DESCRIPTION OF FORTRAN PROGRAM DAWNA FOR ANALYSIS OF MUZZLE BLAST FIELD

Prepared by
Advanced Technology Laboratories, Inc.
Merrick & Stewart Avenues
Westbury, New York 11590

April 1976

Approved for public release; distribution unlimited.

USA BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND
Destroy this report when it is no longer needed. Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.
Description of FORTRAN Program DAWNA for Analysis of Muzzle Blast Field

J. Ranlet and J. Erdos

Advanced Technology Laboratories, Inc.
Merrick & Stewart Aves
Westbury, New York 11590

USA Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland 21005

US Army Materiel Development & Readiness Command
5001 Eisenhower Avenue
Alexandria, VA 22333

Approved for public release; distribution unlimited.

The FORTRAN IV Program DAWNA described in this report solves the set of partial differential equations governing the flow on the axis of symmetry between the blast wave and the Mach disc of a muzzle blast field. A complete description of the method of solution of the governing equations and statement of the boundary conditions can be found in BRL Contractor's Report No. 297.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. PROGRAM STRUCTURE</td>
<td>2</td>
</tr>
<tr>
<td>III. DESCRIPTION OF INPUT</td>
<td>8</td>
</tr>
<tr>
<td>A. Punched Card Format</td>
<td>8</td>
</tr>
<tr>
<td>B. Restart Capability</td>
<td>13</td>
</tr>
<tr>
<td>IV. DESCRIPTION OF OUTPUT</td>
<td>15</td>
</tr>
<tr>
<td>A. Output Format</td>
<td>15</td>
</tr>
<tr>
<td>B. Identification of Variables</td>
<td>16</td>
</tr>
<tr>
<td>C. Sample Input and Output</td>
<td>16</td>
</tr>
<tr>
<td>V. PROGRAM LISTING</td>
<td>22</td>
</tr>
<tr>
<td>DISTRIBUTION LIST</td>
<td>71</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>FIG.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCHEMATIC OF GRID NETWORK</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>CDC SCOPE 2.1 OPERATING SYSTEM CONTROL CARDS ILLUSTRATING THE USE OF THE PROGRAM RESTART CAPABILITY</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>TYPICAL INPUT IN CARD IMAGE FORMAT</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>FIRST PAGE OF PROGRAM OUTPUT</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>SECOND PAGE OF PROGRAM OUTPUT</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>TYPICAL OUTPUT OF FINITE DIFFERENCE CALCULATION</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>TYPICAL OUTPUT OF ACOUSTIC WAVE ANALYSIS</td>
<td>21</td>
</tr>
</tbody>
</table>
The FORTRAN IV program DAWNA solves the set of partial differential equations governing the flow on the axis of symmetry between the blast wave and the Mach disc of a muzzle blast field. A complete description of the method of solution of the governing equations and statement of the boundary conditions is given in Reference (1). Program DAWNA is a revised edition of program DAWN previously described in Reference (2). The two main refinements included in program DAWNA are the following:

1) An initialization technique has been developed that allows the computer program to self-start without the use of empirical relationships (such as the initial locations of the flow discontinuities).

2) An acoustic analysis which enables the continuation of the blast field calculation to very late times.

Besides the major analytical refinements presented above, various other modifications have been included in the present computer code which minimize possible sources of numerical error as well as making the program more convenient for the user. The more pertinent of these include the use of automatic grid subdivision, a streamline trace for the determination of the properties upstream of the Mach disc and the option of choosing the units of both the input and output parameters.
The main program acts as an executive routine which calls the principal subroutines, as follows. Subroutine INDATA is called once to define the finite difference grid and initialize the dependent variables. Subroutines SHOCK, CONTACT and INT PT are called sequentially to advance the solution from time t to time t+Δt. The dependent variables are then reinitialized, printed and the above sequence repeated. This loop is terminated when the selected number of time steps have been completed, at which time the final solution can be written out in TAPE12 (at the option of the user) and the program execution is stopped.

Subroutines SHOCK and CONTACT accomplish the solution at the two shocks and the contact surface by the method of characteristic technique described in Reference (1). The three surfaces of discontinuity are identified by the indices 1, 2 and 3 which refer to the Mach disc, contact surface and blast wave, respectively. Subroutine INT PT accomplishes the solution at the interior grid points by the finite difference algorithm devised by MacCormack (Reference 3). Within subroutine INT PT the index K=1 denotes the solution between the Mach disc and contact and K=2 denotes the solution between the contact and the blast wave. The indices LOOP=0 and 1 refer to the first and second iterates of the MacCormack algorithm.

The subroutines associated with the acoustic wave computation are called from subroutine ACBUS which only returns control to the main program at the completion of the run.

The functions of all the subroutines are summarized in Table I. The main program variables are identified in Table II.
<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CL</td>
<td>Calculates flow properties on the centerline of the supersonic jet exhaust plume from the muzzle exit conditions and the centerline Mach number at the location of the Mach disc.</td>
</tr>
<tr>
<td>2. CONTACT</td>
<td>Calculates the position and velocity of the contact surface and the flow properties on both sides of the contact surface.</td>
</tr>
<tr>
<td>3. FS</td>
<td>Sets the flow variables on the upstream side of the blast wave to the specified ambient (i.e., &quot;infinity&quot;) conditions.</td>
</tr>
<tr>
<td>4. INDATA</td>
<td>Reads the input data and defines accordingly the finite difference grid and the initial values of the dependent variables.</td>
</tr>
<tr>
<td>5. INT PT</td>
<td>Computes the new finite difference solution at the interior grid points and the new location of the grid points at each succeeding time step.</td>
</tr>
<tr>
<td>6. LPOINT</td>
<td>Locates the intersection of a characteristic surface and a time plane, and performs the necessary interpolations of variables from the solution at the grid points.</td>
</tr>
<tr>
<td>7. MUZZLE</td>
<td>Determines the pressure, speed of sound and Mach number at the muzzle exit as a function of time from a spline-fit of the corresponding tabular input data.</td>
</tr>
<tr>
<td>NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>8. PUNCH</td>
<td>Writes the final time step on TAPE12 for restarting the calculation (see Section III(B) for an explanation of the restart capability of the program).</td>
</tr>
<tr>
<td>9. SETN</td>
<td>Reinitializes the arrays in which the new solution at time t+Δt will be stored.</td>
</tr>
<tr>
<td>10. SHOCK</td>
<td>Calculates the position and velocity of a moving shock and the flow properties on the downstream side of the shock. The index K in the calling sequence is used to denote the Mach disc (K=1) and the blast wave (K=3).</td>
</tr>
<tr>
<td>11. SPLINE</td>
<td>Performs a spline-fit of tabular data (see Reference 4).</td>
</tr>
<tr>
<td>12. SPLINT</td>
<td>Uses the spline-fit coefficients to interpolate data at arbitrary values of the independent argument. (First and second derivatives of the dependent variable are also calculated, but not used in the present program.)</td>
</tr>
<tr>
<td>13. STEP</td>
<td>Computes the maximum allowable time step, Δt, based on the Courant, Friedrichs and Lewy criterion.</td>
</tr>
<tr>
<td>14. INTER</td>
<td>Interpolates data when adding or deleting a grid point.</td>
</tr>
<tr>
<td>15. INIT</td>
<td>Determines the initial location and fluid properties of the Mach disc, contact surface and blast wave.</td>
</tr>
<tr>
<td>16. ACOUS</td>
<td>Calls the various subroutines associated with the integration of the acoustic wave equation.</td>
</tr>
<tr>
<td>17. INT1A</td>
<td>Performs the integration of the acoustic wave equation using a fourth-order Runke-Kutta method.</td>
</tr>
</tbody>
</table>
18. **DERSUB**

Subroutine which evaluates the derivative used in the Runge-Kutta integration.

19. **CHSUB**

Dummy subroutine called from subroutine INT1A.

---

**TABLE II**

**PRINCIPAL PROGRAM VARIABLES**

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR(1)</td>
<td>Number of grid points in region 1 (i.e., from Mach disc to the contact).</td>
</tr>
<tr>
<td>IR(2)</td>
<td>Total number of grid points (including points on both sides of both shocks and the contact except for the free stream side of the blast wave).</td>
</tr>
<tr>
<td>Z</td>
<td>Axial distance (in feet).</td>
</tr>
<tr>
<td>P</td>
<td>Natural logarithm of pressure (in psf).</td>
</tr>
<tr>
<td>U</td>
<td>Gas velocity (in fps).</td>
</tr>
<tr>
<td>S</td>
<td>Entropy (in ft$^2$/sec$^2$-$^0$R)</td>
</tr>
<tr>
<td>A</td>
<td>Speed of sound (in fps).</td>
</tr>
<tr>
<td>RH</td>
<td>Gas density (in slugs/ft$^3$).</td>
</tr>
<tr>
<td>W</td>
<td>Velocity of surface of discontinuity.</td>
</tr>
<tr>
<td>GAM1, GAM2</td>
<td>Ratio of specific heats for regions 1 and 2, respectively.</td>
</tr>
<tr>
<td>FORTRAN NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CP1, CP2</td>
<td>Specific heat at constant pressure for regions 1 and 2, respectively.</td>
</tr>
<tr>
<td>RJet</td>
<td>Radius of gun bore.</td>
</tr>
<tr>
<td>XME</td>
<td>Muzzle exit Mach number.</td>
</tr>
<tr>
<td>PE</td>
<td>Muzzle exit pressure (in psf).</td>
</tr>
<tr>
<td>AE</td>
<td>Sound speed at muzzle exit (in fps).</td>
</tr>
<tr>
<td>PF</td>
<td>Ambient pressure (in psf).</td>
</tr>
<tr>
<td>UINF</td>
<td>Ambient gas velocity (in fps) (usually 0.0).</td>
</tr>
<tr>
<td>AINF</td>
<td>Speed of sound in ambient gas (in fps).</td>
</tr>
<tr>
<td>RHINF</td>
<td>Ambient gas density (slugs/ft³).</td>
</tr>
<tr>
<td>TIME</td>
<td>Current time (seconds).</td>
</tr>
<tr>
<td>TIMEF</td>
<td>Final time (seconds).</td>
</tr>
<tr>
<td>KK</td>
<td>Maximum number of time steps.</td>
</tr>
<tr>
<td>KO</td>
<td>Current number of completed time steps.</td>
</tr>
<tr>
<td>LL</td>
<td>Interval in the number of time steps between printing of complete flow field solution.</td>
</tr>
<tr>
<td>IPUNCH</td>
<td>Index for option to write final solution on TAPE12.</td>
</tr>
<tr>
<td>FORTRAN NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DT</td>
<td>Time increment.</td>
</tr>
<tr>
<td>DZ</td>
<td>Increment in axial distance.</td>
</tr>
</tbody>
</table>

Definition of other program variables should be self-evident from the context of their usage.
### A. Punched Card Format

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Format</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>1-5</td>
<td>starting step of run</td>
</tr>
<tr>
<td>15</td>
<td>6-10</td>
<td></td>
<td>final step of run</td>
</tr>
<tr>
<td>15</td>
<td>11-15</td>
<td></td>
<td>print interval</td>
</tr>
<tr>
<td>15</td>
<td>16-20</td>
<td></td>
<td>output file creation index (0 - no output file created; 1 - output file created on TAPE12)</td>
</tr>
<tr>
<td>15</td>
<td>21-25</td>
<td></td>
<td>restart indicator (0 - initial program submittal; 1 - program reads initial step from TAPE10)</td>
</tr>
<tr>
<td>15</td>
<td>26-30</td>
<td></td>
<td>option for moving origin of coordinate system (0 - fixed region; 1 - moving origin)</td>
</tr>
<tr>
<td>15</td>
<td>31-40</td>
<td></td>
<td>input unit index - specifies the units of the input (0 - metric; 1 - english)</td>
</tr>
<tr>
<td>15</td>
<td>41-45</td>
<td></td>
<td>output unit index - determines the units of the output (0 - metric; 1 - english; 2 - non-dimensional)</td>
</tr>
<tr>
<td>15</td>
<td>46-50</td>
<td></td>
<td>acoustic wave index (0 - normal program operation; 1 - only acoustic wave calculation performed)</td>
</tr>
<tr>
<td>2</td>
<td>E10.0</td>
<td>1-10</td>
<td>starting time of run (seconds)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>11-20</td>
<td>final time of run (seconds)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>21-30</td>
<td>multiplicative factor on maximum time step calculated from stability theory (recommended value is 1.0)</td>
</tr>
<tr>
<td>Card Number</td>
<td>Format</td>
<td>Columns</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>2</td>
<td>E10.0</td>
<td>31-40</td>
<td>multiplicative factor on minimum grid spacing as determined by initial number of grid points (recommended value is $\leq 1.0$)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>41-50</td>
<td>multiplicative factor on minimum grid spacing to determine maximum grid spacing (recommended value is $\geq 2.0$)</td>
</tr>
<tr>
<td>3</td>
<td>E10.0</td>
<td>1-10</td>
<td>radius of gun bore (m or ft)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>11-20</td>
<td>coordinate system index (0 - planar; 1 - cylindrical; 2 - spherical)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>21-30</td>
<td>ratio of distance from muzzle to origin of coordinate system to the distance from muzzle to the Mach disc</td>
</tr>
</tbody>
</table>

If the restart indicator is equal to one on card number 1, card number 4 is not required.

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Format</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>15</td>
<td>1-5</td>
<td>last data point in region 1 (shown as IR1 in Figure 1)</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>6-10</td>
<td>last data point in region 2 (shown as IR2 in Figure 1)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>11-20</td>
<td>specific heat ratio in region 1</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>21-30</td>
<td>Specific heat ratio in region 2</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>31-40</td>
<td>specific heat at constant pressure in region 1 ($m^2/sec^2<em>o_K$ or $ft^2/sec^2</em>o_R$)</td>
</tr>
</tbody>
</table>
FIGURE 1. SCHEMATIC OF GRID NETWORK
<table>
<thead>
<tr>
<th>Card Number</th>
<th>Format</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>E10.0</td>
<td>41-50</td>
<td>specific heat at constant pressure in region 2 ( \text{m}^2/\text{sec}^2 \text{K or } \text{ft}^2/\text{sec}^2 \text{R} )</td>
</tr>
<tr>
<td>5</td>
<td>E10.0</td>
<td>1-10</td>
<td>ambient pressure ( \text{N/m}^2 ) or ( \text{lb/ft}^2 )</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>11-20</td>
<td>ambient gas velocity ( \text{m/sec or ft/sec} ) (typically 0.0)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>21-30</td>
<td>ambient speed of sound ( \text{m/sec or ft/sec} )</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>31-40</td>
<td>blast wave Mach number at which acoustic wave analysis commences</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>1-5</td>
<td>INUM-number of points in table of muzzle exit properties (maximum of 25 points)</td>
</tr>
</tbody>
</table>

There are "INUM" of the following cards describing the muzzle exit properties.

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Format</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7a</td>
<td>E10.0</td>
<td>1-10</td>
<td>time (seconds)</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>11-20</td>
<td>pressure ( \text{N/m}^2 ) or ( \text{lb/ft}^2 )</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>21-30</td>
<td>speed of sound ( \text{m/sec or ft/sec} )</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>31-40</td>
<td>Mach number</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>1-5</td>
<td>INUMP-number of points in table describing the plume centerline Mach number distribution</td>
</tr>
</tbody>
</table>
The following cards describe the plume centerline Mach number distribution. A maximum of 35 points are allowed. Input the location of the point (ZZ) followed by the centerline Mach number (ZMCL) for that point. Input 4 pairs to a card and continue on the next card until all "INUMP" points have been input. A sample of the first card is shown below:

9a E10.0 1-10 ZZ(1) - axial distance from muzzle in bore radii

E10.0 11-20 ZMCL(1) - plume centerline Mach number at distance ZZ(1) from nozzle

E10.0 21-30 ZZ(2)

E10.0 31-40 ZMCL(2)

E10.0 41-50 ZZ(3)

E10.0 51-60 ZMCL(3)

E10.0 61-70 ZZ(4)

E10.0 71-80 ZMCL(4)

If the acoustic wave index is equal to zero on card 1, card number 10 is not required.

