GENERAL ATOMIC DIVISION
CENTRAL DYNAMICS CORPORATION

John Jay Hopkins Laboratory for Pure and Applied Science
P.O. Box 699, San Diego 12, California

CAMD-553-Revised
Category II

No. 116

A CONTINUOUS TWO-DIMENSIONAL EULERIAN HYDRODYNAMIC CODE

This document, which was prepared primarily for internal use at 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
Army Contract DA-14-005-MC-1161(x)

Project No. 095
January, 1967
FOREWORD

The OIL computer code described herein is as it existed on October 26, 1964. The code has been in continuous development for three years and in its presented form has been applied successfully by General Atomic to the kind of problems discussed in this report. However, the development and improvement in both the physics and mathematics of the code are being continued, so that duplication of results (or even close agreement) between problems run with the code as published and the code as it existed either before or after this time is not necessarily to be expected.

General Atomic has exercised due care in preparation, but does not warrant the merchantability, accuracy, and completeness of the code or of its description contained herein. The complexity of this kind of program precludes any guarantee to that effect. Therefore, any user must make his own determination of the suitability of the code for any specific use and of the validity of the information produced by use of the code.
CONTENTS

1. INTRODUCTION AND ACKNOWLEDGEMENTS ............................. 1

2. CLAM
   2.1. General Description ........................................ 1
   2.2. Input Description for CLAM ................................. 6
   2.3. Output from CLAM ........................................... 12

3. OIL
   3.1. Basic Equations ............................................. 16
   3.2. Special Subroutines ......................................... 31
   3.3. Logic of OIL .................................................. 33
   3.4. List of Common for OIL ..................................... 40

4. TEST PROBLEMS
   4.1. Comparison of Techniques for the Continuous Eulerian Transport 53
   4.2. Comparison of SHELL (PIC Transport) and OIL (Continuous Transport) 54
   4.3. One-dimensional Test Problems ............................... 54

5. FORTRAN LISTINGS
   5.1. CLAM ....................................................... 72
   5.2. OIL ........................................................ 100

REFERENCES ..................................................................... 139
1. **INTRODUCTION AND ACKNOWLEDGEMENTS**

The OIL code is very closely related to familiar particle-in-cell codes\(^{(1,2,3)}\) and has been developed by modification of the General Atomic particle-in-cell code named SHELL. The basic difference lies in the method of the mass transport, the OIL scheme being a continuous mass transport, while the SHELL scheme is discrete. The initial work on a continuous version of SHELL was undertaken several years ago by B. E. Freeman and the author. Since early 1963 the development of the continuous Eulerian has been continued by J. M. Walsh and the author.

The author is deeply appreciative of the work and consultation given by B. E. Freeman and J. M. Walsh. The assistance of D. R. Yates and R. H. Fisher in the automatic plotting routines used for the OIL code is also much appreciated.

Since its original use for the hypervelocity impact calculations\(^{(4)}\), the OIL code has been successfully adapted to several other high energy fluid dynamic applications.

Detailed descriptions of various problems, especially results for hypervelocity impact, are given in Ref. 4.

2. **CLM**

2.1. General Description

CLM is the generator code for the OIL code, and is used to generate initial values of the variables for each cell in the grid. (An exception arises for certain simple initial conditions, described in Section 3.2, where it is possible to bypass CLM and use instead a more economical routine called SETUP.)

The function of CLM is illustrated by a simple example (Fig. 1). We wish to generate the following grid: A right circular cylinder of density 1. \(\text{g/cm}^3\), radius 24 cm, and height 12 cm is located at position \(z = 20\) cm along the axis. A projectile (right cylinder) of density 1. \(\text{g/cm}^3\), radius 6 cm, and height 12 cm, is located at positions 8-20 along the axis. The projectile has an initial axial velocity of \(1 \times 10^6\) cm/sec. The cells are 1 cm on a side.
Fig. 1
| GAS GAMMA = 1.5 RHO TARGET = RHO PELLET = 1. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0024 1.0       | 0.0             | 24.0            | 32.0            | 0.0             | 2.0             | 0.0             | 0.0             | 0.0             | 0.0             | 0.0             |
| 1132 1.0       | 0.0             | 3.0             | 80.0            | 1.0             | 1.0             | 1.0             | 1.0             | 1.0             | 1.0             | 1.0             |
| 01 1.0         | 0.0             | 6.0             | 8.0             | 0.0             | 20.0            | 0.0             | 0.0             | 0.0             | 0.0             | 0.0             |
| 01 1.0         | 0.0             | 7.0             | 12.0            | 13.0            | 14.0            | 15.0            | 16.0            | 17.0            | 18.0            | 19.0            |
| 01 1.0         | 0.0             | 8.0             | 22.0            | 23.0            | 24.0            | 0.0             | 0.0             | 0.0             | 0.0             | 0.0             |

**GAS GAMMA = 1.5 RHO TARGET = RHO PELLET = 1.**

<table>
<thead>
<tr>
<th>PROB. NO. 1.000</th>
<th>I=24</th>
<th>J=30</th>
</tr>
</thead>
<tbody>
<tr>
<td>X(I) I=0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td>5.0</td>
<td>6.0</td>
<td>7.0</td>
</tr>
<tr>
<td>10.0</td>
<td>11.0</td>
<td>12.0</td>
</tr>
<tr>
<td>15.0</td>
<td>16.0</td>
<td>17.0</td>
</tr>
<tr>
<td>20.0</td>
<td>21.0</td>
<td>22.0</td>
</tr>
<tr>
<td>Y(J) J=0.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td>5.0</td>
<td>6.0</td>
<td>7.0</td>
</tr>
<tr>
<td>10.0</td>
<td>11.0</td>
<td>12.0</td>
</tr>
<tr>
<td>15.0</td>
<td>16.0</td>
<td>17.0</td>
</tr>
<tr>
<td>20.0</td>
<td>21.0</td>
<td>22.0</td>
</tr>
<tr>
<td>25.0</td>
<td>26.0</td>
<td>27.0</td>
</tr>
<tr>
<td>30.0</td>
<td>31.0</td>
<td>32.0</td>
</tr>
<tr>
<td>D(X) I=1.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**GAS GAMMA = 1.5 RHO TARGET = RHO PELLET = 1.**
### PACKAGE NO. 1
1 PARTICLE /CELL

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DENSITY</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>ENERGY</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VELOCITY</td>
<td>0.01</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

RECTANGLE---GEN--- 0.  6.  8.  20.
I(I)= 1  J(J)= 8  I(IN)= 7  J(IN)= 21

72 (X) PARTICLES  PE=6.785839-02  PM=1.357169+03

### PACKAGE NO. 2
1 PARTICLE /CELL

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

RECTANGLE---GEN--- 0.  24.  20.  32.
I(I)= 1  J(J)=20  I(IN)=24  J(IN)=32

288 (X) PARTICLES  PE=0.  PM=2.171467+04

M. = 0.  MX = 2.30718+04  M. + MX = 2.30718+04
PARTICLES --- 0 DOT  360 X  360 TOTAL
TAPE DUMP AT TIME  0.  CYCLE 0.
The coordinates $x(i)$ and $y(j)$ of each cell are calculated from the
$DX$'s and $DY$'s that are loaded. In addition, the area term $T(i) = \frac{\pi}{6}(x(i)^2 - x(i-1)^2)$ are calculated.

The next step is to process the packages (see Section 2.2); there are
two in this example. Package Number (1) is the projectile and Package
Number (2) is the target.

We specify the coordinates of each package. In addition, we specify
the number of particles per cell in each package. We assign a density, the
two velocity components, and specific internal energy for the package in
question. These may be constants or functions of both $z$ and $r$.

After specifying the coordinates of the package, particles ($N\$, where $N$ may range from 1 to 20) are placed in each cell. Each of the particles
is then assigned a density, the two velocity components and specific
internal energy. The cell is divided into $N$ equal parts, and the $N$
particles placed at the center of these areas. The mass of each particle
is then the density (prescribed along with the package input) times the
volume of the small subdivision cell of the cell $(K)$. The mass of cell $(K)$
then is the sum of the $N$ particle masses, and both momenta components are
calculated as the sum of the individual momenta components of each particle.
The internal energy of each cell $(K)$ is the specific internal energy times
the particle mass summed over all $N$ particles.

In the output routine these cell quantities are then converted to the
two velocity components and the specific internal energy.

Particles are created, whether these data are for an OIL or a SHELL
run, for the purpose of describing the many possible geometries and possible
energy, velocity, and density distributions. If this is an OIL run, the
particles are not written on a tape as they would be if this were a SHELL
run. One continues to process each cell $(K)$ within the given package and
the $N$ particles in cell $(K)$.

In this example, it is sufficient to specify only one particle per
cell, since the package boundaries coincide with the cell boundaries and
the density, velocities, and internal energy are constant.
After processing all packages, one then writes this information about the grid and the cells on a tape that will be the starting conditions for the OIL or SHELL code (see Section 2.3.)

The output from CLAM, excluding the long print, is shown on the listings 8 and 4.

2.2. Input Description for CLAM

CLAM and OIL are written in cylindrical coordinates with axial symmetry. In the following discussion and description, X refers to the coordinate R and Y to the coordinate Z. An asterisk before the word signifies that the data are in floating point; otherwise they are fixed point.

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Reader card, any BCD information in col. 2-72.</td>
</tr>
<tr>
<td>2</td>
<td>* 1-10</td>
<td>Contain the problem number. If less than zero, this will be a particle run using SHELL; if greater than zero, this will be a continuous run using OIL.</td>
</tr>
<tr>
<td></td>
<td>* 11-20</td>
<td>IMAX, the number of cells in the X-direction (maximum of 50).</td>
</tr>
<tr>
<td></td>
<td>* 21-30</td>
<td>JMIX, the number of cells in the Y-direction (maximum of 100).</td>
</tr>
<tr>
<td></td>
<td>* 31-40</td>
<td>Blank</td>
</tr>
<tr>
<td></td>
<td>* 41-50</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>* 51-60</td>
<td>S8 (not used as input in CLAM)</td>
</tr>
<tr>
<td></td>
<td>* 61-70</td>
<td>S9 (not used in CLAM)</td>
</tr>
<tr>
<td></td>
<td>* 71-72</td>
<td>N7 (N7 = binary tape number)</td>
</tr>
</tbody>
</table>

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 data cards.

A(1) indicates DY data.

A(0) indicates that we are loading DX data.

3-4 Indicate the number of zones that will have the same DX or DY values that appear in columns 11-20.

5-6 Indicate the number of zones that will have the same DX or DY values that appear in columns 21-30.
<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>7-9</td>
<td>Indicate the number of zones that will have the same IX or IY values that appear in columns 31-40.</td>
</tr>
<tr>
<td></td>
<td>9-10</td>
<td>Indicate the number of zones that will have the same IX or IY values that appear in columns 41-50.</td>
</tr>
<tr>
<td>*</td>
<td>11-20</td>
<td>The value of IX or IY</td>
</tr>
<tr>
<td>*</td>
<td>21-30</td>
<td>The value of IX or IY</td>
</tr>
<tr>
<td>*</td>
<td>31-40</td>
<td>The value of IX or IY</td>
</tr>
<tr>
<td>*</td>
<td>41-50</td>
<td>The value of IX or IY</td>
</tr>
<tr>
<td>4</td>
<td>* 1-10</td>
<td>$\Pi_1 =$ the tape number of one of the scratch tapes to be used in CLAM and OIL</td>
</tr>
<tr>
<td>*</td>
<td>11-20</td>
<td>$\Pi_2 =$ the other tape number for the scratch tape. OIL requires two scratch tapes if using the particle transport.</td>
</tr>
<tr>
<td>*</td>
<td>21-30</td>
<td>$\Pi_4 =$ maximum number of particles plus one per tape record, that CLAM will generate (maximum value = 130)</td>
</tr>
<tr>
<td>*</td>
<td>31-40</td>
<td>Switch (not used)</td>
</tr>
</tbody>
</table>

Now we begin loading the necessary data to generate a package. The maximum number of packages that may be generated is 72; to increase the maximum requires changing the dimension statements.

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  (M$^2$), the number of particles per cell to be generated, where 1 $\leq$ N $\leq$ 20.
* 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  TEMP(3) is loaded into SS (SS contains the number (1-6) of the fit for the density, internal energy, and velocity that will be assigned to each particle as it is generated in this package.
* 41-70  Blank.
Following the first card of each package are five other types of cards:

1. Generate (geometry of package)
2. Delete card (if needed); there may be more than one
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 card (1) above, CLAM has the following geometric options for generating or deleting:

1. A rectangle - a 4 in column 1
   - a (1) in column 7 means generate this rectangle
   - a (0) in column 7 means delete this rectangle
   * 11-20 $X_1$ = the left R coordinate of the rectangle
   * 21-30 $X_2$ = the right R coordinate of the rectangle
   * 31-40 $Y_1$ = the lower Z coordinate of the rectangle
   * 41-50 $Y_2$ = the upper Z coordinate of the rectangle

2. A triangle - a 6 in column 1
   - a (1) in column 7 means generate this triangle
   - a (0) in column 7 means delete this triangle.
   * 11-20 $X_1$
   * 21-30 $Y_1$
   * 31-40 $X_2$ --Note: Vertices (1-3) can be in any order
   * 41-50 $Y_2$
   * 51-60 $X_3$
   * 61-70 $Y_3$

3. An ellipse or circle - a 41 in column (1-2)
   - a (1) in column 7 means generate this ellipse or circle
   - a (0) in column 7 means delete this ellipse or circle
* 11-20 The semi-axis in the X-direction if an ellipse or the radii if for a circle
* 21-30 The semi-axis in the Y-direction if an ellipse or zero if 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.

4. A perturbed ellipse - a 61 in columns (1-2)
   a (1) in column 7 means generate this perturbed ellipse.
   a (0) in column 7 means delete this perturbed ellipse.

* 11-20 Semi-axis in the X-direction
* 21-30 Semi-axis in the Y-direction
* 31-40 0.
* 41-50 Y-coordinate of center of perturbed ellipse
* 51-60 X-coordinate of point of perturbation
* 61-70 Y-coordinate of point of perturbation.

Following the geometry cards are the following:
- 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 $\rho$ or $I$ or velocity will remain the same as the previous package, then a 51, 52 or 53 card is not required.

* 11-20
* 21-30
* 31-40 - Contains the values to be used in the analytical expressions for the density, energy, and velocities.
* 41-50
* 51-60
* 61-70

This data is then loaded into the following arrays:
- TABR(1-6) for density
- TABI(1-6) for internal energy
- TABUV(1-6) for velocity
Finally, a 2 in column 1 signifies the completion of loading all input cards for the CLAM code.

There are six subroutines (FIT1 - FIT6) used for computing \( p, I, U, \) and \( V \). The standard input to these subroutines is as follows:

\[ TY = Y \text{ coordinate of particle } N \]
\[ TX = X \text{ coordinate of particle } N \]

The modified coordinates \( TTY \) and \( TTX \) are computed as follows:

\[ TTY = Y \text{ coordinate} = TY - YC \text{ (relative to } YC) \]
\[ TTX = X \text{ coordinate} = TX - XC \text{ (relative to } XC) \]

The standard output from the subroutines is as follows:

\( WSR = p \) (density) of particle \( N \)
\( WSi = I \) (specific internal energy) of particle \( N \)
\( WSU = U \) (radial velocity component) of particle \( N \)
\( WSV = V \) (axial velocity component) of particle \( N \).

Below are the equations, or analytical fits, for the six subroutines. Any or all may be changed. Each equation is followed by the FORTRAN mnemonic.

1. **FIT 1**

\[
R = (x^2 + y^2)^{\frac{1}{2}} \\
WS = (TTX^2 + TTY^2)^{\frac{1}{2}} \\
p = A + B (Y - C) \\
WSR = TABR(1) + TABR(2) \times (TTY - TABR(3)) \\
I = A + B (Y-C) \\
WSi = TABi(1) + TABi(2) \times (TTY - TABi(3)) \\
U = 0 \\
WSU = 0 \\
V = A + B (Y - C) \\
WSV = TABUV(1) + TABUV(2) \times (TTY - TABUV(3))
\]

2. **FIT 2**

\[
R = (x^2 + y^2)^{\frac{1}{2}} \\
WS = (TTX^2 + TTY^2)^{\frac{1}{2}} \\
p = \left( \frac{X-A}{B} \right)^2 + \left( \frac{Y-C}{D} \right)^2
\]
WSR = \frac{(TTX - TABB(1))^2 + (TTY - TABB(3))^2}{TABB(2)}

I = A + BX + CX^2 + DY + EY^2

WSI = TABI(1) + TABI(2) * TTX + TABI(3) * TTY^2

+ TABI(4) * TTY + TABI(5) * TTY^2

U = C + D * Y

WSU = TABUV(3) + TABUV(4) * TTY

V = A + B * Y

WSB = TABUV(1) + TABUV(2) * TTY

3. **PUT 3**

R = \sqrt{X^2 + Y^2}

WS = \sqrt{TTX^2 + TTY^2}

\rho = A + BR + CR^2

WSR = TABB(1) + TABB(2) * WS + TABB(3) * WS^2

I = A + BR + CR^2

WSI = TABI(1) + TABI(2) * WS + TABI(3) * WS^2

U = \frac{X}{R} \left( \frac{A + BR + CR^2}{D + ER + FR^2} \right)

WSU = \frac{TTX}{WS} \left( \frac{TABUV(1) + TABUV(2) * WS + TABUV(3) * WS^2}{TABUV(4) + TABUV(5) * WS + TABUV(6) * WS^2} \right)

V = \frac{Y}{R} \left( \frac{A + BR + CR^2}{D + ER + FR^2} \right)

WSV = \frac{TTX}{WS} \left( \frac{TABUV(1) + TABUV(2) * WS + TABUV(3) * WS^2}{TABUV(4) + TABUV(5) * WS + TABUV(6) * WS^2} \right)

Fits 4, 5, and 6 are dummy routines. Although particles are not used in the SHELL code if it is a continuous run (problem number greater than zero), the use of particles in CLAM provides the method for assigning mass, energy, to each cell.
2.3. **OUTPUT FROM CLAM**

The output from the CLAM 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 tape.

The normal system of units are the centimeter-gram-shake, where the units of energy are jerks/ε and the pressure in units of jerks/cm³ (1 jerk = 10¹⁶ ergs and 1 shake = 10⁻¹⁶ 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 OIL or SHELL calculation. Below is a complete list of generated data from CLAM that is written on the binary output tape.

<table>
<thead>
<tr>
<th>Z</th>
<th>Equiv.</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROB</td>
<td>--</td>
<td>Equals problem number, input to CLAM</td>
</tr>
<tr>
<td>2</td>
<td>Cycle</td>
<td>--</td>
<td>Equals cycle number = 0</td>
</tr>
<tr>
<td>3</td>
<td>DT</td>
<td>shake</td>
<td>Set to 0 by CLAM</td>
</tr>
<tr>
<td>4</td>
<td>Prints</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Print-L</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DUMPT</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>C Stop</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>PIDY</td>
<td>--</td>
<td>Equals π = 3.1415927</td>
</tr>
<tr>
<td>9</td>
<td>TMZ</td>
<td>grams</td>
<td>Total mass (x) generated by CLAM</td>
</tr>
<tr>
<td>10</td>
<td>GM</td>
<td>--</td>
<td>Set to 0 by CLAM</td>
</tr>
<tr>
<td>11</td>
<td>GAMD</td>
<td>--</td>
<td>Set to 0 by CLAM</td>
</tr>
<tr>
<td>12</td>
<td>GAMX</td>
<td>--</td>
<td>Set to 0 by CLAM</td>
</tr>
<tr>
<td>13</td>
<td>ETH</td>
<td>jerk</td>
<td>Total energy in system</td>
</tr>
<tr>
<td>14</td>
<td>FFA</td>
<td>--</td>
<td>Set to 0 by CLAM</td>
</tr>
<tr>
<td>15</td>
<td>FFB</td>
<td>--</td>
<td>Set to 0 by CLAM</td>
</tr>
<tr>
<td>16</td>
<td>TMZ</td>
<td>grams</td>
<td>Total mass (x) generated by CLAM; this version of CLAM does not generate (.)</td>
</tr>
<tr>
<td>17</td>
<td>Tmax</td>
<td>grams</td>
<td>Total mass (x) generated by CLAM</td>
</tr>
<tr>
<td>18</td>
<td>XMAX</td>
<td>cm</td>
<td>= X(MAX)</td>
</tr>
<tr>
<td>Z</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td><strong>TMAX</strong> cm = <strong>2 XMAX</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td><strong>TMAM</strong> cm = <strong>2. Y MAX</strong> (note <strong>Y MAX</strong> is not in <strong>Z</strong> block)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td><strong>ANM</strong> grams = <strong>minimum mass/2. of the dot particles</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td><strong>AMX</strong> grams = <strong>minimum mass/2. of the X particles</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td><strong>IMN</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td><strong>IMIN</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td><strong>PEF</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td><strong>MIMA</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td><strong>CVIS</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td><strong>NPRI</strong> -- Set equal to 6 in CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td><strong>NPRI</strong> -- CLAM sets <strong>NPRI</strong> = <strong>N4</strong> (check definition of <strong>N4</strong>(<em>Z(54))</em>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td><strong>NC</strong> -- Fixed value of cycle number, set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td><strong>NPC</strong> -- Used as indices in CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td><strong>NRC</strong> -- Used as indices in CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td><strong>DMAX</strong> -- Input to CLAM = maximum number of zones in <strong>X</strong> direction for this run</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td><strong>DMAX</strong> -- Equal <strong>DMAX</strong> + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td><strong>JMAX</strong> -- Input to CLAM = maximum number of zones in <strong>Y</strong> direction for this run</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td><strong>JMAXA</strong> -- = <strong>JMAX</strong> + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td><strong>KMAX</strong> -- = (<strong>IMAX</strong>)(<strong>JMAX</strong> + 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td><strong>KMAXA</strong> -- = <strong>KMAX</strong> + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td><strong>NMAX</strong> -- = total number of particles + 1 that CLAM has generated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td><strong>ND</strong> -- = total number of dot particles + 1 that CLAM has generated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td><strong>MID</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td><strong>IMAX</strong> -- = <strong>IMAXA</strong> + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td><strong>MOD</strong> -- Used as index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td><strong>NOPR</strong> -- Set equal to <strong>N3</strong> (Note definition of <strong>N3</strong> (*Z(53)))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td><strong>N1MAX</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td><strong>NJMAX</strong> -- Set to 0. by CLAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Equiv.</td>
<td>Units</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>--------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>-</td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>48</td>
<td>12</td>
<td>-</td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>49</td>
<td>13</td>
<td>-</td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>50</td>
<td>14</td>
<td>-</td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>51</td>
<td>N1</td>
<td>--</td>
<td>= scratch tape number</td>
</tr>
<tr>
<td>52</td>
<td>N2</td>
<td>--</td>
<td>= scratch tape number</td>
</tr>
<tr>
<td>53</td>
<td>N3</td>
<td>--</td>
<td>= number of particle records of length E that CLAM has generated</td>
</tr>
<tr>
<td>54</td>
<td>N4</td>
<td>--</td>
<td>= number of particles + 1 to be stored on each particle tape record</td>
</tr>
<tr>
<td>55</td>
<td>N5</td>
<td>-</td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>56</td>
<td>N6</td>
<td>--</td>
<td>= number of particles on the last particle tape record</td>
</tr>
<tr>
<td>57</td>
<td>N7</td>
<td>--</td>
<td>= binary tape designation number</td>
</tr>
<tr>
<td>58</td>
<td>N8</td>
<td></td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>59</td>
<td>N9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>N10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>N11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>NRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>TRAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>XNRG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>SN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>DXN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>RADER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>RADET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>RADEB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>DTRAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>RESPCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>RSTOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>SHELL</td>
<td></td>
<td>A counter that may be used to distinguish between codes</td>
</tr>
<tr>
<td>74</td>
<td>BBOUND</td>
<td></td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>75</td>
<td>TOZONE</td>
<td></td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>Z</td>
<td>Equiv.</td>
<td>Units</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>76</td>
<td>ECK</td>
<td></td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>77</td>
<td>SBOUND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>X1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>Y1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>Y2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>CABLN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>VISC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>GMAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>WSGD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>WSGX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>GMADR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>GMAXR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>S1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>S2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>S4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>S5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>S6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>S7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>S8</td>
<td></td>
<td>Used for storage of FIT number for each package in CLAM</td>
</tr>
<tr>
<td>98</td>
<td>S9</td>
<td></td>
<td>Set to 0. by CLAM</td>
</tr>
<tr>
<td>99</td>
<td>S10</td>
<td></td>
<td>Set to 0. by CLAM</td>
</tr>
</tbody>
</table>

Z(100) through Z(150) is also set to 0. by CLAM.
3. The Eulerian equations we wish to solve are the following:

(A) \[ \frac{\partial p}{\partial t} + \nabla \cdot (pu) = 0 \]

(B) \[ \frac{\partial pu}{\partial t} + \nabla \cdot (puu) = -\nabla P \]

(C) \[ \frac{\partial E}{\partial t} + \nabla \cdot (pE) = -7 \cdot (\vec{F}_t) \]

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 particles or continuous mass across cell boundaries.

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

\[ \frac{\partial p}{\partial t} = -\frac{\partial pu}{r \partial r} - \frac{\partial pv}{\partial z} \]  

(1)

\[ \rho \frac{\partial u}{\partial t} = -\frac{p}{r} \]  

(2)

\[ \rho \frac{\partial v}{\partial t} = -\frac{\partial p}{\partial z} \]  

(3)

\[ \rho \frac{\partial E}{\partial t} = -\frac{\partial P u}{r \partial r} - \frac{\partial p v}{\partial z} \]  

(4)

\[ P = f(p, I) \]  

Equation of state  

(5)

\[ p = \text{density of cell (K) in g/cm}^3 \]

\[ r = \text{r coordinate in cm.} \]

\[ z = \text{z coordinate in cm.} \]

\[ u = \text{radial component of velocity in cm/shake} \]

\[ v = \text{axial component of velocity in cm/shake} \]

\[ P = \text{material pressure in jerks/cm}^3 \]

\[ E = \text{total specific energy in jerks/g} \]

\[ I = \text{specific internal energy in jerks/g} \]

(1 jerk = $10^{16}$ ergs)
The five variables listed are all located at the center of the cell (Fig. 2). For complete description of all quantities used, see Section 3.4 on List of Common for OIL.

