 2013
C  OPTION FOR REMOVING SMALL (OR NEGATIVE)
C  INTERNAL ENERGIES FOR MATERIAL (X)
2008 IF(AIX(K)-Z(145))2004,2011,2011
2004 SUM=SUM+AIX(K)*AMX(K)
      AIX(K)=0.0
C  OPTION FOR REMOVING SMALL (OR NEGATIVE)
C  INTERNAL ENERGIES FOR MATERIAL (DOT)
2011 IF(AID(K)-Z(145))2012,2000,2000
2012 SUM=SUM+AID(K)*AMD(K)
      AID(K)=0.
2000 IF(AMX(K)+AMD(K))4400,4400,4401
4401 IF(AMX(K))4402,4402,4403
4402 DKE(K)=-1.
      GO TO 2013
4403 IF(AMD(K))2009,2009,2010
2009 DKE(K)=-2.
      GO TO 2013
2010 DKE(K)=1.
      GO TO 2013
4400 DKE(K)=0.
2013 K=K+IMAX
2001 CONTINUE
      ETH=ETH-SUM-ETH1
      Z(104)=Z(104)+SUM
C  CHECK IF REZONE FLAG HAS BEEN SET BY PH2
8000 IF(REZ)8001,8001,8002
C  CHECK IF YOU WANT TO CALL REZONE
8002 IF(REZFCT)8004,8004,8003
8004 REZ=0.
      GO TO 8001
8003 CALL REZONE
8001 RETURN
      END
C

```

PH2 5170
PH2 5180

PH2 5200

PH2 5220

PH2 5260

```

C -----
C
C SUBROUTINE ES
C
C 200 IF(AMX(K))9901,201,202
C DOT MATERIAL ONLY
201 BEL=-1.
   WS3=1.
   WSA=1.
210 WS1=AMD(K)
   WS2=AID(K)
   JJ=116
   DO 701 II=1,10
   PR(II)=Z(JJ)
   JJ=JJ+2
701 CONTINUE
   GO TO 10
202 IF(AMD(K))9901,203,204
C X MATERIAL ONLY
203 BEL=-1.
   WSA=1.
   WS3=-1.
211 CONTINUE
   WS1=AMX(K)
   WS2=AIX(K)
C HERE SET Z BLOCK DATA FOR (X) MATERIAL TO PR BLOCK
   JJ=115
   DO 700 II=1,10
   PR(II)=Z(JJ)
   JJ=JJ+2
700 CONTINUE
   GO TO 10
C MIXED CELL
C Z(115)=RHONOT FOR X
C Z(116)=RHONOT FOR (.)
204 EPSI=.5
   WSA=EPSI
   WS3=-1.
   BEL=1.
   NN=0
   GO TO 211
10 CONTINUE
   RHOW=WS1/(TAU(I)*DY(J))
   RHOW=RHOW/WSA
   ETA=RHOW/PR(1)
   VOW=1.0/ETA
11 P1=WS2*RHOW*PR(2)
12 P2=WS2
   P3=ETA*ETA*PR(3)
14 P4=PR(4)/(P2/P3+1.)*WS2*RHOW
15 P5=PR(5)*(ETA-1.)
16 IF(ETA-1.)50,100,100
50 IF(VOW-PR(6))55,55,75
55 IF(WS2-PR(7))100,100,75
75 P7=PR(8)*(VOW-1.)
   IF(P7-88.)4002,4002,4003

```