10 E10.0 1-10 the location of the contact surface (m or ft)

E10.0 11-20 the location of the blast wave (m or ft)

E10.0 21-30 the initial time step to be used by the Runge-Kutta integration routine (usually taken to be the final time step obtained by the stability requirements of the finite difference portion of the program) (seconds)
TH 184

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Format</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>E10.0</td>
<td>31-40</td>
<td>the constant in the Mach disk equation printed out at the last finite difference step</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>41-50</td>
<td>the constant in the blast wave equation printed out at the last finite difference step</td>
</tr>
<tr>
<td></td>
<td>E10.0</td>
<td>51-60</td>
<td>the constant in the velocity curve fit printed out at the last finite difference step</td>
</tr>
</tbody>
</table>

B. **Restart Capability** - The program possesses the capability of using a solution previously stored on disk or tape to initialize the necessary variables at each grid point. To create such a file, the output file creation index (card 1) must be set equal to one (1). When the final time step is reached the necessary variables are output on TAPE12 and may be saved by the appropriate use of control cards.

To make use of this file, the restart indicator (card 1) should be set equal to one (1). With the exception of card 2 which must be changed to correspond to the time step saved on the file and card 4 which must be omitted, the program input deck is the same. The variables are then read from TAPE10 and are used for the initiation of the program. At the completion of this run, the variables can again be output on TAPE12 if desired. Care must be taken to assure that the files and tapes correspond to the desired input and output data. Typical control cards that illustrate the use of the restart capability are shown in Figure (2).
CREATION OF OUTPUT FILE

JOB CARD
CHARGE CARD
REQUEST(TAPE12,*PF)
ATTACH(LGO,DAWN,CY=1,ID=ATLXXX)
MAP(PART)
LGO.
REWIND(TAPE12)
CATALOG(TAPE12,NAMESTEPXXX,CY=1,ID=ATLXXX)
EOR

RESTART USING OUTPUT FILE

JOB CARD
CHARGE CARD
REQUEST(TAPE12,*PF)
ATTACH(LGO,DAWN,CY=1,ID=ATLXXX)
MAP(PART)
ATTACH(TAPE10,NAMESTEPXXX,CY=1,ID=ATLXXX)
REWIND(TAPE10)
LGO.
REWIND(TAPE12)
CATALOG(TAPE12,NAMESTEPXXX,CY=1,ID=ATLXXX)
EOR

FIGURE 2. CDC SCOPE 2.1 OPERATING SYSTEM CONTROL CARDS ILLUSTRATING THE USE OF THE PROGRAM RESTART CAPABILITY
A. **Output Format** - The first page of the program output prints a narrative which informs the user of various input parameters such as the initial and final time steps, print interval, free stream properties etc. On the second page of output, the input values of the muzzle exit conditions and the plume centerline Mach number distribution are printed out in tabular form. The units of the input variables are specified by the input unit index as described previously in Section III.

On the following pages of the printout, starting with the initial step, the fluid dynamic properties are printed at each grid point. As a convenience to the user the locations of the Mach disc, contact surface and blast wave are specified in the output by the letters MD, C and BW respectively. At intermediate time steps where a complete output is not desired the velocity and position of the discontinuities are printed. The program has the capability of printing out the fluid variables in either metric or English units or in a non-dimensional form. The output unit index described in Section III controls which unit option is used. A table illustrating the unit option is contained in the following sub-section.

The form of the output for the acoustic wave computation is similar to that of the finite difference portion of the program except that only non-dimensional output is used. When the program reaches the time for switching to the acoustic analysis (i.e., when $W(3)/a_w < 1.10$), the final finite difference time step is output along with the necessary parameters for resubmitting an acoustic wave run.
B. Identification of Variables

STEP - total number of steps taken
TIME - total elapsed time (seconds)

<table>
<thead>
<tr>
<th>Description</th>
<th>Output Unit Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Z axial distance</td>
<td>meter</td>
</tr>
<tr>
<td>P pressure</td>
<td>N/m²</td>
</tr>
<tr>
<td>U velocity</td>
<td>m/sec</td>
</tr>
<tr>
<td>S entropy</td>
<td>m²/sec² oₚK</td>
</tr>
<tr>
<td>A speed of sound</td>
<td>m/sec</td>
</tr>
<tr>
<td>RH density</td>
<td>kg/m³</td>
</tr>
<tr>
<td>TEMP temperature</td>
<td>°K</td>
</tr>
<tr>
<td>MACH Mach number</td>
<td>-</td>
</tr>
<tr>
<td>W(1) velocity of Mach disc</td>
<td>m/sec</td>
</tr>
<tr>
<td>W(2) velocity of contact</td>
<td>m/sec</td>
</tr>
<tr>
<td>W(3) velocity of blast wave</td>
<td>m/sec</td>
</tr>
<tr>
<td>Z(1) location of Mach disc</td>
<td>meter</td>
</tr>
<tr>
<td>Z(2) location of contact</td>
<td>meter</td>
</tr>
<tr>
<td>Z(3) location of blast wave</td>
<td>meter</td>
</tr>
</tbody>
</table>

C. Sample Input and Output
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>10</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.3</td>
<td>E-021</td>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
<td>0.00913</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>0.40</td>
<td>7</td>
<td>151.25</td>
<td>1.40</td>
<td>9507.08</td>
<td>6006.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2120.0</td>
<td>0.0</td>
<td>1120.0</td>
<td>1.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.876</td>
<td>E-031</td>
<td>0.2981E+064</td>
<td>2.49E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.924</td>
<td>E-031</td>
<td>1.776E+063</td>
<td>9.386E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.972</td>
<td>E-031</td>
<td>0.918E+063</td>
<td>7.152E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.02</td>
<td>E-031</td>
<td>0.172E+063</td>
<td>5.246E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.068</td>
<td>E-039</td>
<td>5.173E+053</td>
<td>3.547E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.117</td>
<td>E-038</td>
<td>9.409E+053</td>
<td>2.008E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.165</td>
<td>E-038</td>
<td>4.269E+053</td>
<td>0.591E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.213</td>
<td>E-037</td>
<td>9.668E+052</td>
<td>9.272E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.261</td>
<td>E-037</td>
<td>4.03E+052</td>
<td>8.504E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.309</td>
<td>E-036</td>
<td>7.934E+052</td>
<td>8.140E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.357</td>
<td>E-036</td>
<td>2.439E+052</td>
<td>7.041E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.405</td>
<td>E-035</td>
<td>7.510E+052</td>
<td>7.592E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.453</td>
<td>E-035</td>
<td>3.093E+052</td>
<td>7.379E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.502</td>
<td>E-034</td>
<td>9.135E+052</td>
<td>7.192E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.550</td>
<td>E-034</td>
<td>5.585E+052</td>
<td>7.031E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.598</td>
<td>E-034</td>
<td>2.389E+052</td>
<td>6.090E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.640</td>
<td>E-033</td>
<td>9.509E+052</td>
<td>6.765E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.694</td>
<td>E-033</td>
<td>6.913E+052</td>
<td>6.650E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.742</td>
<td>E-033</td>
<td>4.561E+052</td>
<td>6.552E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.076</td>
<td>E-038</td>
<td>9.808E+052</td>
<td>3.228E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.076</td>
<td>E-033</td>
<td>2.268E+042</td>
<td>0.932E+031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3. Typical Input in Card Image Format**
MUZZLE BLAST ANALYSIS

STARTING STEP = 0
FINAL STEP = 10
PRINT INTERVAL = 10
STARTING TIME = 0. SEC.
FINAL TIME = .3000E-02 SEC.
LAST PT. REGION NO. 1 = 7
LAST PT. REGION NO. 2 = 13
MOVING COORDINATE SYSTEM OPTION = 0
RADIUS OF THE JET = .91300E-02 FT.
FLOW INDEX = 2
ORIGIN OF SPHERICAL SYSTEM = .400E+00 TIMES RJET
FREE STREAM PRESSURE = .21200E+04 LBS/FT/FT
FREE STREAM VELOCITY = 0. FT/SEC
FREE STREAM SPEED OF SOUND = .11200E+04 FT/SEC
SPECIFIC HEAT RATIO (REGION 1) = 1.25
SPECIFIC HEAT RATIO (REGION 2) = 1.40
SPECIFIC HEAT AT CONSTANT PRESSURE (REGION 1) = 9507.1 FT**2/SEC**2-DEG.R
SPECIFIC HEAT AT CONSTANT PRESSURE (REGION 2) = 6006.0 FT**2/SEC**2-DEG.R
BARREL EXIT PRESSURE = .12981E+07 LBS/FT/FT
BARREL EXIT MACH NO. = .10000E+01
BARREL EXIT SPEED OF SOUND = .42493E+04 FT/SEC

FIGURE 4. FIRST PAGE OF PROGRAM OUTPUT
<table>
<thead>
<tr>
<th>TIME</th>
<th>PRESSURE</th>
<th>SOUND SP</th>
<th>MACH</th>
<th>X/R</th>
<th>MACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>1.00E+01</td>
<td></td>
<td>1.00E+01</td>
<td>1.00E+00</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.4800E-04</td>
<td>1.117E+07</td>
<td>3.939E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.9600E-04</td>
<td>1.092E+07</td>
<td>3.715E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.1440E-03</td>
<td>1.017E+07</td>
<td>3.525E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.1920E-03</td>
<td>9.517E+06</td>
<td>3.355E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.2410E-03</td>
<td>8.941E+06</td>
<td>3.201E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.2890E-03</td>
<td>8.427E+06</td>
<td>3.059E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.3370E-03</td>
<td>7.967E+06</td>
<td>2.927E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.3850E-03</td>
<td>7.403E+06</td>
<td>2.850E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.4330E-03</td>
<td>6.793E+06</td>
<td>2.814E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.4810E-03</td>
<td>6.244E+06</td>
<td>2.784E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.5290E-03</td>
<td>5.751E+06</td>
<td>2.759E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.5770E-03</td>
<td>5.309E+06</td>
<td>2.738E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.6260E-03</td>
<td>4.914E+06</td>
<td>2.719E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.6740E-03</td>
<td>4.559E+06</td>
<td>2.703E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.7220E-03</td>
<td>4.239E+06</td>
<td>2.689E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.7700E-03</td>
<td>3.951E+06</td>
<td>2.677E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.8180E-03</td>
<td>3.691E+06</td>
<td>2.665E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.8660E-03</td>
<td>3.456E+06</td>
<td>2.655E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
<tr>
<td>.9040E-03</td>
<td>3.227E+05</td>
<td>2.645E+04</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
<td>1.00E+01</td>
</tr>
</tbody>
</table>

**Figure 5.** Second page of program output.
***Figure 6. Typical Output of Finite Difference Calculation***
### Figure 7. Typical Output of Acoustic Wave Analysis

<table>
<thead>
<tr>
<th>Z/D</th>
<th>P</th>
<th>RMD</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>.39490E+01</td>
<td>10000E+01</td>
<td>10000E+01</td>
</tr>
<tr>
<td>.18401E+03</td>
<td>10000E+01</td>
<td>10011E+01</td>
<td>42083E+02</td>
</tr>
<tr>
<td>.18736E+03</td>
<td>10016E+01</td>
<td>10024E+01</td>
<td></td>
</tr>
<tr>
<td>.18912E+03</td>
<td>10032E+01</td>
<td>10040E+01</td>
<td>64911E+02</td>
</tr>
<tr>
<td>.19088E+03</td>
<td>10053E+01</td>
<td>10061E+01</td>
<td>89150E+02</td>
</tr>
<tr>
<td>.19268E+03</td>
<td>10081E+01</td>
<td>10085E+01</td>
<td>11496E+01</td>
</tr>
<tr>
<td>.19452E+03</td>
<td>10117E+01</td>
<td>10113E+01</td>
<td>14252E+01</td>
</tr>
<tr>
<td>.19638E+03</td>
<td>10159E+01</td>
<td>10146E+01</td>
<td>17204E+01</td>
</tr>
<tr>
<td>.19829E+03</td>
<td>10207E+01</td>
<td>10182E+01</td>
<td>20381E+01</td>
</tr>
<tr>
<td>.20023E+03</td>
<td>10260E+01</td>
<td>10220E+01</td>
<td>23811E+01</td>
</tr>
<tr>
<td>.20221E+03</td>
<td>10316E+01</td>
<td>10258E+01</td>
<td>27524E+01</td>
</tr>
<tr>
<td>.20423E+03</td>
<td>10374E+01</td>
<td>10296E+01</td>
<td>31547E+01</td>
</tr>
<tr>
<td>.20630E+03</td>
<td>10432E+01</td>
<td>10333E+01</td>
<td>35899E+01</td>
</tr>
<tr>
<td>.20840E+03</td>
<td>10492E+01</td>
<td>10369E+01</td>
<td>40539E+01</td>
</tr>
<tr>
<td>.21055E+03</td>
<td>10553E+01</td>
<td>10405E+01</td>
<td>45406E+01</td>
</tr>
<tr>
<td>.21274E+03</td>
<td>10617E+01</td>
<td>10442E+01</td>
<td>50438E+01</td>
</tr>
<tr>
<td>.21497E+03</td>
<td>10684E+01</td>
<td>10482E+01</td>
<td>55575E+01</td>
</tr>
<tr>
<td>.21726E+03</td>
<td>10754E+01</td>
<td>10525E+01</td>
<td>60755E+01</td>
</tr>
<tr>
<td>.21959E+03</td>
<td>10830E+01</td>
<td>10572E+01</td>
<td>65970E+01</td>
</tr>
<tr>
<td>Bw</td>
<td>.22198E+03</td>
<td>10910E+01</td>
<td>10623E+01</td>
</tr>
</tbody>
</table>

**END CONDITION MET**
PROGRAM DNAA (INPUT, OUTPUT, PUNCH, TAPE5=INPUT, TAPE6=OUTPUT, 
1TAPE10, TAPE12)
COMMON/A/Z(200), P(200), U(200), RH(200), S(200), A(200), N(3)
COMMON/B/Z(200), PN(200), UN(200), RHN(200), SN(200), AN(200), N(3)
COMMON/D/IR(2), GAM, XJ, CP, DI, DT
COMMON/F/RJET, XME, PE, AE, PINF, AINF, UINF, RHINF
COMMON/G/KK, LL, KO, TIME, TIMEF
COMMON/IPU/ IPUNCH
COMMON/IZERO/IZERO, IMOVE, OZERO, DTZER, FACT
COMMON/GA/GAM1, GAM2, CP1, CP2
COMMON/BLAST/IB*, MCON, SPEED, RJET2
COMMON/DEBUG/IDBUG
COMMON/UNITS/IUNIT, IUDUM, IUNOUT, IUOUT, FTME, PUNIT, DEGRK, RHUNIT
COMMON/BNINAL/HMAICH
DATA H1/2H, M2/2MMD, H3/2H, C1, H4/2MHD/
DIR=0.0
CALL INDATA
FACT1=CP2*(GAM2-1.0)/GAM2
FACT2=FACT1
FACT1=CP1*(GAM1-1.0)/GAM1
3 CONTINUE
IR1=IR(1)
IR1P=IR1+1
IR2=IR(2)
IR2P=IR2+1
IF (KD, GE, KK) GO TO 4
IF (KO/LL)*LL.NE.KO) GO TO 5
4 CONTINUE
WRITE(6,7) KO, TIME
7 FORMAT((1H1,10X,*STEP =*,1D4,10X,*TIME =*,E13.5,1X,4HSEC.,//)
WRITE(6,8)
8 FORMAT((4X*11X,*ZE,13X,*PA,13X,*UE,13X,*SE,13X,*AE,12X,*RM))-111X:*TEMP*,10X,*MACH*)
I=IR(2)+1
FOUM=AINF*TIME
DO 9 I=1,11
FACREG=FACT1
IF (I, GT, IR1) FACREG=FACT2
MS=H1
IF (I, EQ, 1, OR, I, EQ, 2) MS=H2
IF (I, EQ, 1, OR, I, EQ, IR1) MS=H3
IF (I, GE, IR2) MS=H4
GO TO (22, 24, 26), IUOUT
22 CONTINUE
25=Z(I)*FTME
PS=EXP(P(I))*PUNIT
US=U(I)*FTME
AS=A(I)*FTME
RS=RH(I)*RHUNIT
SS=S(I)*FTME**2/DEGRK
TEMP=(PS/(RS*FACREG))/FTME**2/DEGRK
GO TO 50
24 CONTINUE
25=Z(I)
PS=EXP(P(I))
US=U(I)
AS=A(I)
RS=RH(I)
SS=S(I)