Rewriting Eq. (4):

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

or

\[ \rho \frac{\partial r}{\partial t} + \rho u \frac{\partial u}{\partial t} + \rho v \frac{\partial v}{\partial t} = - \frac{P}{r} \frac{\partial u}{\partial r} - u \frac{\partial P}{\partial r} - \frac{v}{\partial z} \frac{\partial P}{\partial z} - P \frac{\partial v}{\partial z} \]

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

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

Rewriting the momentum equations as

\[ \rho \frac{\partial (u^2)}{\partial t} = - 2u \frac{\partial P}{\partial r} \quad \text{and} \quad \rho \frac{\partial (v^2)}{\partial t} = - 2v \frac{\partial P}{\partial z} \]

the radial momentum equation becomes in difference form

\[ \rho \frac{\partial (u^2)}{\partial t} = 2u_{j-1/2,j-1/2} \frac{r_{j+1/2} - r_{j-1/2}}{\Delta r_{j+1/2}} \]

and the axial momentum equation becomes

\[ \rho \frac{\partial (v^2)}{\partial t} = v_{j-1/2,j-1/2} \frac{[P_{j-3/2,j-1} - P_{j+1/2,j-1/2}]}{\Delta z_{j}} \]

and the change in specific internal energy becomes

\[ \rho \frac{\partial u}{\partial t} = - P \left[ \frac{v_{j-1/2,j+1/2} - v_{j-1/2,j-1/2}}{2 \Delta z_{j}} + \frac{u_{j+1/2,j-1/2} - u_{j-1/2,j-1/2}}{\Delta (r_{j})^2} \right] \]

Defining the velocity on the right-hand side of the momentum equations at time \( n+\frac{1}{2} \) results in
i is the right boundary and j is the top boundary of the cell

\[ x(i) = \sum_{i=1}^{i_{\text{MAX}}} \Delta x(i) \]
\[ y(j) = \sum_{j=1}^{j_{\text{MAX}}} \Delta y(j) \]

The area of cell \((i, j)\) in the \(i\) direction \(\Delta x(i) \Delta y(j)\)

The area of cell \((i, j)\) in the \(j\) direction \(\Delta x(i) \Delta y(j)\)

\(k\) is defined as \((j-1) i_{\text{MAX}} + i + 1\), and is the index for cell quantities.

Fig. 2
\[
\begin{align*}
\frac{\rho_{1,\frac{1}{2},1,\frac{1}{2}}}{\Delta t} & \left[ (u_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1})^2 - (u_{1,\frac{1}{2},j,\frac{1}{2}}^n)^2 \right] = 2r_{1,\frac{1}{2}}(u_{1,\frac{1}{2},j,\frac{1}{2}}^n) \frac{r_{1,\frac{1}{2},1,\frac{1}{2}}}{\Delta r_{1,\frac{1}{2}}} \\
\text{and} \\
\frac{\rho_{1,\frac{1}{2},1,\frac{1}{2}}}{\Delta t} & \left[ (v_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1})^2 - (v_{1,\frac{1}{2},j,\frac{1}{2}}^n)^2 \right] = (v_{1,\frac{1}{2},j,\frac{1}{2}}^n) \frac{r_{1,\frac{1}{2},1,\frac{1}{2}}}{\Delta x_j} \\
\text{and} \\
\frac{\rho_{1,\frac{1}{2},1,\frac{1}{2}}}{\Delta t} & \left[ I_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1} - I_{1,\frac{1}{2},j,\frac{1}{2}}^n \right] \\
= & - \frac{1}{2} \rho_{1,\frac{1}{2},j,\frac{1}{2}} \left[ \frac{v_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1} - v_{1,\frac{1}{2},j,\frac{1}{2}}^n}{\Delta x_j} + \frac{r_{1,\frac{1}{2},1,\frac{1}{2}} - r_{1,\frac{1}{2},1,\frac{1}{2}}}{\Delta r_{1,\frac{1}{2}}} \right] \\
\text{Defining:} \\
u_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1} = & \frac{u_{1,\frac{1}{2},j,\frac{1}{2}}^n + u_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1}}{2}. \\
v_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1} = & \frac{v_{1,\frac{1}{2},j,\frac{1}{2}}^n + v_{1,\frac{1}{2},j,\frac{1}{2}}^{n+1}}{2}. \\
p^n = & \frac{p_{1,\frac{1}{2}}^{n+1} + p_{1,\frac{1}{2}}^n}{2}. \\
p^n = & \frac{p_{1,\frac{1}{2}}^{n+1} + p_{1,\frac{1}{2}}^n}{2}. \\
\text{Equation (2), the radial momentum equation, becomes} \\
u_{(k)}^{n+1} - u_{(k)}^n = & 2\Delta t r_{1,\frac{1}{2}} \delta y(j) \frac{(PL^n - PRR^n)}{AMX(k)} \\
\text{where} \\
PL^n = & \frac{P_{(k)}^n + P_{(cell to the left)}^n}{2}. \\
PRR^n = & \frac{P_{(k)}^n + P_{(cell to the right)}^n}{2}. 
\end{align*}
\]
Equation (3), the axial momentum equation, becomes

$$v^{n+1}(k) - v^n(k) = \Delta t \left( r_1^2 - r_{1-1}^2 \right) \frac{(P^n_{BLO} - P^n_{ABOVE})}{\Delta x(k)}$$

where

$$P^n_{BLO} = \frac{P^n(k) + P^n_{\text{cell below}}}{2}$$

$$P^n_{ABOVE} = \frac{P^n(k) + P^n_{\text{cell above}}}{2}$$

The energy equation (4) becomes

$$T^{n+1}(k) - T^n(k) = \frac{P^n(k) \Delta t}{\Delta x(k)} \left[ \frac{(V^{n+1}_{BLO} + V^{n+1}_{ABOVE} - V^{n+1}_{ABOVE})}{2} \right]$$

$$(r_1^2 - r_{1-1}^2) + [D \frac{\partial}{\partial x}] \left( U_L^R + U_L^{n+1} - U_{RR}^n - U_{RR}^{n+1} \right)$$

where

$$V^{n+1}_{BLO} = \frac{V^n(k) + V^n_{\text{cell below}}}{2}$$

$$V^{n+1}_{ABOVE} = \frac{V^n(k) + V^n_{\text{cell above}}}{2}$$

$$U_L = \frac{u^n_{\text{cell on the left})RL}{2}$$

$$U_{RR} = \frac{u^n_{\text{cell to the right})RR}{2}$$

where

$$RC = r_{i-1}$$

$$RR = r_{i+1}$$

$$RL = r_{i-3/2}$$
The above equations conserve energy exactly, despite finite difference approximations. However, an adjustment at transmissive boundaries of the grid is necessary. This is a work term, which is also taken into account in ETH (the total energy of the system.) For the transmissive boundaries, the pressure gradient is zero and the velocity at the boundary interface is set equal to the velocity of the cell adjacent to the boundary.

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

\[ \frac{P(k) + P(\text{cell to the left})}{2} \cdot u(k) \frac{r_{i-\frac{1}{2}}}{\Delta t} \text{Dy}(j) \]

and the top is

\[ \frac{P(k) + P(\text{cell below})}{2} \cdot v(k) \frac{r_{i}^2 - r_{i-1}^2}{\Delta t} \text{dy}(j) \]

and the bottom, if transmissive, is

\[ \frac{P(k) + P(\text{cell above})}{2} \cdot v(k) \frac{r_{i}^2 - r_{i-1}^2}{\Delta t} \text{dy}(j) \]

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

The velocity terms in the energy equation for those cells at the transmissive boundaries are, at the right = \( u(k) r_{i-\frac{1}{2}} \) and the top = \( v(k) \).

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

\[ \frac{\rho_{n+1}^k - \rho_n^k}{\Delta t} = \left[ \frac{r_{i-1}^k \rho_{i-1}^k u_{i-1}^k}{r_{i-2}^k \Delta r_{i-1}^k} - \frac{r_{i}^k \rho_i^k u_i^k}{r_{i-2}^k \Delta r_i^k} + \frac{\rho_{i-1}^k v_{i-1}^k - \rho_i^k v_i^k}{\Delta A_i^k} \right] \]

where

\[ \Delta A_i^k = \frac{\Delta x}{A_{ij}^k} - \frac{\Delta x}{A_{ij-1}^k} \]

where \( \Delta x \) for all \( j = \pi(r_i^2 - r_{i-1}^2) \).
and
\[ V(k) = \text{volume of cell } k = 2\pi r_{i-\frac{1}{2}}^2 \Delta r_i \Delta z_j \] (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
\[ V(k) r_i = \Delta^r r_{i-\frac{1}{2}} \Delta r_i \] (8)
where \( \Delta^r \) = 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
\[ V(k) r_{i-1} = \Delta^r r_{i-\frac{1}{2}} \Delta r_i \] (9)

Solving Eqs. (8) and (9) for \( r_{i-\frac{1}{2}} \Delta r_i \) and substituting their values into Eq. (6) results in
\[ \frac{\rho^{n+1} - \rho^n}{\Delta t} = \frac{1}{V(k)} (\Delta^Z r_{j-1} \rho_{j-1} v_{j-1} - \Delta^Z r_j \rho_j v_j + \Delta^r r_{i-1} \rho_{i-1} u_{i-1} - \Delta^r r_i \rho_i u_i) \]
or rewriting in terms of mass,
\[ AMX^{n+1}(k) - AMX^n(k) = \Delta t [(Av)^Z_B \rho - (Av)^Z_T \rho + (Av)^Z_L \rho - (Av)^Z_R \rho] \]
where AMX = mass of cell k in grams and (Av) = area times a velocity (the velocity is a function of the velocity of the two cells in question.) B refers to the bottom, T to the top, L to the left, and R to the right of cell k. The \( \rho \) used is the \( \rho \) from the cell from which the flux is coming.

Various techniques for velocity weighting in the mass flux have been tried. Results from these trials are presented in Figs. 5 and 6. The scheme presently being used is as follows. Take the r direction as an example:
The mass to move across \( i \) is between \( i \) and \( ii \) where \( \Delta = i - ii \); thus
\[ \Delta = \tilde{u} \Delta t \]
where \( \tilde{u} \) is the weighted velocity at \( \Delta \). 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
\[ \frac{\Delta}{\Delta t} = \tilde{u} = \frac{u(k) + u(k+1)}{2} \frac{2}{(u(k+1) - u(k))\Delta t} + 1. \]
if \( \tilde{u} > 0 \), use \( \rho(k) \); if \( \tilde{u} < 0 \), use \( \rho(k+1) \) in the mass flux calculation.

Mass, both components of momentum, and the energy across all four sides of the cell 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.

A look ahead, two cells in both directions, is done to remove preferential mass transport because of the initial choice of indexing in the \( r \) direction first. Take the example where the flux out of the top and right are such that their sum would remove more than the mass in the cell. The code would then assign new fluxes such that the top flux would be its fraction of the total flux out times the mass of the cell, and the right flux would be its fraction of the total flux out times the mass of the cell.
To treat a free surface in the continuous Eulerian scheme, we have chosen to use a density cutoff to limit the mass from flowing through $N$ zones in $N$ time steps. If the mass flow across the free surface results in a density which is less than an input number $\left( \sim 10^{-3} \rho_0 \right)$ the flux is held back. To cut the small precursor ahead of the shock front, the velocities are checked against $10^{-8}$ cm/s; if they are smaller than this, they are set to zero.

To ensure that the bottom cells in the projectile will empty as the projectile moves up, a scheme using the $p$ and $v$ from the cell above is used to calculate the flux. This is continued until the initial velocity of the bottom cell of the projectile begins to change because of the shock. After this point is reached, no special procedure is used for the bottom cells of the projectile.

**Boundary Conditions**

These cells adjacent to the axis of symmetry ($r = 0$) have the following boundary conditions; the pressure on the left side of the cell is equal to the pressure of the first cell, and the velocity at the left is set to zero. The pressure at the right interface of a cell whose right neighbor is void is zero, and the velocity is that of the occupied cell; similarly for the case of a void cell above.

The pressure and velocity at a transmittive boundary are the following. The pressure at the boundary is set equal to the pressure at the left or bottom boundary respectively for a right and top transmittive boundary (no acceleration of the border cells) while the velocity is set equal to the border cell velocity.

The top and right boundary of the grid is transmittive; the bottom boundary may be either transmittive or reflective, where the same boundary conditions then will exist as for the top or right and the axis of symmetry.

Two passes are used to solve the change in internal energy due to the work terms. The first pass through, one calculates the new velocities at time $(n+1)$ and simultaneously, the velocities at the interfaces at time $(n)$. The time $(n)$ interface velocities are also used to evaluate the internal energy contributions due to terms involving these velocities (see Eq. 4.)
In a second pass the time (n+1) interface velocities are calculated and the associated contributions to the internal energies are computed. (A look-ahead feature of two cells in both directions would enable one to use only one pass.)

An option exists for correcting negative internal energies if they arise in phase 1. The cell where the maximum negative internal energy occurred is recorded; assuming the rate of change of internal energy with time is essentially constant, we calculate a smaller time step, such that the new internal energy will be positive. We complete the entire cycle, to time (n+1), now set the time step negative, integrate backward to time (n), and now forward with the new smaller Δt to a revised time (n+1).

**Time Control for Code**

The time control for the continuous Eulerian is the same as for the particle in cell scheme, with the exception of the r direction. In the z direction:

\[ Δm_z = ρ \bar{V} Δt \]

Assume \( \bar{V} = v(k) \)
\( ρ = ρ(k) \)
\( Δm_z = AM_z(k) \)

Then

\[ AM_z(k) = \frac{AM_x(k)}{πr^2(1) - r^2(i-1)} \frac{v(k) [r^2(i) - r^2(i-1)] Δt}{DY(j)} \]

\[ = AM_x(k)v(k) \frac{Δt}{DY(j)} \]

or \( |v(k)|Δt ≤ DY(j) \) such that the flux in the z direction will not empty the cell.

In the r direction, the stability is as follows.

\[ Δm_R = ρ \bar{u} Δt \]

Assume \( \bar{u} = u(k) \)
\( ρ = ρ(k) \)
Then
\[ AMx(k) = \frac{AMx(k)}{2\pi(r_{(i)}^{1-\frac{1}{2}})\Delta y(i)\Delta y(j)} u(k)2\pi r(i)\Delta y(j)\Delta t \]
\[ = AMx(k)u(k) \frac{r(i)}{r(i-\frac{1}{2})}\Delta r(i) \Delta t \]
or
\[ u(k)\Delta t \leq \frac{\Delta r(i)r(i-\frac{1}{2})}{r(i)} \leq \frac{TAU(i)}{2\pi r(i)} \]

C = speed of sound, defined as \( \sqrt{\gamma P/\rho} \) for a polytropic form of the equation of state and \( (\partial P/\partial \rho)^{\frac{1}{\gamma}} \) for a real equation of state. Provisions exist for calling either one.

The time control (\( \Delta t \)) conditions are the Courant condition and that the maximum \( |u_{\Delta x}| \) and \( |v_{\Delta y}| \) < \( \frac{1}{\Delta t} \).

Three options for time control are:
1. Code will control the \( \Delta t \), calculated from the Courant and particle velocity scheme, but at a fraction of stability.
2. The \( \Delta t \) loaded at \( t = 0 \) will remain constant, provided option for integrating backward in time to remove negative internal energies from phase 1 is not operating.
3. Code will control the \( \Delta t \), decreasing \( \Delta t \) if
\[ \frac{|u_{\Delta t}|}{\Delta x} \quad \text{or} \quad \frac{|v_{\Delta t}|}{\Delta y} \]

exceeds FFA (an input number), and increasing \( \Delta t \) if it is less than FFB (an input number.)

The stability check is omitted for a cell if the density of that cell is less than some input number. This prevents isolated debris of high velocity and small masses from controlling the the time step.
Corner Coupling

The question investigated here is the correctness of the mass transport which is done neglecting corner coupling. Below is an example of the comparison with a PIC-like transport:

Assume \( u = v \) for all four cells and

\[ v_{\Delta t} = \frac{A_x \text{ or } A_y}{2} \]

Where particles in the PIC scheme located in the shaded area will cross into zone 3 in one time step, the OIL code requires two time steps for mass to move into zone 3, first by the path of zone 1 to 2 or 4, and finally to zone 3.

In the case of very small time steps, it is seen that the above approximation is unimportant. We have chosen to run most of our problems at .5 stability. Further, from early test runs of an impact calculation, results did not change appreciably as the factor was varied from one-fourth to one-half to nine-tenths of stability.

The PIC Transport

The changes required to change from a continuous mass transport to the particle transport are:

1. The problem number must be negative.
2. The transport and rezone subroutines must be replaced with those for PIC transport.

Following is a brief discussion of the particle (PIC) transport characterizing SHELL. Two scratch tapes are required for the particles. SHELL reads in particle records from one tape, processes the particles updating the coordinates, writes them out on the other tape, and interchanges tape numbers.

Five variables are associated with each particle; the mass \( AM \), the \( r \) coordinate \( XL \), and \( Z \) coordinate \( YL \), the \( i \)-coordinate of the cell where the particle lies \( iwl \), and the \( j \)-coordinate of the cell where the particle
lies \( iw2 \). Thus, one computes the cell number where the particle is by
\[
k = (j-1) \times iMAX + i + l.
\]

The particles are moved with an area weighted velocity, which is basically a cell placed with the particle at the center. The overlay of this cell on the four cells in question times the velocity of that cell is summed for the four cells and then the weighted velocity is calculated by dividing through by the total area.

Example: For particle \( N \) the \( u \) component of velocity used to move the particle is
\[
\bar{u} = \frac{4}{\Sigma A_i u_i}
\]
and the \( v \) component is
\[
\bar{v} = \frac{4}{\Sigma A_i v_i}
\]

The particle is then moved with the area weighted velocities. If the particle does not leave the cell \( k \), no additional calculations are needed; process the next particle. If the particle leaves cell \( k \), it carries with it a mass, momentum, and internal energy.

By conserving momentum and total energy, one changes the quantities in the new cell. No changes except removing the particle mass from the mass of cell \( k \) are necessary for updating cell \( k \).

Example: Particle \( N \) moves from cell \( k \) to cell \( L \). Conserving both axial and radial momentum:
\[
(\bar{u}_2)^\prime = \frac{M_2 u_2}{\bar{M}_2} + \frac{m_{11}}{\bar{M}_2}
\]
and
\[
(\bar{v}_2)^\prime = \frac{M_2 v_2}{\bar{M}_2} + \frac{m_{11}}{\bar{M}_2},
\]
where the line above signifies the updated variable. Thus,
\[
\bar{u}_2 = \frac{M_2 u_2}{\bar{M}_2} + \frac{m_{11}}{\bar{M}_2}
\]
\[
\bar{v}_2 = \frac{M_2 v_2}{\bar{M}_2} + \frac{m_{11}}{\bar{M}_2}
\]
However, \( M_2 \) is not available at this stage (since \( M_2 \) has replaced \( M_1 \)); substitute \( M_2 = \frac{\bar{M}_2}{\bar{M}_2} - m \) results in \( u_2 = \frac{M_2}{\bar{M}_2} \) \((u_{11} - u_2) + u_2\); similarly for the \( v \) component. Note that \( u_{11} \) and \( v_{11} \) are \( \frac{\bar{M}_2}{\bar{M}_2} \) set equal to \( u_1 \) and \( v_1 \) unless there has been an elastic bounce off a reflective boundary (requiring the velocities to change sign to conserve momentum); then \( u_{11} \) and \( v_{11} \) are set to \(-u_1 \) and \(-v_1 \).

To calculate the new internal energy in cell \( (2) \) requires that we conserve total energy and momentum, resulting in the expression:
\[
M_1 (u_1^2 + v_1^2) + \frac{\bar{M}_1}{\bar{M}_2} (u_{11}^2 + v_{11}^2) + \frac{\bar{M}_1}{\bar{M}_2} (u_2^2 + v_2^2) =
\]
\[
(M_1 - m) I_1 + \frac{1}{2} \left[ (u_1^2 + v_1^2) + (u_{11}^2 + v_{11}^2) + (M_2 + m) I_2 \right] + \frac{1}{2} (M_2 + m) (u_{11}^2 + v_{11}^2)
\]
Solving for \( I_2 \) yields
\[
I_2 = \frac{m}{\bar{M}_2} \left( I_1 - I_2 + \frac{(u_{11} - u_2^2) + (v_{11} - v_2^2)}{2} \right) \left( 1 - \frac{m}{\bar{M}_2} \right)
\]
Substituting the new values for \( \bar{u}_2 \) and \( \bar{v}_2 \) from conserving momentum results in
\[
I_2 = I_2 + \frac{m}{\bar{M}_2} \left( I_1 - I_2 + \frac{(u_{11} - u_2^2) + (v_{11} - v_2^2)}{2} \right) \left( 1 - \frac{m}{\bar{M}_2} \right)
\]
Thus, after each particle is moved, the two cells involved are updated. After a particle record has been processed, the particles, with their new coordinates and i and j value of the new cell location, are then written on another tape, which will become the starting conditions for the next transport cycle. An option exists to call rezone if particles leave the top or the right of the grid.

**Viscosity**

The movement of mass across the cell boundaries give rise to force which is effective in reducing fluctuations that arise from the differencing technique (ref 1). This is of the form of a "true" viscosity, being proportional to the velocity gradient. This viscosity is present at all times, both in compressions and rarefactions. No additional (that is, a controllable) artificial viscosity is present in the version of oil.
3.2. Special Subroutines

The REZONE routine for the particle transport is different from that required for the continuous mass transport version. If material leaves the grid out of the right or top, a trigger is set to call REZONE.

The REZONE multiplies all dimensions by two, so that four old cells become one in the new grid. The total number of cells remains the same, and the target is doubled in depth and width by adding mass at the sides and back surface. This scheme does conserve total energy, and by conserving momentum, also, new velocities and internal energy are calculated. This will cause the total internal energy of the system to rise slightly.

For the particle rezone, it is not necessary to multiply all dimensions by two, but rather change the Δx's and Δy's by any prescribed amount. One adds new material with the same density, internal energy, and velocity distributions as are available in CLAM. The number of particles per cell to add is also an input number. For a more complete description, see Reference 3.

A subroutine SETUP is available to generate the initial grid (bypasses the generator code CLAM) if both the target and projectile are of the same density. The projectile must be a right circular cylinder. This routine assumes that all Δx's are the same and all Δy's are the same. An asterisk before the symbol implies it is floating point.

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>LOCATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Z(111)</td>
<td>111</td>
<td>Initial density g/cm³</td>
</tr>
<tr>
<td>* Z(112)</td>
<td>112</td>
<td>Initial pellet velocity cm/sh.</td>
</tr>
<tr>
<td>* DX(1)</td>
<td>7845</td>
<td>Δx in cm</td>
</tr>
<tr>
<td>* DY(1)</td>
<td>7897</td>
<td>Δy in cm</td>
</tr>
<tr>
<td>IMAX</td>
<td>33</td>
<td>Maximum number of zones in the r direction</td>
</tr>
<tr>
<td>JMAX</td>
<td>35</td>
<td>Maximum number of zones in the z direction</td>
</tr>
<tr>
<td>11</td>
<td>47</td>
<td>The i value of the exterior radius of the projectile</td>
</tr>
<tr>
<td>12</td>
<td>48</td>
<td>The j value of the top of projectile +2</td>
</tr>
<tr>
<td>* PROB</td>
<td>1</td>
<td>Any positive number for the problem</td>
</tr>
<tr>
<td>* BK(3)</td>
<td>237</td>
<td>Must be a positive number</td>
</tr>
<tr>
<td>*PK(4)</td>
<td>238</td>
<td>Set = 1</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>*PK(5)</td>
<td>239</td>
<td>Right boundary (i) of projectile</td>
</tr>
<tr>
<td>*PK(6)</td>
<td>240</td>
<td>Bottom (j)+1 of projectile</td>
</tr>
<tr>
<td>*PK(7)</td>
<td>241</td>
<td>Top (j) of projectile</td>
</tr>
<tr>
<td>*PK(8)</td>
<td>242</td>
<td>Set = 1</td>
</tr>
<tr>
<td>*PK(9)</td>
<td>243</td>
<td>Right (i) boundary of target</td>
</tr>
<tr>
<td>*PK(10)</td>
<td>244</td>
<td>Bottom (j)+1 of target</td>
</tr>
<tr>
<td>*PK(11)</td>
<td>245</td>
<td>Top (j) of target</td>
</tr>
</tbody>
</table>

And the usual input data to start OIL code from a CLAM tape.

A. Example of Input for OIL Using the Subroutine SETUP.

<table>
<thead>
<tr>
<th>OIL INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAS GAMMA= 1.5 RHO TARGET = RHO PELLET =1.</td>
</tr>
<tr>
<td>23531.01</td>
</tr>
<tr>
<td>1121.01</td>
</tr>
<tr>
<td>1111.1</td>
</tr>
<tr>
<td>13821.</td>
</tr>
<tr>
<td>21871.</td>
</tr>
<tr>
<td>245132.</td>
</tr>
<tr>
<td>33132.</td>
</tr>
<tr>
<td>789711.</td>
</tr>
<tr>
<td>2731.</td>
</tr>
<tr>
<td>6541.</td>
</tr>
<tr>
<td>1421.</td>
</tr>
<tr>
<td>2421.</td>
</tr>
<tr>
<td>4728.</td>
</tr>
<tr>
<td>11311.</td>
</tr>
<tr>
<td>13510.</td>
</tr>
<tr>
<td>1721.</td>
</tr>
</tbody>
</table>

This will make a binary tape that is equivalent to that made by the example in CLAM (see Section 2.). See Section 5.2 for format for CARDS subroutine.
3.3. Logic of OIL

The logic involved in following a given cell (K) from time t to t+Δt, or from cycle n to n+1.

\[ k = (j-1) \max + i + 1 \]

\[ \tau(1) = \pi(x_j^2 - x_{j-1}^2) \]

Fig. 3

1. **CUT Routine**

Here one calculates the pressure (P) for cell (k) where \( P(k) = f[\rho(k), T(k)] \) where \( \rho(k) = \frac{\text{AM}(k)}{\tau(1) \Delta y(j)} \). The speed of sound \( \frac{\gamma P}{\rho} \) or \( \frac{(\partial P}{\partial y} \frac{\gamma}{2} \) is then calculated and the Courant condition for stability, that \( \frac{\min (\Delta x \ or \ \Delta y)}{\max (\Delta x \ or \ \Delta y)} < \frac{1}{\Delta t} \) and the particle velocity criteria that the \( \min (U_x or V_y) < \frac{1}{\Delta t} \) are calculated. From these stability checks a new \( \Delta t \) is calculated or \( \Delta t \) remains the same (see options under description "common for OIL, symbol CABN."). The cycle number and the time are now advanced. (A radiation time step is also calculated, although radiation is not being used in this version of the code.)