```

4003 P7=88.0
4002 CONTINUE
      P8=EXP(P7)
      P9=1./P8
      P10=PR(9)*((VOW-1.):**2)
      IF(P10-88.)4000,4000,4001
4001 P10=88.
4000 CONTINUE
      P11=EXP(P10)
      P12=1./P11
      WSC=P1+(P4+P5*P9)*P12
      GO TO 120
100 P6=PR(10)*((ETA-1.):**2)
      WSC=P1+P4+P5+P6
120 CONTINUE
119 IF(WSC)999,999,500
999 WSC=0.
      WS6X=.5
500 IF(BEL)501,502,502
C     CELL IS NOT MIXED
501 P(K)=WSC
      GO TO 600
502 IF(WS3)503,509,509
503 WS3=1.
      WSA=1.-WSA
      P(K)=WSC
      EPSI=1.-WSA
      GO TO 210
C     N1 = MAX. NO. OF CYCLES FOR ITERATION
509 IF(N1-NN)420,13,13
13 NN=NN+1
      IF(P(K))510,510,511
510 P(K)=0.
      GO TO 400
511 WS1=ABS(P(K)-WSC)
      IF(WS1/ABS(P(K))>.05)420,420,410
410 IF(P(K)-WSC)400,420,401
420 P(K)=(P(K)+WSC)/2.
      DKE(K)=EPSI
      GO TO 600
C     FEF = EPSILON TO INCREASE OR DECREASE PARTIAL
C     VOLUMES FOR THE ITERATION
401 EPSI=EPSI+FEF
      IF(EPSI>.99)710,710,711
711 EPSI=.99
710 WSA=EPSI
      WS3=-1.
      GO TO 211
400 EPSI=EPSI-FEF
      IF(EPSI<-.01)712,712,713
712 EPSI=.01
713 WSA=EPSI
      WS3=-1.
      GO TO 211
9901 NK=200
      NR=9999
      CALL DUMP

```

600 WSGX=.5
RETURN
END

SUBROUTINE EDIT

E D I T

100 IF(SWITCH)102,104,102

102 CALL SSWTCH(4,K000FX)
GO TO(122,104),K000FX

FIRST CYCLE OF THE RUN (SENSE LIGHT 3 ON)

104 CALL SLITET(3,K000FX)
GO TO(106,108),K000FX

106 CALL SLITE (3)
GO TO 126

108 IF(CYCLE-CSTOP)110,122,122

110 IF(REZ)9901,112,124

112 IF(AMOD(CYCLE,DUMPT7))114,124,114

114 IF(AMOD(CYCLE,PRINTL))116,126,116

116 IF(SWITCH)118,120,118

118 CALL SSWTCH(5,K000FX)
GO TO(128,120),K000FX

120 IF(AMOD(CYCLE,PRINTS))140,128,140

NORMAL OR FORCED STOP ON THIS CYCLE

122 CALL SLITE (1)

EXECUTE WTAPE - DUMP VARIABLES ONTO TAPE 7

124 GO TO 1

SET SENSE LIGHT TO INDICATE TAKING OF LONG PRINT

126 CALL SLITE (4)

EXECUTE SP - WRITE THE SHORT PRINT INFORMATION

128 GO TO 6000

EXECUTE PLOT - PLOT THE FILLED CELL DISTRIBUTION

130 GO TO 1000

132 CALL SLITET(4,K000FX)
GO TO(134,136),K000FX

EXECUTE LP - WRITE THE LONG PRINT INFORMATION

134 GO TO 5000

TEST FOR AN ENERGY CHECK VIOLATION

136 IF(ABS(ECK)-DMIN)140,140,9905

140 CALL SLITET(1,K000FX)
GO TO(142,144),K000FX

142 REWIND N7

CALL SLITE (1)

144 GO TO 10000

***** SUBROUTINE WTAPE *****
1 IF(DUMPT7)30,3,3
3 BACKSPACE N7
WS=555.0
WRITE(N7)WS,CYCLE,N3
WRITE(N7){Z(L),L=1,MZ}
6 WRITE(N7){U(I),V(I),AMD(I),AMX(I),AID(I),AIX(I),
1P(I),DKE(I),I=1,KMAXA)

EDIT0050

EDIT1010

EDIT1130

EDIT1140

EDIT1150

EDIT1160

EDIT1170

EDIT1180

EDIT1190

EDIT1200

EDIT1210

EDIT1220

EDIT1230

EDIT1240

EDIT1250

EDIT1260

EDIT1270

EDIT1280

EDIT1290

EDIT1300

EDIT1310

EDIT1320

EDIT1330

EDIT1340

EDIT1350

EDIT1360

EDIT1370

EDIT1380

EDIT1410

EDIT1420

EDIT1430

EDIT1440

EDIT1450

EDIT1460

EDIT1470

EDIT1480

EDIT1490

EDIT1510

EDIT1520

EDIT1530

EDIT1540

EDIT1550

EDIT1580

EDIT1620