TEMP=(PS/(RS*FACREG))
GO TO 50

26 CONTINUE
ZS=Z(1)/RJET2
PS=EXP(P(I))/PINF
US=U(I)/AINF
AS=A(I)/AINF

RS=RH(I)/RMINF

TEMP=(PS/R5)*FACINF/FACREG
IF(1.GT.IR1) GO TO 30
S5=GAM1*(S(I)-S(1))/CP1
GO TO 40

30 CONTINUE
S5=GAM2*(S(I)-S(1R2))/CP2

50 CONTINUE

XM5=U(1)/A(1)
WRITE(6,10) H5,Z5,PS,US,SS,AS,H5,TEMP,XM5

2000 FORMAT(8X,5E14.5)
9 CONTINUE

10 FORMAT(15,1X,A2,3E14.5)
WRITE(6,12)

12 FORMAT(/)
IF(KO.GE.KK) GO TO 6
WRITE(6,21)

21 FORMAT(11X,M(1)*10*X*M(2)*10*X*M(3)*10*X*Z(1)*10*X*Z(2)*10*X*Z(3)*
110*X*TIME**,11X,*DT*)

5 CONTINUE
Z1=Z(1)/RJET2
Z2=Z(1R1)/RJET2
Z3=Z(1R2)/RJET2
IF(Z3.GT.1.0)

1B=CON=Z3 *(1,0=(AINF/M(3))*2)*SQRT(ALOG(Z3 ))
IF(Z3.GT.1.0) EN=CON=TIME*1.0E+06/(RJET2/AINF)*(B*CON*
1SQRT(ALOG(Z3 ))-Z3)*1.0E+06
ZMD=0.69*XME=SQRT(GAM1*PE/EXP(P(2)))
GO TO 32,34,36,1000

32 CONTINUE

W1DUM=W(1)*FTME
W2DUM=W(2)*FTME
W3DUM=W(3)*FTME
Z1DUM=Z(1)*FTME
Z2DUM=Z(1R1)*FTME
Z3DUM=Z(1R2)*FTME
GO TO 30

34 CONTINUE

W1DUM=W(1)
W2DUM=W(2)
W3DUM=W(3)
Z1DUM=Z(1)
Z2DUM=Z(1R1)
Z3DUM=Z(1R2)
GO TO 30

36 CONTINUE

W1DUM=W(1)/AINF
CONTINUE
WHITE(6,11) M1DOM,M2DOM,M3DOM,Z1DOM,Z2DOM,Z3DOM,TIME,DT
11 FORMAT(6X,8E14.5,2F8.2)
IF(1B6,EQ,0) GO TO 14
13 LL=LLDOM
ZCON=Z(IN1)
ZBn=Z(IN2)
RCT=Z(IN1P)-FDUM
SGN=SIGN(1,0,RCT)
DO 60 I=1,IR2
RCT=Z(I)-FDUM
SGN1=SIGN(1,0,RCT)
IF(SGN,EQ,SGN1) GO TO 59
15 IACT=1
16 IRCTM=11
17 RCTM=Z(IRCTM)-FDUM
TERM=(Z(IRCTM)/RJET2)**2*(RH(IRCTM)/RHlNF-1.0-U(IRCTM)/SPEED)
TERM=(Z(IRCTM)/RJET2)**2*(RH(IRCTM)/RHlNF-1.0-U(INCTM)/SPEED)
RATHCT=RCT/(RCT+RCT4)
RMORCT=TERM+RATRCT*(TERM1-TERM)
GO TO 70
59 CONTINUE
SGM=SGN1
60 CONTINUE
RMORCT=1.0
70 CONTINUE
WRITE(6,1000) CMD,DM,CMD,RMORCT
1000 FORMAT(6X,1X,*THE CONSTANT IN THE MACH DISC EQUATION IS=,E13.5,
1//,1X,*THE CONSTANT IN THE BLAST WAVE EQUATION IS=,E13.5,
1//,1X,*THE CONSTANT IN THE VELOCITY CURVE FIT IS=,E13.5)
CALL ACOUS(ZCON,ZbN,DT,CMD,RMORCT)
14 CONTINUE
CALL SETN
CALL STEP
CALL SHOCK(1)
CALL SHOCK(3)
CALL CONTACT(2)
CALL INT PT
IF(IMOVE,EO,0) GO TO 20
25 Z ZERO=ZN(1)+FACT
26 DIZERO=NN(1)+FACT
27 DTZER=NN(1)-A(1)/DT
28 CONTINUE
I1=IN(2)+1
DO 1 I=1,11
2 (I)=Z N(I)
P (I)=P N(I)
U (I)=U N(I)
8 (I)=S N(I)
A (I)=A N(I)
1 RH(I)=RH N(I)
DO 2 I=1,3
2 \( w(i) = w(n(i)) \)
\( k0 = k0 + 1 \)
TIME = TIME + DT
IF((\( x(3)/a_{\text{inf}} \)) \( \leq \) 0.9MACH) IBM = 1
IF(IBM.EQ.0) GO TO 3
CALL MUZZLE(TIME,PEXIT,AEXIT,XEXIT)
COND = Z(1)/(XEXIT*SORT(GAM1*PEXIT/PINF))/KJET2
LLDUM = LL
LL = 1
GO TO 3
6 CONTINUE
IF(IPUNCH.NE.0) CALL PUNCH
STOP
END
SUBROUTINE CL(IT1, BT)

COMMON/A/Z(200), P(200), U(200), RH(200), S(200), A(200), W(3)
COMMON/B/ZN(200), PN(200), UN(200), RHN(200), SN(200), AN(200), A(3)
COMMON/D/IR(2), GAM, XJ, CP, DZ, DT
COMMON/F/RJE1, XME, PE, AE, PINF, AINF, UINF, RNINF
COMMON/GA/GAM1, GAM2, CP1, CP2
COMMON/G/KK, LL, KU, TIME, TIMES
COMMON/MCL/ZZ(35), XMCL(35), EMCL(35), INUMP
COMMON/IN/PM(25), AM(25), XM(25), TM(25)
COMMON/DEBUG/IDBD

DIMENSION ZM(1), XM(1), DXM(1), D2XM(1)

DATA MAX, ERRVAL/1, 0, 0.01/

ZE=0.0
ITER=0
RATIO=0.5
IF(BI.EQ.0.0) UN(1)=U(1)
TINIT=TIME+DT
T=TM(I)
IF(IDBGL.EQ.0.0) WRITE(6,1) IT1, Z, RATIO, TIME, DT, TINIT, T, BT
CONTINUE
TDSM=S/RATIO*TINIT*(1.0-RATIO)*T
CALL MUZZLE(TDSM, PE, AE, XME)
UE=0.5*(XME*AE+UN(1))
TDSMP=TINIT*(ZN(1)-ZE)/UE
ERR=(TDSMP-TDSM)/TDSM
IF(IDBG1.EQ.0.0) WRITE(6,1) ITER, TDSM, TDSMP, PE, AE, XME, UE, ZN(1), ERR
IF(TDSMP.LT.11, UN, TDSMP, GT, TINIT) GO TO 100
IF(ABS(ERR).LT.11, ERRVAL) GO TO 300
ITER=ITER+1
IF(ITER.GT.20) GO TO 200
RATIO=(11-TDSMP)/(11-TINIT)
GO TO 20
CONTINUE
WRITE(6,1000)
1000 FORMAT(1X, "STREAMLINE IN SUBROUTINE CL IS OUT OF BOUNDS")
CALL EXIT
CONTINUE
WRITE(6,2000)
2000 FORMAT(1X, "MANY ITERATIONS IN SUBROUTINE CL")
CALL EXIT
CONTINUE
GAM=GAM1

I=1
ZZ1=ZN(1)/RJET+1,
G1=GAM-1,
XME=XME*XME
F1=1.0+G1*XME/2,
G3=GAM/G1
G4=1.0/G1
ZM(1)=ZZ1
CALL SPLINTZZ1, XMCL, INUMP, ZM, MAX, XM, DXM, D2XM, EMCL
XMCL=XM(1)
XMCL2=XMCL*XMCL
F2=1.0+G1*XME/2,
PPE=(F1/F2)*G3
RHRP=(F1/F2)*G4
PA(1)=PPE*PE
\begin{verbatim}
TM 184

PME = GAM*PE/AE
MHM(1) = RMH * RHE
AN(I) = SQRT(GAM*PN(1)/RHN(1))
UN(I) = XMCL*AN(I)
SN(I) = S(1) + CP1*(((ALOG(PN(1)) - P(1)))/GAM1 = ALOG(RHN(1)/RH(1)))
PN(1) = ALOG(PN(1))

IF(IDEBUG,EU,E) WRITE(6,1) I,PN(I),RHN(I),AN(I),UN(I),SN(I),
1S(1),P(1),RH(1)
RETURN
1 FORMAT(115,9E13,5)
END
\end{verbatim}
SUBROUTINE CONTCT(n)

COMMON/A/Z(200), U(200), RH(200), S(200), A(200), n(3)
COMMON/B/ZN(200), PN(200), UN(200), RHN(200), SN(200), AN(200), n(3)
COMMON/C/ZI(200), PI(200), UI(200), RHI(200), SI(200), AI(200)
COMMON/D/IR(2), GAM, XJ, CP, DZ, DT
COMMON/E/BT, AL
COMMON/GA/GAM1, GAM2, CP1, CP2
COMMON/ZZERO/ZZERO, IMOVE, DZZERO, DTZZERO, FACT

I=IR(1)
IT1=1
AL=1.
BT=0,

W(K)=W(K)
UN(1)=UN(K)
UN(I+1)=UN(I)

ZI(I)=Z(I)+(AL* W(K)+BT*W(K))*DT
1 ZI(I+1)=ZI(I)
U1=UN(K)
U2=U1
GAM=GAM1
CP=CP1
CALL LPOINT(I, 1, 1, 1)
GAM=GAM2
CP=CP2
CALL LPOINT(I+1, 1, 1, 1)
R1=ZI(1)-ZZERO
R2=ZI(2)-ZZERO
A1=GAM1*(AL/AN(1))+BT/VN(1))
B1=GAM2*(AL/AN(2)+BT/AN(I+1))

IF(IMOVE, 0, 0) GO TO 200
A4=GAM1*XJ*(AL*(U1(1)-ZZERO))/R1+BT*(UN(1)-ZZERO)/R1
A4=A4+DT
B4=GAM2*XJ*(AL*(U1(2)-ZZERO))/R2+BT*(UN(1+1)-ZZERO)/R2
B4=B4+DT
GO TO 210

200 A4=(AL*U1(1)/R1+BT*UN(I)/R1)*DT*XJ*GAM1
B4=(AL*U1(2)/R2+BT*UN(I+1)/R2)*XJ*GAM2+DT

210 PC1=PI(1)-(U1=U1(1))*A1-A4
PC2=PI(2)+U2=U1(2))*61-B4
APC1=EXP(PC1)
APC2=EXP(PC2)
ER1=(APC1+APC2)/(APC1+APC2)*2.0
IF(ABS(ER1), 0, 0) GO TO 7

IT1=IT1+1
IF(111, 0, 0) GO TO 777

111 FORMAT(5X, *TUU MANY ITERATIONS IN SUBROUTINE CONTCT*)
STOP

777 IF(IT1, GT, 2) GO TO 14

BET=UN(2)
ER1=ER1
BET1=BET
BET1=1.01*BET
GO TO 15

14 DTM=BET1-ER1*(BET-BET1)/(ER-ER1)
ER1=ER1
BET1 = BET
BET = DUM
15 \text{RHN}(K) = \text{BET}
\text{ZN}(I) = Z(I) + 5*(W(K) + \text{RHN}(K)) \times Q1
\text{GO TO 1}
7 \text{PN}(I) = \text{PC1}
\text{PN}(I+1) = \text{PN}(I)
\text{UN}(I) = U1
\text{UN}(I+1) = U2
\text{SN}(I) = S(I)
\text{SN}(I+1) = S(I+1)
\text{CK} = \text{ALOG}\left(\text{RHN}(I) \right) = P(I) / \text{GAM}1
\text{RHN}(I) = \text{EXP}\left(\text{RHN}(I)\right)
\text{CK} = \text{ALOG}\left(\text{RHN}(I+1) \right) = P(I+1) / \text{GAM}2
\text{RHN}(I+1) = \text{PN}(I+1) / \text{GAM2} = (\text{SN}(I+1) = S(I+1)) / \text{CP2} + \text{CK}
\text{RHN}(I+1) = \text{EXP}\left(\text{RHN}(I+1)\right)
\text{AN}(I) = \text{SQRT}\left(\text{GAM1} \times \text{EXP}(\text{PN}(I)) / \text{RHN}(I)\right)
\text{AN}(I+1) = \text{SQRT}\left(\text{GAM2} \times \text{EXP}(\text{PN}(I+1)) / \text{RHN}(I+1)\right)
\text{IF}(\text{BT} < \text{GT} \times 0.1) \text{ GO TO 9}
\text{BT} = 0.5
\text{AL} = 0.5
\text{IT} = 1
\text{GO TO 2}
9 \text{RETURN}
\text{END}
SUBROUTINE FS
COMMON/B/ZN(200),PN(200),UN(200),RMN(200),SN(200),AN(200),WN(3)
COMMON/D/IR(2),GAM,XJ,CP,DZ,DT
COMMON/F/RJET,XME,PE,AE,PINF,AINF,UNINF,RHINF
COMMON/GAM1,GAM2,CP1,CP2
1=IR(2)+1
PN(1)=PINF
UN(1)=UNINF
RMN(I)=RHINF
SN(I)=0,
AN(I)=SQRT(GAM2*PN(I)/RMN(I))
PN(1)=ALUG(PN(1))
RETURN
END
SUBROUTINE INDATA
COMMON/A/((200), P(200), U(200), RH(200), S(200), A(200), N(3))
COMMON/B/((200), P(200), UN(200), RN(200), SN(200), AN(200), N(3))
COMMON/D/((200), GAM, XJ, CP, DZ, DT
COMMON/F/JET, XME, PE, AE, PINF, AINF, UINF, RINF
COMMON/G/KK, LL, KU, TIME, TIMEF
COMMON/Z/ZM, ZMAX
COMMON/DTST/DST
COMMON/UNIT/UNIT
COMMON/BLAST/BLAST
COMMON/BANAL/BANAL
COMMON/UNIT/UNIT
COMMON/NUMITS/NUMITS
COMMON/DIMENSION/DIMENSION
DATA H1/HSEC,
DATA H2/SEC,
DATA H3/SEC,
DATA H4/SEC,
DATA H5/SEC,
DATA H6/SEC,
DATA H7/SEC,
DATA H8/HSEC,
DATA HSEC/HSEC,
DATA DEGK/DEGK,
DATA FACT/FACT

READ(5,1) KO, KK, LL, IPUNCH, IRRSTR, IMOVE, IUNIT, IUNOUT, IBN
IUDUM=IUNIT+1
IOUT=IUNOUT+1
HDUM=H1
IF(IMOVE, EQ, 1) HDUM=H2
READ(5,2) TIME, TIMEF, DTSTAB, DZMIN, DZMAX
READ(5,2) JET, XJ, FACT
IF(IUNIT, EQ, 0) KJET=JET/FTME
IF(IIRRSTR, GT, 0) GO TO 6
READ(5,3) IH(1), IH(2), GAM, GAM2, CP, CP2
IF(IUNIT, EQ, 1) GO TO 6
CP1=CP1/FTME**2/DEGK
CP2=CP2/FTME**2/DEGK
CONTINUE
READ(5,2) PINF, UINF, AINF, BANAL
IF(IUNIT, EQ, 1) GO TO 11
PINF=PINF/PUNIT
UINF=UINF/FTME
AINF=AINF/FTME
FACP=PUNIT
FA=FTME
CONTINUE
READ(5,1) INUM
DO 10 I=1, INUM
READ(S,2) TM(I),PM(I),AM(I),XMM(I)