2. **EDIT Routine**

The OIL code has four separate editing-like routines all included in the routine called EDIT. A section called short print displays the time, cycle number, the total internal and kinetic energy and total mass, and various other integral quantities such as momenta, and mass in various
angles (see Sect. 3.4.) A plot routine is also available in the EDIT routine which places an (x) (in an equal cell size grid, corresponding to the actual grid in OIL) if there is any mass in the cell; thus it is a display of mass movement from time to time. A long print routine may also be called for that edits on each page a column from the OIL grid which contains the coordinates of the column, and the cell quantities as a function of the row coordinates (see column identification in Sect. 5.2 FORTRAN listing of EDIT.)

The last option is a dump routine which dumps all necessary data for restarting the problem. This data may also be used for the automatic plotting routines.

Various input numbers (see Sect. 3.4) specify the frequency that these routines will be called for, and an input number specifying a cycle number to stop.

In the following discussion, please refer to Fig. 3.

3. PHI Routine

Here we integrate the two momentum equations and the change in internal energy due to the work terms. No material is moved at this time, and the transport terms are dropped. Using the new pressures and the time step which were computed in CDT, we now prepare to integrate the equations.

\( P_L(j) \), the pressure at interface (i-1) and \( u_L(j) \), the velocity at interface (i-1) are available from the previous column sweep on i-1.

\[
P_L(j) = \frac{P(k) + P(k-1)}{2}.
\]

\[
u_L(j) = \frac{r_{i-3/2} u^n_{(k-1)} + r_{i-1} u^n_k}{2}.
\]

The PBLO term, which was the PABOVE for cell (kB) and VBLO, VABOVE for cell kB, is also available for interface j-1.

\[
PBLO = \frac{P(k) + P(kB)}{2}.
\]

\[
VBLO = \frac{v(k) + v(kB)}{2}.
\]
Now we calculate terms at interface $i$ and $j$:

$$P_{R} = \frac{P(k+1) + P(k)}{2}, \quad U_{RR} = \frac{R_{i-1} u^n(k) + R_{i+1} u^n(k+1)}{2}.$$

$$P_{\text{ABOVE}} = \frac{P(k) + P(k+1)}{2}, \quad V_{\text{ABOVE}} = \frac{v(k) + v(k+1)}{2}.$$

Now we can integrate the two momentum equations

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial P}{\partial x}$$

or

$$u^{n+1}(k) = u^n(k) + \left(\frac{P(k) - P_{\text{ABOVE}}}{\Delta x(k)}\right) r_{i-\frac{1}{2}} \Delta t$$

and

$$\rho \frac{\partial v}{\partial t} = - \frac{\partial P}{\partial z}$$

or

$$\gamma^{n+1}(k) = \gamma^n(k) + \left(\frac{P_{\text{ABOVE}} - P(k)}{\Delta x(k)}\right) r_{i-\frac{1}{2}} \Delta t$$

Now one can add the work term due to velocities at cycle $N$ to the change in internal energy.

$$I^{n+1}_k = I^n_k + \frac{P(k)}{\Delta x(k)} \left[\left(u^n_{(j)} - U_{RR}^n\right) r_{i-\frac{1}{2}} \Delta t + \left(V_{\text{ABOVE}}^n - P_{\text{ABOVE}}^n\right) r_{i-\frac{1}{2}} \Delta t \right]$$

$$\rho \frac{\partial r}{\partial t} = - \rho \left(\frac{\partial v}{\partial z} + \frac{1}{r} \frac{\partial \phi}{\partial \theta}\right)$$

Now we make one more pass through the grid, this time omitting the momentum equations but calculating the velocity terms at the interface, where again we only have to calculate the data at interface $i$ and $j$. 
The specific internal energy is checked during both passes for negative values. If a negative value is found, we assume \( \frac{dI}{dt} \) is constant over the time step, and recompute a new \( \Delta t \) (not placing it in the \( \Delta t \) storage) that will prevent \( I \) from going negative. After the completion of integrating all values to cycle \( n+1 \), an option exists for removing these negative energies. If one selects the option, \( \Delta t \) is set equal to \(-\Delta t\), the code integrates backward two passes to cycle \( n \), then replacing \( \Delta t \) with the new, smaller \( \Delta t \) and now forward in time with two passes using the smaller \( \Delta t \).

4. **PH2 Routine**

Here we move the mass and approximate the transport terms in the momentum and energy equations that were omitted in Phase 1.
AVMP = mass flow across the right boundary
AMUR = radial momenta across the right boundary
AMVR = axial momenta across the right boundary
DELER = total specific energy across the right boundary
AMAY = mass flow across the bottom boundary
AMMU = radial momenta across the bottom boundary
AMNV = axial momenta across the bottom boundary
DELEB = total specific energy across the bottom boundary
GAMC(J) = mass flow across the left boundary
FLEFT(J) = radial momenta across the left boundary
YAMC(J) = axial momenta across the left boundary
SIGC(J) = total specific energy across the left boundary
AMPY = mass flow across the top boundary
AMUT = radial momenta across the top boundary
AMVT = axial momenta across the top boundary
DELET = total specific energy across the top boundary

Again, following a typical cell k (Fig. 4) the masses, the momentas, and the energies are now available at the left and bottom boundaries of cell (k) from the previous column sweep and the cell below.

Now, begin by calculating the mass flow at the top of cell (k).

\[ V_{\text{ABOVE}} = \frac{V(k) + V(kA)}{2} \]

then form

\[ \frac{V_{\text{ABOVE}}}{1 + \left[ \frac{V(kA) - V(k)}{\Delta y(j)} \right] \Delta t} \]

as the weighted velocity to use in the flux equation and store it in \( V_{\text{ABOVE}} \).

If it is positive, use \( \rho(k) \); if it is negative, we use \( \rho(kA) \). Now calculate the mass flow across the top as \( \Delta m_T = \rho(M) A_{(j)} V \Delta t \) where

- \( \overline{V} = V_{\text{ABOVE}} \)
- \( M = \text{donor cell} \)
or

\[ \Delta m_T = AMP = \frac{AMx(m)}{DY(j)} \cdot V_{ABOVE} (\Delta t) \]

Now we calculate the mass flow at the right boundary of cell \( k \). \( uRR \) is defined as \( \frac{u(k) + u(k+1)}{2} \). Then form

\[
URR = \frac{URR}{1 + \left[ \frac{u(k+1) - u(k)}{\Delta x(i)} \right] \Delta t}
\]

as the weighted velocity to use in the flux equation and store it in \( uRR \). The mass flow across \( i \) is then

\[ \Delta m_R = p(M)A(i)u \Delta t \]

where

- \( u = uRR \)
- \( N = i \) value of donor cell
- \( M = \) donor cell
- \( \text{PIDTS} = \frac{1}{n \Delta t} \)

or

\[ \Delta m_R = AMP = \frac{AMx(M)}{TAU(N)} \frac{x(i)}{\text{PIDTS}} 2(uRR) \]

Now check to see if these masses will more than empty the cell, since it is possible that the left and bottom flux were both negative. Search cells ahead in both directions to remove preferential mass movement. As an example, suppose the flux at the TOP (\( AMP \)) is positive and the flux at the right is positive, where their sum is larger than the mass in cell \( (k) \); then normalize the fluxes the following way. The flux out of the top is

\[
\frac{F_T[AMx(k)]}{F_T + F_R}
\]

and the flux out of the right is equal to

\[
\frac{F_R[AMx(k)]}{F_T + F_R}
\]

where \( F \) is a symbol for the mass flux.
The momenta associated with these masses are now computed. The sign of
the flux specifies the zone where the mass came from, thus at the top; the
radial component equals \( \text{AMUT} = \text{AMFY}(u(N)) \) and the axial \( \text{AMVT} = \text{AMFY}(v(N)) \)
where \( (N) \) = cell number of the donor cell.

The momenta for the right is \( \text{AMUR} = \text{AMMP}(u(N)) \) for the radial and
\( \text{AMVR} = \text{AMMP}(v(N)) \) for the axial component, where again \( (N) \) = cell number of
the donor cell.

The total specific energy that those mass fluxes carry is also
calculated at this time; for the top it is equal to

\[
\text{DELET} = I(n) + \frac{u^2(N) + v^2(N)}{2}.
\]

and for the right it is equal to

\[
\text{DELER} = I(n) + \frac{u^2(N) + v^2(N)}{2}.
\]

where again \( (N) \) = cell number of the donor cell.

The mass now in cell \( (k) \) is equal to \( \text{DELM} = \text{AMX}(k) + \text{GAMC}(j) + \text{AMMY} - \\
\text{AMFY} - \text{AMMP} \) which equals the original mass plus the mass flow across the
left, the bottom, and less the mass flow across the top and the right.

The total axial momenta that have come into or left cell \( (k) \) is
\( \text{SIGMV} = \text{YAMC}(j) + \text{AMVV} - \text{AMVT} - \text{AMVR} \) the momenta crossing the left
boundary plus the momenta crossing the bottom boundary less the momenta
crossing the top and the right boundary.

The total radial momenta that has come into or left cell \( (k) \) is
\( \text{SIGMU} = \text{FLKT}(j) + \text{AMMU} - \text{AMUT} - \text{AMUR} \) momenta crossing the left and bottom
boundary less the momenta crossing the top and the right.

Similarly we calculate the total energy that these fluxes have carried
\( \text{DELEK} = (\text{GAMC}(j)) \text{SIGC}(j) + (\text{AMMY}) \text{Deleb} - (\text{AMFY}) \text{Delet} - (\text{AMMP}) \text{Deler} \) the mass times the total specific energy at the left plus the similar term
for the bottom less the similar terms for the top and the right.
40

Now by conserving momenta and total energy, calculate the new specific internal energy and velocities of cell \( (k) \).

\[
M_{UL} + M_{UB} - M_{UT} - M_{UR} + M_{UK} = (\text{TOTAL MASS})\bar{u}
\]

where total mass = DEIM and

\[
M_{VL} + M_{VB} - M_{VT} - M_{VR} + M_{VK} = (\text{TOTAL MASS})\bar{v}
\]

and the new specific internal energy

\[
I(k) = \frac{E_L + E_B - E_T - E_R + E_k}{DEIM} - \frac{\bar{u}^2 + \bar{v}^2}{2}.
\]

and now AMX(k) set = to DEIM.

The subscripts L, B, T, R, refer to the left, bottom, top, and right. Now the values that we calculated at the right for cell \( (k) \) are now set to the left values for cell \( (k+1) \) and the top values for cell \( (k) \) now become the bottom values for cell \( (kA) \).

The limits of the DO loops on \( i \) and \( j \) are 11 and 12. A check is done to see if mass or energy has moved beyond 11 or 12 and then the counter is increased by 1. This check is also done in PHI. By using this scheme, we process only the active mesh at all times, in all the subroutines.

3.4. List of Common (OIL)

The location refers to the location of that symbol relative to the beginning of common. Since the beginning of common is assigned the same location for each subroutine, a program (CARDS) is available for changing any word in common. The z block is first in common for OIL.

Note that if one should change the dimensions of the arrays, he must be careful and also make the necessary changes of the locations in the following tables.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Location</th>
<th>No. of Words</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AID</td>
<td>706</td>
<td>1</td>
<td>--</td>
<td>Not used, since this is a one-material (x) code</td>
</tr>
<tr>
<td>AIX</td>
<td>707</td>
<td>3500</td>
<td>jerks/g</td>
<td>Specific internal energy (x) for cell (k)</td>
</tr>
<tr>
<td>AM</td>
<td>4207</td>
<td>130</td>
<td>g</td>
<td>Mass of particle (N) for PIC transport only</td>
</tr>
<tr>
<td>AMD</td>
<td>4337</td>
<td>1</td>
<td>--</td>
<td>Not used in this one-material code</td>
</tr>
<tr>
<td>AMK UR(16)</td>
<td>220</td>
<td>15</td>
<td>g</td>
<td>Storage (EDIT) for summing masses in given angles for editing</td>
</tr>
<tr>
<td>AMX</td>
<td>4338</td>
<td>3500</td>
<td>g</td>
<td>Total (x) mass in cell k</td>
</tr>
<tr>
<td>AREA</td>
<td>7838</td>
<td>1</td>
<td>--</td>
<td>Tag, used in PH2 (EUL and PIC)</td>
</tr>
<tr>
<td>BIG</td>
<td>7839</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>BOUNCE</td>
<td>7840</td>
<td>1</td>
<td>--</td>
<td>Tag used in particle PH2</td>
</tr>
<tr>
<td>CABLN</td>
<td>82</td>
<td>1</td>
<td>--</td>
<td>If &lt; 0 code controls Δt but at Z(139) of instability</td>
</tr>
</tbody>
</table>

Caution: You must load a Δt for this option

If = 0, code will control the Δt, decreasing Δt if \( \frac{\Delta u}{\Delta x} \) or \( \frac{\Delta v}{\Delta y} \) exceed FFA (an input number) and increasing Δt if \( \frac{\Delta u}{\Delta x} \) or \( \frac{\Delta v}{\Delta y} \) is less than FFB (an input number) |

This holds provided SN ≠ 0 If > 0, DT loaded will remain constant.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Location</th>
<th>No. of Words</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDXN</td>
<td>7841</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>DDVK</td>
<td>7842</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>DKE</td>
<td>7843</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>DVK</td>
<td>7844</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>DX</td>
<td>7845</td>
<td>52</td>
<td>cm</td>
<td>DX(i) = X(i) - X(i-1)</td>
</tr>
<tr>
<td>DY</td>
<td>7897</td>
<td>100</td>
<td>cm</td>
<td>DY(j) = Y(j) - Y(j-1)</td>
</tr>
<tr>
<td>E</td>
<td>7997</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>FD</td>
<td>7998</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>FS</td>
<td>7999</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>FX</td>
<td>8000</td>
<td>1</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>OUT</td>
<td>8001</td>
<td>1</td>
<td>--</td>
<td>Tag in particle PH2</td>
</tr>
<tr>
<td>P</td>
<td>8002</td>
<td>3500</td>
<td>jerks/cm³</td>
<td>Material pressure in cell k</td>
</tr>
<tr>
<td>PABOVE</td>
<td>11502</td>
<td>1</td>
<td>jerks/cm³</td>
<td>( \frac{P(k) + P_{\text{cell above}}}{2} )</td>
</tr>
<tr>
<td>Variable</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PBLO</td>
<td>11503</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIDTS</td>
<td>11504</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FK UR(31)</td>
<td>235</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PL,PR,GAMC</td>
<td>405</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH(lOO),SIG(C)</td>
<td>405</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPABOV</td>
<td>11505</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRR</td>
<td>11506</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUL</td>
<td>11507</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QDT</td>
<td>11508</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QK</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td>11509</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReZ</td>
<td>11510</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHO</td>
<td>11511</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL</td>
<td>11512</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RR</td>
<td>11513</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIG</td>
<td>11514</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td>11515</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWITCH</td>
<td>11516</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAB</td>
<td>205</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLM</td>
<td>11517</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAU</td>
<td>11518</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAUDETTS</td>
<td>11570</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAUDETX</td>
<td>11571</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>11572</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>15072</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. At

\[ \frac{1}{\Delta t} = \frac{1}{\Delta t} \text{ in (PhI)}; \]

2. \[ \text{Radial momentum in certain angles for editing} \]

3. \[ \text{Axial momentum in certain angles for editing} \]

4. \[ \text{Pressure in Phi, flux in Phi2, etc.} \]

5. \[ \text{Density} \]

6. \[ \text{Not used} \]

7. \[ \text{Not used} \]

8. \[ \text{Not used} \]

9. \[ \text{Not used} \]

10. \[ \text{Density} \]

11. \[ \text{Not used} \]

12. \[ \text{Not used} \]

13. \[ \tan(\alpha) \text{ (Table of, used in EDIT routine)} \]

14. \[ \text{Not used} \]

15. \[ \text{Not used} \]

16. \[ \text{Not used} \]

17. \[ \text{R component of velocity in cell (k)} \]

18. \[ \text{R component of velocity in cell (k)} \]

19. \[ \text{R component of velocity in cell (k)} \]

20. \[ \text{R component of velocity in cell (k)} \]
### UL, UR, Fleet.

| 205 | 200 | cm²/sh | \( \frac{U(k)RC + U(k-1)RCC}{2} \) where RCC = \( \frac{X(i-1) + X(i-2)}{2} \) and RC = \( \frac{X(i) + X(i-1)}{2} \) |

| URR  | 15073 | 1 | cm²/sh | \( \frac{[U(k)RC + U(k+1)RR]}{2} \). |
| UT   | 15074 | 1 | -- | Signal in PHI (decrease \( \Delta t \)). |
| UU   | 15075 | 1 | sh | New \( \Delta t \) if PHI integrates back for \( I < 0 \), set originally to \( 10^{-5} \). |
| UUU  | 15076 | 1 | -- | Not used. |
| UTEF | 15077 | 1 | cm/sh | R velocity component used to move particle when using PIC transport. |
| UVMAX | 15078 | 1 | l/sh | \( \frac{[\text{Max velocity}] + \text{Min}(\Delta X \text{ or } \Delta Y)}{2} \). |
| V    | 15079 | 3500 | cm/sh | Axial (Z) component of velocity for cell (k). |
| VABOVE | 18579 | 1 | cm/sh | \( \frac{[V(k) + V(\text{cell above})]}{2} \). |
| VBL0 | 18580 | 1 | cm/sh | \( \frac{[V(k) + V(\text{cell below})]}{2} \). |
| VEL  | 18581 | 1 | -- | Used as a tag in PHI and Max(\( y-1 \)) in CDT, and tag in JUL, PH2. |
| VK   | 18582 | 1 | cm/sh | Axial component of velocity in cell (k) for PIC transport. |
| VT   | 18583 | 1 | -- | Not used. |
| VTEF | 18584 | 1 | cm/sh | Z velocity component used to move particle when using PIC transport. |
| VV   | 18585 | 1 | -- | Not used. |
| VVABOV | 18586 | 1 | -- | Not used. |
| VVBLO | 18587 | 1 | -- | Not used. |
| W2   | 18588 | 1 | -- | Not used. |
| W3   | 18589 | 1 | -- | Not used. |
| WPS  | 18590 | 1 | -- | Working Storage. |
| WS   | 18591 | 1 | | |
| WSA  | 18592 | 1 | | |
| WSB  | 18593 | 1 | | |
| WSC  | 18594 | 1 | | |
| XX  | 152 | 53 cm | XX(2) = X(1) |
| XL  | 18595 | 130 cm | R coordinate of particle N |
| XLF | 18725 | 1 | Fraction of area on left to use in velocity weighting for PIC PH2. |
| XW  | 18726 | 1 cm | R coordinate of particle N at cycle (n-1) |
| XR  | 19727 | 1 | Fraction of area to the right to use in velocity weighting for PIC PH2. |
| X1  | 75 | 1 | -- | Not used |
| X2  | 79 | 1 | -- | Not used |
| Y   | 606 | 100 cm | Y(j) = top dimension of zone (i,j) |
| YL  | 18728 | 130 cm | Z coordinate of particle N |
| YLW | 18358 | 1 | -- | Fraction of area below to be used in velocity weighting for PIC PH2. |
| YN  | 13359 | 1 cm | Z coordinate of particle N at cycle (n-1) |
| YU  | 18060 | 1 | -- | Fraction of area above to be used in velocity weighting for PIC PH2. |
| YY  | 605 | 1 cm | YY(j+1) = Y(j) |
| (YY(2) = Y(1)) |
| Y1  | 80 | 1 | -- | Not used |
| Y2  | 81 | 1 | -- | Not used |
| Z   | 1 | 150 | -- | See pages where Z(1) through Z(150) are defined |
| ZMAX | 12361 | 1 cm | An up-to-date value of largest Y coordinate of all particles used in PIC PH2. |
| i  | 18362 | 1 | -- | Index (Working Storage) |
| ii  | 18363 | 1 |
| iF  | 18364 | 1 |
| iR  | 18365 | 1 |
| iWS | 18366 | 1 |
| iiSA | 18367 | 1 |
| iSB | 18368 | 1 |
| ...c | 18369 | 1 | -- |
| il  | 47 | 1 | -- |

The right boundary of the active grid + 2, MAX(i1) = iMAX
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i2</td>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td>i3</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>i4</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>IWI.</td>
<td>18870</td>
<td>130</td>
</tr>
<tr>
<td>J</td>
<td>19000</td>
<td>1</td>
</tr>
<tr>
<td>JN</td>
<td>19001</td>
<td>1</td>
</tr>
<tr>
<td>JP</td>
<td>19002</td>
<td>1</td>
</tr>
<tr>
<td>JR</td>
<td>19003</td>
<td>1</td>
</tr>
<tr>
<td>K</td>
<td>19004</td>
<td>1</td>
</tr>
<tr>
<td>KDT</td>
<td>41</td>
<td>1</td>
</tr>
<tr>
<td>KN</td>
<td>19005</td>
<td>1</td>
</tr>
<tr>
<td>KP</td>
<td>19006</td>
<td>1</td>
</tr>
<tr>
<td>KR</td>
<td>19007</td>
<td>1</td>
</tr>
<tr>
<td>KFM</td>
<td>19008</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td>19009</td>
<td>1</td>
</tr>
<tr>
<td>M</td>
<td>19010</td>
<td>1</td>
</tr>
<tr>
<td>MA</td>
<td>19011</td>
<td>1</td>
</tr>
<tr>
<td>MB</td>
<td>19012</td>
<td>1</td>
</tr>
<tr>
<td>MC</td>
<td>19013</td>
<td>1</td>
</tr>
<tr>
<td>MD</td>
<td>19014</td>
<td>1</td>
</tr>
<tr>
<td>ME</td>
<td>19015</td>
<td>1</td>
</tr>
<tr>
<td>IIZ</td>
<td>19016</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>19017</td>
<td>1</td>
</tr>
<tr>
<td>NK</td>
<td>19018</td>
<td>1</td>
</tr>
<tr>
<td>NKMAX</td>
<td>19019</td>
<td>1</td>
</tr>
<tr>
<td>NK1</td>
<td>19020</td>
<td>1</td>
</tr>
<tr>
<td>NO</td>
<td>19021</td>
<td>1</td>
</tr>
<tr>
<td>NR</td>
<td>19022</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
iW2 19023 130  --  = (j) value of cell (k) where particle (N) is located, used in PIC PH2.
12  1  150  Many  Fixed value of Z block
Fleft 205 100  Equivalenced, used for flux terms
YANC 304 100  in Eulerian continuous (PH2)
SIGC 504 100  

<table>
<thead>
<tr>
<th>Location</th>
<th>Symbol</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z(1)</td>
<td>PROB</td>
<td>--</td>
<td>Problem number (if positive, this is an OIL run; if negative, this is a PIC run)</td>
</tr>
<tr>
<td>Z(2)</td>
<td>CYCLE</td>
<td>--</td>
<td>Cycle number (floating point value)</td>
</tr>
<tr>
<td>Z(3)</td>
<td>DT</td>
<td>sh</td>
<td>$\Delta t$ hydro = $t^N - t^{n-1}$</td>
</tr>
<tr>
<td>Z(4)</td>
<td>PRINTS</td>
<td>--</td>
<td>Cycle frequency for short print</td>
</tr>
<tr>
<td>Z(5)</td>
<td>PRINTL</td>
<td>--</td>
<td>Cycle frequency for long print</td>
</tr>
<tr>
<td>Z(6)</td>
<td>DUMP7</td>
<td>--</td>
<td>Cycle frequency for binary tape dumps</td>
</tr>
<tr>
<td>Z(7)</td>
<td>CSTOP</td>
<td>--</td>
<td>Cycle number at which problem stops</td>
</tr>
<tr>
<td>Z(8)</td>
<td>PIDX</td>
<td>--</td>
<td>= $\pi = 3.1415927$</td>
</tr>
<tr>
<td>Z(9)</td>
<td>TMZ</td>
<td>g</td>
<td>Total ((x + )) mass at (t = 0) (calculated in CLAM)</td>
</tr>
<tr>
<td>Z(10)</td>
<td>GAM</td>
<td>--</td>
<td>Not used</td>
</tr>
<tr>
<td>Z(11)</td>
<td>GAMD</td>
<td>--</td>
<td>$1/\gamma - 1$ Computed in Input</td>
</tr>
<tr>
<td>Z(12)</td>
<td>CAMX</td>
<td>--</td>
<td>$1/\gamma_x - 1$</td>
</tr>
<tr>
<td>Z(13)</td>
<td>ETH</td>
<td>Jerks</td>
<td>Total energy (computed in CLAM 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.</td>
</tr>
<tr>
<td>Z(14)</td>
<td>FFA</td>
<td>--</td>
<td>Upper limit for stability and to calculate $\Delta t$, only if CABLE = 0.</td>
</tr>
<tr>
<td>Z(15)</td>
<td>FFB</td>
<td>--</td>
<td>Lower limit for stability and to calculate $\Delta t$, only if CABLE = 0.</td>
</tr>
<tr>
<td>Z(16)</td>
<td>TMXZ</td>
<td>g</td>
<td>Total ((x)) mass ((t = 0)) calculated in CLAM</td>
</tr>
<tr>
<td>Z(17)</td>
<td>TXMZ</td>
<td>g</td>
<td>Total ((x)) mass ((t = 0)) calculated in CLAM</td>
</tr>
<tr>
<td>Z(18)</td>
<td>XMAX</td>
<td>cm</td>
<td>$= X (#MAX)$</td>
</tr>
<tr>
<td>Z(19)</td>
<td>TXMAX</td>
<td>cm</td>
<td>2 (XMAX) (t = 0). calculated in CLAM</td>
</tr>
<tr>
<td>Z(20)</td>
<td>TYNAX</td>
<td>cm</td>
<td>2 (YMAX) (t = 0). calculated in CLAM</td>
</tr>
<tr>
<td>Z(2')</td>
<td>AMDM</td>
<td>g</td>
<td>Min((\cdot)) particle mass/2.; calculated in CLAM</td>
</tr>
</tbody>
</table>
Z(22) AM0M g
Z(23) DNI --
Z(24) IDMIN --
Z(25) PEF --
Z(26) DTNA sh
Z(27) CVIS --
Z(28) NPR --
Z(29) NPR1 --
Z(30) NC --
Z(31) NFC --
Z(32) NRC --
Z(33) iMAX --
Z(34) iMAXA --
Z(35) JMAX --
Z(36) JMAXA --
Z(37) KMAX --
Z(38) KMAXA --
Z(39) NMAX --
Z(40) ND --
Z(41) KDT --
Z(42) iMAX --
Z(43) NOG --
Z(44) NOPR --
Z(45) NIMAX --
Z(46) NJMAX --
Z(47) 11 --
Z(48) 12 --
Z(49) 13 --
Z(50) 14 --
Z(51) N1 --
Z(52) N2 --

Min (x) particle mass/2. Calculated in CLAM

(ETH - E)n-NPC/ETH

IR (ECK) Note Z(76) > IDMIN, problem will stop
and the edit routine will call dump.