```

7 WRITE(N7)(X(K),TAU(K),K=1,IMAX)
WRITE(N7)(Y(K),K=1,JMAX)
WS=666.0
WRITE(N7)WS,WS,WS
WRITE (6,8120)NC
30 GO TO 126
C*** END OF WTAPE SUBROUTINE *****
C
C
C*** SUBROUTINE S P *****
C      SIZE OF TABLE
6000 NK=12
      TAB(1)=0.02
      TAB(2)=0.04
      TAB(3)=0.06
      TAB(4)=0.08
      TAB(5)=0.10
      TAB(6)=0.15
      TAB(7)=0.20
      TAB(8)=0.25
      TAB(9)=0.30
      TAB(10)=0.4
      TAB(11)=0.5
      TAB(12)=1.0
6010 DO 6012 I=1,18
6012 PR(I)=0.0
      NK1=NK+2
      DO 6014 I=1,NK1
C      TEMPORARY USE PARTICLE STORAGE FOR EDITING
      AM(I)=0.
      XL(I)=0.
      YL(I)=0.
      AMK(I)=0.0
      PK(I)=0.0
6014 QK(I)=0.0
      DO 6028 K=2,KMAX
      WSB=(U(K)**2+V(K)**2)/2.0
6015 IF(AMD(K))9917,6019,6017
6017 PR(1)=AMD(K)*AID(K)+PR(1)
      PR(2)=AMD(K)*WSB+PR(2)
6018 PR(4)=AMD(K)+PR(4)
6019 IF(AMX(K)+AMD(K))9917,6028,6020
6020 I=NK1
      IF(V(K))6026,6026,6022
6022 WSA=ABS(U(K))/V(K)
      DO 6024 I=1,NK
      IF(TAB(I)-WSA)6024,6026,6026
6024 CONTINUE
      I=NK+1
6026 WS=AMX(K)
6027 AMK(I)=AMK(I)+AMX(K)+AMD(K)
      XL(I)=XL(I)+U(K)*AMD(K)
      YL(I)=YL(I)+V(K)*AMD(K)
      PK(I)=PK(I)+U(K)*AMX(K)
      QK(I)=QK(I)+V(K)*AMX(K)
      PR(5)=PR(5)+AIX(K)*AMX(K)
      PR(6)=PR(6)+WSB*AMX(K)

```

EDIT1780

EDIT1800

EDIT1810

EDIT1820

EDIT1830

EDIT1840

EDIT1850

EDIT1860

EDIT1870

EDIT1880

EDIT1890

EDIT1900

EDIT1910

EDIT1920

EDIT1930

EDIT1940

EDIT1950

EDIT1960

EDIT1970

EDIT1980

EDIT1990

EDIT2000

EDIT2010

EDIT2020

EDIT2030

EDIT2040

EDIT2050

EDIT2060

EDIT2070

EDIT2080

EDIT2090

EDIT2120

EDIT2130

EDIT2140

EDIT2150

EDIT2160

EDIT2170

EDIT2180

EDIT2190

EDIT2200

EDIT2210

EDIT2220

EDIT2230

EDIT2240

EDIT2250

EDIT2260

	PR(8)=PR(8)+AMX(K)	EDIT2270
6028	CONTINUE	EDIT2280
	PR(3)=PR(1)+PR(2)	EDIT2290
	PR(7)=PR(5)+PR(6)	EDIT2300
	XNRG=PR(7)	EDIT2310
	PR(9)=PR(1)+PR(5)	EDIT2320
	PR(10)=PR(2)+PR(6)	EDIT2330
	PR(11)=PR(3)+PR(7)	EDIT2340
	PR(12)=PR(4)+PR(8)	EDIT2350
	IF(ETH)7002,7002,7003	
7002	WSA=0.	
	GO TO 7000	
7003	CONTINUE	
	WSA=(ETH-PR(11))/ETH	EDIT2360
	IF(NPC)7000,7000,7001	
7000	NPC=1	
7001	PR(18)=(WSA-DNN)/FLOAT(NPC)	
	ECK=PR(18)	EDIT2380
	DNN=WSA	EDIT2390
	NPC=0	EDIT2400
C	*** FOR PELLET PROBLEMS ONLY ****	EDIT2410
	SUMD=0.	
	SUMX=0.	
	DO 800 I=1,13	EDIT2430
	SUMX=SUMX+QK(I)	
	SUMD=SUMD+YL(I)	
800	CONTINUE	EDIT2450
C	RADET=POSITIVE AXIAL MOMENTUM OF X	
C	VABOVE=POSITIVE AXIAL MOMENTUM OF.	
	RADET=SUMX	
	VABOVE=SUMD	
801	SUM=0.0	EDIT2470
	SUMD=0.	
	DO 820 K=2,KMAX	
	IF(AMX(K))810,810,802	EDIT2490
802	IF(U(K))810,810,803	EDIT2500
803	SUM=SUM+AMX(K)*U(K)	EDIT2510
810	IF(AMD(K))820,820,821	
821	IF(U(K))820,820,823	
823	SUMD=SUMD+AMD(K)*U(K)	
820	CONTINUE	
	RADER=SUM	EDIT2530
	VBLO=SUMD	
	PR(19)=0.0	EDIT2540
	DO 8029 I=1,NK	EDIT2550
6029	PR(I+19)=PR(I+18)+AMK(I)	EDIT2560
	PR(NK+20)=0.0	EDIT2570
	PR(NK+21)=0.0	EDIT2580
	JJ=2(147)	
	SUMX=0.	
	SUMD=0.	
	DO 811 I=1,IMAX	
	K=I+1	
	DO 813 J=1,JJ	
	IF(AMX(K))816,816,814	
814	IF(U(K))816,816,817	
817	SUM=SUM+U(K)*AMX(K)	