IF(I.EQ.1) TM1=TM(I)
TM(I)=TM(I)-1
IF(IUNIT.EQ.1) GO TO 10
PM(I)=PM(I)/PUNIT
AM(I)=AM(I)/TIME

10 CONTINUE
CALL SPLINE(TM,PM,INUM,EMP)
CALL SPLINE(TM,AM,INUM,EMA)
CALL SPLINE(TM,XMM,INUM,EMM)
CALL MUZZLE(TIME,PE,AE,XME)
READ(S,1) INUM
READ(S,2) (ZZ(I),ZMCL(I),I=1,INUMP)
IF(18M.EQ.0) GO TO 7
READ(S,2) ZCON,ZBM,DT,CONMD,BWCON,RHORCT
IF(IUNIT.EQ.1) GO TO 7
ZBM=ZBM/FTME
ZCON=ZCON/FTME

7 CONTINUE
IF(IRSTRI.GT.0) GO TO 300

9 CONTINUE
RHINF=GAM2*PINF/AINF/AINF
SPEED=AINF
RJET2=RJET*2

IF(18M.EQ.1) CALL ACOUS(ZCON,ZBM,DT,CONMD,RHORCT)
IF(IUNIT.EQ.1) GO TO 320
CP1DUM=CP1*FTME**2*DEGRK
CP2DUM=CP2*FTME**2*DEGRK
PEDUM=PE*PUNIT
AEDUM=AE*FTME
RJDUM=RJET*FTME
PIDUM=PINF*PUNIT
UIDUM=UINF*FTME
AIDUM=AINF*FTME
GO TO 325

320 CONTINUE
CP1DUM=CP1
CP2DUM=CP2
PEDUM=PE
AEDUM=AE
RJDUM=RJET
PIDUM=PINF
UIDUM=UINF
AIDUM=AINF

325 CONTINUE
C*******************************************************
C WRITE INPUTS
C*******************************************************
WRITE(6,1632)
WRITE(6,200) KO,KK,LL,TIME,HB,TIMEF,HB,IR,IMOVE
J=J+4,5
WRITE(6,201) RJDUM,H3(IUDUM),J,FACT,HDUM
WRITE(6,202) PIDUM,H4(IUDUM),UIDUM,H5(IUDUM),AIDUM,H5(IUDJM)
WRITE(6,206) GAM1,GAM2,CP1DUM,MB(IUDUM),M7(IUDUM),CP2DUM,MB(IUDUM)
WRITE(6,207) PEDUM,H4(IUDUM),XME,AEDUM,H5(IUDUM)
WRITE(6,205)
WRITE (6, 204)
INUMQ = INUM
IF (INUMP, LT, INUMQ) INUMQ = INUMP
DO 326 I = 1, INUMQ
PDUMMY = PM(I) * FACP
ADUMMY = AM(I) * FACA
WRITE (6, 500) T(I), PDUMMY, ADUMMY, XM(I), ZZ(I), ZMCL(I)
326 CONTINUE
INUMQ = INUMQ + 1
IF (INUMP, EQ, INUMQ) GO TO 307
307 WRITE (6, 501) (ZZ(I), ZMCL(I), I = 1, INUMQ)
GO TO 308
308 CONTINUE
DO 327 I = INUMQ, INUM
PDUMMY = PM(I) * FACP
ADUMMY = AM(I) * FACA
WRITE (6, 502) T(I), PDUMMY, ADUMMY, XM(I)
327 CONTINUE
309 CONTINUE
DO 12 I = 1, INUMP
ZZ(I) = ZZ(I) + 1.0
12 CONTINUE
C**********************************************************************
C INITIALIZE FLOW PROPERTIES
C**********************************************************************
CALL SPLINE(ZZ, ZMCL, INUMP, EMCL)
IF (IRSTRT, GT, 0) GO TO 342
CALL INIT
IR1 = IR(1)
IR1 = IR(1) + 1
IR2 = IR(2)
IR2 = IR(2) + 1
IR4 = IR(4)
IR4 = IR(1) + 1
IR6 = IR(6)
IR6 = IR(2) + 1
P2 = EXP(P(2))
PIR1 = EXP(P(IR1))
CK = ALOG(RH(1)) * P(1) / GAM1
DO 330 I = 3, IR4
RAT = FLOAT(I - 2) / FLOAT(IR1 - 2)
Z(I) = Z(2) + (Z(IR1) - Z(2)) * RAT
U(I) = U(2) + (U(IR1) - U(2)) * RAT
S(I) = RAT * ALOG10(S(IR1) / S(2))
S(I) = S(I) * S(1)
P(I) = P2 + (PIR1 - P2) * RAT
RH(1) = ALOG(P(1)) / GAM1 = (S(I) - S(1)) / CP1 + C2
W(1) = EXP(RH(1))
A(1) = SQRT(GAM1 * P(1) / RH(1))
P(1) = ALOG(P(1))
330 CONTINUE
P2 = EXP(P(IR1))
PIR2 = EXP(P(IR2))
CK = ALOG(RH(INF)) = ALOG(P(INF)) / GAM2
DO 340 I = IR5, IR6
RAT = FLOAT(I - IR1) / FLOAT(IR2 - IR1)
Z(I) = Z(IR1) + (Z(IR2) - Z(IR1)) * RAT
-34-
TM 184

U(I)=U(IR1)+((U(IR2)-U(IR1)))*RAT
S(I)=RAT+ALOG(10*(S(IR2)/S(IR1)))
S(I)=10.*S(I)
S(I)=S(IR1)+S(I)
P(I)=P2*(PIR2-P2)*RAT
RH(I)=ALOG(P(I))/GM2-(S(I)-S(IR2))/CP2+CK
RH(I)=EXP(RH(I))
A(I)=SHTG(M2*P(I)/RH(I))
P(I)=ALOG(P(I))

340 CONTINUE
342 CONTINUE
RETURN
300 CONTINUE

C*****************************************************************************
C READ RESTANT VARIABLES
C*****************************************************************************
READ(10) IR(1),IR(2),GAM1,GAM2,CP1,CP2
READ(10) (A(N),N=1,3)
READ(10) ZERO,DOZERO,DTZER
I1=IR(2)+1
READ(10) (Z(I),P(I),U(I),RH(I),S(I),A(I),I=1,11)
IDUM1=IR(1)
IDUM2=IR(2)
ZDUM1=(Z(IDUM1)-Z(2))/FLOAT(IDUM1-2)
ZDUM2=(Z(IDUM2)-Z(IDUM1))/FLOAT(IDUM2-IDUM1-1)
DZMIN=DZMAX*DZMIN*1,001
GO TO 9
1 FORMAT(14I5)
2 FORMAT(8E10,0)
3 FORMAT(2I5,4E10,0)
22 FORMAT(4E15,0)
203 FORMAT(8E12.4)
1632 FORMAT(1HI////14X*U Z Z L E B L A S T A N A L Y S I S*///)
200 FORMAT(9X,*STARTING STEP **14/9X,*FINAL STEP **14/9X,*PRINT
INTERVAL **14/9X,*STARTING TIME **E13.5,2X,4X,
2/9X,*FINAL TIME **E13.5,2X,4X,/9X,
3*LAST PT, REGION NO, 1 **14/9X,*LAST PT, REGION NO, 2 **14/9X,
4 9X,MOVING COORDINATE SYSTEM OPTION **15/) 
201 FORMAT(9X,*RADIUS OF THE JET **E13.5,2X,4X,/9X,
1*FLOW INDEX **12/9X,*ORIGIN OF SPHERICAL SYSTEM **E10.5,* TIMES *
144/)
202 FORMAT(9X,*FREE STREAM PRESSURE **E13.5,2X,4X,/9X,*FREE STREAM VELOCITY **E13.5,2X,4X,/9X,*FREE STREAM SPEED OF
1 SOUND **E13.5,2X,4X,/) 
205 FORMAT(1HI////15X,MUZZLE EXIT CONDITIONS*19X,PLUME CENTERLINE MACH
NUMBER/) 
204 FORMAT(5X,*MUZZLE PRESSURE**4X,SOUND SPEED**16X**4X,MACH*) 
206 FORMAT(9X,*SPECIFIC HEAT RATIO (REGION 1) **F4.2,/9X,*SPECIFIC HEAT RATIO (REGION 2) **F4.2,/9X,*SPECIFIC HEAT AT CONSTANT PRESSURE
1 REGION (REGION 1) **F7.1,2X,4X,/) 
2 **F7.1,2X,4X,/) 
3 /*9X,*SPECIFIC HEAT AT CONSTANT PRESSURE(REGION
42) **F7.1,2X,4X,/) 
207 FORMAT(9X,*BARREL EXIT PRESSURE **E13.5,2X,4X,/9X,*BARREL
1 EXIT MACH NO, **E13.5,/9X,*BARREL EXIT SPEED OF SOUND **E13.5,2X,
2,4X,/) 
500 FORMAT(4E12.4,8X,2E12.4)
501 FORMAT(56X,2E12.4)
502 FORMAT(4E12.4)
END
SUBROUTINE INT PT
COMMON/A/Z(200), P(200), U(200), RH(200), S(200), A(200), N(3)
COMMON/B/ZN(200), PN(200), UN(200), RHN(200), SN(200), AN(200), NN(3)
COMMON/C/ZI(200), PI(200), UI(200), RMI(200), SI(200), AI(200)
COMMON/D/IR(2), GAM, XJ, CP, DZ, DT
COMMON/E/BT, AL
COMMON/GA/GAM1, GAM2, CP1, CP2
COMMON/IZERO/IZZERO, IMOVE, DZZERO, DZZER, FACT
AL*5, BT*5
K=1
GAM=GAM1
CP=CP1
NCL=2
NCM=IR(1)
88 CONTINUE
NCLI=NCL+1
NCM1=NCM-1
M=NCM+1
DO 20 I=NCL, NCM
U(I)=U(I)
P(I)=P(I)
S(I)=S(I)
A(I)=A(I)
20 HRI(I)=RH(I)
DX=1./FLOAT(NCM=NCL)
LOOP=0
DEL=ZN(NCM)-ZN(NCL)
DD=1./DEL
8 XI=0,
IF(LOOP,LT,1) GO TO 30
WK=W(K)
WKP=W(K+1)
GO TO 31
30 WK=W(K)
WKP=W(K+1)
31 CONTINUE
DO 1 I=NCLI, NCM
XI=XI+DX
NM1=I-LOOP
NP1=NM1+1
EE=OU*EXP(PI1)/RHI(I)
CC=DD*(UI1)*XI+1,=WK =XI*WK
FF=DD*GAM
PX=(PI(NP1)-PI(NM1))/DX
UX=(UI(NP1)-UI(NM1))/DX
SX=(SI(NP1)-SI(NM1))/DX
HI=Z(I)
IF(LOP1,GT,0) R1=Z(N(I)
R1=R1=DZERO
URELI=UI(I)=FLOAT(IMOVE)*DZZERO
PT=(CC*PX+FF*UX+XJ*GAM*URELI/R1)
UT=(CC*UX+EE*PX)
ST=(CC*SX)
ZN(I)=ZN(NCM)+XI*DEL
IF(LOOP, EQ, 1) GO TO 7
PN(I)=PI1+PI*DT
UN(1) = UI(1) + UT * DT
C
SN(1) = SI(1) + SI * DT
IO = 1
IF (UN(1) .LT. 0.) IU = -1
CALL LPOINT(1, 0., IU, M)
SN(1) = SI(M)
CK = ALUG(TH(1)) = P(1)/GAM
RHN(1) = PN(1)/GAM = (SN(1) - S(1))/CP * CK
RHN(1) = EXP(RHN(1))
AN(1) = SORT(GAM * EXP(PN(1))/RHN(1))
GO TO 2
7 PN(1) = 5 * (PI(1) + P(1) + PT * DT)
C
UN(1) = 5 * (UI(1) + U(1) + UT * DT)
C
SN(1) = 5 * (SI(1) + S(1) + SI * DT)
IO = 1
IF (UN(1) .LT. 0.) IO = -1
CALL LPOINT(1, 0., IU, M)
SN(1) = SI(M)
CK = ALUG(TH(1)) = P(1)/GAM
RHN(1) = PN(1)/GAM = (SN(1) - S(1))/CP * CK
RHN(1) = EXP(RHN(1))
AN(1) = SORT(GAM * EXP(PN(1))/RHN(1))
2 CONTINUE
1 CONTINUE
IF (LOOP.EQ.1) GO TO 10
DO 3 I = NCL, NCM
U I(1) = UI N(I)
P I(1) = P N(I)
S I(1) = SN N(I)
A I(1) = AN N(I)
RH(1(I) = RHN(I)
3 CONTINUE
LOOP = 1
GO TO 8
10 IF (K.EQ.2) GO TO 66
K = K + 1
NCL = NCM + 1
NCM = 1H(2)
GAM = GAM2
CP = CP2
GO TO 88
66 RETURN
END
SUBROUTINE LPOINT(N, OPT, IO, M)

COMMON /A/Z(200), P(200), U(200), R(200), S(200), A(200), M(3)
COMMON /B/ZN(200), PN(200), UN(200), RHN(200), SN(200), AN(200), MN(3)
COMMON /C/ZI(200), PI(200), UI(200), RHI(200), SI(200), AI(200)
COMMON /D/IR(2), GAM, XJ, CP, DZ, DT

COMMON /E/H, AL

KJ=1
IF(OPT.EQ.0.) KJ=0
FIO=FLOAT(IO)
I=N

4 K=I-I0
E1=HT*(UN(I)+OPT*AN(I))
EM1=AL*(U(I)+OPT*A(I))+E1
EM1L=AL*(U(K)+OPT*A(K))+E1
ZB=(Z(I)+Z(K))/2.
KIP=0

88 CONTINUE
RAT=(ZH-Z(I))/(Z(K)-Z(I))
IF(ABS(RAT).LT.1.E-10) GO TO 1
WRITE(6,111)
111 FORMAT(SX,*CHARACTERISTIC SHOT BACK IS OUT OF RANGE IN SUBROUTINE
LPOINT*)
WRITE(6,113) I,K,J,IO,ZB,Z(K),Z(I)
113 FORMAT(* I,K,IO,ZB,Z(K),Z(I)*115,3E14,4)
1 EM1=EM1N+RAT*(EM1L-EM1N)
ZBI=ZH
ZB=ZN(I)-E^1*DT
IF(KJ.GT.0) GO TO 2
IF(FIU*Z(I)=FIO=ZB) 3,2,2
3 KJ=1
IO=IO
GO TO 4
2 CONTINUE
TESTZ=ABS((ZH-ZBI)/(Z(K)-Z(I)))
IF(TESTZ.LT.01) GO TO 86
KIP=KIP+1
IF(KIP.LE.15) GO TO 88
WRITE(6,112)
112 FORMAT(SX,*100 MANY ITERATIONS IN SUBROUTINE LPOINT*)
STOP
86 CONTINUE
ZH(I)=ZB
U I(M)=U (I)+RAT*(U (K)-U (I))
SI(M)=S(1)+RAT*(S(K)-S(I))
P I(M)=P (I)+RAT*(P (K)-P (I))
CK=ALOG(RHI(I))/P(I)/GAM
RHI(I)=PI(M)/GAM-(SI(M)-S(I))/CP+CK
RHI(I)=EXP(RHI(M))
A(M)=SORT(GAM*EXP(PI(M))/RHI(M))
RETURN
END