Not used

z(n-1)

If < 0, bottom boundary is transmissive;
otherwise reflective boundary.

Index (Working storage)

""

Cycle number in fixed point.

Number of cycles between short prints

Index

Maximum number of zones in R direction

iMAX + 1

Maximum number of zones in Z direction

jMAX + 1

(kMAX)(jMAX) + 1

kMAX + 1

Total number of particles + 1, generated in
CLAM for PIC problem only.

Total number of (.) particles + 1 generated
in CLAM

Defined previously

Not used

Index

Index

New iMAX before adding new zones

New jMAX before adding new zones

Defined previously

Defined previously

Not used

Not used

Scratch tape number for particles if this is a
PIC run.

Scratch tape number for particles if this is a
PIC run.
<table>
<thead>
<tr>
<th>Variable (Z)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N3</td>
<td>Number of particle records generated if this is a PIC run.</td>
</tr>
<tr>
<td>N4</td>
<td>Number of particles per record (MAX = 127) if this is a PIC run.</td>
</tr>
<tr>
<td>N5</td>
<td>Not used</td>
</tr>
<tr>
<td>N6</td>
<td>Number of particles on last particle record if this is a PIC run</td>
</tr>
<tr>
<td>N7</td>
<td>Not used</td>
</tr>
<tr>
<td>N8</td>
<td>Not used</td>
</tr>
<tr>
<td>N9</td>
<td>Not used</td>
</tr>
<tr>
<td>N10</td>
<td>= i value of zone that is controlling Δt</td>
</tr>
<tr>
<td>N11</td>
<td>= j value of zone that is controlling Δt</td>
</tr>
<tr>
<td>NRM</td>
<td>= maximum number of Rad cycles/Hydro (input number)</td>
</tr>
<tr>
<td>TRAD</td>
<td>sh NR * Δt Rad = Δt Hydro; not used in this version</td>
</tr>
<tr>
<td>XNRG</td>
<td>Jerks Total energy of (x) material</td>
</tr>
<tr>
<td>SN</td>
<td>If = 0 code will decrease Δt to correct for I &lt; 0, if ≠ 0, those I &lt; 0 are left alone</td>
</tr>
<tr>
<td>DXN</td>
<td>Not used</td>
</tr>
<tr>
<td>RADER</td>
<td>g-cm/sh Total positive radial momentum (x only)</td>
</tr>
<tr>
<td>RADET</td>
<td>g-cm/sh Total positive axial momentum (x only)</td>
</tr>
<tr>
<td>RADB</td>
<td>g-cm/sh Total positive radial momentum (x) for material under target</td>
</tr>
<tr>
<td>DTRAD</td>
<td>-- Not used</td>
</tr>
<tr>
<td>REZFCT</td>
<td>-- If ≠ 0, PH2 will not trigger rezone</td>
</tr>
<tr>
<td>RSTOF</td>
<td>-- Not used in continuous version</td>
</tr>
<tr>
<td>SHELL</td>
<td>-- Not used</td>
</tr>
<tr>
<td>BBOUND</td>
<td>-- Not used in this version</td>
</tr>
<tr>
<td>TOZONE</td>
<td>g/cm³ Minimum density for mass flow at free surface</td>
</tr>
<tr>
<td>ECK</td>
<td>Energy check [\left(\frac{\text{ETH} - E}{\text{ETH}}\right)^n - \left(\frac{\text{ETH} - E}{\text{ETH}}\right)^{n-NPC}/NPC]</td>
</tr>
<tr>
<td>SBOUND</td>
<td>-- Fraction of (\Delta) in mass weighting velocity</td>
</tr>
<tr>
<td>X1</td>
<td>-- Not used</td>
</tr>
<tr>
<td>X2</td>
<td>-- Not used</td>
</tr>
<tr>
<td>Y1</td>
<td>-- Not used</td>
</tr>
<tr>
<td>Z(81)</td>
<td>Y2</td>
</tr>
<tr>
<td>Z(82)</td>
<td>CABLN</td>
</tr>
<tr>
<td>Z(83)</td>
<td>VISC</td>
</tr>
<tr>
<td>Z(84)</td>
<td>T</td>
</tr>
<tr>
<td>Z(85)</td>
<td>GMAX</td>
</tr>
<tr>
<td>Z(86)</td>
<td>WSGD</td>
</tr>
<tr>
<td>Z(87)</td>
<td>WSGX</td>
</tr>
<tr>
<td>Z(88)</td>
<td>GMADR</td>
</tr>
<tr>
<td>Z(89)</td>
<td>GMAXR</td>
</tr>
<tr>
<td>Z(90)</td>
<td>S1</td>
</tr>
<tr>
<td>Z(91)</td>
<td>S2</td>
</tr>
<tr>
<td>Z(92)</td>
<td>S3</td>
</tr>
<tr>
<td>Z(93)</td>
<td>S4</td>
</tr>
<tr>
<td>Z(94)</td>
<td>S5</td>
</tr>
<tr>
<td>Z(95)</td>
<td>S6</td>
</tr>
<tr>
<td>Z(96)</td>
<td>S7</td>
</tr>
<tr>
<td>Z(97)</td>
<td>S8</td>
</tr>
<tr>
<td>Z(98)</td>
<td>S9</td>
</tr>
<tr>
<td>Z(99)</td>
<td>S10</td>
</tr>
<tr>
<td>Z(100)</td>
<td>( f )</td>
</tr>
<tr>
<td>Z(101)</td>
<td>Jerks</td>
</tr>
<tr>
<td>Z(102)</td>
<td>g-cm/sh</td>
</tr>
<tr>
<td>Z(103)</td>
<td>g-cm/sh</td>
</tr>
<tr>
<td>Z(104)</td>
<td>Jerks</td>
</tr>
<tr>
<td>Z(105)</td>
<td>--</td>
</tr>
<tr>
<td>Z(106)</td>
<td>--</td>
</tr>
<tr>
<td>Z(107)</td>
<td>--</td>
</tr>
<tr>
<td>Z(108)</td>
<td>--</td>
</tr>
<tr>
<td>Z(109)</td>
<td>--</td>
</tr>
<tr>
<td>Z(110)</td>
<td>Jerks/g</td>
</tr>
<tr>
<td>Z(111)</td>
<td>g/cm³</td>
</tr>
<tr>
<td>Z(112)</td>
<td>cm/sh</td>
</tr>
<tr>
<td>Z(113)</td>
<td>--</td>
</tr>
<tr>
<td>Z(114)</td>
<td>--</td>
</tr>
<tr>
<td>Z(115)</td>
<td>$c/\text{cm}^3$</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
</tr>
<tr>
<td>Z(116)</td>
<td>--</td>
</tr>
<tr>
<td>Z(117)</td>
<td>jerks/g</td>
</tr>
<tr>
<td>Z(118)</td>
<td>--</td>
</tr>
<tr>
<td>Z(119)</td>
<td>jerks/cm$^3$</td>
</tr>
<tr>
<td>Z(120)</td>
<td>--</td>
</tr>
<tr>
<td>Z(121)</td>
<td>--</td>
</tr>
<tr>
<td>Z(122)</td>
<td>jerks/g</td>
</tr>
<tr>
<td>Z(123)</td>
<td>--</td>
</tr>
<tr>
<td>Z(124)</td>
<td>--</td>
</tr>
<tr>
<td>Z(125)</td>
<td>--</td>
</tr>
<tr>
<td>Z(126)</td>
<td>jerks/cm$^3$</td>
</tr>
<tr>
<td>Z(127)</td>
<td>--</td>
</tr>
<tr>
<td>Z(128)</td>
<td></td>
</tr>
<tr>
<td>Z(129)</td>
<td></td>
</tr>
<tr>
<td>Z(130)</td>
<td></td>
</tr>
<tr>
<td>Z(131)</td>
<td></td>
</tr>
<tr>
<td>Z(132)</td>
<td></td>
</tr>
<tr>
<td>Z(133)</td>
<td></td>
</tr>
<tr>
<td>Z(134)</td>
<td></td>
</tr>
<tr>
<td>Z(135)</td>
<td></td>
</tr>
<tr>
<td>Z(136)</td>
<td></td>
</tr>
<tr>
<td>Z(137)</td>
<td></td>
</tr>
<tr>
<td>Z(138)</td>
<td>$g/\text{cm}^3$</td>
</tr>
<tr>
<td>Z(139)</td>
<td>--</td>
</tr>
<tr>
<td>Z(140)</td>
<td>--</td>
</tr>
<tr>
<td>Z(141)</td>
<td>--</td>
</tr>
<tr>
<td>Z(142)</td>
<td></td>
</tr>
<tr>
<td>Z(143)</td>
<td></td>
</tr>
<tr>
<td>Z(144)</td>
<td></td>
</tr>
<tr>
<td>Z(145)</td>
<td></td>
</tr>
<tr>
<td>Z(146)</td>
<td></td>
</tr>
<tr>
<td>Z(147)</td>
<td>--</td>
</tr>
<tr>
<td>Z(148)</td>
<td>$A \times 10^5\text{cm/sec}$</td>
</tr>
<tr>
<td>Z(149)</td>
<td>B</td>
</tr>
<tr>
<td>Z(150)</td>
<td>$c$</td>
</tr>
</tbody>
</table>
See Ref. 5 for a more detailed description of the equation of state. For condensed states, 

$$P = \left[ a + \frac{b}{E_{0}^{2} + 1} \right] \frac{E}{V} + A\mu + B\mu^{2}$$

for expanded states, 

$$P = aE_{0} + \left[ \frac{bE_{0}}{E_{0}^{2} + 1} + A\mu \frac{E}{E_{0}^{2} + 1} \right] e^{-\beta(V - 1)}$$

$$\rho = \frac{1}{V}, \eta = \frac{E}{E_{0}}, \text{ and } \mu = \eta - 1.$$  

$E$ = specific internal energy  

Condensed form for states where  

$$\frac{V}{V_{0}} < 1 \text{ and } E < E_{S}$$

expanded form for states where  

$$\frac{V}{V_{0}} > 1 \text{ and } E > E_{S}$$

The following table contains the constants for the above equations.
EQUATION OF STATE DATA

<table>
<thead>
<tr>
<th>Loc.</th>
<th>Def.</th>
<th>Unit</th>
<th>$p$ (g/cm$^3$)</th>
<th>$a$</th>
<th>$E_0$ (g jersks)</th>
<th>$b$</th>
<th>$A$ (g jersks)</th>
<th>$V_0$</th>
<th>$E_s$ (g jersks)</th>
<th>$z$</th>
<th>$f$</th>
<th>$o$</th>
<th>$B$ (g jersks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>19.7</td>
<td>.5</td>
<td>2.25 (-5)</td>
<td>1.04</td>
<td>3.00 (-4)</td>
<td>.11</td>
<td>0.135 (-6)</td>
<td>10</td>
<td>10.5</td>
<td>2.5</td>
<td>(-4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cu</td>
<td>3.9</td>
<td>.5</td>
<td>3.25 (-5)</td>
<td>1.5</td>
<td>1.39 (-4)</td>
<td>.13</td>
<td>0.138 (-6)</td>
<td>5</td>
<td>5.0</td>
<td>1.1</td>
<td>(-4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fe</td>
<td>7.86</td>
<td>.5</td>
<td>9.5 (-6)</td>
<td>1.5</td>
<td>1.279 (-4)</td>
<td>.21</td>
<td>0.244 (-6)</td>
<td>5</td>
<td>5.0</td>
<td>1.5</td>
<td>(-4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Al</td>
<td>2.7</td>
<td>.5</td>
<td>5. (-6)</td>
<td>1.63</td>
<td>7.52 (-5)</td>
<td>.11</td>
<td>0.3 (-6)</td>
<td>5</td>
<td>5.0</td>
<td>6.5</td>
<td>(-5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Be</td>
<td>1.845</td>
<td>.55</td>
<td>1.75 (-5)</td>
<td>.62</td>
<td>1.173 (-4)</td>
<td>.11</td>
<td>0.5 (-5)</td>
<td>5</td>
<td>5.0</td>
<td>5.5</td>
<td>(-5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ti</td>
<td>4.51</td>
<td>.5</td>
<td>7. (-6)</td>
<td>.63</td>
<td>1.03 (-4)</td>
<td>.10</td>
<td>0.35 (-6)</td>
<td>5</td>
<td>5.0</td>
<td>5.</td>
<td>(-5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ni</td>
<td>8.96</td>
<td>.5</td>
<td>9. (-6)</td>
<td>1.33</td>
<td>1.912 (-4)</td>
<td>.11</td>
<td>0.285 (-6)</td>
<td>5</td>
<td>5.0</td>
<td>1.5</td>
<td>(-4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mo</td>
<td>10.2</td>
<td>.5</td>
<td>4.5 (-6)</td>
<td>1.02</td>
<td>2.713 (-4)</td>
<td>.10</td>
<td>0.23 (-6)</td>
<td>5</td>
<td>5.0</td>
<td>1.65</td>
<td>(-4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Th</td>
<td>11.69</td>
<td>.4</td>
<td>2.5 (-6)</td>
<td>.36</td>
<td>5.31 (-5)</td>
<td>.15</td>
<td>2. (-6)</td>
<td>9</td>
<td>0.88</td>
<td>5.</td>
<td>(-5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch$_2$</td>
<td>.92</td>
<td>.6</td>
<td>7. (-6)</td>
<td>.0</td>
<td>7.5 (-5)</td>
<td>1.1</td>
<td>2.4 (-6)</td>
<td>10</td>
<td>5.0</td>
<td>2.</td>
<td>(-6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pb</td>
<td>11.34</td>
<td>.4</td>
<td>1.5 (-6)</td>
<td>2.5</td>
<td>4.664 (-4)</td>
<td>1.1</td>
<td>2.6 (-7)</td>
<td>13</td>
<td>15.0</td>
<td>1.5</td>
<td>(-5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$Z(\lambda)$, $Z(\lambda)$, $Z(\lambda)$ Where

$C_0 = C_0 + AP^2$

$C_0$ in $10^{-5} \text{ cm/sec}$

$P$ in megabars
4. TEST PROBLEMS

4.1. Technique

A series of one-dimensional impact problems was undertaken to select the most appropriate scheme for the velocity weighting in the mass transport equation. An iron projectile traveling at a velocity of $1 \times 10^6$ cm/sec striking an iron target was used as the standard test problem. The projectile had five zones of 1-cm each; the target had 95 cones of 1-cm each. The equation of state used was that for iron.\(^{(5)}\)

Four different velocity weighting techniques were investigated to determine the best $\bar{u}$ to use in the mass transport expression $\rho u \Delta t$. These are illustrated below, for the case where the flow is from cell $k$ to cell $(k+1)$.

The four different approaches were as follows:

1. The donor cell scheme: $\bar{u} = u_k$

2. The Rich scheme\(^{(6)}\) (no density terms) $\bar{u} = u_k + \frac{u_{k+1} - u_{k-1}}{4}$

3. The OIL scheme $\bar{u} = \frac{u_k + u_{k+1}}{2} \frac{1 + \frac{[u_{k+1} - u_k] \Delta t}{\Delta x}}{1 + \frac{[u_{k+1} - u_k] \Delta t}{\Delta x}}$

and 4, another scheme where $\bar{u} = \frac{u_k + u_{k+1}}{2} \left[1 - \frac{[u_{k+1} - u_k] \Delta t}{\Delta x}\right]$.

The donor cell, Fig. 5, looks very good in the neighborhood of the shock front; however, behind the front (the rarefaction side) instability sets in. Rich's scheme (without his density weighting) is very similar to scheme 4. The present scheme 3 now used in the OIL code was chosen as the best representation of the shock front and the rarefaction (Fig. 6). However, the results indicate the possibility of using the donor cell (scheme 1) near the shock front and one of the other three schemes in the rest of the problem.
4.2. Comparison of SHELL and OIL

An iron projectile, 3-cm-diam by 3-cm long, strikes an iron target at a velocity of $4 \times 10^6$ cm/sec. The problem was run using both SHELL (PIC transport) and OIL (continuous mass transport.) The starting conditions and cell sizes were the same for both. There were 72 cells in the projectile and 2976 in the target, and 16 particles per cell were used in the SHELL run.

Figure 7 is a plot of the total positive axial momentum and the total positive radial momentum (in the target) vs time for the two problems. Figure 8 shows the shock pressure vs position as a function of time for the two problems. The agreement between the two techniques is very good, with the SHELL scheme overshooting the theoretical shock pressure by a somewhat larger amount. Figures 9 and 10 are mass flux plots of the two schemes vs zone number, again at $45^\circ$. Figure 11 is a pressure and compression plot of the two schemes as a function of zone number along a ray $45^\circ$ from the initial center at the projectile target interface. Figure 12 is the same information at a later time of 9.2 μsec. Throughout all the comparisons, the two schemes are seen to be in very good agreement, with the SHELL scheme exhibiting some undesirable oscillatory behavior due to the discrete nature of the particle population in the cells. Finally, the SHELL run required longer machine time by a factor of 15 over the OIL run.

4.3. One-dimensional Test Problems

Figure 13, a comparison of the two schemes for a plane wave free expansion problem, indicates good agreement between OIL and theory. A hot gas extending to 30 cm with a $p$ of .8 g/cm$^3$, a $\gamma$ of 5/3, and specific internal energy of $3 \times 10^{-3}$ joules/g, with a rigid wall on the left, was suddenly released and free to rarefy to zero pressure. The slight rise in mass in the leading edge is due to the constraint adopted for preventing mass transport to diffuse n cells in n time steps (see Section 3.1).

A second "shock tube" problem consisted of a rigid boundary on the left, a hot gas of $p = .8$ g/cm$^3$, a $\gamma$ of 5/3, and a specific internal energy of $9 \times 10^{-3}$ joules/g, which was allowed to shock a cold region extending from 30 cm to 40 cm, consisting of a $p$ of 1.2 g/cm$^3$ and a $\gamma$ of 5/3. In
addition to the OIL and SHELL versions of this flow, a third comparison was made using a plane-geometry version of a one-dimensional Lagrangian (SPITTER) code. The results are given in Figs. 14 and 15, where pressure and velocity are plotted as functions of distances in both the hot and cold material. The comparison between OIL and SPUTTER is very good, while SHELL, again, exhibits an oscillatory behavior.

Figure 16 shows the initial setup of a spherical blast problem which was run using the three codes, SHELL, OIL, and a spherical version of SPUTTER. Figure 17 is the pressure versus position for the three cases, and Fig. 18 is a plot of density versus position. In the latter plot particularly, SPUTTER displays a decidea advantage over the two Eulerian methods. This is due to the fact that the SPUTTER (Lagrangian) zoning was on an equal mass basis and the resolution is accordingly better at big R. Hence, there are many zones to represent the density discontinuity or shock location. The SHELL and OIL results are very similar, with the OIL results being substantially smoother.

Figures 19 and 20 are pressures vs position along the r and z axis for the SHELL and OIL codes. The results for OIL are especially interesting in that they are very nearly spherical, where the results for SHELL are not as smooth and are slightly different along the two axes, partially because of the preferential movement of particles that exist in the PIC transport of SHELL.

The spherical character of the OIL solution is also borne out by further tests, such as the close agreement, to about one-tenth of one percent, of the two components of velocity along a ray $45^\circ$ from the axis of symmetry. The fact that an initially spherical problem remains spherical is gratifying as evidence that the differencing or transport schemes are not introducing significant preferential treatments in the radial or axial directions.
REGION ① \( \rho = 0.8, \gamma = 5/3 \)
\( I = \text{SPECIFIC INTERNAL ENERGY} = 9 \times 10^{13} \text{ ERG} \)
\( U = V = 0 \)

REGION ② \( \rho = 1.2, \gamma = 5/3 \)
\( I = U + V = 0 \)

Fig. 16
NOTE: THE BELOW SET UP
DIMENSION, EQUIVALENCE AND COMMON IS
TO BE USED WITH ALL SUBROUTINES IN
CLAM, WITH THE EXCEPTION OF MAIN ROUTINE.

5.1. FORTRAN IV LISTINGS OF CLAM

INPU0020
INPU0030
INPU0040
INPU0050
INPU0060
INPU0070
INPU0080
INPU0090
INPU0100
INPU0110
INPU0120
INPU0130
INPU0140
INPU0150
INPU0160
INPU0170
INPU0180
INPU0190
INPU0200
INPU0210
INPU0220
INPU0230
INPU0240
INPU0250
INPU0260
INPU0270
INPU0280
INPU0290
INPU0300
INPU0310
INPU0320
INPU0330
INPU0340
INPU0350
INPU0360
INPU0370
INPU0380
INPU0390
INPU0400
INPU0410
INPU0420
INPU0430
INPU0440
INPU0450
INPU0460
INPU0470
INPU0480
INPU0490
INPU0500
INPU0510
INPU0520
INPU0530
INPU0540
NOTE. ALTHOUGH THE DIMENSIONS FOR THE CELL QUANTITIES ARE 4300 IN CLAM, THE DIMENSIONS FOR OIL ARE 3500. THAT IS, KEEP (IMAX)(JHAX) +1 LESS THAN 3499.
74.

**SUBFC MAIN**

**LIST,DECK,FOR**

**CMAIN**

**CALL**

**PLK**

**IF**

**CLAM**

**IF**

**CALL**

**INPUT**

**CALL**

**OUTPUT**

**EXIT**
SIBFIC INPUT

LIST, DECK, REF

SUBROUTINE INPUT

C
C
C

**** NOTE (1 MATERIAL ONLY (X))
MZ=150
C CLEAR Z BLOCK.
DO 30 I=1,MZ
30 Z(I)=0.0
C READ IN HEADING CARD
READ (5,8012)IW,IV
IW=I
WRITE (6,8012)(IW)!
WRITE (6,8100)
C
C READ IN PROBLEM CONSTANTS
C PROB=PROBLEM NO. AIMAX=IMAX,
C A1MAX=AJMAX, QCD0FL IS NOT USED-SET
C TO ZERO. SHELL SET=2., 58, 59 ARE
C ZERO, SET N7 TO=TAEPE NO.
READ (5,8004)PROB,AIMAX,AJMAX,QCD0FL,SHELL,S8,S9,N7
IF(K7)40,40,50
C MAX. NUMBER OF ZONES IN R DIRECTION.
MI=50
C MAX. NUMBER OF ZONES IN Z DIRECTION.
MJ=100
C MAX. NUMBER OF PARTICLES/CELL.
NXP=400
C SIZE OF TABLE (TAB)
JTM=500
C MAXIMUM I*J
C MAX. NUMBER OF CELLS.
MJ=4299
C CALCULATE ADDITIONAL INDICES FOR CLAM AND OIL.
70 IMAX=AIMAX
JMAX=AJMAX
IMAXA=IMAX+1
JMAXA=JMAX+1
KMAX=(IMAX*JMAX)+1
KMAXA=KMAXA+1
WRITE (6,6048)(PROB,IMAX,JMAX)
C CHECK INPUT NOS. CONCERNED WITH GRID SIZE.
101 IF(IMAX-MI)102,102,9901
102 IF(JMAX-MJ)104,104,9902
104 IF(KMAX-MIJ)1106,106,9903
106 NOD=1
NPC=1
NRC=0
C READ IN DX AND DY

I=0
J=0
X(I)=0.0
Y(J)=0.0

2000 READ (5,81C2)IWSA,IWSB,N1,N2,N3,M4,(TEMPIK),X(I),Y(I)
L=1

C COUNT NO. OF DIFFERENT DX OR DY.
2003 L=L+1
2004 L=L+1
2006 L=L+1
2008 IF(IWSB)2010,2010,2030

C PROCESS THE DX AND DY VALUES.
2010 DO 2014 N=1,L
NK=IZ(N+50)
DO 2012 K=1,NK
I=I+1
DX(I)=TEMP(N)
X(I)=X(I-1)+DX(I)
2012 CONTINUE
2014 CONTINUE
GO TO 2050

C CALC THE Y AND DX VALUES.
2030 DO 2034 N=1,L
NK=IZ(N+50)
DO 2032 K=1,NK
J=J+1
DY(IJ)=TEMP(N)
Y(IJ)=Y(IJ-1)+DY(IJ)
2032 CONTINUE
2034 CONTINUE
2050 IF(IWSA)2052,2000,2052
C IF(L) READ MORE DX OR DY DATA CARDS.
2052 IF(LJ-JMAX)9905,2053,9905
C CHECK INPUT NUMBERS.
2053 IF(I-I MAX)9906,2054,9906
2054 CONTINUE
READ (5,6004)IWS,IWSB,N5,N8,SWITCH
C N1, AND N2 ARE THE 2 SCRATCH TAPES.
N1=WS
N2=WSA
REMIN N1
REMIN N2
C N4=MAX. NUMBER OF PARTICLES-1 PER RECORD.
N4=W SB
NPRI=N4
NPRR=N4
C  CALCULATE THE AREA*S(TAU)=PI*(I(I-1))**2*
C  *
DO 1008 I=1,IMAX
   WSA=WSA+XI(I)**2
1008  TAU(I)=WS*(WSI-WSB)
C  WRITE OUT XI, Y, DX, DY, AND TAU VALUES.
C  WRITE (6,9066)IMAX,(DX(I),I=1,IMAX)
WRITE (6,9067)IMAX,(DY(I),I=1,IMAX)
WRITE (6,9092)IMAX,(TAU(I),I=1,IMAX))
C  PLOY IS REALLY PI*(3.1415927).
C  PLOY=WS
C  SET VELOCITIES, INTERNAL ENERGIES AND MASSES
C  TO 0.
DO 1014 I=1,KMAX
   U(I)=0.0
   V(I)=0.0
   AIX(I)=0.0
   AMX(I)=0.0
1014  CONTINUE
C  SET TOTAL ENERGY TO ZERO.
ETH=0.0
C  INITIALIZE MIN. MASS PARTICLE TO A LARGE NO.
AMDN=1.E+28
AMXM=AMDN
GO TO 2016
C  EROR
9901  NK=101
       GO TO 9999
9902  NK=102
       GO TO 9999
9903  NK=104
       GO TO 9999
C  JMAX DOES NOT EQUAL THE SUM OF THE INPUT J
9905  NK=2052
       GO TO 9999
C  IMAX DOES NOT EQUAL THE SUM OF THE INPUT I
9906  NK=2053
9999  WRITE (6,8888)NK,I,J,X,L,M,N
       PRINT 8888,NK,I,J,X,L,M,N
       CALL DUMP
2016  RETURN
78.