```

816 IF(AMD(K))813,813,818
818 IF(U(K))813,813,819
819 SUMD=SUMD+U(K)*AM(K)
813 K=K+IMAX
811 CONTINUE
PBLO=SUMX
PABOVE=SUMD
WRITE (6,8116)PRO,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
WRITE (6,8117)(PR(I),I=1,8)
WRITE (6,8118)(PR(I),I=9,12)
WRITE (6,8119)RADER,RADET,UVMAX,ETH,ECK
WRITE(6,8203)RADET,VABOVE,RADER,VBLO,PBLO,PABOVE
WRITE (6,9040)N10,N11
WRITE (6,8124)(I,AMK(I),PR(I+19),PK(I),QK(I),I=1,NK1)
6090 GO TO 130
C**** END OF S P SUB. OUTINE *****
C
C
C**** SUBROUTINE PLOT * *****
1000 GO TO 1030
1030 WRITE (6,8116)PRO,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
JMAX=JMAX
WRITE (6,8307)DX(1),DY(1),XMAX,Y1,Y2,Y(JMAX)
M=1
IF(JMAX-52)1034,1036,1036
1034 M=IABS(51-JMAX)/2
1036 DO 1040 I=1,M
WRITE (6,8308)
1040 CONTINUE
1044 J=I2
1100 K=(J-1)*IMAX+1
1105 DO 1180 I=1,I1
K=K+1
1126 PR(I)=PLOT(1)
C TEST FOR DOT PARTICLE
1148 IF(AMD(K))9917,1150,1152
C TEST FOR X PARTICLE
1150 IF(AMX(K))9917,1156,1160
C TEST FOR MIXED CELL
1152 IF(ANX(K))9917,1162,1164
C X PARTICLE ONLY
1160 PR(I)=PLOT(2)
GO TO 1180
C DOT PARTICLE ONLY
1162 PR(I)=PLOT(3)
GO TO 1180
C MIXED CELL
1164 PR(I)=PLOT(4)
GO TO 1180
1166 PR(I)=PLOT(1)
1180 CONTINUE
1200 IF(MOD(J,5))1210,1204,1210
1204 IF(DY(J)-DY(J-1))1206,1208,1206
1206 WRITE(6,8211)DY(J),J,(PR(I),I=1,I1)
GO TO 1224
1208 WRITE(6,8201)J,(PR(I),I=1,I1)
GO TO 1224

```

EDIT2590

EDIT2600

EDIT2610

EDIT2620

EDIT2630

EDIT2640

EDIT2650

EDIT2660

EDIT2670

EDIT2680

EDIT2690

EDIT2700

EDIT2710

EDIT2720

EDIT2740

EDIT2750

EDIT2760

EDIT2770

EDIT2780

EDIT2790

EDIT2810

EDIT2830

EDIT2860

EDIT2870

EDIT2880

EDIT2890

EDIT2900

EDIT2910

EDIT2920

EDIT2950

EDIT2960

EDIT2990

EDIT3000

EL 3030

EDIT3060

EDIT3070

EDIT3080

EDIT3100

EDIT3120