-39-
SUBROUTINE MUZZLE(TIME, PE, AE, XME)
COMMON/INIT/P(M(25), AM(25), XMM(25), IM(25))
COMMON/FIT/EMP(25), EMA(25), EMM(25), INUM
DIMENSION TD(1), PD(1), DPD(1), D2PD(1)
MAX=1
TD(1)=TIME
CALL SPLINT(TM, PM, INUM, TD, MAX, PD, DPD, D2PD, EMP)
PE=PD(1)
CALL SPLINT(TM, AM, INUM, TD, MAX, PD, DPD, D2PD, EMA)
AE=PD(1)
CALL SPLINT(TM, XMM, INUM, TD, MAX, PD, DPD, D2PD, EMM)
XME=PD(1)
RETURN
END
SUBROUTINE PUNCH
COMMON/A/Z(200),P(200),U(200),RH(200),S(200),A(200),M(3)
COMMON/D/H(2),GAM,XJ,CP,DZ,DT
COMMON/GA/GAM1,GAM2,CP1,CP2
COMMON/ZZERO/ZZERO,MOVE,DZERO,DTZER,FACT
WRITE(12) M(1),IR(2),GAM1,GAM2,CP1,CP2
WRITE(12) (A(N),N=1,3)
WRITE(12) ZZERO,DZERO,DTZER
I1=IR(2)+1
WRITE(12) (Z(1),P(1),U(1),RH(1),S(1),A(1),I=1,11)
RETURN
END
SUBROUTINE SELN

COMMON/A/Z(200),P(200),U(200),RH(200),S(200),A(200),W(3)
COMMON/B/ZH(200),PN(200),UN(200),RN(200),SN(200),AN(200),WN(3)
COMMON/D/IR(2),GAM,XJ,CP,DZ,DI

I1=1K(2)+1
DO 1 I=1,11
   A N(I)=A (I)
   P N(I)=P (I)
   U N(I)=U (I)
   S N(I)=S (I)
   RH N(I)=RH(I)
1 Z N(I)=Z (I)
DO 2 I=1,3
   WN(I)=W(I)
RETURN
END
SUBROUTINE SMOCK(K)
COMMON/A/Z(200),P(200),U(200),RM(200),S(200),A(200),M(3)
COMMON/B/ZN(200),PN(200),UN(200),RHN(200),SN(200),AN(200),MN(3)
COMMON/C/Z(200),PI(200),UI(200),RHI(200),SI(200),AI(200)
COMMON/D/IR(2),GAM,X,J,CP,OZ,DT
COMMON/E/IR(2),AL
COMMON/GA/GAM1,GAM2,CP1,CP2
COMMON/ZZERO/ZZERO,MOVE,ZZERO,ZZZER,FACT
COMMON/DEBUG/DEBUG
COMMON/BLAS1/IBA,IBC,CON,SPEED,RJET2
REAL MREL, MREL1,MREL2
!
M=1
IF(K,EQ,3)GO TO 3
GAM=GAM1
CP=CP1
!
I=1
N=2
OPT=1.0
GO TO 4
3 CONTINUE
GAM=GAM2
CP=CP2
I=IR(2)+1
N=IR(2)
OPT=1.0
4 CONTINUE
G2=GAM+1.
G1=GAM-1.
AL=1.
BT=0.
WN(K)=W(K)
MREL=(U(I)-W(K))/A(I)
IO=OPT
L=TN-IFIX(OPT)
IFLAG=0
STEP=0.52
IFLG=0
2 CONTINUE
IF(ABS(MREL).LE.1.0,AND.IT.GT.2)IFLG=1
IF(IFLG,EQ,1)MREL=SIGN(1.0,MREL)
ZM(I)=Z(1)+(AL*A(K)+BT*W(K))*DT
1 ZK(N)=ZN(I)
IF(K,EQ,3)GO TO 6
CALL CL(IT1,BT)
WN(K)=UN(I)-MREL*AN(I)
GO TO 5
6 CONTINUE
CALL FS
IF(IBw,EQ,0)WN(K)=UN(I)-MREL*AN(I)
ZDUM=ZN(N)/RJET2
IF(IBw,EQ,1)WN(K)=SPEED/SORT(1.0=8wCON*(SORT(ALOG(ZDUM))/ZDUM))
IF(IBw,EQ,1,AND.BT.EQ,0.0)GO TO 8
5 CONTINUE
V1=UN(I)-WN(K)
RHI=RHN(I)
P1=PN(I)
THXmxVl/AN(I)

XM1=VI/AN(I)
XM12=XM1*XM1
V12=G2*XM12/(1+XM12+2)
V2=VI/V12
RH2=RH1+V12
P2=ALOG((2*GAM*XM12)=G1)/G2)*P1
U2=V2+WN(K)
AP2=EXP(P2)
IF (FLGM.EQ.1) GO TO 7
IF (IEQ.EQ.1.AND.K.EQ.3) GO TO 7
CALL LPINT(N,OPT,10,M)
A6=GAM/(AL+2)*HI=AN(N)
IF (IOMOVE.EQ.0) GO TO 200
A7=GAM*XI/(AL+2)*OZ=U1(1)-OZ=U1(1)/(Z1(1)=Z1(1))
1+BT*(UN(N)=OZ=U1(1)/(Z1(1)-OZ=U1(1))
A7=A7*DI
GO TO 210
200 A7=KJ*GAM*DI*(AN(K)-AN(K))/(Z1(1)=Z1(1))
IT=IT+1
GO TO 210
210 P2S=P1(1)=OP1*A6*(U2=U1(I))
P2S=P2S-A7
AP2S=EXP(P2S)
ERROR=(AP2=AP2S)/(AP2S)*2.0
IF (IDLEogle.Q.EQ.2) WRITE(6,10) IT, I, N
IF (IDLEogle.Q.EQ.2) WRITE(6,20) MREL, ZN(I), ZN(N), AN(K), UN(I), AN(I),
V1, RHI, P1, XM1
IF (IDLEogle.Q.EQ.2) WRITE(6,20) XM12, V12, V2, RH2, P2, V2, AP2, A6, A7, P2S
IF (IDLEogle.Q.EQ.2) WRITE(6,20) AP2S, ERRUW, XJ
IF (ABS(ERROR),LT,1.04) GO TO 7
S=SIGN(1., ERROR)
SSTEP=SIGN(1., S)
IF (ABS(MREL),LE,1.05.AND.ABS(STEP),NE.0,01) STEP=0.0025*SSTEP
IT=IT+1
IF (IT.GT.75) GO TO 110
IF (IT.GT.2) GO TO 40
50 CONTINUE
MREL= MREL
ER1=ERROR
S2=S1
MREL=MREL+STEP
GO TO 2
40 CONTINUE
IF (S1.NE.S2.UN.2 IFMAG.EQ.1) GO TO 45
IF (ABS(ERROR).LE,ABS(ER1)) GO TO 50
STEP=STEP
GO TO 50
110 CONTINUE
WRITE(*,111) K, MREL, U2, RH2, P2
111 FORMAT(5X,*YOU MANY ITERATIONS IN SUBROUTINE SHUCK*,2X,*K*,15,2X,
1*MREL=*,E13,5,2X,*U2=*,E13,5,2X,*RH2=*,E13,5,2X,*P2=*,E13,5)
STOP
120 CONTINUE
WRITE(6,121)
121 FORMAT(5X,*ABSOLUTE VALUE OF MACH NUMBER IS LESS THAN ONE*)
CALL EXIT
45 CONTINUE
MREL2=MREL1-ER1.*(MREL-MREL1)/ERROR-ER1)
MREL1=MREL
ER1=ERROR
MREL=MREL2
IFLAG=1
GO TO 2

7 UN(N)=U2
RHN(N)=RH2
PN(N)=P2
SN(N)=SN(I)+CP*((PN(N)-PN(I))/GAM=ALOG(RHN(N)/RHN(I))
AN(N)=SORT(GAM*A/P2/RH2)
IF(BT.GT.0.) GO TO 9
8 CONTINUE
BT=.5
AL=.5
IT=1
IFLAG=0
STEP=0.02
IFLG0=0
GO TO 2
9 RETURN
10 FORMAT(14I5)
20 FORMAT(10E13.5)
END
SUBROUTINE SPLINE(X,Y,N,EM)
DIMENSION X(N),Y(N)
DIMENSION SH(25),G(25),EM(25)
DATA CB(1),G(1)/-0.5,0.0/1
NO=N-1
IF(NO.LT.2) GO TO 20
DO 10 I=2,NO
A = (X(I)-X(I-1))/b,
C = (X(I+1)-X(I))/b,
W = 2.0*(A+C)-4*SH(I-1)
SB(I) = C/N
F = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
10 G(I) = (F-A*C(I-1))/W
20 EM(N) = G(N-1)/(2.+SB(N-1))
DO 30 I=2,N
K = N+1-I
30 EM(K) = G(K)=SB(K)*EM(K+1)
RETURN
END
SUBROUTINE SPLINT(X,Y,N,Z,MAX,YINT,DYDX,DY2DX,EM)
DIMENSION X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX),DY2DX(MAX)
DIMENSION EM(25)
DATA SR/0/
INTEGER SR
III = SR
DO 140 I=1,MAX
K=2
60 IF(Z(I)-X(I)) 70, 60, 90
   YINT(I)=Y(I)
   SK = X(K)-X(K-1)
   GO TO 130
70 IF(Z(I),GE,1.X(I)-X(2)) GU TO 120
   WRITE(6,1000) Z(I)
   SR= 16
   GO TO 120
80 IF(Z(I),LE,1.X(N)-X(N-1)) GU TO 120
   WRITE(6,1000) Z(I)
   SR= 16
   GO TO 120
90 IF(Z(I),X(K)) 120,100,110
100 YINT(I)=Y(K)
   SK = X(K)-X(K-1)
   GO TO 130
110 IF(K-N) 90,90,90
120 CONTINUE
   SK = X(K)-X(K-1)
   YINT(I) = EM(K-1)*(X(K)-Z(I))*3/6/EM+EM(K)*Z(I)=X(K-1)*3/6/EM
   SK+EM(K)*SK /6.)*(Z(I)=X(K-1))*EM(K)-E(K-1)
   SK*EM(K)-EM(K-1)*SK/6.
   DYDX(I)=EM(K-1)*(X(K)-Z(I))/SK+EM(K)*(Z(I)=X(K-1))/SK
   DY2DX(I)=EM(K-1)*(X(K)-Z(I))/SK+EM(K)*(Z(I)-X(K-1))/SK
130 CONTINUE
   M= MAXO(N,MAX)
C   IF(SR,GE,16) WRITE(6,1010) N,MAX,(X(I),Y(I),Z(I),YINT(I),DYDX(I),
C   *DY2DX(I),I=1,M)
   SR= III
RETURN
1000 FORMAT (5$4H SPLINT USED FUR EXTRAPULATION, EXTRAPULATED VALUE =,
   +G14.6)
1010 FORMAT (2X,21HNU. OF POINTS GIVEN =,13,5OH, NO. OF INTERPOLATED PO
   +NTS =,13/10X,1MX,19X,1HY,16X,11HY=INTERPOL.,9X,11HY=INTERPOL.,
   +8X,14HDYDX=INTERPOL.,6X,15HDY2DX=INTERPOL.+(6E20,8))
END
SUBROUTINE STEP

COMMON/A/Z(200),P(200),U(200),RH(200),S(200),A(200),W(3)
COMMON/D/IR(2),GAM,XJ,CP,DZ,DT
COMMON/G/LL,KO,TIME,TIMEF
COMMON/DZ/DZMIN,DZMAX
COMMON/GAM1,GAM2,CP1,CP2
COMMON/DTSTAB/DTSTAB

DT = L.E+06
K = I
NCL = 2
NCM = IR(1)

5 CONTINUE

DZ = (Z(NCM) - Z(NCL))/FLOAT(NCM - NCL)
IF(DZ .GT. DZMIN) GO TO 10
IF(NCL .NE. 2) GO TO 20
IF(IR(1) .LE. 4) GO TO 10
GAM = GAM1
CP = CP1
CALL INTER(1, IR(1) , -1)
IR(1) = IR(1) + 1
GAM = GAM2
CP = CP2

20 IF(IR(2) = IR(1) , LE. 3) GO TO 10
CALL INTER(IR(1), IR(2), -1)
NCM = IR(2)
GO TO 5

10 CONTINUE

IF(DZ .LT. DZMAX) GO TO 40
IF(IR(2) .GE. 199) GO TO 40
IF(NCL .NE. 2) GO TO 30
GAM = GAM1
CP = CP1
CALL INTER(1, IR(1) , +1)
IR(1) = IR(1) + 1
GAM = GAM2
CP = CP2
CALL INTER(IR(1), IR(2), +1)
NCM = IR(2)
GO TO 5