C FORMAT
8004 FORMAT(7E10.5,(2)
8012 FORMAT(I1,71H) IS THE CLAM PROGRAM AND THERE IS AN ERROR.
1
8048 FORMAT(1H/9H PROGRAM YU.FY.3,1ZL1.241=12,24X,2HJ=12)
8064 FORMAT(1H/10H X(1) I=0,12/(5F16.6))
8065 FORMAT(1H/10H Y(J) J=0,12/(5F16.6))
8066 FORMAT(1H/11H NX(I) I=1,12/(5F16.6))
8067 FORMAT(1H/11H NY(J) J=1,12/(5F16.6))
8092 FORMAT(1H/13H ARE(A(I) I=1,12/(5F16.6))
8100 FORMAT(1H/14H (SHELL INPUT))
8102 FORMAT(211,417,4E10.4)
8888 FORMAT(1H+/26H1 INPUT ERROR IN STATEMENTS, 12X, 12H INDICES ARE 617)
END
C  **** NOTE (1 MATERIAL ONLY (X))
C  READ IN GEOMETRY ETC.
NPP=7
NPR=NPP-1
TPIDY=PIDY*2.0
ND=0
NX=0
NT=1
NYY=1
C FIRST CARD OF EACH PACKAGE.
READ (5,8008)IX,LX,MX,TEMP(1),TEMP(2),TEMP(3)
C INITIALIZE THE NUMBER OF PACKAGES TO 0.
NPKG=0
2015 IF(IX-1)9901,2018,2018
2016 IX=I
LX=L
MX=M
C IF THERE ARE NO MORE PACKAGES GO COMPUTE TOTAL VALUES
2017 IF(IX-2)2018,7000,9902
2018 J=0
NFKG=NPKG+1
C SET PACKAGE MASS AND ENERGY TO 0.
PE=0.0
PM=0.0
C ORIGIN FOR THE RADIUS VECTORS TO BE USED
VC=TEMP(1)
XC=TEMP(2)
C S8 CONTAINS THE FIT NUMBER FOR THE
C PACKAGE IN QUESTION.
S8=TEMP(3)
WRITE (6,8100)(NPKGMX)
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)
IWS=1
IF(I-5)2021,2040,2022
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)2060,9903,2022
C IF GREATER, EITHER A TRIANGLE OR PERTURBED ELLIPSE.
2022 IF(I)9904,2030,2024
C A PERTURBED ELLIPSE.
2024 IWS=7
GO TO 2030

2026 IWS=3
2027 IF(L)9905,2030,2028
2028 IWS=5
2029 A TRIANGLE.
2030 IF(M)9906,2034,2032
2032 IWS=IWS+1
2034 J=J+1
2036 TAB STORAGE CONTAINS THE COORDINATES OF
C GEOMETRY.
ITAB(J)=IWS
DO 2036 N=1,NPR
J=J+1
2036 TAB(J)=TEMP(N)
GO TO 2020
C ONE ONLY RHO, I, U OR V ALLOWED PER PACKAGE
C IF=, THIS IS A DENSITY CARD.
2040 IF(L-1)9907,2046,2042
C IF GREATER, EITHER A VELOCITY OR ENERGY CARD.
2042 IF(L-3)2052,2058,9908
C IF=, THIS IS A VELOCITY CARD, IF LESS, THIS IS A
C ENERGY CARD.
2046 DO 2048 N=1,6
2048 TABR(N)=TEMP(N)
GO TO 2020
C DENSITY
2052 DO 2054 N=1,6
2054 TABI(N)=TEMP(N)
GO TO 2020
C ENERGY
2058 DO 2059 N=1,6
2059 TABUV(N)=TEMP(N)
GO TO 2020
C VELOCITY (U AND V)
C OUTPUT DENSITY, ENERGY, AND VELOCITY PARAMETERS
C ALL CARDS FOR THIS PACKAGE HAVE
C BEEN READ IN.
2060 IF(J-JTM)2070,2070,9915
C NO. OF PACKAGES EXCEED (72), NOTE
C JTM SET=TO 500 IN INPUT, THUS MAX.
C NO. OF PACKAGES = 72, UNLESS DIMENSIONS
C ARE CHANGED.
2070 WRITE (6,8C36)(TABR(II),II=1,6)
WRITE (6,8038)(TABI(II),II=1,6)
WRITE (6,8040)(TABUV(II),II=1,6)
C COMPUTE BOUNDARIES OF GEOMETRIES FOR EFFICIENCY IN
C GENERATING OR DELETING PARTICLES
2000 CALL PH2
C COMPUTE I(0), I(N), J(0) AND J(N), FROM PREVIOUSLY
C COMPUTED VALUES FOR UPPER AND LOWER LIMITS IN
C THE CELL MESH SCAN
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
   IXX=1
   IWS=IMAX-1
3800 IF(IWS)9929,3820,3801
3801 DO 3808 N=1,IWS
   IF(XIN)-GXN)3802,3804,3804
3802 IXN=IXN+1
   IXX=IXX+1
3804 IF(XIN)-GX)3806,3806,3808
3806 CONTINUE
   IF(XIN)3812,3812,3814
3812 IXN=1
3814 IF(IMAX-IXX)3816,3818,3818
   IXX=IMAX
3816 IF(YIN)-GY)3819,3817,3817
   IYN=IYN+1
3817 IF(YIN)-GYX)3815,3815,3813
3818 IYY=IYY+1
3813 CONTINUE
   IF(YIN)3824,3824,3826
3824 IYN=1
3826 IF(JMAX-IYX)3828,3830,3830
3828 IYY=JMAX
3830 IF(IYN)-IYX)3834,3834,9931
3834 WRITE 16,80441IXN,IXX,IXY
C SCAN CELL MESH TO DETERMINE IF PARTICLES ARE TO BE
C GENERATED OR DELETED
C GENERATE PARTICLES
4000 CALL PH3
C REARRANGE X,Y AND M FOR PARTICLES IF NECESSARY
6011 LA=NY-NT
   IF(LA)9947,6020,6022
6020 ND=ND+LA
   GO TO 6024
6022 NX=NX+LA
6024 NT=NY
   ETH=FTH+PE
C REPLACE 606760000000 BY -18115198976
82.

\[
WS = \text{ABS}(-18115198976)
\]

6026 IF(LX)9933, 6028, 6030
C REPLACE 244663000000 BY 22125740032
6028 WS = ABS(22125740032)
6030 WRITE (6,8501) LA, WS, PE, PM
C GO READ IN NEXT PACKAGE
6050 GO TO 2016
7000 NMAX = NT
C NMAX = MAX. NUMBER OF PARTICLES + 1.
C YOU HAVE PROCESSED ALL PACKAGES, ALL
C PARTICLES, NOW GO TO THE OUTPUT.
IF(M(2))4051, 4050, 4051
4050 N3 = NRC
GO TO 4060
4051 NRC = NRC + 1
N3 = NRC
C N3 = NO. OF PARTICLE RECORDS OF
C N4 WORDS.
IF(PROB)4052,4052,4060
4052 WRITE (N2)(AM(I), XL(I), YL(I), I1W(I), IW2(I), I=2, NPRI)
4060 N6 = NMAX - (N4 - 1)*(N3 - 1)
NPR = N3
REWIND N2
GO TO 10000
C ERROR
9901 NK = 2015
GO TO 9999
9902 NK = 2017
GO TO 9999
9903 NK = 2021
GO TO 9999
9904 NK = 2022
GO TO 9999
9905 NK = 2027
GO TO 9999
9906 NK = 2030
GO TO 9999
9907 NK = 2040
GO TO 9999
9908 NK = 2042
GO TO 9999
9915 NK = 2060
GO TO 9999
9929 NK = 3800
GO TO 9999
9930 NK = 3818
GO TO 9999
9931 NK = 3830
GO TO 9999
9933 NK = 6026
GO TO 9999
9947 NK=6011
9999 WRITE (6,8888)NK
PRINT 8888,NK
CALL DUMP
10000 RETURN
C
FORMATS
8008 FORMAT (211,I5,E13.5,5E10.5)
8036 FORMAT(1H07X,8H DENSITY 9X,1P6E16.6)
8038 FORMAT(1H07X,8H ENERGY 9X,1P6E16.6)
8940 FORMAT(1H07X,8H VELOCITY9X,1P6E16.6/1HO/)
8044 FORMAT(1H /6H I(1)=12,4X,5HJ(1)=12,4X,5H(N)=12,4X,5HJ(N)=12)
81000FORMAT(1HO//12H PACKAGE ND.13,120,15H PARTICLES/CELL//33X,2HA114XPH1 2660
1,2HA214X,2HA314X,2HA414X,2HA514X,2HA6)
85010FORMAT(1HO//12H (A3,1H) PARTICLES22X,4HPE =1PE12.6,16X,4HPM =E1PH1 2680
12.6)
8888 FORMAT(23H1PH1 ERROR IN STATEMENTI5)
END
SUBROUTINE PH2
CALCULATE THE PACKAGE GEOMETRIES

GENERATING OR DELETING PARTICLES

J=VALUE OF LAST COORDINATE READ IN.
JT=J

INITIALIZE OUTER BOUNDARIES.
GXN=XMAX
GYN=YMAX
GX=0.0
GY=0.0

NPP=7(SET IN PHI).

DO 3700 J=1,JTNPP

IWS STORED IN ITAB ARRAY IN PHI.
IF IWS=2(A TRIANGLE), IF=4(A RECTANGLE),
IF=6(A ELLIPSE OR CIRCLE. IF IWS=8, A
PERTURBED ELLIPSE. IF IWS IS LESS THAN
THese VALUES, THE DEFINITION STILL HOLDS, BUT
NOW DELETE THIS GEOMETRY.

KK=(ITAB(J)-I)/2
3007 IF(KK)9919,3010,3008
3008 IF(KK-2)3100,320093009
3009 IF(KK-4)3400,9920,9920

TRIANGLE
VERTICES CAN BE INPUTED IN ANY ORDER,
X COORDINATE FIRST.
SEARCH FOR THE LARGEST X(WSE) AND
SMALLEST X(WSD).

3010 IF(ITAB(J+1)-TAB(J+3))3011,3012,3013
3011 WSE=TAB(J+3)
WSD=TAB(J+1)
GO TO 3014
3012 TAB(J+1)=TAB(J+1)*1.0000001+1.0E-8
3013 WSE=TAB(J+1)
WSD=TAB(J+3)
GO TO 3014
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
3018 WSE=TAB(J+5)
GO TO 3024
3019 TAB(J+5)=TAB(J+5)*0.9999999-1.0E-8
3020 WSD=TAB(J+5)

ARRANGE VERTICES IN ASCENDING ORDER
3024 IF(TAB(J+2)-TAB(J+4))3036,3034,3038
3034 TAB(J+2)=TAB(J+2)*1.0000001+1.0E-8
GO TO 3036
3036 WSA=TAB(J+1)
WSB=TAB(J+2)
TAB(J+1)=TAB(J+3)
TAB(J+2)=TAB(J+4)
TAB(J+3)=WSA
TAB(J+4)=WSB
3038 IF(TAB(J+4)-TAB(J+6))3042,3040,3044
3040 TAB(J+6)=TAB(J+6)*0.99999999-1.0E-8
GO TO 3044
3042 WSA=TAB(J+3)
WSB=TAB(J+4)
TAB(J+3)=TAB(J+5)
TAB(J+4)=TAB(J+6)
TAB(J+5)=WSA
TAB(J+6)=WSB
GO TO 3024
C WSF=MINIMUM VALUE OF Y
C WSG=MAXIMUM VALUE OF Y
3044 WSF=TAB(J+6)
WSG=TAB(J+2)
C COMPUTE SLOPES
SLA=(TAB(J+4)-TAB(J+2))/(TAB(J+3)-TAB(J+1))
SLB=(TAB(J+6)-TAB(J+2))/(TAB(J+5)-TAB(J+1))
3053 IF(SLA-SLB)3054,9921,3058
3054 IF(SLA)3056,9922,3056
3056 IF(SLB)3064,9923,3062
3058 IF(SLA)3062,9924,3056
3062 WSA=TAB(J+3)
WSB=TAB(J+4)
WSC=SLA
TAB(J+3)=TAB(J+5)
TAB(J+4)=TAB(J+6)
SLA=SLB
TAB(J+5)=WSA
TAB(J+6)=WSB
SLB=WSC
3064 IF(TAB(J+3)-TAB(J+5))3066,9925,3068
3066 ITAB(J)=ITAB(J)+2
IWS=ITAB(J)-3
GO TO 3069
3068 IWS=ITAB(J)-1
3069 KE=J+1
KF=KE+5
C REPLACE 272545000000 BY 25058082816
WS= ABS( 25058082816)
IF(IWS)3072,3070,3072
C REPLACE 242543000000 BY 21836333056
3070 WS= ABS( 21836333056)
3072 WRITE (6,8016)WS,(TAB(N),N=KE,KF)
WS=TAB(J+2)-SLB*TAB(J+1)
86.

\[
\begin{align*}
\text{TAB}(J+1) &= \text{TAB}(J+2) - \text{SLA} \times \text{TAB}(J+1) \\
\text{TAB}(J+6) &= (\text{TAB}(J+6) - \text{TAB}(J+4)) / (\text{TAB}(J+5) - \text{TAB}(J+4)) \\
\text{TAB}(J+5) &= \text{TAB}(J+4) - \text{TAB}(J+6) \times \text{TAB}(J+3) \\
\text{TAB}(J+2) &= \text{SLA} \\
\text{TAB}(J+3) &= \text{WS} \\
\text{TAB}(J+4) &= \text{SLB} \\
\text{GO TO} 3600 \\
\end{align*}
\]

C    RECTANGLE

3100  \text{ITAB}(J) = \text{ITAB}(J+2) \\
      \text{IWS} = \text{ITAB}(J) - 5 \\
C    \text{REPLACE} 272545000000 BY 25058082816 \\
      \text{WS} = \text{ABS}(25058082816) \\
      \text{IF}(\text{IWS} = 3110, 3105, 3110) \\
C    \text{REPLACE} 242543000000 BY 21836333056 \\
      \text{3105 WS} = \text{ABS}(21836333056) \\
3110  \text{WRITE} (6, 8020) \text{WS}, \text{TAB}(J+1), \text{TAB}(J+2), \text{TAB}(J+3), \text{TAB}(J+4) \\
      \text{WSD} = \text{TAB}(J+1) \\
      \text{WSE} = \text{TAB}(J+2) \\
      \text{WAF} = \text{TAB}(J+3) \\
      \text{WGS} = \text{TAB}(J+4) \\
      \text{GO TO} 3600 \\
\]

C    ELLIPSE OR CIRCLE

3200  \text{IF}(\text{ABS}(\text{TAB}(J+1) - \text{TAB}(J+2)) - 1.0E-8) \leq 3300, 3302 \\
      \text{3202 IF}(\text{ITAB}(J+2) = 9926, 3300, 3203) \\
C    \text{ELLIPSE WITH NO PERTURBATION} \\
      \text{3203 ITAB}(J) = \text{ITAB}(J) + 2 \\
      \text{IWS} = \text{ITAB}(J) - 7 \\
C    \text{REPLACE} 272545000000 BY 25058082816 \\
      \text{WS} = \text{ABS}(25058082816) \\
      \text{IF}(\text{IWS} = 3210, 3205, 3210) \\
C    \text{REPLACE} 242543000000 BY 21836333056 \\
      \text{3205 WS} = \text{ABS}(21836333056) \\
3210  \text{WRITE} (6, 8024) \text{WS}, \text{TAB}(J+1), \text{TAB}(J+2), \text{TAB}(J+3), \text{TAB}(J+4) \\
      \text{WSD} = \text{TAB}(J+3) - \text{TAB}(J+1) \\
      \text{WSE} = \text{TAB}(J+3) + \text{TAB}(J+1) \\
      \text{WSF} = \text{TAB}(J+4) - \text{TAB}(J+2) \\
      \text{WSG} = \text{TAB}(J+4) + \text{TAB}(J+2) \\
      \text{TAB}(J+1) = \text{TAB}(J+1)**2 \\
      \text{TAB}(J+2) = \text{TAB}(J+2)**2 \\
      \text{GO TO} 3600 \\
\]

C    CIRCLE

3300  \text{ITAB}(J) = \text{ITAB}(J+4) \\
      \text{IWS} = \text{ITAB}(J) - 9 \\
      \text{TAB}(J+2) = \text{TAB}(J+1) \\
C    \text{REPLACE} 272545000000 BY 25058082816 \\
      \text{WS} = \text{ABS}(25058082816) \\
      \text{IF}(\text{IWS} = 3310, 3305, 3310) \\
C    \text{REPLACE} 242543000000 BY 21836333056 \\
      \text{3305 WS} = \text{ABS}(21836333056) \\
3310  \text{WRITE} (6, 8028) \text{WS}, \text{TAB}(J+1), \text{TAB}(J+3), \text{TAB}(J+4) \\
      \text{WSD} = \text{TAB}(J+3) - \text{TAB}(J+1) \\
      \text{WSE} = \text{TAB}(J+3) + \text{TAB}(J+1) \\
      \text{WSF} = \text{TAB}(J+4) - \text{TAB}(J+2) \\
      \text{WSG} = \text{TAB}(J+4) + \text{TAB}(J+2) \\
      \text{TAB}(J+1) = \text{TAB}(J+1)**2 \\
      \text{TAB}(J+2) = \text{TAB}(J+2)**2 \\
      \text{GO TO} 3600 \\
\]
GO TO 3215

ELLIPSE WITH PERTURBATION

3400 ITAB(J)=ITAB(J)+4

WS=1.0-(TAB(J+5)/TAB(J+1))**2

IWSA=ITAB(J)+7

OTAB(J+7)=(TAB(J+4)+TAB(J+2)*SQRT(WS))/

1

((TAB(J+5)-TAB(J+1)))**2)

IWS=ITAB(J)-11

KE=J+1

KF=KE+6

C REPLACE 272545000000 BY 25058082816

WSA=ABS(25058082816)

IF(IWS)3410,340593410

C REPLACE 242543000000C BY 21836333056

3405 WSA=ABS(21836333056)

3410 WRITE(6,8032)WSA,(TAB(N),N=KE,KF)

3415 IF(WS)-9927,9927,3420

3420 IF(TAB(J+3))9928,9925,9928

3425 TAB(J+3)=TAB(J+7)

ITAB(J+7)=IWSA

WSA=TAB(J+2)+TAB(J+1)/4.0

WSD=0.0

WSE=TAB(J+1)+TAB(J+1)/4.0

WSF=TAB(J+4)-WSA

WSG=TABLE(J+4)+WSA

C DETERMINE BOUNDARIES OF GEOMETRIES

3600 IF(WSD-GXN)3602,3604,3604

C MAXIMUM (X)

3602 GXN=WSD

3604 IF(WSE-GXX)3608,3608,3606

C MINIMUM (X)

3606 GXX=WSE

3608 IF(WSF-GYN)3610,3612,3612

C MAXIMUM (Y)

3610 GYN=WSF

3612 IF(WSG-GYX)3700,3700,3614

C MINIMUM (Y)

3614 GYX=WSG

3700 CONTINUE

J=JT

GO TO 10000

C ERROR

9919 NK=3007

GO TO 9999

9920 NK=3009

GO TO 9999

9921 NK=3053

GO TO 9999

9922 NK=3054

GO TO 9999
9923  NK=3056
     GO TO 9999
9924  NK=3058
     GO TO 9999
9925  NK=3064
     GO TO 9999
9926  NK=3202
     GO TO 9999
9927  NK=3415
     GO TO 9999
9928  NK=3420

9999  WRITE (6,8888)NK
     PRINT 8888,NK
     CALL DUMP

10000  RETURN

8016  FORMAT(15HOTRIANGLE    ---- A3,7H -------1PE16.6)
8020  FORMAT(15HORECTANGLE    ---- A3,7H -------1.6E16.6)
8024  FORMAT(15HOELLIPSE    ---- A3,7H -------1PE16.6)
8028  FORMAT(15HOCIRCLE    ---- A3,7H -------1PE16.6)
8032  FORMAT(15HOP ELLIPSE   ---- A3,7H -------1PE16.6)
8888  FORMAT(23:1PH2 ERROR IN STATEMENTS)
     END
SUBROUTINE PH3
C GENERATE (OR DELETE) THE PARTICLES
C
C SCAN CELL MESH TO DETERMINE IF PARTICLES ARE TO
C BE GENERATED OR DELETED
C
C SAVE CURRENT VALUES OF COUNTERS.
4000 IA=I
JA=J
IJ=K
JT=L
IF(IX-1)<9932,4010,9932
4010 IF(MX-MNP)<4012,4012,9935
IFIIF GREATER, YOU TRIED TO GENERATE MORE THAN
C 400 PARTICLES / CELL.
C
4012 WS=MX
FMX=SQR(T(WS))
MXS=FMX+.5
4011 IF(MX*MXS-MX)<9936,4013,9936
C IF(GREATER OR LESS) THE NO. OF PARTICLES / CELL
C THAT YOU REQUESTED WAS NOT N SQ. WHERE
C N IS FROM 1 TO 20.
4013 Mxa=1-MX
TFMX=.5/FMX
WPIDY=TPIDY/FMX
4015 IF(MXA)<4018,4024,9938
C IF GREATER, YOU HAVE FAILED TO SPECIFY THE
C NO. OF PARTICLES TO GENERATE.
4018 NY=NT
DO 5700 I=IX,IXX
C COMPUTE THE COORDINATE OF THE PARTICLE
C UNDER CONSIDERATION
5700 WS5=DX(I)/FMX
C THE VOLUME OF THE SUBDIVIDED CELL =
C PI(2.*XLI(NIDY/N*DY/N).
TABX(I)=X(I)-TFMX*DX(I)
4019 IF(MXA)<4020,4024,9938
4020 DO 4022 K=2,MX
C WE START AT THE RIGHT AND TOP OF CELL(K).
C SET UP ARRAY FOR X COORDINATES OF THE
C PARTICLES.
4022 TABX(K)=TABX(K-1)-WS5
C J LOOP, LIMITS OF Y FOR THIS PACKAGE.
4024 DO 5700 J=IY,IVX
TAB=WPIDY*WS5*DY(J)
C TAM= 2PI/N*DX/N*DY
C
E=0.0
IIIWS=0
IWS=0
IB=0
WS=0.0
TABY=0
TABY(I)=Y(I)-YMX*DY(I)

C
MXS=N
DO 4026 K=2,MAX

C
SET UP ARRAY FOR Y COORDINATES OF THE
C
PARTICLES.

C
4026 TABY(K)=TABY(K-1)-WS

C
K USED FOR THE CEL QUANTITIES.

K=(J-1)*MAX+I+1

C
4028 IB=IB/MXS

C
IB=IB+1

C
IB=MOD(1B,MXS)

C
TX=X COORDINATE OF PARTICLE IN QUESTION.

C
TX=TX(I)

C
TY=Y COORDINATE OF PARTICLE IN QUESTION.