```

1210 IF(DY(J)-DY(J-1))1212,1214,1212                                EDIT3130
1212 WRITE(6,8222)DY(J),(PR(I),I=1,I1)
      GO TO 1224                                                        EDIT3150
1214 WRITE(6,8202)(PR(I),I=1,I1)
1224 J=J-1                                                            EDIT3170
1226 IF(J)1230,1230,1100                                              EDIT3180
1230 PR(1)=PLOT(5)
      WRITE(6,8201)J,(PR(1),I=1,I1)
      WRITE(6,8302)(I,I=0,IMAX,5)                                     EDIT3220
1240 GO TO 132                                                         EDIT3290
C*** END OF PLOT SUBROUTINE *****                                EDIT3300
C
C
C*** SUBROUTINE L P *****                                EDIT3330
5000 WRITE(6,8116)PROC,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4          EDIT3340
5004 DO 5030 I=1,I1
      CALL SLITE(4)                                                    EDIT3360
      J=I2+1
      K=I2*IMAX+I+1
      DO 5046 L=1,I2
      J=J-1                                                            EDIT3400
      K=K-IMAX                                                         EDIT3410
5012 IF(AMX(K)+AMD(K))5017,5046,5014                                  EDIT3420
5014 CALL SLITET(4,K000FX)                                             EDIT3430
      GO TO(5016,5019),000FX
5016 WRITE(6,8135)I,X(I),DX(I)                                        EDIT3450
5019 IF(AMX(K))5030,5030,5031
5030 WSB=AMD(K)/(TAU(I)*DY(J))
      WSB=WSB/Z(116)
      WSA=0.
      GO TO 5034
5031 IF(AMD(K))5032,5032,5033
5032 WSA=AMX(K)/(TAU(I)*DY(J))
      WSA=WSA/Z(115)
      WSB=0.
      GO TO 5034
5033 WSA=(AMX(K)+AMD(K))/(TAU(I)*DY(J))
      WSB=-.11111
5034 WSC=P(K)*1.E+4
C FIRST COLUMN J
C SECOND COLUMN RADIAL VELOCITY CM./SH.
C THIRD COLUMN AXIAL VELOCITY CM./SH.
C FOURTH COLUMN PRESSURE IN MEGABARS
C FIFTH COLUMN DKE(STATUS OF MIXED CELL)
C SIXTH COLUMN MASS (OF X MATERIAL) IN GRAMS.
C SEVENTH COLUMN SPECIFIC INTERNAL ENERGY FOR (X) IN JERKS/GR
C EIGHTH COLUMN MASS (OF . MATERIAL) IN GRAMS.
C NINTH COLUMN SPECIFIC INTERNAL ENERGY FOR (.) IN JERKS/GR.
C TENTH COLUMN DENSITY IN G/CC.
C ELEVENTH COLUMN WSB
C TWELFTH COLUMN Y (J) TOP COORDINATE OF CELL IN CM.
5018 WRITE(6,8108)J,U(K),V(K),WSC,THETA(K),AMX(K),AIX(K),
      LAMD(K),AID(K),WSA,WSB,Y(J)
5046 CONTINUE                                                         EDIT3460
5050 CONTINUE                                                         EDIT3490
      GO TO 136                                                         EDIT3500
C *** END OF L P SUBROUTINE *****                                EDIT3510

```