30 DO 7 I = NCL , NCM

7 CONTINUE

IF(K .EQ. 2) GO TO 8
IF(IR(2) = IR(1) , LE. 2) GO TO 8
K = K + 1
NCL = NCM + 1
NCM = IR(2)
GO TO 5

-48-
8 CONTINUE
DT=DT*DTSTAB
TT=TIME+DT
IF(TT,LT,TIMEF) RETURN
DT=TIMEF-TIME
KK=K0
RETURN
END
SUBROUTINE INTER(I1, IF, IOPT)
COMMON/Z(200), P(200), U(200), RH(200), S(200), A(200), M(3)
COMMON/U*N(200), P(200), U*N(200), RH(N(200)), S(N(200), A(200)), M(N(3))
COMMON/1R(2), GAM, XJ, CP, DZ, OT
COMMON/LR, ZE, PE, AE, PNF, AIMF, UINF, RHINF
I1=I1+1
IFN=IF+IOPT
J1=I1
IDUM=1
IF(IF, EJ, IR(2)) IDUM=IR(2)+1
DO 1 I=I1, IFN
RAT=LUAT(I-I11)/FLOAT(I=I11)
ZN(I)=Z(I11)+Z(I1)+Z(I11)-Z(I1111)*RAT
DO 10 J=J1, IF
JU=J
IF(Z(J)=Z(N(I))) 10, 3, 2
10 CONTINUE
IF(Z(I1)+.0001, 61, ZN(I)) GO TO 3
WRIT(6, 100)
100 FORMAT(* ERRUM IN SUBROUTINE INTER*)
STOP
3 U N(I)=U (JU)
S N(I)=S (JU)
P N(I)=P (JU)
A N(I)=A (JU)
RMN(I)=RM(JU)
J1=JU
GO TO 1
2 JL=JU-1
RAT=(ZN(I)-Z(JL))/Z(JU)=Z(JL))
UN(I)=U(JL)+U(JU)-U(JL)**RAT
RMN(I)=RM(JL)+(RM(JU)-RM(JL))**RAT
P2=EXP(P(JL))
SN(I)=S(IDUM)+CP**((4LOG(PN(I))=P(IDUM))/(GAM=ALOG(RMN(I)/RH(IDUM))))
AN(I)=SORT(GAM*PN(I)/RMN(I))
PN(I)=ALOG(PN(I))
J1=JU
1 CONTINUE
1H1=IF1+1
1H1=IR(2)+1
DO 4 I=1F1, 1H21
J1=I+IOPT
ZN(J)=Z(I)
U N(J)=U (I)
S N(J)=S (I)
P N(J)=P (I)
A N(J)=A (I)
4 RMN(J)=RM(I)
IR(2)=1H2(2)+IOPT
1H21=IR(2)+1
DO 5 I=1I2, 1R21
Z (I)=Z(N(I)
U (I)=U N(I)
S (I)=S N(I)
P (I)=P N(I)
-50-
A(I) = A N(I)
5 RH(I) = RHN(I)
RETURN
END
SUBROUTINE INIT
COMMON/A(200), P(200), U(200), RH(200), S(200), A(200), n(3)
COMMON/B(200), PH(200), UH(200), RHU(200), SN(200), AN(200), n(3)
COMMON/DX/IR(2), GAMS, XJ, CP, DZ, DT
COMMON/FNJET, XME, PE, AE, PINF, AINF, UINF, RHINF
COMMON/GA/GAM1, GAM2, CP1, CP2
COMMON/DZ/DZMIN, DZMAX
COMMON/IT/EMP(25), EMA(25), EM(25), INUM
COMMON/INIT/P(25), AM(25), XMM(25), TM(25)
COMMON/MCL/Z3(35), ZCL(35), EMCL(35), IJUMP
COMMON/GAM, LL, K(1), TIME, TIMEF
COMMON/ZZERO, ZZERO, IMOVE, DZERO, DIZ, FACT
COMMON/DBG/10HUG
DIMENSION ZM(1), XM(1), DXM(1), D2X(1)
DATA TSTEP, ENHVAL, MAX/1, 1, 0.03, 0.001, 1/
DATA IJUMP/1, 2/
C**************************************************************************************
C SUBROUTINE INIT INITIALIZE ALL POINTS UPSTREAM AND DOWNSTREAM OF
C THE DISCONTINuITIES
C**************************************************************************************
DMIN=RJET
JDN=TINT(XJ+1, 5)
JME=JDN+1
G1=GAM1/1.0
G3=GAM1/G1
G4=1.0/G1
G5=GAM1+1.0
G6=GAM2+1.0
G7=GAM2+1.0
IP1=IR(1)
IP2=IP1+1
IP3=IR(2)
IP4=IP3+1
ZET=Z(1)*RJET-RJET
TINIT=T(1)+TM(INUM)-TM(1))/TSTEP
ITER=1
IF (IMOVE.EQ.1) GO TO 4
ZZERO=FACT*RJET
DZZERO=0.0
DIZZERO=0.0
4 CONTINUE
IF (IMOVE.EQ.1, AND, FACT.EQ.0.0, AND, JDN, NE, 1) IJUMP=3.0
5 CONTINUE
T1=TM(1)
CALL MUZLE(T1, P1, A1, XM1)
U1=XM1*A1
T2=TINIT
CALL MUZLE(T2, P2, A2, XM2)
U2=XM2*A2
DELT=TINIT-TM(1)
ITERZ=1
ZLOK=ZET
ZUP=ZET+U1*TINIT
10 CONTINUE
ZMD=ZLOK+0.2*(ZUP-ZLOK)
C**************************************************************************************
C DETERMINE MACH NUMBER AT MACH DISC-POINt 1
C******************************************************************************
ZM(1)=ZMD/Rjet+1.0
CALL SPLINT(ZZ,ZMC1,INUMP,ZM,MAX,XM,DXM,DXM,EMC1)
XMD=X(1)
C******************************************************************************
C LOCATE ORIGIN OF STREAMLINE AT MUZZLE EXIT
C******************************************************************************
ITEMS=1
RATIO=0.5
20 CONTINUE
TDUMS=RATIO*TINIT+(1.0-RATIO)*T1
UE=RATIO*U2+(1.0-RATIO)*U1
TDUMP=0.05*(ZMD-ZL)/UE
ERR=(TDUMS-TDUMP)/TDUMS
IF(TDUMP,LT,T1,OK,TDUMP,GT,TINIT) GO TO 650
IF(ABS(ERF),LT,ERFVAL) GO TO 300
ITER=ITER+1
IF(ITER,GT,20) GO TO 200
RATIO=(T1-TDUMP)/(T1-TINIT)
GO TO 20
200 CONTINUE
IF(IDEBUG.EQ,3) WRITE(6,2000)
2000 FORMAT(1X,100,May, MANY ITERATIONS FOR LOCATION OF STREAMLINE IN SUBRO
1LINE INIT*)
CALL EXIT
300 CONTINUE
PE=RATIO*P2+(1.0-RATIO)*P1
AE=RATIO*A2+(1.0-RATIO)*A1
XME=RATIO*XME2+(1.0-RATIO)*XME1
C******************************************************************************
C DETERMINE PROPERTIES AT POINT 1
C******************************************************************************
F1=1.0+0.5*G1*XME**2
F2=1.0+0.5*G1*XMD**2
PPE=(F1/F2)**G3
RMRE=(F1/F2)**G4
P(1)=PPE*PE
RM=GM1*PE/AE/AE
RH(1)=RMRE*RM
A(1)=SQR(T(GAM1*P(1)/RH(1))
U(1)=XMD*1.0
WTER=U(1)-1.10*A(1)
IF(IMOVE,GT,0) GO TO 15
ZERO=ZMD+FACT
DZERO=WTER+FACT
DZZERO=0.0
15 CONTINUE
IF(IDEBUG.EQ,3) WRITE(6,1)
1 particle, AE, XME, P(1), RH(1), A(1), U(1), WTER
C******************************************************************************
C DETERMINE PROPERTIES AT POINT 2
C******************************************************************************
XPT1=1.10
XI=(2.0*GM1*XPT1**2*G1)/G5
P(2)=XI*P(1)
FXI=(G5*XI+G1)/(G1*XI+G5)
U(2)=WTER+(U(1)-WTER)/FXI
C******************************************************************************
TM 184

RH(2) = FXI * RM(1)
CONSMD = RH(2) * (ZMD = ZZERO) ** JM
IF( IDEBUG.EQ.3 )
WRITE(*,1) XMPT1, XI, P(2), FXI, U(2), RM(2)
1

C DETERMINE BLAST WAVE PROPERTIES
C*****************************************************************************
X1B = P(2) / PINF
FX11 = (G7 * X1B + a) / (a * GAM2)
VSB = SURF(FX11 * GAM2 * PINF / RHINF)
FX12 = (G7 * X1B + a) / (a * GAM1)
RH(1R2) = FX12 * RHINF
UDUM2 = VSB = VSB / FX12
ZSHK = ZE + VSB ** DELT
CONSBN = RH(IN2) * UDUM2 * (ZSHK = ZZERO) ** JM
C*****************************************************************************

C DETERMINE PROPERTIES AT MD SIDE OF CONTACT
C*****************************************************************************

CONS1 = CONSMD * DELT / RH(2)
GO TO (102, 101, 100), JDOM
100 CONTINUE
ZC = 0.5 * (ZMD + ZSHK)
ITERZC = 1
30 CONTINUE
YDUM = ZC ** 3 - ZC ** 2 * ZZERO * ZC + ZZERO ** 2 * ZC
ZCUM = ZC - (YDUM / YDOM)
EZRZC = (ZC - ZCUM) / ZC
IF( IDEBUG.EQ.3 ) WRITE(*,3) ITERZC, ZC, ZCDUM, YDOM, YDOM, ERZC
3 IF( ITERZC.GT.20 ) GO TO 40
ITERZC = ITERZC + 1
IF( ABS(ERZC), LE, ERRVAL ) GO TO 50
ZC = ZCDUM
GO TO 30
40 CONTINUE
IF( IDEBUG.EQ.3 ) WRITE(*,5000)
5000 FORMAT(1X, * TOO MANY ITERATIONS FOR CONTACT POSITION IN SUBROUTINE INIT *)
GO TO 650
101 CONTINUE
ZC = (ZZERO * SURF(ZZERO ** 2 + 4.0 ** CONST)) / 2.0
GO TO 50
102 CONTINUE
ZC = CONST
50 CONTINUE
UC1 = ZC / DELT
PC1 = P(2)
RHC1 = RH(2)
C*****************************************************************************
C DETERMINE PROPERTIES AT BW SIDE OF CONTACT
C*****************************************************************************
PC2 = PC1
UC2 = UC1
RHC2 = CONSBN / (UC2 * (ZC = ZZERO) ** JM)

C*****************************************************************************
C CHECK SOLUTION USING DENSITY AT BW SIDE OF CONTACT
C*****************************************************************************
ITERZ=ITERZ+1
ERROR=(RH(1R2)-RHC2)/RH(1R2)
IF(IDEBUG,EQ,3)
WRITE(6,3) ITERZ,XIB,VSDB,RH(1R2),RHC2,ERROR,ZMD,ZC,ZSH<
IF(ABSERROR.LE.ERRTVAL) GO TO 700
IF(ITERZ.GT.20) GO TO 600
IF(ERROR,GTLT.0,0) GO TO 500
ZLOW=ZMD
GO TO 10
500 CONTINUE
ZUP=ZMD
GO TO 10
600 CONTINUE
IF(IDEBUG,EQ,3) WRITE(6,3000)
3000 FORMAT(1X,TOO MANY ITERATIONS FOR SHOCK VELOCITY IN SUBROUTINE IN)
110* CONTINUE
650 CONTINUE
IF(ITERZ.GT.20) GO TO 675
TINIT=TJUMP*TINIT
GO TO 5
675 CONTINUE
WRITE(6,4000)
4000 FORMAT(1X,TOO MANY ITERATIONS IN SUBROUTINE INIT FOR THE INITIAL)
1TIME STEP*)
CALL EXIT
700 CONTINUE
IF(ZC,LT.ZMD,OR.ZC,GTZSHK) GO TO 600
IF(ZMD,GE.0MIN) GO TO 750
TINIT=TJUMP*TINIT
GO TO 5
750 CONTINUE
C**************************************************************************
C DETERMINE ALL FLOW PROPERTIES AT THE DISCONTINUITIES
C**************************************************************************
TIME=TINIT
W(1)=WTER
Z(1)=ZMD
S(1)=0.0
A(1)=SQR(T(GAM1*P(1)/RH(1))
P(1)=ALOG(P(1))
Z(2)=Z(1)
S(2)=S(1)+CP1*((ALOG(P(2))-P(1))/GAM1=ALOG(RH(2)/RH(1)))
A(2)=SQR(T(GAM1*P(2)/RH(2))
P(2)=ALOG(P(2))
Z(IH1)=ZC
S(IH1)=S(2)
P(IH1)=ALOG(PC1)
RH(IH1)=RH(C1)
U(IH1)=UC1
A(IH1)=SQR(T(GAM1*PC1/RHC1))
W(2)=UC1
Z(IR2P)=ZSHK
P(IR2P)=ALOG(PINF)
U(IR2P)=UINF
RH(IR2P)=RHINF
S(IR2P)=ZFINF
-55-
A(IR2P)=SQRT(GAM2*PINF/RHINF)
Z(IR2)=Z(IR2P)
P(IR2)=P(2)
U(IR2)=UDUM2
S(IR2)=CP2*([P(IR2)-P(IR2P)]/GAM2*ALOG(RH(IR2)/RHINF))
Z(IR1P)=Z(IR1)
P(IR1P)=ALOG(PC2)
U(IR1P)=UC2
RH(IR1P)=RH(IR2)
S(IR1P)=S(IR2)
A(IR1P)=SQRT(GAM2*PC2/RHC2)
IDUM1=IN(1)
IDUM2=IR(2)
ZDUM1=(Z(IDUM1)-Z(2))/FLOAT(IDUM1-2)
ZDUM2=(Z(IDUM2)-Z(IDUM1))/FLOAT(IDUM2-IDUM1-1)
DZMIN=DZMIN*AMIN1(ZDUM1, ZDUM2)*0.999
DZMAX=DZMAX*DZMIN*1.001
RETURN
1 FORMAT(10E13.5)
2 FORMAT(14(I5))
3 FORMAT(115,8E13.5)
END
SUBROUTINE ACOUS(ZCON,ZBw,DT,COMM,MRHOR,CT)
C***************************************************************
C VAR(1) IS TIME
C VAR(2) IS Z
C***************************************************************
COMMON/F/RJET,XME,PE,AE,PINF,AINF,UINF,RHINF
COMMON/G/KK,LL,KO,TME,TFINAL
COMMON/BLAST/LL,KBACON,SPEED,RJET2
COMMON/KUTA/,CUVAR(2),VAR(2),CI,DER(2),II
DIMENSION ERRVAL(1),ELT(1),ELE1(1),ELE2(1)
DIMENSION RFIT(20),PFIT(20),RFITU(20),UFIT(20)
DIMENSION EMFITP(20),EMFITR(20),EMFITU(20)
DIMENSION XDUM(1),YDUM(1),DYDUM(1),D2YDUM(1)
DATA IFIT,ITEXT=0/1,
DATA (RFIT(T),I=1,20)/-50.0,-45.0,-40.0,-35.0,-30.0,-27.5,-25.0,
1 -22.5,-20.0,-17.5,-15.0,-12.5,-10.0,-7.5,0,5.0,-2.5,0,2.5,5.0,7.5/
2 -10.0,-2.5,0,2.5,5.0,7.5/
DATA (PFIT(T),I=1,20)/0.0,-0.15,-0.24,-0.42,-0.62,-0.82,-0.92,-0.92,0.0,-0.92,
1 1.03,-1.20,-1.53,-1.92,-2.33,-2.83,-3.22,3.72
2 4.23,5.0,
DATA (RFITU(T),I=1,20)/-50.0,-45.0,-40.0,-35.0,-30.0,-27.5,-25.0,
1 -22.5,-20.0,-17.5,-15.0,-12.5,-10.0,-7.5,0,5.0,-2.5,0,2.5,5.0,7.5/
2 -10.0,-2.5,0,2.5,5.0,7.5/
DATA (UFIT(T),I=1,20)/0.0,0.04,0.095,0.162,0.22,0.30,0.35,0.41,
1 0.52,0.65,0.77,0.825,0.91,0.97,1.0,0.955,
2 0.755,0.55,0.045,0.725/
DATA H/2H,,HOL2/2H,,HOL3/2H,,HOL4/2H C/ EXTERNAL DERSUB,CHSUB
DO 8 I=1,IFIT
RHOFIT(I)=PFIT(I)/GAM2
8 CONTINUE
CALL SPLINE(RFIT,PFIT,IFIT,EMFITP)
CALL SPLINE(RFIT,RHOFIT,IFIT,EMFITR)
CALL SPLINE(RFITU,UFIT,IFIT,EMFITU)
IMD=0
N=1
ITEXT=0
NT=1
ELT(1)=TFINAL
SPEC=0.0
C1=DT
CIMAX=0.0
ELE1(1)=0.0001
ELE2(1)=0.0
VAR(1)=TIME
VAR(2)=ZBw
II=0
3 CONTINUE
CALL INT1A(11,N,NT,CI,SPEC,CIMAX,IERH,VAR,CUVAR,DER,ELE1,ELE2,ELT,
1 ERRVAL,DERSUB,CHSUB,ITEXT)
K0=K0+1
GO TO (1,2,3,4,5)
1 CONTINUE
IF((KO,GE,4),OK,VAR(1),GE,TFINAL) GO TO 5
IF((KO/LL,LL,NE,KO) GO TO 150
5 CONTINUE
C** CALCULATION OF THE MACH DISC LOCATION
C*************************************************************************
IF(IND.EQ.1) GO TO 6
CALL MUZZLE(TDUM,PE,AE,XME)
PRAT=PE/PINF
IF(PRAT,GE,0.0) GO TO 7
IMD=1
GO TO 6
7 CONTINUE
ZMD=CONMD*XME*SORT(GAM1*PRAT)
IF(ZMD,LE,1.0) IMD=1
6 CONTINUE
C** CALCULATION OF THE ACOUSTIC WAVE PROPERTIES
C*************************************************************************
WRITE(6,4000)
IF(IND.EQ.0) WRITE(6,10) MOL3,ZMD
MOL=MOL4
DO 100 I=1,20
RATIO=FLUA(I-1)/19.0
RDUM=ZCON+RATIO*(ZDUM-ZCON)
XDUM(I)=(RDUM-SPEED*T0UM)/(HJET2*ALOG(RDUM/RJET2))
IF(XDUM(I),GE,RFIT(I)) GO TO 20
PRESS=1.0
RHO=1.0
UVAL=0.0
GO TO 30
20 CONTINUE
CALL SPLINT(RFIT,PFIT,IFIT,XDUM,IDUM,YDUM,DYDUM,D2YDUM,EMFITP)
PRESS=YDUM(I)*RJET2/RDUM(I)*0
CALL SPLINT(RFIT,RMOFIT,IFIT,XDUM,IDUM,YDUM,DYDUM,D2YDUM,EMFITR)
RHO=YDUM(I)*RJET2/RDUM(I)*0
CALL SPLINT(RFIT,UFIT,IFIT,XDUM,IDUM,YDUM,DYDUM,D2YDUM,EMFITU)
UVAL=RHO=1.0-YDUM(I)*RJET2/RDUM(I)*2*RHURCT
30 CONTINUE
IF(I,EQ,20) MOL=MOL2
ZDUMMY=RDUM/RJET2
WRITE(6,10) MOL,ZDUMMY,PRESS,RHO,UVAL
MOL=MOL1
100 CONTINUE
IF(KO,GE,KK,OR,VAR(I),GE,FINAL) GO TO 4
150 CONTINUE
GO TO(3,3,4,4),11
2 CONTINUE
WRITE(6,2000) IERR
CALL EXIT
4 CONTINUE
WRITE(6,3000)
STOP
10 FORMAT(1X,A2,4E13.5)
1000 FORMAT(1I1,10X,*STEP **,14,10X,*TIME **,E13.5,//)
2000 FORMAT(//,1X,20MINI_NON=CONVERGENCE,IERR=12)
3000 FORMAT(//,1X,17MEND CONDITION MET)
-58-
TM 184