C
TY=TY(I)

C
GENERATE / DELETE THE PARTICLE

C
ID=0

C
IG=0

C
DO 4200 L=1,JA,NPP

C
KK=ITAB(L)

C
IF(KK-5),4062,4064,4200

C
TRIANGLE

C
4062 WSX=(TY-TAB(L+1))/TAB(L+2)

C
IF(WSX-TX)4064,4066,4072

C
4064 WSX=TY-TAB(L+3)/TAB(L+4)

C
IF(WSX-TX)4200,4066,4066

C
4066 WSY=TAB(L+6)*TX+TAB(L+5)

C
IF(KK-2),14066,4072

C
4068 IF(WSY-TY)4200,4072,4070

C
4070 GO TO (4074,4076,4078,4080,4082,4076),KK

C
4072 IF(WSY-TY)4070,4070,4020

C
4074 ID=1

C
GO TO 4200

C
4076 IG=1

C
GO TO 4200

C
4078 KK=KK-4

C
4077 IF(KK-8),14079,4094,9939

C
4079 GO TO (4080,4080,4090,4090,4092,4092,4094),KK

C
RECTANGLE

C
4080 IF(TAB(L+1)-TX)4082,4082,4200

C
4082 IF(TAB(L+2)-TX)4200,4084,4084

C
4084 IF(TAB(L+3)-TY)4086,4086,4086

C
4086 IF(TAB(L+4)-TY)4200,4088,4088

C
4088 GO TO (4074,4076,4078),KK

C
ELLIPSE WITH NO PERTURBATION

PH3 1260
PH3 1270
PH3 1280
PH3 1290
PH3 1300
PH3 1310
PH3 1320
PH3 1330
PH3 1340
PH3 1350
PH3 1360
PH3 1370
PH3 1380
PH3 1390
PH3 1400
PH3 1410
PH3 1420
PH3 1430
PH3 1440
PH3 1450
PH3 1460
PH3 1470
PH3 1480
PH3 1490
PH3 1500
PH3 1510
PH3 1520
PH3 1530
PH3 1540
PH3 1550
PH3 1560
PH3 1570
PH3 1580
PH3 1590
PH3 1600
PH3 1610
PH3 1620
PH3 1630
PH3 1640
PH3 1650
PH3 1660
PH3 1670
PH3 1680
PH3 1690
4090 KK=KK-2
   IF((TX-TAB(L+3))**2/TAB(L+1)+(TY-TAB(L+4))**2
1/TAB(L+2)-1.0)4088,4088,4200
   C   CIRCLE
4092 KK=KK-4
   01F((TX-TAB(L+3))**2+(TY-TAB(L+4))**2-TAB(L+1))
1 4088,4088,4200
   C   ELLIPSE WITH PERTURBATION
4094 KK=KK-6
   01F((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 CONTINUE
   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(IWS)123,22,23
22 IWS=1
23 IWS=1
   NYY=NYY+1
   CALL P114 P113
   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(IWS)4335,4335,24
24 IWS=-1
4333 IF(AMX(K))9951,4335,4334
   C   CALCULATE PACKAGE ENERGY.
4334 E=(U(K)**2+V(K)**2)/(AMX(K))*.5+AIX(K)
   C   SET THE PARTICLE COORDINATES INTO THE
   C   PROPER ARRAYS.
4335 XL(N)=TX
   YL(N)=TY
   C   SET I AND J OF CELL LOCATION OF PARTICLE).
1W1(N)=I
1W2(N)=J
   C   CALCULATE PARTICLE MASS AS
   C   =2PI/N=DX/N=OY*X(N)*RHO.
   C   AM(N)=TAM*TX*WSR
4341 IF(LX)19945,342,434^4
4342 WS=AM(N)*WSI
   IF(AM(N)-ANDH)16,15,15
16 AMD=AM(N)
15 CONTINUE
   PN=PM+AM(N)
AM(N)=-AM(N)  
GO TO 4346  
4344 WS=AM(N)*WSI  
IF 'AM(NI-AMX(M)18,17,17  
18 AMX(M)=AM(N)  
17 AIX(K)=AIX(K)+WS  
PM=PM+AM(N)  
C SUM UP MASS, BOTH COMPONENTS OF MOMENTA  
C AND TOTAL INTERNAL ENERGY IN CELL K.  
AMX(K)=AMX(K)+AM(N)  
4346 U(K)=U(K)+ABS(AM(N))*WSU  
V(K)=V(K)+ABS(AM(N))*WSV  
IF(NY-NPRR)4800,14,9945  
14 NRC=NRC+1  
NPRR=NPRR+NPRI-1  
IF(PROB)5000,5000,5001  
C WRITE PARTICLES ON TAPE IF THIS IS TO  
C BE A PIC RUN.  
5000 WRITE (N2) (AM(N),XL(N),YL(N),IX1(N),IX2(N),N=2,NPRI)  
5001 NYY=1  
3 DU 2 N=2,NPRI  
C SET PARTICLE ARRAYS TO ZERO.  
XL(N)=0.0  
YL(N)=0.0  
AM(N)=0.0  
IX1(N)=0  
IX2(N)=0  
2 CONTINUE  
4800 IF(MX=1B)9946,4880,4028  
C A L C U L A T E E N E R G Y F O R P K G  
4880 IF(IWS)4900,5700,4900  
4900 IF(AMX(K))9951,5700,4910  
4910 PEE=(U(K)**2+V(K)**2)/(AMX(K)**2+5+AIX(K))  
4930 IF(E)4950,4910,4940  
4940 PEE=PEE-E  
4950 PE=PE+PEE  
5700 CONTINUE  
4 GO TO 10000  
C E R R O R  
9932 NK=4000  
GO TO 9999  
9935 NK=4010  
GO TO 9999  
9936 NK=4011  
GO TO 9999  
9937 NK=4015
GO TO 9999

9938 NK=4019
GO TO 9999

9939 NK=4077
GO TO 9999

9940 NK=4201
GO TO 9999

9941 NK=4310
GO TO 9999

9945 NK=4341
GO TO 9999

9946 NK=4803
GO TO 9999

9951 NK=4905

9999 WRITE (6,8888)NK,I,J,K,L,M,N
PRINT 8888,NK,I,J,K,L,M,N
CALL DUMP

10000 RETURN

8888 FORMAT(1H+/26H1 P H 3 ERROR IN STATEMENTI5,12X,12H INDICES ARE617)PH3 2750

END
SUBROUTINE PH4 DETERMINES WHICH ONE OF THE SIX POSSIBLE FITS TO CALL FOR, TO ASSIGN A DENSITY ENERGY, AND VELOCITIES TO PARTICLE N. THE FIT NUMBER IS SPECIFIED IN COLUMNS 31-40 OF THE FIRST CARD OF EACH PACKAGE (SEE SECTION 2.1)
SUBROUTINE FIT1
C
C    WS=SQRT(TTX**2+TTY**2)
C    DENSITY
C    WSR=TABR(1)+TABR(2)*(TTY-TABR(3))
C    ENERGY
C    WSI=TA81(1)+TA81(2)*TTY-TA81(3))
C    VELOCITIES
C    WS=TA8UV(1)+TA8UV(2)*TTY-TA8UV(3))
C    WSU=0.0
C    WSV=WS
C    RETURN
C    END

SUBROUTINE FIT2
C
C    WS=SQRT(TTX**2+TTY**2)
C    DENSITY
C    WSR=((TTX-TABR(1))/TABR(2))**2+((TTY-TABR(3))/
C    TABR(4))**2
C    ENERGY
C    WSI=TA81(1)+TA81(2)*TTX+TA81(3)*TTX**2
C    +TA81(4)*TTY+TA81(5)*TTY**2
C    VELOCITIES
C    WSV=TA8UV(1)+TA8UV(2)*TTY
C    WSU=TA8UV(3)+TA8UV(4)*TTY
C    RETURN
C    END
$IBFTC FIT3 LIST,DECK,REF
SUBROUTINE FIT3
C
C THIS FIT FOR SIN KZ/KZ ************
WS=SQR(TTX**2+TTY**2)
C DENSITY
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

$IBFTC FIT4 LIST,DECK,REF
SUBROUTINE FIT4
RETURN
END

$IBFTC FIT5 LIST,DECK,REF
SUBROUTINE FIT5
RETURN
END

$IBFTC FIT6 LIST,DECK,REF
SUBROUTINE FIT6
RETURN
END
**IBFTC** OUTPUT LIST DECK REF

**SUBROUTINE OUTPUT**

**COMMON **** OUTPUT ******

**NOTE (1 MATERIAL ONLY ((X)))**

**PACKAGES HAVE BEEN READ IN AND PROCESSED**

**COMPUTE TOTAL ENERGIES AND TOTAL MASSES**

\[ E = \text{ETH} \]

\[ \text{WRITE (6,8104)} \]

7001 ND=ND+1

\[ \text{IF(E)6000,6000,6001} \]

6000 AMDM=0.0

AMXM=0.0

GO TO 7016

6001 AMDM=AMDM/2.0

AMXM=AMXM/2.0

7013 IF(AMDM)9901,9901,7014

7014 IF(AMXM)9902,9902,7016

7016 ETH=0.0

TMDZ=0.0

TMXZ=0.0

DO 7012 I=2,KMAX

7005 IF(ANX(I))9904,T7012,9905

7006 CONTINUE

SUM UP TOTAL \((X)\) MASS IN GRID.

\[ \text{TMDZ} = \text{ANX(I)} + \text{TMDZ} \]

C

CALCULATE SPECIFIC INTERNAL ENERGY/CELL \((K)\).

\[ \text{AIX}(I) = \text{AIX}(I)/\text{AMX}(I) \]

C

CALCULATE radial and Axial velocities by

CONSERVING BOTH COMPONENTS OF MOMENTA.

\[ \text{U}(I) = \text{U}(I)/\text{WS} \]

\[ \text{V}(I) = \text{V}(I)/\text{WS} \]

C

SUM UP TOTAL ENERGY IN SYSTEM.

\[ \text{ETH} = \text{ETH} + ((\text{U}(I))**2 + \text{V}(I)**2)/2.0 + \text{AIX(I)} * \text{WS} \]

GO TO 7012

7012 CONTINUE

TMDZ=TMDZ+TMXZ

WRITE (6,8072)ETH,E,TMDZ,TMXZ,TMZ

IWS=ND-1

IWSA=NMAX-ND

IWSB=NMAX-1

WRITE (6,8073)(IWS,IWSA,IWSB)

C

PUT INPUT ON BINARY TAPE 7

7113 REWIND N7

C

WRITE TAPE FOR OIL CODE.
C OUTP1350
7163 N3=0 OUTP1360
7162 WS=555.0 OUTP1370
WRITE (N7)WS,CYCLE,N3 OUTP1380
WRITE (N7)Z(I),I=1,MZ OUTP1390
7131 WRITE (N7)(U(K),ViK),AMX(K),AIX(K),AIX(K),K=1,KMAXA) OUTP1400
GO TO 7140 OUTP1410
7140 CONTINUE OUTP1420
WRITE (NT)WSCYCLE,N3 OUTP1430
WRITE (NT)(4Z(I),I=1*MZ) OUTP1440
7131 WRITE (N7)(U(K),ViK),AMX(K),AIX(K),AIX(K),K=1,KMAXA) OUTP1450
GO TO 7140 OUTP1460
C OUTP1470
C WRITE PARTICLES ON DUMP TAPE FOR PIC RUN.
7150 DO 7160 I=1,N3 OUTP1480
7150 READ (N2)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,NPRI) OUTP1490
7161 WRITE (N7)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,NPRI) OUTP1500
7160 CONTINUE OUTP1510
7161 WRITE (NT)WS,WS$WS$WS OUTP1520
REWINO N7 OUTP1530
WRITE (6,8120)TPNC OUTP1540
J=JMAXA OUTP1550
CALL SLITE (0) OUTP1560
DO 7517 J=1,JMAX OUTP1570
CALL SLITE (1) OUTP1580
JMAXA OUTP1590
K=IWS+1 OUTP1600
DO 7517 J=1,JMAX OUTP1610
J=J-1 OUTP1620
J=J-1 OUTP1630
K=K-JMAX OUTP1640
7170 IF(AMX(K))9905,7517,7175 OUTP1650
7175 CALL SLITE(T,K000FX) OUTP1660
GO TO(7180,7185),K000FX OUTP1670
C PRINT OUT CELL QUANTITIES.
7180 WRITE (6,8080)I,X(I),DX(I) OUTP1680
7180 WRITE (6,8080)J,Y(J),DY(J),U(K),V(K),J(K),AIX(K) OUTP1690
1,V(K),AMX(K) OUTP1700
7517 CONTINUE OUTP1710
IF(QOO0FL)7520,7520,7616 OUTP1720
7616 REWINO N2 OUTP1730
GO TO 7520 OUTP1820
C ERROR
9901 NK=7013 OUTP1830
GO TO 9999 OUTP1840
9902 NK=7014 OUTP1850
GO TO 9999 OUTP1860
9904 NK=7005 OUTP1870
GO TO 9999 OUTP1880
9905 NK=7170 OUTP1890
C OUTP1900
9999 WRITE (6,8888)NK, I, J, K, L, M, N
PRINT 8888, NK, I, J, K, L, M, N
CALL DUMP
7520 RETURN
FORMAT OUTP1910
C
FORMAT OUTP1920
80720 FORMAT(1H ///6H THE =1PE16.9,7X,3HE =E16.9///5H M. =E11.5,5X,4HMXOUTP1960
1 =E11.5,7X,7HM, +MX =E11.5) OUTP1970
8073 FORMAT(1HO/17HOPARTICLES - - =12,4H DOT114,2H XI14,6H TOTAL) OUTP1980
8080 FORMAT(1HO///3HOI=I2,10X,2H=1PE13.7,10X,3HOX=E13.7/3HO J10X,1HY13OUTP1990
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 TIME10.1,7X,5HCYCLE14) OUTP2070
8888 FORMAT(1H+/26H1OUTPUT ERROR IN STATEMENT15,12X,12H INDICES ARE6I71OUTP2080.
END OUTP2090
*** NOTE THE FOLLOWING SET OF DIMENSIONS, COMMON AND EQUIVALENCE CARDS ARE TO BE USED FOR ALL SUBROUTINES WITH THE EXCEPTION OF MAIN AND CARDS.

**DIMENSION**

- AM(130), XL(130), YL(130),
- U(350), V(350), AMX(350), AlX(350),
- P(350),
- 3 IWL(130), IW2(130),
- 4DX(52), X(53), XX(54), OY(100), Y(100), YY(101),
- 5TAB(15), AMK(15), PK(15), QK(15), Z(150), 1Z(150),
- 6TAU(52), PL(200), PR(200), UL(200), UR(200),
- 7LEFT(100), YAMC(100), SIGC(100), GAMC(100),
- COMMON Z, XX, UR, PR, YY
- COMMON AIX, AM, AMD, AMX, AREA
- COMMON BIG, BOUNCE, DDXN, DDVK, DKE, DQV
- COMMON DX, DY, E, FD, FS, FX
- COMMON OUT, P, PABOVE, PBLU, PIDS, PPABOV
- COMMON PRR, PUL, WDT, RC, REZ, RHO
- COMMON RLRR, SIG, Q000FL, SWITCH, TABLM
- COMMON TAU, TAUDTS, TAUDTX, U, UK, URR
- COMMON UT, UU, UUU, UTEN, UVMAX, V
- COMMON VABOVE, VBLO, YEL, VK, VT, VTEF
- COMMON VV, VVABOV, VYBLO, W2, W3, WPS
- COMMON WS, WSA, WSB, WSC, XL, XLF
- COMMON XN, XR, YL, YLY, YN, YU
- COMMON ZMAX, I, VII, IN, IR, IWS
- COMMON IWSA, IWSB, IWSC, IWI, J, JN
- COMMON KRM, L, MA, MB, MC
- COMMON MD, ME, MZ, M, NK, N, NMAX
- COMMON NK1, NO, NR, IW2

**EQUIVALENCE**

- Q1Z, PROB), (Z2), CYCLE), (Z3), DT),
- (Z4), PRINL), (Z5), PRINT), (Z6), DUMP1T), (Z7), CSTOP),
- (Z8), PIDY), (Z9), TMZ), (Z10), GAN), (Z11), GAMD),
- (Z12), GAMX), (Z13), ETH), (Z14), FFA), (Z15), FFB),
- (Z16), TMDZ), (Z17), TMXZ), (Z18), XMAX), (Z19), TXMAX),
- (Z20), TYMAX), (Z21), ANDM), (Z22), AMMX), (Z23), DNN),
- (Z24), DMIN), (Z25), FEF), (Z26), DTNAI), (Z27), CVIS),
- (Z28), NPI), (Z29), NPIR), (Z30), NCI), (Z31), NPC),
- (Z32), NRC), (Z33), IMAX), (Z34), IHMAX), (Z35), JMAX),
- (Z36), JMAXA), (Z37), KMAX), (Z38), KMAXA), (Z39), NMAX)
OEQUIVALENCE  (Z(40),ND),  (Z(41),KDT),  (Z(42),IXMAX),  PH2 0530
1(Z(43),NOU),  (Z(44),NOPR),  (Z(45),NIMAX),  (Z(46),NMAX),  PH2 0540
2(Z(47),I1),  (Z(48),I2),  (Z(49),I3),  (Z(50),I4),  PH2 0550
3(Z(51),I1),  (Z(52),I2),  (Z(53),I3),  (Z(54),I4),  PH2 0560
4(Z(55),N1),  (Z(56),N2),  (Z(57),N3),  (Z(58),N4),  PH2 0570
5(Z(59),N1),  (Z(60),N10),  (Z(61),N11),  (Z(62),N2),  PH2 0580
6(Z(63),TRAD),  (Z(64),XNRG),  (Z(65),SN),  (Z(66),DNX),  PH2 0590
7(Z(67),RADER),  (Z(68),RADET),  (Z(69),RADAR),  (Z(70),DTRAD),  PH2 0600
8(Z(71),REZFT),  (Z(72),RSTOP),  (Z(73),SHELL),  (Z(74),BOUND),  PH2 0610
9(Z(75),TZONE),  (Z(76),ECK),  (Z(77),SBOUND),  (Z(78),X1),  PH2 0620
OEQUIVALENCE  (Z(79),X2),  (Z(80),Y1),  (Z(81),Y2),  PH2 0630
1(Z(82),CABLN),  (Z(83),VISC),  (Z(84),T),  (Z(85),GMAX),  PH2 0640
2(Z(86),WSGD),  (Z(87),WSG),  (Z(88),GMDR),  (Z(89),GMAXR),  PH2 0650
3(Z(90),S1),  (Z(91),S2),  (Z(92),S3),  (Z(93),S4),  PH2 0660
4(Z(94),S5),  (Z(95),S6),  (Z(96),S7),  (Z(97),S8),  PH2 0670
5(Z(98),S9),  (Z(99),S10),  PH2 0680
OEQUIVALENCE  (XX(2),XX(1)), (UR,UL,FLEFT), (UR(100),YAMC), PH2 0700
1(PR(100),SIGC),(PR,PL,GAMC),(UR,TAB), PH2 0710
2(UR(16),AMK),  (UR(31),PK),  (UR(46),QK),  (YY(2),Y(1)), PH2 0720

NOTE, THERE ARE 2 SPECIAL SUBROUTINES (FORTRAN 4) USED
IN THE OIL CODE, SUBROUTINE SLITE SERVES THE SAME FUNCTION
AS TURNING ON SENSE LIGHTS, AND SLITET SERVES THE
FUNCTION OF TESTING THE SENSE LIGHTS.

NOTE, IF AN ERROR (SEE THE END OF THE SUBROUTINES) OCCURS, THE
SUBROUTINE WILL CALL FOR A DUMP. BY CHECKING THE
VALUES OF NR AND NK, ONE CAN READILY IDENTIFY THE
STATEMENT NUMBER AND THE SUBROUTINE WHERE THE ERROR OCCURRED.

NK WILL CONTAIN THE STATEMENT NUMBER, AND NR IS
AN IDENTIFICATION FOR THE SUBROUTINE AS FOLLOWS,
NR=1  INPUT
NR=2  CDT
NR=3  PH1
NR=4  PH2
NR=6  EDIT
**Note:** Material only (XI)  

Input reads oil dump tape or  

Will call subroutine setup which  

Will make a dump tape for certain types of problem  

Also calculates dx and dy and equation of state data  

Call input  

Cdt routine calculates dt (hydro time step)  

And pressures, advance cycle no. etc.  

10 Call Cdt  

In edit, determine whether to execute a long  

Print, a short print, a tape dump, etc. and  

Calculate total energy in system (compare  

With eth) total mass, integrate total  

Components of mon. mass.  

Call edit  

Call slitet(1, k000fx)  

Sense lite 1 signifies this  

Is the last cycle of this run $\cdots$  

Lite turned on in the edit routine $\cdots$  

Go to (30, 20), k000fx  

Phi, integrate the momenta eqs. integrate  

Energy equation (only changes due to work  

Terms). No movement of mass here  

20 Call Phi  

Transport mass across boundaries (solve  

Mass transport eq. transport terms in  

The momenta and energy eqs. left out of  

Phi, here approximated by mass movement. conserve  

Mass, momenta and total energy.  

Call ph2  

Go to 10  

30 Call Exit  

End
SUBROUTINE INPUT
C
C TURN ON SENSE LITE 3.
CALL SLITE (3)
C
C READ HEADER CARD (COLUMNS 2-72).
READ (5,8004)IWS
WRITE (6,8004)IWS
C
CALL DATA.
6 CALL CARDS
C
C IF PK(3) = OR GREATER THAN ZERO, CALL ROUTINE
C SET-UP, OTHERWISE, BINARY OIL TAPE HAS BEEN MADE.
C READ IN DATA FROM OIL DUMP TAPE, OR
C GENERATE A DUMP TAPE FOR OIL, AND
C CALCULATE DX AND DY FROM THE X AND
C Y VALUES FROM TAPE.
88888 CALL CARDS
8887 CONTINUE
C
READ TAPE
C
GO READ BINARY TAPE.
GO TO 1000
C
READ IN REMAINING INPUT CARDS
10 CONTINUE
C
CALL CARDS
GO TO 2000
C
C SET THE PRESSURES TO ZERO.
40 DO 45 K=1,KMAXA
45 P(K)=0.0
C
INTEGRATE BACKWARDS ON CYCLE, TIME AND NO. OF
C CYCLES BETWEEN ENERGY CHECK, SINCL THESE
C ARE ADVANCED IN CDT.
T=T-DTNA
NC=NC-1
CYCLE=NC
NPC=NPC-1
UVMAX=0.0
C
CALCULATE THE DX'S, SINCE THESE ARE NOT ON
C TAPE.
DO 50 I=1,IMAX
50 DX(I)=X(I)-X(I-1)
C
CALCULATE THE DY'S, SINCE THESE ARE NOT ON
C TAPE.
DO 55 J=1,JMAX

55 DY(J)=Y(J)-Y(J-1) J=MZ-8
C PRINT Z BLOCK.
62 DO 80 I=1,J,8
K=I+7
DO 65 J=I+K
IF(Z(J))70,65,70
65 CONTINUE
GO TO 80
70 K=I+7
WRITE (6,8111)I,(Z(L),L=I,K)
80 CONTINUE
GO TO 10000
C
C READ BINARY TAPE.
1000 MZ=150
IWS=0
1003 REWIND 7
1004 READ(7)PR(1),PR(2),N3
NR=N3+5
1006 IF(IPR(1)-555.0)1010,1016,1010
1010 IWS=IWS+1
1011 IF(MOD(IWS,3))9902,9902,1003
1016 IF(IPR(2))110191018,1018
C CHECK HERE FOR THE CORRECT CYCLE NUMBER.
1018 IF(PR(2)-PR(2)1023t1023vl020 INPU1500
C SKIP OVER, LOOK AT NEXT CYCLE.
1020 DO 1022 L=2, NR
1022 READ(7)
GO TO 1004
1023 READ(7)(Z(I),I=1,MZ)
C CHECK FOR THE CORRECT PROBLEM NO.
IF(ABS(PROB-PK(I))-.01)1024,1024,9901
1024 READ(7)(UI(I),VI(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
READ(7)(Y(I),TAU(I),I=1,JMAX)
READ(7)(X(I),I=0,JMAX)
C NOTE, INITIALIZE N1 TO 2, AND N2 TO 3
C NOTE, N1 AND N2 ARE ONLY USED FOR BOOK-KEEPING ******
N1=2
N2=3
C NOTE, IF PROBLEM NO. IS NEGATIVE, THIS IS
C A PIC TRANSPORT, CHECK YOUR P42 ROUTINE,
C AND READ THE PARTICLES FROM TAPE ONTO
C SCRATCH TAPE N1.
IF(IPROB11,1,1034
1 REWIND 2
REWIND 3
DO 1025 .I=1,N3
1025
READ(7)(AM(N),XL(N),YL(N),IW1(N),IW2(N);N=2,N4)
WRITE(2)(AM(N),XL(N),YL(N),IW1(N),IW2(N);N=2,N4)
1025 CONTINUE
1034 READ(7)PR(1),PR(2),PR(3)
REWIND 2
1036 IF(PR(1)=555.0)9504,1040,1038
1038 IF(PR(2)=666.0)9905,1040,9905
1040 GO TO 10
C**** END OF READ TAPE ********************************************
C
C CALCULATE MAX. GAMMA AND GAMMA/(GAMMA-1.).
C
2000 IF(WSGX)9906,2010,2005
2005 GAMX=1.0/WSGX-1.0 INPU1680
2010 WSGX=(GAMX+1.0)/GAMX INPU1690
2012 IF(WSGD)9907,2020,2015
2015 GAMD=1.0/WSGD-1.0 INPU1700
2020 WSGD=(GAMD+1.0)/GAMD INPU1710
2025 GMAX=WSGX INPU1720
2030 GO TO 40 INPU1730
C**** END OF RES **********************************************
C
C ERROR
9901 NK=1023 INPU1740
GO TO 9999 INPU1750
9902 NK=1011 INPU1760
GO TO 9999 INPU1770
9904 NK=1036 INPU1780
GO TO 9999 INPU1790
9905 NK=1038 INPU1800
GO TO 9999 INPU1810
9906 NK=2000 INPU1820
GO TO 9999 INPU1830
9907 NK=2012 INPU1840
9999 NR=1 INPU1850
CALL DUMP INPU1860
10000 RETURN INPU1870
C
C FORMATS
8000 FORMAT(7E10.3,I2)
8004 FORMAT(I17,I1H)
8111 FORMAT(I4,8014)
END
SUBROUTINE CARDS

DIMENSION TABLE(1),CARD(7),LABLE(1)

COMMON TABLE

C 2 IN COLUMN 1, ROUTINE WILL FIX THE
C FLOATING PT. NO.
C 1 IN COLUMN 1, MEANS THIS IS LAST CARD TO
C READ IN.
EQUIVALENCE(TABLE(1),LABLE(1))

WRITE (6,10)

1 READ (5,11)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
WRITE (6,12)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
DO 4 I=1,NUMWPC
J=LOC+I-1
IF(IEND-2)2,5,2
5 LABLE(J)=IFIXICARD(I))
GO TO 4
4 CONTINUE
2 TABLE(J)=CARD(I)
3 RETURN
C FORMATS
10 FORMAT(20H1 OIL INPUT CARDS///)
11 FORMAT(1,15,I1,L0P7E9.4)
12 FORMAT(1H 14,17,I3,L0P7E14.6)
END
SUBROUTINE SETUP

C WILL ONLY GENERATE (1) MATERIAL.
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(J) 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(111).
  RHO=Z(111)
C LOAD INITIAL PELLET VELOCITY INTO Z(112).
  VTEF=Z(112)
C CLEAR ALL CELL ARRAYS.
DO 1 K=1,KMAX
  U(K)=0.0
  V(K)=0.0
  P(K)=0.0
  AMX(K)=0.0
  AIX(K)=0.0
1 CONTINUE
C CALCULATE DX, X, TAU
DO 10 I=2,IMAX
  DX(I)=DX(I-1)
  X(I)=X(I-1)+DX(I)
  WS=X(I)**2
  PIDY=3.1415927
  TAU(I)=WS*PIDY
10 CONTINUE
Y(I) = DY(I)

C CALCULATE DY AND Y.
DO 20 J = 2, JMAX
Y(J) = Y(J-1) + DY(I)
DY(J) = DY(I)
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) = RHO*DY(J)*TAU(I)
V(K) = VTEF
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
AMX(K) = RHO*DY(J)*TAU(I)
60 K = K + IMAX
50 CONTINUE

IMAX = IMAX
JMAX = JMAX
SHLLE = 2.0
CYCLE = 0.0
DT = 0.0
NMAX = 0
N1 = 2
N2 = 3
N3 = 0
N4 = 127
XMAY = X(IMAX)
TMAX = XMAX*2.0
YMAX = Y(JMAX)
TMAX = YMAX*2.0
REWIND 7
WS = 555.0
C WRITE OUTPUT FOR OIL ON TAPE.
WRITE ( 7) NS, CYCLE, N3
WRITE ( 7) ( Z(I), I = 1, 150)
WRITE ( 7) ( V(I), ANX(I), AIX(I), P(I), I = 1, KMAY)
WRITE ( 7) ( X(0), (X(I), TAU(I), I = 1, IMAX)
WRITE ( 7) ( Y(I), I = 0, JMAX)
WS = 666.0
WRITE ( 7) WS, WS, WS
REWIND 7
RETURN
SUBROUTINE CDT

C CHECK COUJANT-condition and particle velocity.
C RECORD I AND J OF ZONE WHERE DT IS BEING CONTROLLED.