```

C
C
C          ERROR
9901 NK=110
      GO TO 9999
C          ENERGY CHECK
9905 NK=136
      GO TO 9999
C          NEGATIVE MASS
9917 NK=6015
      GO TO 9999
9920 NK=904
      GO TO 9999
9921 NK=912
      GO TO 9999
9922 NK=918
      GO TO 9999
9923 NK=922
      GO TO 9999
9924 NK=926
9999 NR=6
      CALL DUMP
10000 RETURN
C
C          FORMATS
8108 FORMAT(I3,1X,1P11E10.3)
81160FORMAT(8H1PROBLE: 3X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HDTRAD1
12X,2HNR6X,2HN14X,2HN24X,2HN34X,2HN4/(F7.1,I11,2X,1P4E16.7,I10,2X,4
2I6))
81170FORMAT(1H0//17X2HAI16X,2HAK14X,5HAI+AK15X,2HAM/4H DOT3X,1P4E18.7/3
1H X4X,4E18.7)
81180FORMAT(12X,13H-----5X,13H-----5X,13H-----5
1X,13H-----/7H TOTALS1P4E18.7)
81190FORMAT(2H0 //16X,5HRADEB13X,5HRADER13X,5HRADET12X,7HMAX VEL13X,3HTED
1HE12X,9HREL ERROR/7X,1P6E18.7////)
8120 FORMAT(1H0//21H TAPE 7 DUMP ON CYCLEI5////)
81240FORMAT(3H K12X,5HAM(K)11X,9HSUM AM(K)11X,4HP(K)13X,4HQ(K)/(I3,4X,
11P4E18.7))
8131 FORMAT(1H ///11H DY(J) J=1,I2//(10F12.3))
8133 FORMAT(1H ///11H Y(J) J=0,I2//(10F12.3))
81350FORMAT(1H ///4H I =I3,6X,6HX(I) =F12.3,6X,7HDX(I) =F12.3//3H J8X,
11HU9X,1HV9X,3HF/A7X,3HDI7X,3HAMX7X,3HAIX7X,3HAMD7X,3HAID7X,4HETAX
2X6X,4HETAD6X,1HZ/)
8201 FORMAT(I10,2H I54A2)
8202 FORMAT(10X,2H I54A2)
8203 FORMAT(3X,1P6E18.7)
8211 FORMAT(F7.1,I3,2H I54A2)
8222 FORMAT(F7.1,3X,2H I54A2)
8302 FORMAT(I12,10I10)
83070FORMAT(5H X1 =1PE12.6,3X,4HX2 =E12.6,3X,6HXXMAX =E12.6,6X,4HY1 =E12
1.6,3X,4HY2 =E12.6,3X,6HYMAX =E12.6)
8308 FORMAT(1H /)
9040 FORMAT(1H / 6I6)
      END
C
C
C

```

```

EDIT3520
EDIT3530
EDIT3540
EDIT3550
EDIT3560
EDIT3570
EDIT3580
EDIT3590
EDIT3600
EDIT3610
EDIT3620
EDIT3630
EDIT3640
EDIT3650
EDIT3660
EDIT3670
EDIT3680
EDIT3690
EDIT3700
EDIT3710
EDIT3720
EDIT3740
EDIT3750
EDIT3760
EDIT3770
EDIT3790
EDIT3800
EDIT3810
EDIT3820
EDIT3830
EDIT3840
EDIT3850
EDIT3860
EDIT3870
EDIT3880
EDIT3890
EDIT3900
EDIT3910
EDIT3920
EDIT3930
EDIT3950
EDIT3960
EDIT3970
EDIT3980
EDIT3990
EDIT4000
EDIT4010
EDIT4020
EDIT4030
EDIT4040

```

SUBROUTINE REZONE

REZ00010

NOTE, THIS VERSION ASSUMES THAT WE ARE
ADDING (.) MATERIAL

***** A 2 MATERIAL OIL CODE *****

A 2 MATERIAL REZONE SUBROUTINE

CONSERVE MOMENTUM AND TOTAL ENERGY, INCREASE

ALL LINEAR DIMENSIONS BY 2. (THUS 4 CELLS

IN THE OLD GRID ARE COMBINED INTO 1 FOR

THE NEW GRID.)

NIMAX=IMAX/2

REZ00990

NJMAX=JMAX/2

REZ01000

DO 10 J=1,NJMAX

REZ01010

K=(J-1)*NIMAX+2

REZ01020

L=(J-1)*2*IMAX+2

REZ01030

DO 11 I=1,NIMAX

REZ01040

M=L+IMAX

REZ01050

12 WSA=AMX(L)+AMX(M)+AMX(L+1)+AMX(M+1)

REZ01060

WSAD=AMD(L)+AMD(M)+AMD(L+1)+AMD(M+1)

IF (WSA+WSAD) 100,100,101

100 AMX(K)=0.

AIX(K)=0.

AMD(K)=0.

AID(K)=0.

U(K)=0.

V(K)=0.

GO TO 9901

101 CONTINUE

8900 CONTINUE

WSB=(AMX(L)+AMD(L))*(U(L)**2+V(L)**2)

WSB=WSB+(AMX(M)+AMD(M))*(U(M)**2+V(M)**2)

WSB=WSB+(AMX(L+1)+AMD(L+1))*(U(L+1)**2+V(L+1)**2)

WSB=WSB+(AMX(M+1)+AMD(M+1))*(U(M+1)**2+V(M+1)**2)

U(K)=U(L)*(AMX(L)+AMD(L))+U(M)*(AMX(M)+AMD(M))

+U(L+1)*(AMX(L+1)+AMD(L+1))+U(M+1)*(AMX(M+1)+AMD(M+1))

U(K)=U(K)/(WSA+WSAD)

V(K)=V(L)*(AMX(L)+AMD(L))+V(M)*(AMX(M)+AMD(M))

+V(L+1)*(AMX(L+1)+AMD(L+1))+V(M+1)*(AMX(M+1)+AMD(M+1))

V(K)=V(K)/(WSA+WSAD)

AIX(K)=AIX(L)*AMX(L)+AIX(M)*AMX(M)+AIX(L+1)*

REZ01140

+AIX(M+1)*AMX(M+1)+AIX(L+1)*

REZ01150

AMD(K)=WSA

REZ01160

AID(K)=AID(L)*AMD(L)+AID(M)*AMD(M)+

+AID(L+1)*AMD(L+1)+AID(M+1)*AMD(M+1)

AMD(K)=WSAD

WS=U(K)**2+V(K)**2

REZ01170

E=AIX(K)+AID(K)+WSB/2.0

IF (AMD(K)+AMX(K)) 9901,9901,500

500 IF (AMD(K)) 501,501,502

X ONLY

501 AIX(K)=E/AMX(K)-.5*WS

DKE(K)=-2.

AID(K)=0.

GO TO 9901

502 IF (AMX(K)) 503,503,504