4000 FORMAT(9X,3HZ/D,10X,1HP,12X,3HRM0,10X,1MU)
END
SUBROUTINE INT1A(N, NT, CI, SPEC, CIMAX, IERR, VAR, CVAR, DER, ELE1,
ELE2, ELT, ERRVAL, ODFSUB, CHSUB, ITEXT)

********* DOCUMENT DATE 08-01-68 SUBROUTINE REVISED 08-01-68 *********

DIMENSION S1VAR(20), SEL1(20), ELE1(20), ELE2(20), DER(1),
FDERV(20), SDY(20), SDY(20), VINC(20), ERRVAL(20), ERRV(20),
ELE2(10), SEL1(13), KELV(20), STEP(3)

DIMENSION VAR(21), CVAR(21)
INTEGER TCX(15)
INTEGER CODE, TPSM, SUMHAF, STEP, TEST, DCODE
REAL Kl

C BEGIN INITIALIZATION
IF (11 .GT. 0) GO TO 520
1 TP=0
SSPEC=SIGN(SPEC, CI)
SCIMAX=SIGN(CIMAX, CI)
VAR=VAR(1)
IF (CI .EQ. 0.0) GO TO 151
IF (SSPEC .EQ. 0.0) GO TO 7
IF (ABS(SCIMAX) .GT. ABS(SSPEC) OR, SCIMAX .EQ. 0.0)
1 SCIMAX=SSPEC

C TEST TO SEE IF VAR IS ZERO
IF (ABS(VAR1), GT, 1.0E-11) GO TO 2
TP=SSPEC
GO TO 7
2 IF (((VAR1/SSPEC) .GT. 1.0E-13) GO TO 4
3 K1=0.0
GOTO6
4 K1=1.0
6 TP=VAR1-AMOD(VAR1, SSPEC)
IF (ABS(1P-VAR1), LT, 1.0E-12) K1=1.0
TP=TP+K1*SSPEC
IF (ABS((TP-VAR1)/VAR1) .LT. 1.0E-11) TP = TP + SPEC

C TEST FOR DIRECTION OF INTEGRATION
7 K1=1.0
IF (CI .LT. 0.0) K1=-1.0
CI=CI*K1
CIMAX=SCIMAX*K1
TP=TP*K1
VARK=VAR1*K1

C SET UP STORAGE FOR INTERNAL USE
NP1=N+1
NELT=1
REMAIN=0.0
NHA=0
NTSSNT
SUMHAF=0
LOUP=0
DO 91 I=1,3
91 STEP(I)=0
IERR=1
DO 8 I=1, NP1
8 CVAR(I) = VAR(I)
DO 101 I=1, N
101 SEL1(I)=ELE1(I)
IF (NT .EQ. 0) GO TO 13
100 IF (NT .EQ. 1) GO TO 10
NTM1=NT-1

-60-
```
ELTK = K1 * ELT(1)
DO 9 I = 1, N1
ELTK2 = K1 * ELT(I + 1)
IF (ELTK .LT. ELTK2) GO TO 9
GO TO 500
9 ELTK = ELTK2
10 CONTINUE
ELTK = K1 * ELT(NELT)
IF (VARK .LT. ELTK) GO TO 11
IF (NELT .EQ. NT) GO TO 13
NELT = NELT + 1
GO TO 10
11 NELTL = NT - NELT + 1
GO TO 1012
13 NELTL = 0
12 DO 14 I = 1, N
14 RELMIN(I) = SELE1(I) / 128, 0
IF (NT .EQ. 0) GO TO 995
DO 995 I = 1, NT
995 SELT(I) = ELT(I)
996 CALL DERSUB (II, CUVAR, DER, N, VAR)
IF (II .EQ. 4) GO TO 120
DO 15 I = 1, N
15 FDERY(I) = DER(I + 1)
II = 1
TEST = 0
DO 300 I = 1, 15
300 TEX(I) = 0
TEX(1) = 1
TEX(2) = 1
K3 = 1
IF (IEXT) <= 35, 63, 635
151 WRITE (6, 1000)
1000 FORMAT ("//11 CI IS ZERO")
STOP
C END OF INITIALIZATION
520 II = 1
TPSH = 0
LTSH = 0
VARK = VAR(I) * K1
CIK = CI * K1
S1 = VARK + CIK
IF (SSPEC .EQ. 0, 0) GO TO 525
KK = 1
IF (NELTL .EQ. 0) GO TO 17
IF (ELTK = TPK) 16, 16, 17
16 CV = ELTK
CODE = 1
GO TO 18
17 CV = TPK
CODE = 2
18 IF (ABS(CV), LT, 1, E-12) GO TO 530
IF (CV = S1) 20, 20, 19
19 IF (ABS((CV = S1) / CV), GE, 1E-11) GO TO 535
20 IF (NELTL .EQ. 0) GO TO 540
IF (ABS((ELTK = TPK) / CV), LT, 1E-11) GO TO 550
IF (CODE .EQ. 1) GO TO 545
-61-
```
SHORT INTERVAL DUE TO BOTH SPEC

IF (VARK=0) GO TO 540
IF (REMANT<>VARK) GO TO 550

60 TO 560

SPEC IS ZERO

IF (ABS(REMAINT)>.GE.1.E-12) GO TO 560
IF (NELTL.EQ.0) GO TO 565
IF (ABS(ELT)-.LT.1.0E-12) GO TO 565
GO TO 545

21 S2=ELTK-S1
IF (S2) 545,545,22
22 IF (ABS(S2/ELTK).LT.1.E-12) GO TO 545
GO TO 565

SHORT INTERVAL IS DUE TO ELT BLOCK

GO TO 560

DELTA= SELT(NELT)

IF (VARK=0) GO TO 540
IF (REMANT<>VARK) GO TO 550

60 TO 560

BEGIN RUNGE-KUTTA

TEST=0
555 DO 24 I=1,N
24 $1VAR(1)= VAR(I+1) 
575 CUVAH(1)=VAR(1)
576 DO 25 I=1,N
SDY(I)=DER(I+1)
25 CUVAR(I+1)=S1VAR(1)+(DX*DER(I+1))/2.0
CUVAR(1)=CUVAR(1)+DX/2.0
CALL DERSUB (II,CUVAH,DER,N,VAR)
IF (II .EQ. 4) GO TO 120
580 DO 26 I=1,N
SDY(I)=SDY(1)+2.0*DER(I+1)
26 CUVAR(I+1)=S1VAR(1)+(DX*DER(I+1))/2.0
CALL DERSUB (II,CUVAH,DER,N,VAR)
IF (II .EQ. 4) GO TO 120
585 DO 27 I=1,N
SDY(I)=SDY(1)+2.0*DER(I+1)
27 CUVAR(I+1)=S1VAR(1)+DX*DER(I+1)
CUVAR(1)=CUVAR(1)+DX/2.0
CALL DERSUB (II,CUVAH,DER,N,VAR)
IF (II .EQ. 4) GO TO 120
590 DO 90 I=1,N
SDY(I)=(SDY(I)+DER(I+1))/6.0
90 CONTINUE
IF (LOOP) 28,28,29
28 DO30 I=1,N
SDY1(I)=SDY(1)
YINCR(I)=0.0
30 DER(I+1)=FDERV(I)
DX=DX/2.0
LOOP=1
GO TO 575
C
C LOOP WAS NOT ZERO
C
29 DO 31 I=1,N
31 YINCR(1)=YINCR(1)+SDY(1)
IF (LOOP .EQ. 2) GO TO 33
DO 32 I=1,N
S1VAR(1)=VAR(I+1)+DX*YINCR(1)
32 CUVAH(I+1)=S1VAR(1)
CUVAR(1)=VAR(I+1)+DX
LOOP=2
CALL DERSUB (II,CUVAH,DER,N,VAR)
IF (II .EQ. 4) GO TO 120
GO TO 576
33 LOOP=0
H=2.0*DX
DO 34 I=1,N
ERRVHM(I)=(YINCR(1)/2.0-SDY1(I))/15.0
ERRVAL(I)=H*ERVVHM(I)
34 S1VAR(I)=S1VAR(1)+DX*SDY(I)+ERRVAL(I)
C
C S1VAR HOLD THE APPROXIMATE ANSWERS
C
IF (SCIMAX) 36,35,36
36 IF (ABS(SCIMAX-C1),LT,1.0E-12) GO TO 38
35 IF(ABS(H-C1),GT,1.0E-12) GO TO 38
GO TO 575

1100 CONTINUE

IF (NHAFT<GT, 1) GO TO 999
NHT=NHT+1

IF (NHT<GT, 13) GO TO 998
ACV=VAR(I)+C1
ACVK=ACV*K1
IF (NCLT<EQ, 0) GO TO 1102
NLT=NLT

1101 ELTK1=SELT(NLT)*K1
IF (ACVK<LT, ELTK1) GO TO 1103
NLT=NLT+1

IF (NLT<EQ, NHT) GO TO 1106
GO TO 1101

1102 SELT(NLT)=ACV
GO TO 1105

1103 NLTPI=NLT+1
I=NHT

1108 SELT(I)=SELT(I-1)
IF (1<EQ, NLTPI) GO TO 1106
I=I+1
GO TO 1108

1106 SELT(NLT)=ACV

1105 NLTTL=NLTTL+1

TEX(9)=0
TEX(10)=1
ELTK1=K1*SELT(NLT)
GO TO 999

C

DOUBLE PRECISION UPDATING

620 LOOP=0

DH=H
DO 51 I=1,N
PHI=ERV0VH(I)+YINC1(I)/2.0
DPHI=PHI
51

CUVAR(I+1)=VAR(I+1)+DH*DPHI
CUVAR(I)=VAR(I)+DH
CALL DEHSUB(I,CUVAR,DER,N,VAR)
IF (11<EQ, 4) GO TO 120
CALL CHSUG(I+1,CI,VAR,CUVAR,DER)
IF (I=1) 54,600,121

121 TEST=0
54 DO 57 I=1,N
57 DE1R(I)=D1R(I+1)
SUMHAF=SUMHAF+NHAFT=STEP(1)
STEP(1)=STEP(2)
STEP(2)=STEP(3)
STEP(3)=NHAFT
NHAFT=0
IERI=1
IF (SUMHAF=8) 63,63,510
63 DO 59 I=1,NP1
59 VAR(I)=CUVAR(I)
TEX(12)=1

501 KK3=4
IF (11EXT<EQ, 1) GO TO 637
58 IF (TEST ,EQ. 1) GO TO 520
120 RETURN
C
C RECOMPUTE INTERVAL
C
600 TEST=0
   NHAF=0
   III=1
   DX=CI
   TEX(11)=1
   KK3=3
   IF (TEST ,EQ. 1) GO TO 636
70 CIK=CI*K1
   DO 60 I=1,N
   DER(I+1)=FDERV(I)
60   CVAR(I)= VAR(I)
   CVAR(N+1)= VAR(N+1)
   IF (TPS ,EQ. 0) GO TO 61
   TP=TPSPEC
   TPK=TP*K1
   TPSH=0
61 IF (LTSH ,EQ. 0) GO TO 555
   NELT=NELT-1
   REMAIN=0,0
   NELTL=NELTL+1
   ELTK=SEL1(NELT)*K1
   GO TO 555
636 WRITE(6,183) VAR(1),DX
   GO TO 102
635 IF (TEX(1),EQ. 1) WRITE(6,171) VAR(1)
   IF (TEX(2),EQ. 1) WRITE(6,172) CI,CIMAX,SPEC
   IF (TEX(3),EQ. 1) WRITE(6,173) H
   IF (TEX(4),EQ. 1) WRITE(6,174) H
   IF (TEX(5),EQ. 1) WRITE(6,175) H
   IF (TEX(6),EQ. 1) WRITE(6,176) H
   IF (TEX(7),EQ. 1) WRITE(6,178) VAR(1)
   IF (TEX(8),EQ. 1) WRITE(6,179) CI
   IF (TEX(9),EQ. 1) WRITE(6,180) NVAR, CI
   IF (TEX(10),EQ. 1) WRITE(6,185) NVAR, DX
   IF (TEX(11),EQ. 1) WRITE(6,183) VAR(1),DX
   IF (TEX(12),EQ. 1) WRITE(6,179) VAR(1)
   IF (TEX(13),EQ. 1) WRITE(6,180)
   IF (TEX(14),EQ. 1) WRITE(6,181)
   IF (TEX(15),EQ. 1) WRITE(6,182)
102 DO 320 I=3,13
320 TEX(I)=0
   GO TO (120,99,70,58),KK3
171 FORMAT (35H INITIALIZATION STARTS AT VAR(1)=E16.8/)
172 FORMAT (4H CI=E15.8,9H CIMAX=E15.8,8H SPEC=E15.8/)
173 FORMAT (37H DX IS THE FULL COMPUTING INTERVAL CI/)
174 FORMAT (28H DX IS A SHORTENED INTERVAL E15.8,25H DUE TO A CRITICAL VALUE/)
175 FORMAT (28H DX IS A SHORTENED INTERVAL E15.8,21H DUE TO A SPEC VALUE/)
176 FORMAT (28H DX IS A SHORTENED INTERVAL E15.8,39H DUE TO BOTH A SPEC AND CRITICAL VALUE/)
177 FORMAT (27H CI HAS BEEN LENGTHENED TO E16.8/)
178 FORMAT (5H VAR(1,2,32H) HAS CAUSED CI TO BE HALVED TO ,E16.8/)
179 FORMAT (27H VAR(1) HAS BEEN UPDATED TO ,E16.8/)
180 FORMAT (3IH ERROR RETURN=ELT NOT MONOTONIC/)
181 FORMAT (55H ERROR RETURN=HAVE HALVED 9 TIMES OVER LAST 3 INTERVALS 1/)
182 FORMAT (45H ERROR RETURN=HAVE HALVED 9 CONSECUTIVE TIMES/)
183 FORMAT (3IH INTERVAL RECOMPUTED AT VAR(1)=,E16.8,9M WITH DX=,E16.8 1/)
184 FORMAT (25H DX IS SHORTENED INTERVAL,E16.8,28H DUE TO A PREVIOUS EL 1T VALUE/)
185 FORMAT (5H VAR(,12,32H) HAS CAUSED DX TO BE HALVED TO ,E16.8,38H BU 1T NOT CI SINCE CI ALREADY SHORTENED/) 

500 IERR=2
TEX(13)=1
TEST=0
GO TO 63

505 IERR=3
TEX(15)=1
TEST=0
GO TO 501

510 IERR=4
TEST=0
TEX(14)=1
GO TO 63
END
SUBROUTINE DERSUB
COMMON/KUTTA/ CUVAH(2), VAH(2), CI, DEH(2), II
COMMON/F/RJET, X, E, PE, AE, PINF, AINF, UINF, HMINF
COMMON/BLAST/ IBW, MWCON, SPEED, RJET2
RDIST=CUVAH(2)/RJET2
DER(2)=AINF/SQRT(1, 0=BMCON/(SQRT(ALOG(RDIST))*RDIST))
RETURN
END
SUBROUTINE CMSUB
RETURN
END
REFERENCES