3000 VEL=0.0
3005 DO 3050 I=1,11
3010 K=I+1
3015 DO 3050 J=1,12
3020 IF(ANX(K)J9901,3050,3025
3025 CALL ES
3030 IF(ABS(P(K))-1.0E-20)3035,3040
3035 P(K)=0.0
3040 IF(WSGX-VEL)3045,3050
3045 VEL=WSGX
3050 K=K+INAX
3055 KDT=I
3060 UVXAX=.0
3070 DO 3255 I=1,11
3075 K=I+1
3095 DO 3255 J=1,12
3100 KP=K.INAX
3115 IF(ANX(K)J9901,3255,4
3120 SIG=DY(J)
3125 SIG=DY(J)
3130 IF(SIG(148)<Z(138))3130,3130
3130 SIG(148)=Z(138)+Z(150)*WSA*Z(150)
3140 WS=SQRT(GAMMA*TAU(I)*DY(J)*ABS(P(K))/AMX(K))
GO TO 3205
3150 IF(SIG(148)>4000)4000,4001
4000 WS=SQRGAMMA*TAU(I)*DY(J)*ABS(P(K))/AMX(K))
GO TO 3205
4001 WS=ABS(P(K))+1.0E+4
110.

```
WS=WS*1.E-3
3205 WS=WS/SIG
3210 IF(UVMAX-WS)3215,3220,3220
3215 N10=I
N11=J
UVMAX=WS
3220 IF(NMAX)1,1,2
C EULERIAN CHECK FOR RADIAL PARTICLE VELOCITY.
1 CONTINUE
3 WS=ABS(U(K))/TAU(I)*X(I)/.5*KIY
GO TO 3225
C PIC CHECK FOR RADIAL PARTICLE VELOCITY.
2 WS=ABS(U(K))/DX(I)
3225 IF(UVMAX-WS)3230,3235,3235
3230 UVMAX=WS
N10=I
N11=J
3235 WS=ABS(V(K))/DY(J)
3240 IF(UVMAX-WS)3245,3250,3250
3245 N10=I
N11=J
UVMAX=WS
3250 CONTINUE
3255 K=K+IMAX
IF(UVMAX)9912,9912,3260
C FOR OPTIONS ON CABLE, CHECK
C SECTION 3.4 IN GAMM-5580.
3260 IF(CABLE)90,91,3300
90 DT=.5/VEL/UVMAX*Z(139)
GO TO 3295
91 WS=UVMAX*DT
WS=0.5/VEL
3265 IF(FFA-WSA)327c,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/WS/0.9
3295 KDT=0
C INTEGRATE THE TIME AND CYCLE COUNTER.
3300 T=T+DTNA
85 IF(DTRAD)9911,80,81
80 NR=NR
84 WS=NR
TRAD=DT/WS
GO TO 82
81 IMS=DT/DTRAD
NR=IMS+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
C
   NEGATIVE MASS
9901 NK=3020
   GO TO 9999
9909 NK=3305
   GO TO 9999
9910 NK=3310
   GO TO 9999
C
   THE DT WILL BE 0. OR NEGATIVE, STOP
9912 NK=1
   GO TO 9999
9911 NK=85
9999 NR=2
   CALL DUMP
3325 RETURN
8000 FORMAT (1THOCHANGE DT ..., T=1PE9.3,11H
  DT(N)=1PE9.3,13H  DTCDT=1PE9.3,13H)
SUBROUTINE PHI

VELOCITIES, ENERGIES, PRESSURES ARE AT THE CENTER OF THE CELL.
(2) PASSES THRU PHI ARE REQUIRED. NO MASS IS MOVED IN PHI.

********* NOTE 1 MATERIAL ONLY (X) ***********

=========================================================================

NRT=0
NRC=0
UU=1.E+15
UT=0.0

YOU WILL GET BACK HERE IF AIX WAS LESS THAN 0. AND PROVIDED SN=0.

8000 VEL=1.0

INITIALIZE MID-POINTS OF FIRST AND SECOND CELL IN R DIRECTION.

3301 RC=DX(I)/2.0
RR=(X(1)+X(2))/2.0
3304 K=2

AXIS OF SYMMETRY BOUNDARY CONDITIONS.
DO 3302 J=IJMAX
PL(J)=P(K)
UL(J)=0.0
3302 K=K+IMAX

FIRST PASS THRU, CALCULATE U AND V AT CYCLE N+1, AND THE WORK TERMS USING U AND V FROM CYCLE N.
SECOND PASS THRU, CALCULATE ONLY THE CONTRIBUTION TO THE CHANGE IN INTERNAL ENERGY FROM WORK TERMS EVALUATED FROM U AND V AT CYCLE N+1.

DO 3360 I=1,II
K=I+1
IF(CVISI.7002,7003,7003
7002 VBLO=V(K)
PBL=0.0
GO TO 7004
7003 VBLO=0.0
PBL=P(K)
7004 TAUDTS=TAU(I)*DT
II= MAX.(I) OF DISTURBANCE IN R DIRECTION.
C I2= MAX(J) OF DISTURBANCE IN Z DIRECTION.
C DO LOOP IN J DIRECTION
DO 3348 J=1,12
PIDTS=1.0/(PIDY*DT*DY(J))
C K= INDEX OF CELL IN QUESTION.
C N= INDEX OF CELL ABOVE.
N=K+IMAX
3305 IF(ANX(K))9902,3340,3306
3306 IF(IMAX=I9903,3311,3310
3310 IF(ANX(K+1))9904,3312,3314
C WE ARE AT THE RIGHT BOUNDARY, SET PRESSURE
C GRADIENT TO 0. IN R DIRECTION, MODIFY ETH.
C FOR RIGHT BOUNDARY BEING TRANSMITTIVE.
3311 PRR=PL(J)
3307 ETH=ETH-PRR*U(K)/PIDTS*RC
GO TO 3313
C RIGHT BOUNDARY CONDITION FOR THE MOMENTUM EQ.
C ADJACENT TO EMPTY CELL.
3312 PRR=0.0
3313 URR=RC*U(K)
GO TO 3316
C CALCULATE PRESSURE AT INTERFACE(I) AND
C (RU) FOR WORK TERM.
3314 PRR=(P(K)+P(K+1))/2.0
3315 URR=(U(K)+RC+U(K+1)+RR)/2.0
3316 IF(AMX(K))9905,3318,3320
C SET PRESSURE GRADIENT TO 0. THIS IS FOR TOP
C BOUNDARY BEING TRANSMITTIVE.
3318 PABOVE=PBLO
C MODIFY ETH FOR TOP BOUNDARY CONDITION.
3319 ETH=ETH-PABOVE*V(K)/2.0*TAUDTS
GO TO 3323
3320 IF(AMX(N))9906,3324,3322
C TOP BOUNDARY CONDITION (EMPTY CELL ABOVE.)
C TOP BOUNDARY CONDITION FOR VELOCITY (EMPTY CELL ABOVE).
3322 PABOVE=0.0
3323 VABOVE=V(K)
GO TO 3328
C CALCULATE PRESSURE AT INTERFACE(J)
3324 PABOVE=(P(K)+P(N))/2.0
3325 IF(CVIS)7001,3325,3325
7001 IF(I=J3325,7000,9905
C BOTTOM BOUNDARY IS TRANSMITTIVE, SET PRESSURE
C GRADIENT TO 0.
C AND MODIFY ETH.
7000 PBLO=PABOVE
ETH=ETH+PBLO*V(K)/2.0*TAUDTS
C VELOCITY AT INTERFACE(J)
3328 IF(VEL)9907,3404,3400
3325 IF(VEL)9907,3404,3400
C COMPUTE DELTA U AND DELTA V.

3400 V(K)=V(K)+(PLO-PABOVE)*TAUDTS/(AMX(K))
  IF(ABS(V(K))=1.E-08)3401,3402
3401 V(K)=0.0
3402 U(K)=U(K)+(PL(J)-PRR)/(AMX(K))*RC/PIDTS*2.0
  IF(ABS(U(K))=1.E-08)3403,3404
3403 U(K)=0.0

C CHECK FOR ADVANCING COUNTERS OF THE ACTIVE GRID IN THE R DIRECTION.

3404 IF(I-11)6016,6005,6005
6005 IF(U(K)16605,6606,6605
6605 NRC=1
6606 IF(V(K)15607,6004,6604
6607 NRC=1
6608 IF(AIX(K))601,6016,6015
6015 CONTINUE

C HERE CALCULATE CHANGE IN INTERNAL ENERGY DUE TO WORK TERMS ONLY.

W$=VBL$-V$ABOVE)*TAUDTS/2.0*P(K)
RHO=W$+UL(J)-URR)/PIDTS*P(K)

C CONVERT TO SPECIFIC INTERNAL ENERGY.

W$=AIX(K)+RHO/AMX(K)
GO TO 1000

C CHECK FOR NEGATIVE INTERNAL ENERGIES.

1000 IF(W$X)1011,1001,1001
1001 AIX(K)=W$X
GO TO 3342
1011 JT=1.0

C COMPUTE NEW DT STORE IN UU ASSUMING THAT DI/DT WILL BE THE SAME FOR A SMALLER TIME STEP, THE NEW DT IS CHOSEN SUCH THAT AIX AT N+1=2/3 OF AIX AT N.

WSA=2.0*AIX(K)/3.0*DT/AIX(K)-WS$X)
1013 IF(W$A-UU)1014,1001,1001
1014 UU=W$A
GO TO 1001

C CELL (K) IS EMPTY, SET INTERFACE QUANTITIES, ASSUMING CELL TO THE RIGHT AND TOP ARE NOT VOID.

3340 PRR=0.0
URR=U(K+1)*RR
PABOVE=0.0
VABOVE=V(N)

C SET RIGHT QUANTITIES TO THE LEFT (FOR NEXT COLUMN SWEEP) AND SET ABOVE QUANTITIES TO BELOW FOR NEXT CELL ABOVE.

3342 V$LO=V$ABOVE
  PL(J)=PRR
  UL(J)=URR

GO TO 1013
I15.

```
K=N
3348 PBL=PA

PHI 1880
LL=K-I\text{MAX}

PHI 1890
C CHECK FOR ADVANCING COUNTERS OF THE ACTIVE

PHI 1900
GRID IN Z DIRECTION.

PHI 1910
IF(U(LLL)6000,6001,6000

PHI 1920
6000 NRT=1

PHI 1930
6001 IF(V(LLL)6002,6003,6002

PHI 1940
6002 NRT=1

PHI 1950
6003 IF(AIX(LLL)6017,6018,6017

PHI 1960
6017 NRT=1

PHI 1970
6018 CONTINUE

PHI 1980
3355 RC=RR

PHI 1990
RR=(X(I+1)+X(I+2))/2.0

PHI 2000
CONTINUE

PHI 2010
3360 IF(VEL19911,1000C,3363

PHI 2020
3363 VEL=0.0

PHI 2030
GO TO 3301

PHI 2040
C ERROR

PHI 2050
9902 NK=3305

PHI 2060
GO TO 9999

PHI 2070
9903 NK=3306

PHI 2080
GO TO 9999

PHI 2090
9904 NK=3310

PHI 2100
GO TO 9999

PHI 2110
9905 NK=3316

PHI 2120
GO TO 9999

PHI 2130
9906 NK=3320

PHI 2140
GO TO 9999

PHI 2150
9907 NK=3328

PHI 2160
GO TO 9999

PHI 2170
9911 NK=3361

PHI 2180
9999 NR=3

PHI 2190
CALL DUMP

PHI 2200
C IF SN4(6OT=0.) ANY NEGATIVE ENERGIES WILL

PHI 2210
REMAIN, IF=O, CODE WILL TRY ANOTHER PASS

PHI 2220
WITH A SMALLER DT.

PHI 2230
10000 IF(SN17030,7031,7030

PHI 2240
7031 IF(UUT)7020,7030,7010

PHI 2250
C NEGATIVE ENERGIES HAVE OCCURED, INTEGRATE

PHI 2260
BACK TO CYCLE N WITH (-DT).

PHI 2270
7010 UT=-1.0

PHI 2280
DT=-DT

PHI 2290
C YOU NOW HAVE INTEGRATED BACK TO CYC: n. NUW

PHI 2300
C INTEGRATE TO CYCLE N+1 WITH NEW DT(STORED IN UU).

PHI 2310
GO TO 8000

PHI 2320
7020 UT=0.0

PHI 2330
DT=UU

PHI 2340
NR=DT/TR4AD+1.0

PHI 2350
WS=NR
```

PHI 2360
INCREASE ACTIVE GRID COUNTERS IF NEEDED.

703C I1=I1+NRC
I2=I2+NRT
IF(I1-1MAX)6100,6100,6200
6200 I1=1MAX
6100 IF(I2-1MAX)6201,6201,6202
6202 I2=JMAX
6201 RETURN
END
SUBROUTINE PH2

C Z(110)= CRITICAL ENERGY BETWEEN GAS AND CONDENSED STATE
C Z(111)= INITIAL DENSITY
C Z(112)= INITIAL VELOCITY OF PELLET
C Z(113)= EPSILONICS FOR EMPTYING PELLET ON BASIS OF VELOCITY
C TOZONE = MINIMUM DENSITY FOR MASS FLOW

C
C AMPY=MASS ACROSS TOP BOUNDARY OF CELL
C AMUI=RADIAL MOMENTA OF THIS MASS
C AMVIT=AXIAL MOMENTA OF THIS MASS
C DELET=TOTAL SPECIFIC ENERGY OF THIS MASS
C AMMP=MASS ACROSS RIGHT BOUNDARY OF CELL
C AMUR=RADIAL MOMENTA OF THIS MASS
C AMVR=AXIAL MOMENTA OF THIS MASS
C DELEB=TOTAL SPECIFIC ENERGY OF THIS MASS
C AMMY=MASS ACROSS BOTTOM BOUNDARY OF CELL
C AMMV=AXIAL MOMENTA OF THIS MASS
C AMMV=AXIAL MOMENTA OF THIS MASS
C DELET=TOTAL SPECIFIC ENERGY OF THIS MASS
C GAMC=MASS ACROSS LEFT BOUNDARY OF CELL
C FLEFT=RADIAL MOMENTA OF THIS MASS
C YAMC=AXIAL MOMENTA OF THIS MASS
C SIGC=TOTAL SPECIFIC ENERGY OF THIS MASS
C ==TOTAL SPECIFIC ENERGY OF THIS MASS==

NRT=0
NRC=0
REZ=0.0
CALL SLITE (0)

PIDT=1.0/(PIDY*DT)

101 DO 103 J=1,JMAX
102 GAMC(J)=0.0
FLEFT(J)=0.0
YAMC(J)=0.0
SIGC(J)=0.0
103 CONTINUE

104 DO 547 I=1,II

K=I+1
80 IF(AMX(K))9900,97,81
81 IF(V(K))82,97,97
97 AMHV=0.0
GO TO 98
82 AMMY=AMX(K)*V(K)*DY(J)
83 IF(AMMY+AMX(K))84,85,85
84 AMMY=-AMX(K)
85 IF(CVIS)106,99,99

C BOTTOM BOUNDARY IS TRANSMITTIVE, MATERIAL IS MOVING
C OUT, REMOVE ITS ENERGY FROM ETH.

106 AMMU=AMMY*U(K)
AMMV=AMMY*V(K)
DELEB=AX(K)+(U(K)**2+V(K)**2)/2.0
WS=(U(K)**2+V(K)**2)/2.0
ETH=ETH+AMMY*(AX(K)+WS)
GO TO 107

C BOTTOM BOUNDARY IS REFLECTIVE, NET MOMENTA CHANGE IN Z DIRECTION IS 2 HV.

99 AMMV=2.0*AMMY*V(K)
98 AMMY=0.0
AMMU=0.0
DELEB=0.0

C BEGIN DO LOOP IN J(Z) DIRECTION.

107 DO 546 J=1,12
108 L=K+IMAX
I=I
J=J
AREA=0.0
VEL=0.0
FS=0.0

210 IF(JMAX-J)211,211,207
211 VEL=1.0
GO TO 208

207 IF(AX(L))215,215,214
214 IF(AX(K))216,216,209
216 VABOVE=V(L)
GO TO 212

215 IF(AX(K))205,205,208
205 VABOVE=0.0
GO TO 212

208 VABOVE=V(K)
GO TO 212

209 VABOVE=(V(K)+V(L))/2.0

212 CONTINUE
I=I
J=J
FS=0.0

404 IF(IMAX-I)412,412,405
405 IF(AX(K+1))411,411,409
409 IF(AX(K))410,410,407
410 URR=U(K+1)
GO TO 408

411 IF(AX(K))403,403,406
403 URR=0.0
GO TO 408

C WE ARE AT THE RIGHT BOUNDARY OF THE GRID, THE BOUNDARY CONDITION HERE IS TRANSMITTIVE.

412 FS=1.0
406 URR=U(K+1)
GO TO 408
407 URR=(U(K)+U(K+1))/2.0
408 CONTINUE
109 IF(AREA)9901,301,547
301 IF(VABOVE)300,304,302
302 IF(AMX(K))9900,304,8800
8800 IF(J-1)9900,303,8801
8801 KP=K+IMAX
       IF(AMX(KP)19900,8803,303
C A CHECK HERE TO INSURE THAT THE BOTTOM ZONES
C OF THE PROJECTILE EMPTY (FOR HYPERVELOCITY) UP UNTIL
C THE INITIAL VELOCITY CHANGES DUE TO THE SHOCK.
8803 IF(ABS(VABOVE-Z(112))/Z(112)-Z(113))306,303,303
303 M=K
304 JJ=J
       GO TO 307
307 IF(VEL)9901,305,304
305 IF(AMX(L))9903,304,306
306 M=L
       JJ=J+1
307 IF(VEL)6130,6130,6140
6130 WSA=(V(K)+V(L))/2.0
       WSB=1.0+((J-1)*SBOUND)*DT
       VABOVE=WSA/WSB
       C CALCULATE THE MASS FLUX AT THE TO. OF CELL K.
6140 AMPY=AMX(M)*VABOVE/DY(JJ)*DT
501 IF(U(RR))500,504,502
502 IF(AMX(K))9900,504,503
503 M=K
504 AMMP=0.0
       AMUR=0.0
       AMVR=0.0
       DELET=0.0
       GO TO 508
505 IF(FS)9905,506,504
506 IF(AMX(K+1))9904,504,507
507 M=K+1
508 IF(FS)6100,6100,6110
6100 WSA=(U(K)+U(K+1))/2.0
       WSB=1.0+((K+1)-U(K))/DX(N)*SBOUND)*DT
       URR=WSA/WSB
       C CALCULATE THE MASS FLUX AT THE RIGHT OF CELL K.
6110  DEN=AMX(M)/TAU(N)
     AMP=DEN/PIDS*X(I)/.5*URR
     1  IF(AMP)16,74,8820
8820  IF(GANC(J))14,74,8821
  8821  IF(IFS)16120,6120,74
  6120  IF(AMX(K+1))9903,8822,74
  8822  IF(AMX(K)/TAU(I)*DY(J)+Z(111))8823,74,74
  8823  IF(AIX(K)-Z(110))8824,74,74
  8824  WS=GANC(J)+AMX(K)-TAU(I)*DY(J)+Z(111)
  8825  AMP=WS
     GO TO 74
  8826  AMP=0.0
     74 JTAG=0
     BEGIN CHECKING TO SEE IF THERE IS ANY
     PREFERENTIAL MASS FLUX BECAUSE OF CHOICE OF
     INDEXING DIRECTION.
     2  IF(AMPY)3,4,4
     TOP FLUX IS INTO CELL K.
     3  I TAG=1
        WS=X=AMX(K)
        AMP=AMPY-0.0
        GO TO 64
     4  JTAG=0
     BOTTOM FLUX IS INTO CELL K.
     5  IF(GANC(J))1,6,6
     LEFT FLUX IS INTO CELL K.
     6  WS=AMX(K)
        GO TO 11
     LEFT FLUX IS OUT.
     7  WS=AMX(K)+GANC(J)
        GO TO 11
     BOTTOM FLUX IS OUT OF CELL K.
     9  IF(GANC(J))10,8,48
     LEFT FLUX IS INTO CELL K.
     8  WS=AMX(K)+AMMY
        GO TO 11
     LEFT FLUX IS OUT OF CELL K.
     10 WS=AMX(K)+GANC(J)+AMMY
     11 WSA=AMPY+AMMP
     12 IF(WSA-WS)75,75,13
     CHANGE TOP AND RIGHT FLUX TO BE THE
     OLD FLUX TIMES THE MASS OF THE CELL/THE SUN
     OF THE OLD FLUXES.
     12 AMPY=AMPY*WS/WSA
        AMMP=AMMP*WS/WSA
  75 IF(JTAG)14,73,14
  73 WSA=AMMP
     14 IF(JTAG)15,7000,15
15 AMPY=WSB
   ITAG=0
   GO CHECK CELL ABOVE.
   GO TO 40
C RIGHT FLUX IS INTO CELL K.
16 IF(IFS)7b,17,76
   76 WSC=AMMP
C 1=IMAX, SO CHECK CELL ABOVE K.
   GO TO 40
17 IF(I+1=IMAX)19,13,9908
18 URRR=U(K+1)/2.0
   GO TO 20
19 URRR=U(K+1)+U(K+2))/2.0
20 IF(URRR)39,39,21
C FLUX IS OUT OF THE RIGHT OF CELL(K+1).
21 URRR=URRR/TAU*(I+1)*AMX(I+1)/2.0
   GO TO 26
22 IF(J=1)9909,23,24
23 VBLO=V(K+1)/2.0
   GO TO 26
24 KP=K+1-IMAX
   VBLO=(V(K+1)+V(KP))/2.0
   GO TO 31
25 IF(VBLO)25,30,38
C FLUX IS OUT OF THE BOTTOM OF CELL(K+1).
26 IF(VBLO)25,30,38
27 IF(VEL)28,29,28
28 VAB=V(K+1)/2.0
   GO TO 31
29 KP=K+IMAX+1
   VAB=(V(K+1)+V(KP))/2.0
   IF(VAB)36,36,36
30 VAB=AMX(K+1)/DY(J)*VABL*DT
C FLUX IS OUT OF TOP.
31 IF(VAB)36,36,36
32 WS=AMX(K+1)
33 WSA=URRR-AMMP-VBLO+VAB
34 IF(WSA=WS)77,77,77
35 AMMP=AMMP*WS/WSA
77 JTAG=1
   WSC=AMMP
   AMMP=0.0
   GO TO ?
C FLUX AT TOP IS INTO CELL (K+1).
36 =S=AMX(K+1)
37 WSA=URRR-AMMP-VBLO
   GO TO 34
C FLUX IS IN FROM BOTTOM INTO CELL (K+1).
38 VBLO=0.0
   GO TO 27
C FLUX IS INTO CELL (1..1) FROM RIGHT.
39 URRR=0.0
   GO TO 22
RIGHT FLUX OUT OF CELL (K) IS POSITIVE AND TOP FLUX IS COMING INTO CELL (K) FROM (K+IMAX).

40 IF(VEL)7000 0 41,7000
41 IF(FS)42,43,42

C WE ARE AT THE RIGHT BOUNDARY OF THE GRID.
42 KP=K*IMAX
URT=U(KP)/2.0
GO TO 45
43 KP=K*IMAX
URT=(U(KP)+U(KP+1))/2.0
45 IF(URT)46,46,70

C FLUX AT RIGHT (CELL M) IS NEGATIVE.
46 URT=0.0
GO TO 47
70 KP=K+IMAX
URT=URT/TAU(I)*AMX(KP)/PIDTS*X(I)/.5

C FLUX AT RIGHT (CELL M) IS POSITIVE.
47 IF(J+1-JMAX)48,49,9910
48 KP=K+IMAX
KL=KP+IMAX
VABT=(V(KP)+V(KL))/2.0
GO TO 51
49 KP=K+IMAX
KL=KP+IMAX
VABT=V(KP)/2.0
51 IF(VABT)8810,71,72
8810 IF(AMX(KP))9903,8811,71

C CHECK FOR SOLID OR VAPOR.
8811 IF(AMX(KP))/TAU(I)*DY(J+1)-Z(I11))8812,71,71
8812 IF(AMX(KP)-Z(I10))8813,71,71
8813 VABT=VABT*AMX(KP)/DY(J+1)*DT
8814 WS=-VABT*AMX(KP)-TAU(I)*DY(J+1)*Z(I111)
8815 IF(WS)8816,8817,8817
8816 AMPY=-WS
GO TO 71
8817 AMPY=0.0
71 VABT=0.0
GO TO 60
72 VABT=VABT*AMX(KP)/DY(J+1)*DT
52 IF(GAMC(J+1))54,53,53
53 WS=AMX(KP)
GO TO 55
54 WS=AMX(KP)+GAMC(J+1)
55 WSA=VABT-AMPY+URT
GO TO 57
60 IF(GAMC(J+1))56,57,59
61 WS=AMX(KP)+GAMC(J+1)
GO TO 58
59 WS=AMX(KP)
58 WSA=-AMPY+UR
57 IF(WSA-WS)7000,7000,56
56 AMPY=AMPY*WS/WSA
GO TO 7000
7000 AMP=WSC
309 IF(AMPY)8834,8831,8833
8833 IF(JMAX-J)9911,318,8835
8835 KP=K+IMAX
8836 IF(AMX(KP))9900,8837,318

C **** NOTE ********************************************
C ACROSS FREE SURFACE, HOLD UP MASS FLUX
C UNLESS THIS MASS PRODUCES A DENSITY GREATER THAN TZONE.
C *********************************************************
8837 IF(AMPY/(TAU(I)*DY(J))-TZONE)8838,318,318
8838 AMPY=0.0
GO TO 8831
8834 IF(J-1)991lv325,8839
8839 IF(AMX(K))9900,8840,325
8840 IF(-AMPY/(TAU(I)*DY(J))-TZONE)8841,325,325
8841 AMPY=0.0
GO TO 8831
318 DELM=GAMC(J)+AMMY-AMPY
322 IF(VEL)9901,324,323
323 WS=U(K)**2+V(K)**2

C MATERIAL HAS LEFT THE TOP, TRIGGER RZONE
C FLAG, REMOVE ITS ENERGY FROM ETHTOTAL ENERGY OF SYSTEM.
ETH=ETH-AMPY*(AI(X(K)+WS/2.0)
IF(AMPY/(TAU(I)*DY(J))-TZONE)324,324,6900
6900 REZ=1.0
324 AMUT=AMPY*U(K)
325 AMVT=AMPY*V(K)
GO TO 326
326 CONTINUE

8831 AMUT=AMPY*U(L)
326 IF(AMPY)327,328,328
327 DELT=AI(X(L)+U(L)**2+V(L)**2)/2.0
GO TO 333
328 IF(AMMY)329,330,330
329 DELT=DELEB
GO TO 333
330 IF(GAMC(J))331,332,332
331 DELEB=SIGC(J)
GO TO 333
332 DELEB=AI(X(K)+U(K)**2+V(K)**2)/2.0
C SUM UP RADIAL MOMENTA FOR ALL FLUXES EXCEPT THE
C THE RIGHT AND STORE IN SIGMU.
333 SIGMU=LEFT(J)+AMMU-AMUT
C SUM UP AXIAL MOMENTA FOR ALL FLUXES EXCEPT THE
RIGHT AND STORE IN SIGMV.
SIGMV=YAMC(J1)+AMVM-AMVT
SUM UP TOTAL ENERGY CARRIED BY THESE FLUXES
EXCEPT THE RIGHT FLUX AND STORE IN DELEK.
DELEK=GAMC(J1)*SIGC(J)+AMNY*DELE5-AMPY*DELET

IF(AMMP)8843,518,8844
8844 IF(MAX-I)9911,518,8845
8845 IF(AMX(K+1))9900,8846,518
8846 IF(AMMP/(TAU(I)*DY(J))-TOZONE)8847,518,518
8847 AMMP=0.0
GO TO 518
8843 IF(I-I.)9911,518,8849
8849 IF(-AMMP/(TAU(I)*DY(J))-TOZONE)8850,512,512
8850 AMMP=0.0
GO TO 518
512 DELM=DELM-AMMP+AMX(K)
513 CONTINUE
514 CONTINUE
8828 AUH=AMMP*U(K+1)
ANVR=AMMP*V(K+1)
GO TO 525
518 DELM=DELM-AMMP+AMX(K)
521 CONTINUE
522 IF(FS)9905,524,523
523 WS=U(K)**2+V(K)**2
ETH=ETH-AMMP(AUX(K)+WS/2.0)
IF(AMMP/(TAU(I)*DY(J))-TOZONE)524,524,6901
6901 REZ=1.0
524 AUH=AMMP*U(K)
ANVR=AMMP*V(K)
525 SIGMU=SIGMU-AMUR
SIGMV=SIGMV-AMVR
526 TIC=0.0
527 IF(AMMP)528,529,529
528 DELEB=AUX(K+1)+(U(K+1)**2+V(K+1)**2)/2.0
GO TO 537
529 IF(AMMY)530,531,531
530 DELEB=DELEB
GO TO 536
531 IF(GAMC(J1))532,533,533
532 DELEB=SIGC(J)
GO TO 536
533 IF(AMMP)534,535,534
534 DELEB=DELE5
GO TO 536
535 DELEB=AUX(K+1)+(U(K+1)**2+V(K+1)**2)/2.0
536 TIC=1.0
537 DELK=DELEK-AMMP*DELER
538 IF(TIC)9907,539,550
125.

550 WS=DELER
GO TO 999
539 WS=AIX(K)+*(U(K)**2+V(K)**2)/2.0
998 IF(IDELM)998,543,544
997 DELM=0.0
GO TO 543
C ENK=TOTAL ENERGY OF CELL (K) + ENERGY THAT
C HAS BEEN ADDED AND LOST.
540 ENK=AMX(K)*WS+DELEK
C BY CONSERVING AXIAL MOMENTA, CALCULATE THE NEW
C AXIAL VELOCITY COMPONENT FOR CELL K.
541 U(K)=(SIGMU+AMX(K)*U(K))/DELM
C BY CONSERVING RADIAL MOMENTA, CALCULATE THE NEW
C RADIAL VELOCITY COMPONENT FOR CELL K.
601 V(K)=(SIGMV+AMX(K)*V(K))/DELM
604 IF(U(K))605,604,604
605 NRC=1
606 IF(V(K))607,606,606
607 NRC=1
608 IF(AIX(K))609,610,609
609 NRC=1
610 CONTINUE
603 WS=U(K)**2+V(K)**2
C BY CONSERVING BOTH TOTAL ENERGY AND
C MOMENTA, CALCULATE THE NEW SPECIFIC
C INTERNAL ENERGY FOR CELL K.
542 AIX(K)=ENK/DELM-WS/2.0
543 AMX(K)=DELM
2007 AIX(K)=0.0
U(K)=0.0
V(K)=0.0
P(K)=0.0
C THE RIGHT VALUES OF CELL (K) BECOME THE LEFT
C VALUES OF CELL (K+1).
544 GAMC(J)=AMMP
FLEFT(J)=AMUR
YAMC(J)=ANVR
SIGC(J)=DELER
C THE TOP VALUES OF CELL(K) BECOME THE
C BOTTOM VALUES FOR CELL (K+IMAX).
545 AMMY=AMPY
AMMU=AMUT
AMMV=AMVT
DELEB=DELET
546 K=K+IMAX
LL=K-IMAX
IF(J(LL))6500,6600,6500
IC

6500 NRT=1
6600 IF(VILL)6601,6602,6601
6601 NRT=1
6602 IF(AIX(L1)6611,547,5611
6611 NRT=1
547 CONTINUE
   I1=I1+NRG
   I2=I2+NRG
   IF(IMAX-I1)/6700,6701,6702
6700 I1=IMAX
6701 CONTINUE
6702 IF(JMAX-I2)/6800,6801,6802
6800 I2=JMAX
6801 CONTINUE
6802 GO TO 544
9901 NK=360
   GO TO 9999
9900 NK=302
   GO TO 9999
9903 NK=305
   GO TO 9999
9904 NK=500
   GO TO 9999
9905 NK=500
   GO TO 9999
9906 NK=513
   GO TO 9999
9911 NK=8833
   GO TO 9999
9908 NK=17
   GO TO 9999
9909 NK=22
   GO TO 9999
9910 NK=27
   GO TO 9999
9907 NK=530
9999 NR=4
   CALL DUMP
548 SUM=0.0
2005 DO 2001 I=1,11
   K=I+1
   DO 2000 J=1,12
      IF(AUX(K))/(TAU(I)*DY(J))-TOZONE2010,2008,2009
      C IF ANY RHO (CELL DENSITY) IS LESS THAN TOZONE,
      C SET THE MASS TO ZERO, AND TALLY THE
      C MOMENTAS AND ENERGIES IN THE Z STORAGE, ALSO
      C CHECK FOR NEGATIVE INTERNAL ENERGIES, IF
      C WE FIND SOME, SET THEM TO ZERO AFTER
      C SUBTRACTING THEM FROM TH.
      2009 IF(AUX(K))/(TAU(I)*DY(J))-TOZONE2010,2008,2009
   PH2 4590
   PH2 4600
   PH2 4610
   PH2 4620
   PH2 4630
   PH2 4640
   PH2 4650
   PH2 4660
   PH2 4670
   PH2 4680
   PH2 4690
   PH2 4700
   PH2 4710
   PH2 4720
   PH2 4730
   PH2 4740
   PH2 4750
   PH2 4760
   PH2 4770
   PH2 4780
   PH2 4790
   PH2 4800
   PH2 4810
   PH2 4820
   PH2 4830
   PH2 4840
   PH2 4850
   PH2 4860
   PH2 4870
   PH2 4880
   PH2 4890
   PH2 4900
   PH2 4910
   PH2 4920
   PH2 4930
   PH2 4940
   PH2 4950
   PH2 4960
   PH2 4970
   PH2 4980
   PH2 4990
   PH2 5000
   PH2 5010
   PH2 5020
2010 WS=(U(K)**2+V(K)**2)/2.0
Z(100)=Z(100)+AMX(K)
WS=AMX(K)*(AIX(K)+WS)
Z(101)=Z(101)+WS
ETH=ETH-WS
Z(102)=Z(102)+AMX(K)*U(K)
Z(103)=Z(103)+AMX(K)*V(K)
AMX(K)=0.0
AIX(K)=0.0
P(K)=0.0
U(K)=0.0
V(K)=0.0
GO TO 2000
2008 IF(AIX(K))2004,2000,2000
2004 SUM=SUM+AIX(K)*AMX(K)
AIX(K)=0.0
2000 K=K+IMAX
2001 CONTINUE
2003 ETH=ETH-SUM
Z(104)=Z(104)+SUM
8000 IF(REZ)8001,8001,8002
8002 IF(REZFCT)8004,8004,8003
8004 REZ=0.
GO TO 8001
8003 CALL REZONE
8001 RETURN
END
SUBROUTINE ES

C METALLIC EQUATION OF STATE, SEE
C GA-3216 REPORT.

10 RHOW=AMX(K)/(TAU(I)*DY(J))
ETA=RHOW/Z(115)
VOW=1.0/ETA

11 P1=AMX(K)*RHOW*Z(116)
12 P2=(Z(115)*TAU(I)*DY(J)**2)*AMX(K)
13 P3=AMX(K)**2*Z(117)
14 P4=Z(118)/(P2/P3+1.0)*AMX(K)*RHOW
15 P5=Z(119)*((ETA-1.0)
16 IF(ETA-1.0)*55,100,100
50 IF(VOW-Z(120))55,100,100
55 IF(AMX(K)-Z(122))100,100,75
75 P7=Z(123)*((VOW-1.0)
IF(P7-88.0)4002,4002,4003

4003 P7=88.0
4002 CONTINUE
P8=EXP(P7)
P9=1.0/P8
P10=Z(124)*((VOW-1.0)**2)
IF(P10-88.0)4000,4000,4001

4001 P10=88.0
4000 CONTINUE
P11=EXP(P10)
P12=1.0/P11
P(K)=P1+(P4+P5*P9)*P12
GO TO 119

100 P6=Z(126)*((ETA-1.0)**2)
P(K)=P1+P4+P5*P6
119 IF(P(K))999,999,200
200 WSGX=.5
GO TO 500
999 P(K)=0.0
WSGX=.5+Z(125)
GO TO 500
500 RETURN
END

SUBROUTINE ES

C ** POLYTROPIC EQUATION OF STATE **
P(K)=AMX(K)*AMX(K)/GAMX/(DY(J)*TAU(I))
WSGX=GMAX-1.0
RETURN
END
CONSERVE MOMENTUM AND TOTAL ENERGY, INCREASE
ALL LINEAR DIMENSIONS BY 2. (THUS 4 CELLS
IN THE UL: GRID ARE COMBINED INTO 1 FOR
THE NEW GRID.)

NIMAX = 1MAX/2
NJMAX = JMAX/2
DO 10 J=1,NJMAX
K = (J-1)*NIMAX+1
L = (J-1)*2*NJMAX+2
DO 11 I=1,NIMAX
M = L+IMAX
12 WS = AMXL(J)+AMX(M)+AMX(I+1)+AMX(I)+1)
  U = AMXL(L)+U(L)*2+V(L)*2+AMX(M)+U(M)
  V = AMX(L)+U(L)*2+V(L+1)*2
  U(L) = (U(L)*AMXL(L)+U(M)+AMX(M)+U(L+1)+AMX(L+1)+
  AMX(L1+1)/WSA
  V(L) = AMXL(L)+AMX(M)+V(L+1)*2
  I = AMX(L1+1)/WSA
  AIX(K) = AIX(L)*AMX(L)+AIX(M)+AIX(L1+1)+
  AMX(L1+1)+AMX(M1+1)+AIX(M)+1)
  AIX(K) = WSA
  WS = U(K)*2+V(K)*2
  E = AIX(K)+WSB/2,0
  AIX(K) = E/AMX(K)-5*WS
  IF(K>=214,14,13
  AMX(L) = 0,0
  U(L) = 0,0
  V(L) = 0,0
  AIX(L) = 0,0
  P(L) = 0,0
  AMX(M) = 0,0
  U(M) = 0,0
  V(M) = 0,0
  AIX(M) = 0,0
  P(M) = 0,0
  AMX(L+1) = 0,0
  U(L+1) = 0,0
  V(L+1) = 0,0
  AIX(L+1) = 0,0
  P(L+1) = 0,0
  AMX(M+1) = 0,0
  U(M+1) = 0,0
  V(M+1) = 0,0
  AIX(M+1) = 0,0
  P(M+1) = 0,0
14 K=K+1
L=L+2
11 CONTINUE
10 CONTINUE

C CALCULATE NEW DY AND Y (JMAX IF THEM)
18 DO 999 J=1,JMAX
   DY(J)=DY(J)*2.0
999 CONTINUE

DO 99 J=1,JMAX
   Y(J)=Y(J-1)+DY(J)
99 CONTINUE

16 DX(I)=2.0*UX(I)
   XPX(I)=DX(I)
   WSA=X(I)*2
   TAU(I)=PIDY*WSA
   WS=WSA
98 CONTINUE

IMAX=NMAX
JMAX=NMAX

C PREPARE NOW TO SHUFFLE THE K ARRAYS SUCH
C AS TO PRESERVE K=(J-1)*IMAX+I+1.

DO 20 I=1,JMAX
   J=JMAX+1-I
   K=(J-1)*IMAX+1+IMAX
   L=(J-1)*IMAX+1+IMAX+1
   DO 21 I=1,IMAX
21 CONTINUE
20 CONTINUE

IMAX=NMAX*2
JMAX=NMAX*2
II=NMAX+1
ADD ON NEW MASS WITH DENSITY=Z(111) IN TARGET 
C
59 50  I=1,NJMAX
K=(JJ-1)*IMAX+I+1
DO 50  J=JJ,JMAX
AMX(K)=Z(111)*TAU(I)*DY(J)
60  K=K+IMAX
50 CONTINUE
JJ=(I(147)/2.+2)
JJ=JJ+1
DO 61  I=1,IMAX
K=I+(JJ-1)*IMAX
DO 62  J=JJ,JMAX
AMX(K)=Z(111)*TAU(I)*DY(J)
62  K=K+IMAX
61 CONTINUE
C
RESET ACTIVE GRID MARKERS.
JJ=JJ-1
Z(I47)=JJ
11=IMAX+2
12=NJMAX-2
WS=1+DTNA
NK=NC+1
C
EDIT THE NEW GRID.
WRITE (6,8004)WS,NK,DX(I)
WRITE (6,8007)IMAX,(X(I),I=0,IMAX)
WRITE (6,8009)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8009)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8010)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8011)IMAX,(TAU(I),I=1,IMAX)
WRITE (6,8010)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8011)IMAX,(TAU(I),I=1,IMAX)
KMAX=IMAX+1
IMAX=IMAX+1
JMAX=JMAX+1
KMAX=KMAX+1
RETURN
C
WRITE (6,8004)WS,NK,DX(I)
WRITE (6,8007)IMAX,(X(I),I=0,IMAX)
WRITE (6,8009)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8010)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8011)IMAX,(TAU(I),I=1,IMAX)
WRITE (6,8010)IMAX,(DY(J),J=0,JMAX)
WRITE (6,8011)IMAX,(TAU(I),I=1,IMAX)
KMAX=IMAX+1
IMAX=IMAX+1
JMAX=JMAX+1
KMAX=KMAX+1
RETURN
C
3
```
SIBFIC EDIT LIST,DECK,REF
SUBROUTINE EDIT