```

C   DOT ONLY
503 AID(K) = E/AMD(K) ~ .5* WS
    DKE(K)=-1.
    AIX(K) =0.
    GO TO 9901
C   MIXED CELL
504 DQ=E-(AMX(K)+AMD(K))*WS/2.
    WSE=AIX(K)+AID(K)
    AIX(K)=AIX(K)/AMX(K)*DQ/WSE
    AID(K)=AID(K)/AMD(K)*DQ/WSE
    DKE(K)=1.
9901 IF(K-2)14,14,13
C   SET CELL QUANTITIES OF OLD GRID TO ZERO.
13  AMX(L)=0.0
    AMD(L)=0.
    U(L)=0.0
    V(L)=0.0
    AIX(L)=0.0
    AID(L)=0.
    P(L)=0.0
    DKE(L)=0.
    AMX(M)=0.0
    AMD(M)=0.
    U(M)=0.0
    V(M)=0.0
    AIX(M)=0.0
    AID(M)=0.
    P(M)=0.0
    DKE(M)=0.
    AMX(L+1)=0.0
    AMD(L+1)=0.
    U(L+1)=0.0
    V(L+1)=0.0
    AIX(L+1)=0.0
    AID(L+1)=0.
    P(L+1)=0.0
    DKE(L+1)=0.
    AMX(M+1)=0.0
    AMD(M+1)=0.
    U(M+1)=0.0
    V(M+1)=0.0
    AIX(M+1)=0.0
    AID(M+1)=0.
    P(M+1)=0.0
    DKE(M+1)=0.
14  K=K+1
    L=L+2
11  CONTINUE
10  CONTINUE
C   CALCULATE NEW DY AND Y (JMAX OF THEM).
    I=0
    DO 200 J=1,JMAX,2
    I=I+1
    DY(I)=DY(J)+DY(J+1)
200 CONTINUE
    II=NJMAX+1
    DO 201 J=II,JMAX

```

REZ01210

REZ01220

REZ01230

REZ01240

REZ01250

REZ01260

REZ01270

REZ01280

REZ01290

REZ01300

REZ01310

REZ01320

REZ01330

REZ01340

REZ01350

REZ01360

REZ01370

REZ01380

REZ01390

REZ01400

REZ01410

REZ01420

REZ01430

REZ01440

```

      DY(J)=DY(I)
201  CONTINUE
      WS=0.
      DO 202 J=1,JMAX
      Y(J)=DY(J)+WS
      WS=Y(J)
202  CONTINUE
C    CALCULATE THE NEW DX,S AND TAU,S
      I=0
      DO 203 J=1,IMAX,2
      I=I+1
      DX(I)=DX(J)+DX(J+1)
203  CONTINUE
      II=NIMAX+1
      DO 204 J=II,IMAX
      DX(J)=DX(I)
204  CONTINUE
      WS=0.
      WSA=0.
      DO 205 I=1,IMAX
      X(I)=DX(I)+WS
      WS=X(I)
      WSB=X(I)**2
      TAU(I)=PIDY*(WSB-WSA)
      WSA=WSB
205  CONTINUE
      IMAX=NIMAX
      JMAX=NJMAX
C    PREPARE NOW TO SHUFFLE THE K ARRAYS SUCH
C    AS TO PRESERVE K=(J-1)*IMAX+I+1.
      DO 20 N=1,JMAX
      J=JMAX+1-N
      K=(J-1)*IMAX+1+IMAX
      L=(J-1)*(IMAX+IMAX)+1+IMAX
      DO 21 I=1,IMAX
1000  AMX(L)=AMX(K)
      DKE(L)=DKE(K)
      AMD(L)=AMD(K)
      AIX(L)=AIX(K)
      AID(L)=AID(K)
      U(L)=U(K)
      V(L)=V(K)
      P(L)=P(K)
      IF(J-1)1002,1002,1001
1001  AMX(K)=0.0
      AMD(K)=0.
      AIX(K)=0.0
      AID(K)=0.
      DKE(K)=0.
      V(K)=0.0
      U(K)=0.0
      P(K)=0.0
1002  K=K-1
      L=L-1
      21  CONTINUE
      20  CONTINUE
      IMAX=NIMAX*2

```

REZ01620
REZ01630

REZ01640
REZ01650
REZ01660
REZ01670
REZ01680
REZ01690

REZ01700

REZ01710
REZ01720
REZ01730
REZ01740
REZ01750

REZ01760

REZ01770
REZ01780
REZ01790
REZ01800
REZ01810/
REZ01820
REZ01830
REZ01840

RE201850

```

JMAX=NJMAX*2
II=NIMAX+1
JJ=NJMAX+1
C ADD ON NEW MASS WITH DENSITY=Z(111) IN TARGET
DO 50 I=1,NIMAX
K=(JJ-1)*IMAX+I+1
DO 50 J=JJ,JMAX
AMD(K)=Z(111)*TAU(I)*DY(J)
60 K=K+IMAX
50 CONTINUE

```

```

JJ=(Z(147)/2.+2)
JJ=JJ+1
DO 61 I=II,IMAX
K=I+1+(JJ-1)*IMAX
DO 62 J=JJ,JMAX
AMD(K)=Z(111)*TAU(I)*DY(J)
62 K=K+IMAX
61 CONTINUE

```

```

C RESET ACTIVE GRID MARKERS.
JJ=JJ-1
Z(147)=JJ
I1=I1/2+2
I2=I2/2+2
WS=T+UTNA
KK=NC+1

```

```

C EDIT THE NEW GRID.
WRITE (6,8004) WS, NK, DX(1)
WRITE (6,8007) IMAX, (X(I), I=0, IMAX)
WRITE (6,8006) JMAX, (Y(J), J=0, JMAX)
WRITE (6,8009) IMAX, (DX(I), I=1, IMAX)
WRITE (6,8010) JMAX, (DY(J), J=1, JMAX)
WRITE (6,8011) IMAX, (TAU(I), I=1, IMAX)
KMAX=IMAX*JMAX+1
IMAXA=IMAX+1
JMAXA=JMAX+1
KMAXA=KMAX+1
RETURN

```

80040FORMAT(1H ////22H PROBLEM REZONED AT T=1PE12.6,6X,5HCYCLEI4,5X,3HOREZ02190
1X=E12.6////)

```

8007 FORMAT(1H /10H X(I) I=0,I2/(5F16.6))
8008 FORMAT(1H /10H Y(J) J=0,I2/(5F16.6))
8009 FORMAT(1H /11H DX(I) I=1,I2/(5F16.6))
8010 FORMAT(1H /11H DY(J) J=1,I2/(5F16.6))
8011 FORMAT(1H /15H AREA(I) I=1,I2/(F16.6,4F18.6))
END

```

```

C -----
C -----

```

RE