SENSE LITE (1) INDICATES LAST CYCLE OF THIS
RUN.
SENSE LITE (3) INDICATES FIRST CYCLE OF THIS
RUN.

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)199U1,112,124
112 IF(AMOD(CYCLE,DUMPT7))114,124,114
114 IF(AMOD(CYCLE,PRINTS))11C,126,120
120 IF(AMOD(CYCLE,PRINTS))140,128,140
NORMAL STOP ON THIS CYCLE.

122 CALL SLITE (1)
      DUMP ON TAPE 7.
124 GO TO 1
126 CALL SLITE (4)
128 GO TO 6000
130 GO TO 1000
132 CALL SLITET(4,K000FX)
      GO TO(134,136),K000FX
134 GO TO 5000

CHECK FOR ENERGY CHECK ERROR. WHERE
ECK= PERCENT ERROR/PER CYCLE.
ECK=(ETH-E1/ETH AT CYCLE N-ETH-E1/ETH
AT CYCLE N-NPC ALL DIVIDED BY NPC, NOTE
NPC= NO. OF CYCLES BETWEEN ENERGY CHECK

136 IF(ABS(ECKI-DMIN))140,140,905
140 CALL SLITET(1,K000FX)
      GO TO(142,144),K000FX
142 REMIND 7
144 GO TO 10000

DUMP ON TAPE 7
1 IF(DUMPT7)30,3,3
3 BACKSPACE 7
REMIN 2
REMIN 3
WS=555,0
WRITE (7)NS,CYCLE,N3
WRITE (7)IL(1,L=1,MI)
6 WRITE (7)U(1,K),V(K),AMX(K),AIX(K),P(K),K=1,IMAX)
7 WRITE (7)X(0),X(K),TAU(K),K=1,IMAX)
```
WRITE (7, I(N), K = 3, JMAX)

C AGAIN, IF PROBLEM NO. IS NEGATIVE, WRITE
C PARTICLE RECORDS ON TAPE 7.
IF (PCB = 16, 16, 4)
16 DO 13 I = 1, H3
13 IF (N(I) = 150, 148, 150)
140 CONTINUE
READ (2, AM(N), XL(N), YL(N), IN(N), I2(N), N = 2, M4)
GO TO 152
150 CONTINUE
READ (3, AM(N), XL(N), YL(N), IN(N), I2(N), N = 2, M4)
152 CONTINUE
WRITE (7, I(N), K = 3, JMAX)
GO TO 126
C
C 6000 NK = 12
C TABS ARE TANGENT ALPHAS,
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) = 0.9
NK1 = NK + 2
DO 6014 I = 1, NK1
AMK(I) = 0.0
PK(I) = 0.0
6014 QPK(I) = 0.0
DO 6028 K = 2, KMAX
6017 PR(I) = 0.0
PK(2) = 0.0
PK(4) = 0.0
C CALCULATE KINETIC ENERGY IN CELL K.
WKB = (UK)**2 + VKK**2 + 1/2
6019 IF (AMK(I) < 19, 17, 6028, 6020)
C SEARCH FOR TAN ANGLE THAT VELOCITY VECTORS
MAKE
IF(TAN(I)-W$46.324,6026,6026)
6024 CONTINUE
I=NX+1
6026 WS=AMX(K)
C SUM UP MASS BETWEEN ANGLES.
C AMX(1)=AMX(1)+AMX(K)
C SUM UP RADIAL MOMENTA IN THE ANGLES.
PR(1)=PR(1)+PR(K)
PR(7)=PR(5)+PR(6)
XNRG=PR(7)
PR(9)=PR(1)+PR(5)
PR(10)=PR(2)+PR(6)
PR(11)=PR(3)+PR(7)
PR(12)=PR(4)+PR(8)
WSA=(ETH-PR(11))/ETH
IF(CYCLE)END
9931 NPC=1
9932 PR(18)=(WSA-DN)/FLOAT(NPC)
ECK=PR(18)
DN=AMX
C RESET CYCLE COUNTER BETWEEN ENERGY CHECK.
NPC=0
SUM=0.0
DO 800 I=1,13
SUM=SUM+QK(I)
800 CONTINUE
C RADET= TOTAL POSITIVE AXIAL MOMENTUM IN GRID
RADET=SUM
801 SUM=0.0
DO 801 K=2,KMAX
IF(AMX(K))810,801,802
802 IF(U(K))810,810,803
803 SUM=SUM+AMX(K)*U(K)
810 CONTINUE
C READ IN INITIAL CONDITIONS AND NUMERICAL INTEGRATION
C READ IN = SUM
SUM = C
JD = 0
GO TO 1
K = 0
GO TO 2
1 IF (K.EQ.15) GO TO 12
IF (J.EQ.1) GO TO 13, 12
2 SUM = SUM + U(R, J, XMAX)
13 K = K + 1
11 CONTINUE
C RADE = TOTAL RELATIVE WALL MOUNTMENT DELTA
C INITIAL TARGET-PROJECTILE INTERFACE
RADE = SUM
PR(150) = C, C
GO TO 6029
6029 JD = JD + 1, J = J + 1
PR(150) = PR(150) + SUM, XMAX = XMAX + 0.1
PR(150 + 2) = C, C
PR(150 + 2) = C, C
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 1111) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
6C98 GO TO 130
C**** END OF SUBROUTINE ******************************************************
C
C**** SUBROUTINE PLOT ******************************************************
1000 GO TO 1530
1030 WRITE (6, 8616) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
8616 JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
WRITE (6, 8607) JD, JD, JD, JD, JD, JD, JD, JD, JD, JD
1034 JD = 1351, JD = JD + 1
1036 JD = 1036, JD = JD + 1
WRITE (6, 8608)
1040 CONTINUE
1044 JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1
1105 JD = 1130, JD = JD + 1
1109 JD = JD + 1
1126 PR(1) = A254, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1
1150 IF (XMAX .LT. XMAX + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1, JD = JD + 1
C X PARTICLE ONLY
C REPLACE 6C0000D00000 BY 17179859184
C REPLACE 6C0000D00000 BY 922745680
136.

1160 PR(I)=OR(PR(I), ABS( 92746880) )
GO TO 1180
C REPLACE 6000000000 BY 805306368
1166 PR(I)=OR(PR(I), ABS( 805306368) )
1180 CONTINUE
1200 IF(MOD(J,5))1210,1204Y1210
1204 IF(DY(J)-DY(J-1))1206,1208,1212
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
1210 IF(DY(J)-DY(J-1))12,1214,1212
1212 WRITE (6,8222)DY(J),(PR(I),I=1,I1)
GO TO 1224
1214 WRITE (6,8202)(PR(I),I=1,1I)
1214 J=J-1
1216 IF(J)1230,1230,1100
C REPLACE 604000000000 BY -17716740096
1230 PR(I)=(-ABS(-17716740096))
WRITE (6,8201)J,(PR(1),I=1,1I)
WRITE (6,8302)(I,I=1,IMAX,5)
1240 GO TO 132
C**** END OF PLOT SUBROUTINE ****************************************** EDIT2870
C
C
C**** SUBROUTINE L P ************************************************** EDIT2900
5000 WRITE (6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
5004 DO 5050 I=1,11
CALL SLITE (4)
J=12+1
K=I2*IMAX+I+1
5046 J=J-1
K=K-IMAX
GO TO(5016#5019)LKGOOFX
5016 WRITE (6,8135)I,X(I),DX(I)
C WS= DENSITY OF CELL(K) IN GRAMS/CM. CUBED.
5019 WS=AMX(K)/(TAU(I)*DY(J))
C WSA= COMPRESSION = RHO/RHO NOT. WSA=WS/WS(111)
C WSC= PRESSURE CONVERTED TO MEGABARS. WSC=P(K)*1.E+4
C FIRST COLUMN= (J) THE ROW NO.
C SECOND COLUMN= RADIAL VELOCITY CM./SHAKE
C THIRD COLUMN= AXIAL VELOCITY CM./SHAKE
C FOURTH COLUMN= F/A = PRESSURE IN MEGABARS
C FIFTH COLUMN = AMX = MASS IN GRAMS.
C SIXTH COLUMN = RHO = DENSITY IN GRAMS/CC.
C SEVENTH COLUMN = AIX = SPECIFIC INTERNAL ENERGY JERKS/GM.
C EIGHT COLUMN = COMPRESSION = RHO/RHO NOT
C NINTH COLUMN = Z VALUE (CM.) OF TOP OF CELL

50180 WRITE (6,8108) JU(K),V(K),WSCtAMX(K), WSAV(K), EDIT3060
     IWSA, Y(J) EDIT3070
5046 CONTINUE EDIT3080
5050 CONTINUE EDIT3090
     GO TO 136 EDIT3100

C**** END OF L P SUBROUTINE **************************************************** EDIT3110
C
C
C ERROR
9901 NK=110 EDIT3120
     GO TO 9999 EDIT3130
9905 NK=136 EDIT3140
     GO TO 9999 EDIT3150
C ENERGY CHECK
9910 NK=160 EDIT3160
     GO TO 9999 EDIT3170
C NEGATIVE MASS
9917 NK=6015 EDIT3180
     GO TO 9999 EDIT3190
9920 NK=904 EDIT3200
     GO TO 9999 EDIT3210
9921 NK=912 EDIT3220
     GO TO 9999 EDIT3230
9922 NK=918 EDIT3240
     GO TO 9999 EDIT3250
9923 NK=922 EDIT3260
     GO TO 9999 EDIT3270
9924 NK=926 EDIT3280
     GO TO 9999 EDIT3290
9999 NR=6 EDIT3300
     CALL DUMP EDIT3310
10000 RETURN EDIT3320
C
C FORMATS
8108 FORMAT(13,1X,1P2E14.6,3E15.6,E14.6,E15.6,E14.6) EDIT3330
81160 FORMAT(8HIPROBLEM6X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HTRAD11X,EDIT3340
     2X) 2GHNR6X,2HNI14X,2HN24X,2HN34X,2HN44/(F7.1,111,2X,1P4E16.7,110,2X,4EDIT3350
     4E15.6) EDIT3360
81170 FORMAT(1H0//17X2HAI16X,2HAK14X,5HAI+AK15X,2HAM/4H DOT3X,1P4E18.7/3EDIT3370
     12X,2HNR6X,2HNI14X,2HN24X,2HN34X,2HN44/(F7.1,111,2X,1P4E16.7,110,2X,4EDIT3380
     4E15.6) EDIT3390
81180 FORMAT(12X,13H---------------/7X TOTALS1P4E18.7) EDIT3400
1X,13H---------------/7X TOTALS1P4E18.7) EDIT3410
81190 FORMAT(2HO //16X,5HRADE13X,5HRAD13X,5HRAD12X,7MAX VEL13X,3HTEDIT3420
     1X12X,9HREL ERROR/7X,1P6E18.7////) EDIT3430
8120 FORMAT(1HO//21H TAPE 7 DUMP ON CYCLE15///) EDIT3440
81240 FORMAT(3H K12X,5HAM(K)11X,9HSUM AM(K)11X,4HP(K)13X,4HQ(K)/(13,4X,EDIT3450
     1P4E18.7)) EDIT3460
8131 FORMAT(1H //1IH DY(J) J=1,12\(10F12.3)) EDIT3470
8133 FORMAT(1H //1IH Y(J) J=0,12\(10F12.3)) EDIT3480
81350 FORMAT(1H //4H I =13,6X,6HX(I) =F12.3,6X,7MDX(I) =F12.3\://3H J8X,EDIT3490
     8X,EDIT3500
     8X,EDIT3510
     8X,EDIT3520
136.

11HX:13X,1HY13X,3HF/A12X,3HAMX12X,3HRHO11X,3HAI12X,4HCOMP11X,2H Y/} EDIT3530
8201 FORMAT(I10,2H I54A2) EDIT3540
8202 FORMAT(10X,2H I54A2) EDIT3550
8211 FORMAT(F7.1,13,2H I54A2) EDIT3560
8222 FORMAT(F7.1,3X,2H I54A2) EDIT3570
8302 FORMAT(112,10I10) EDIT3580
8307 FORMAT(5H XI =1PE12.6,3X,4HX2 =E12.6,3X,6HMAX =E12.6,6X,4HY1 =E12EDIT3590
1.6,3X,4HY2 =E12.6,3X,6HYMAX =E12.6) EDIT3600
8308 FORMAT(1H /) EDIT3610
9040 FORMAT(1H / 616) EDIT3620
END EDIT3630
REFERENCES