<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>Defense Documentation Center</td>
<td>US Army Electronics Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: DDC-TCA</td>
<td>ATTN: DRSEL-PD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
<td>Fort Monmouth, NJ 07703</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>Defense Nuclear Agency</td>
<td>US Army Missile Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: DDC-TCA</td>
<td>ATTN: DRSEL-PD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22314</td>
<td>ATTN: DRSMI-R</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRSMI-RDK</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>6</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Materiel Development and Readiness Command</td>
<td>US Army Nuclear Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: DRCDMA-ST</td>
<td>ATTN: DRSEL-PD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5001 Eisenhower Avenue</td>
<td>Fort Monmouth, NJ 07703</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRCDMA-ST</td>
<td>ATTN: DRSMI-R</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5001 Eisenhower Avenue</td>
<td>DRSMI-RDK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22333</td>
<td>Mr. R. Becht (4 cys)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mr. R. Deep</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>2</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Materiel Development and Readiness Command</td>
<td>US Army Mobility Equipment Research &amp; Development Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: DRCDMA-ST</td>
<td>ATTN: DRSMI-R</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5001 Eisenhower Avenue</td>
<td>DRSMI-RDK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22333</td>
<td>Mr. R. Becht (4 cys)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mr. R. Deep</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>4</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRCDMA-ST</td>
<td>ATTN: P. Ehle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5001 Eisenhower Avenue</td>
<td>E. Haug</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22333</td>
<td>Technical Lib</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRSAR-RDG, J. Blick</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Commander</td>
<td>2</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Aviation Systems Command</td>
<td>US Army Armament Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: DRSAV-E</td>
<td>ATTN: Rodman Laboratories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DRSAV-EQA, CPT Schrage</td>
<td>S. Thompson</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DRCQM-AAH, G. Smith</td>
<td>S. Burley</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12th and Spruce Streets</td>
<td>Rock Island, IL 61202</td>
<td></td>
</tr>
<tr>
<td></td>
<td>St. Louis, MO 63166</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Air Mobility Research and Development Laboratory</td>
<td>US Army Armament Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: DRSAV-E</td>
<td>ATTN: Rodman Laboratories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DRSAV-EQA, CPT Schrage</td>
<td>S. Thompson</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DRSQM-AAH, G. Smith</td>
<td>S. Burley</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12th and Spruce Streets</td>
<td>Rock Island, IL 61202</td>
<td></td>
</tr>
<tr>
<td></td>
<td>St. Louis, MO 63166</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Moffett Field, CA 94035</td>
<td>71</td>
<td></td>
</tr>
</tbody>
</table>
# DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Commander</td>
<td>4</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Frankford Arsenal</td>
<td></td>
<td>US Army Watervliet Arsenal</td>
</tr>
<tr>
<td></td>
<td>ATTN: Mr. T. Boldt</td>
<td></td>
<td>ATTN: Tech Lib</td>
</tr>
<tr>
<td></td>
<td>SARFA-U2100</td>
<td></td>
<td>SARWV-PDR-S</td>
</tr>
<tr>
<td></td>
<td>Mr. J. Mitchell</td>
<td></td>
<td>Dr. F. Sautter</td>
</tr>
<tr>
<td></td>
<td>SARFA-U3100, S. Fulton</td>
<td></td>
<td>SARWV-PDR-AMM</td>
</tr>
<tr>
<td></td>
<td>L4100-150-2</td>
<td></td>
<td>Dr. J. Zweig</td>
</tr>
<tr>
<td></td>
<td>Mr. C. Sleischer, Jr.</td>
<td></td>
<td>SARWV-RDD-SE, P. A. Alto</td>
</tr>
<tr>
<td></td>
<td>Philadelphia, PA 19137</td>
<td></td>
<td>Watervliet, NY 12183</td>
</tr>
<tr>
<td>4</td>
<td>Commander</td>
<td></td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Frankford Arsenal</td>
<td></td>
<td>US Army Jefferson Proving Ground</td>
</tr>
<tr>
<td></td>
<td>ATTN: SARFA-U3300</td>
<td></td>
<td>ATTN: STEJP-TD-D</td>
</tr>
<tr>
<td></td>
<td>Mr. S. Hirshman</td>
<td></td>
<td>Madison, IN 47250</td>
</tr>
<tr>
<td></td>
<td>Mr. A. Cianciosi</td>
<td></td>
<td>2 Commander</td>
</tr>
<tr>
<td></td>
<td>SARFA-MDS-D 220</td>
<td></td>
<td>US Army Harry Diamond</td>
</tr>
<tr>
<td></td>
<td>F. Puzycki</td>
<td></td>
<td>Laboratories</td>
</tr>
<tr>
<td></td>
<td>C. Rueter</td>
<td></td>
<td>ATTN: DRXDO-TI</td>
</tr>
<tr>
<td></td>
<td>Philadelphia, PA 19137</td>
<td></td>
<td>DRXDO-DAB, H. J. Davis</td>
</tr>
<tr>
<td>5</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Picatinny Arsenal</td>
<td></td>
<td>US Army Materials and</td>
</tr>
<tr>
<td></td>
<td>ATTN: SARPA-D, S. Wasserman</td>
<td></td>
<td>Mechanics Research Center</td>
</tr>
<tr>
<td></td>
<td>SARPA-DR-D, Mr. A. Loeb</td>
<td></td>
<td>ATTN: DRXMR-ATL</td>
</tr>
<tr>
<td></td>
<td>Mr. D. Mertz</td>
<td></td>
<td>Watertown, MA 02172</td>
</tr>
<tr>
<td></td>
<td>Mr. F. Friedman</td>
<td></td>
<td>1 Commander</td>
</tr>
<tr>
<td></td>
<td>SARPA-D, Mr. Lindner</td>
<td></td>
<td>US Army Natick Research</td>
</tr>
<tr>
<td></td>
<td>Dover, NJ 07801</td>
<td></td>
<td>and Development Command</td>
</tr>
<tr>
<td>5</td>
<td>Commander</td>
<td>1</td>
<td>Director</td>
</tr>
<tr>
<td></td>
<td>US Army Picatinny Arsenal</td>
<td></td>
<td>US Army TRADOC Systems</td>
</tr>
<tr>
<td></td>
<td>ATTN: SARPA-V, E. Walbrecht</td>
<td></td>
<td>Analysis Activity</td>
</tr>
<tr>
<td></td>
<td>Mr. S. Verner</td>
<td></td>
<td>ATTN: ATAA-SA</td>
</tr>
<tr>
<td></td>
<td>SARPA-VE, Dr. Kaufman</td>
<td></td>
<td>White Sands Missile Range</td>
</tr>
<tr>
<td></td>
<td>SARPA-FF-M-MA</td>
<td></td>
<td>NM 88002</td>
</tr>
<tr>
<td></td>
<td>Mr. E. Barrieroes</td>
<td></td>
<td>1 Director</td>
</tr>
<tr>
<td></td>
<td>SARPA-PA-S</td>
<td></td>
<td>US Army TRADOC Systems</td>
</tr>
<tr>
<td></td>
<td>Mr. W. Dingala</td>
<td></td>
<td>Analysis Activity</td>
</tr>
<tr>
<td></td>
<td>Dover, NJ 07801</td>
<td></td>
<td>ATTN: ATAA-SA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>White Sands Missile Range</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NM 88002</td>
</tr>
</tbody>
</table>

72
DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Commander</td>
<td>US Army Research Office</td>
</tr>
<tr>
<td></td>
<td>ATTN: CRD-AA-EH</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 12211</td>
</tr>
<tr>
<td></td>
<td>Research Triangle Park</td>
</tr>
<tr>
<td></td>
<td>NC 27709</td>
</tr>
<tr>
<td>1 Director</td>
<td>US Army BMD Advanced Technology Center</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 1500, West Station</td>
</tr>
<tr>
<td></td>
<td>Huntsville, AL 35807</td>
</tr>
<tr>
<td>1 Commander</td>
<td>US Army Ballistic Missile Defense Systems Command</td>
</tr>
<tr>
<td></td>
<td>Huntsville, AL 35804</td>
</tr>
<tr>
<td>3 Commander</td>
<td>US Naval Air Systems Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: AIR-604</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20360</td>
</tr>
<tr>
<td>3 Commander</td>
<td>US Naval Ordnance Systems Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: ORD-9132</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20360</td>
</tr>
<tr>
<td>2 Commander</td>
<td>David W. Taylor Naval Ship Research &amp; Development Center 2</td>
</tr>
<tr>
<td></td>
<td>ATTN: Tech Lib Aerodynamic Lab</td>
</tr>
<tr>
<td></td>
<td>Bethesda, MD 20084</td>
</tr>
<tr>
<td>3 Commander</td>
<td>US Naval Surface Weapons Center 2</td>
</tr>
<tr>
<td></td>
<td>ATTN: Code GX, Dr. W. Kemper Mr. F. H. Maille Dr. G. Moore</td>
</tr>
<tr>
<td></td>
<td>Dahlgren, VA 22448</td>
</tr>
<tr>
<td>4 Commander</td>
<td>US Naval Surface Weapons Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: Code 031, Dr. K. Lobb Code 312, Mr. R. Regan Mr. S. Hastings</td>
</tr>
<tr>
<td></td>
<td>Code 730, Tech Lib</td>
</tr>
<tr>
<td></td>
<td>Silver Spring, MD 20910</td>
</tr>
<tr>
<td>3 Commander</td>
<td>US Naval Weapons Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: Code 553, Tech Lib</td>
</tr>
<tr>
<td></td>
<td>Code 6003 Dr. W. Hazeltine</td>
</tr>
<tr>
<td></td>
<td>Code 511, Mr. A. Rice</td>
</tr>
<tr>
<td></td>
<td>China Lake, CA 93555</td>
</tr>
<tr>
<td>3 Director</td>
<td>US Naval Research Laboratory</td>
</tr>
<tr>
<td></td>
<td>ATTN: Tech Info Div</td>
</tr>
<tr>
<td></td>
<td>Code 7700, D. A. Kolb</td>
</tr>
<tr>
<td></td>
<td>Code 7720, Dr. E. McClean</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20390</td>
</tr>
<tr>
<td>1 Commander</td>
<td>US Naval Ordnance Station</td>
</tr>
<tr>
<td></td>
<td>ATTN: Code FS13A, P. Sewell</td>
</tr>
<tr>
<td></td>
<td>Indian Head, MD 20640</td>
</tr>
<tr>
<td>1 AFRPL/LKCG (Dr. Horning)</td>
<td>Edwards, CA 93523</td>
</tr>
<tr>
<td></td>
<td>ADTC (ADBPS-12)</td>
</tr>
<tr>
<td></td>
<td>Eglin AFB, FL 32542</td>
</tr>
<tr>
<td>3 AFATL (DLDL)</td>
<td>Eglin AFB, FL 32542</td>
</tr>
<tr>
<td></td>
<td>AFATL (DLRA, F. Burgess; Tech Lib)</td>
</tr>
<tr>
<td></td>
<td>Eglin AFB, FL 32542</td>
</tr>
<tr>
<td>1 AFWL (DEV)</td>
<td>Kirtland AFB, NM 87117</td>
</tr>
<tr>
<td>1 ASD (ASBEE)</td>
<td>Wright-Patterson AFB, OH 45433</td>
</tr>
</tbody>
</table>

73
### DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Director</td>
<td></td>
<td>Advanced Technology Laboratories</td>
</tr>
<tr>
<td></td>
<td>National Aeronautics and Space Administration</td>
<td>1</td>
<td>ATTN: Dr. J. Erdos</td>
</tr>
<tr>
<td></td>
<td>George C. Marshall Space Flight Center</td>
<td></td>
<td>Merrick &amp; Stewart Avenues</td>
</tr>
<tr>
<td></td>
<td>ATTN: MS-1, Lib</td>
<td></td>
<td>Westbury, NY 11590</td>
</tr>
<tr>
<td></td>
<td>R-AERO-AE, Mr. A. Felix</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Huntsville, AL 35812</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>2</td>
<td>ARO, Inc.</td>
</tr>
<tr>
<td></td>
<td>Jet Propulsion Laboratory</td>
<td></td>
<td>ATTN: Tech Lib</td>
</tr>
<tr>
<td></td>
<td>ATTN: Tech Lib</td>
<td></td>
<td>Arnold AFS, TN 37389</td>
</tr>
<tr>
<td></td>
<td>2800 Oak Grove Drive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pasadena, CA 91103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td>ARTEC Associates, Inc.</td>
</tr>
<tr>
<td></td>
<td>National Aeronautics and Space Administration</td>
<td></td>
<td>ATTN: Dr. S. Gill</td>
</tr>
<tr>
<td></td>
<td>Langley Research Center</td>
<td></td>
<td>26046 Eden Landing Road</td>
</tr>
<tr>
<td></td>
<td>ATTN: MS 185, Tech Lib</td>
<td></td>
<td>Hayward, CA 94545</td>
</tr>
<tr>
<td></td>
<td>Langley Station</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hampton, VA 23365</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td>AVCO Systems Division</td>
</tr>
<tr>
<td></td>
<td>National Aeronautics and Space Administration</td>
<td></td>
<td>ATTN: Dr. W. Reinecke</td>
</tr>
<tr>
<td></td>
<td>Lewis Research Center</td>
<td></td>
<td>201 Lowell Street</td>
</tr>
<tr>
<td></td>
<td>ATTN: MS 60-3, Tech Lib</td>
<td></td>
<td>Wilmington, MA 01887</td>
</tr>
<tr>
<td></td>
<td>2100 Brookpark Road</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cleveland, OH 44135</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td>Calspan Corporation</td>
</tr>
<tr>
<td></td>
<td>NASA Scientific and Technical Information Facility</td>
<td></td>
<td>ATTN: Mr. G. A. Sterbutzel</td>
</tr>
<tr>
<td></td>
<td>ATTN: SAK/DL</td>
<td></td>
<td>P.O. Box 235</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 8757</td>
<td></td>
<td>Buffalo, NY 14221</td>
</tr>
<tr>
<td></td>
<td>Baltimore/Washington International Airport, MD 21240</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td>Technical Director</td>
</tr>
<tr>
<td></td>
<td>Colt Firearms Corporation</td>
<td></td>
<td>Colt Firearms Corporation</td>
</tr>
<tr>
<td></td>
<td>ATTN: Dr. T. Stastny</td>
<td></td>
<td>150 Hyshore Avenue</td>
</tr>
<tr>
<td></td>
<td>Cockeyville, MD 21030</td>
<td></td>
<td>Hartford, CT 14061</td>
</tr>
<tr>
<td>1</td>
<td>AAI Corporation</td>
<td>1</td>
<td>General Electric Corporation</td>
</tr>
<tr>
<td></td>
<td>ATTN: Dr. T. Stastny</td>
<td></td>
<td>Armaments Division</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 8757</td>
<td></td>
<td>ATTN: Mr. R. Whyte</td>
</tr>
<tr>
<td></td>
<td>Baltimore/Washington International Airport, MD 21240</td>
<td></td>
<td>Lakeside Avenue</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>Burlington, VT 05401</td>
</tr>
<tr>
<td></td>
<td>Martin Marietta Aerospace</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN: Mr. A. J. Culotta</td>
<td></td>
<td>P.O. Box 5387</td>
</tr>
<tr>
<td></td>
<td>Orlando, FL 32805</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Northrop Corporation Aircraft Division</td>
<td>1</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td></td>
<td>ATTN: Dr. A. Wortman 3901 W. Broadway Hawthorne, CA 90250</td>
<td></td>
<td>Department of Aeronautics and Astronautics</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN: Tech Lib 77 Massachusetts Avenue Cambridge, MA 02139</td>
</tr>
<tr>
<td>1</td>
<td>Winchester-Western Division Olin Corporation New Haven, CT 06504</td>
<td>1</td>
<td>Ohio State University Department of Aeronautics and Astronautical Engineering</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN: Tech Lib Columbus, OH 43210</td>
</tr>
<tr>
<td>1</td>
<td>Sandia Laboratories ATTN: Aerodynamics Dept Org 9320, R. Maydew P.O. Box 5800 Albuquerque, NM 87115</td>
<td>2</td>
<td>Polytechnic Institute of Brooklyn Graduate Center</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN: Tech Lib Dr. G. Moretti Farmingdale, NY 11735</td>
</tr>
<tr>
<td>1</td>
<td>S&amp;D Dynamics Inc. ATTN: Dr. M. Soifer 755 New York Avenue Huntington, NY 11743</td>
<td></td>
<td>Princeton University Forrestal Laboratories</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN: Dr. M. Summerfield Princeton, NJ 08540</td>
</tr>
<tr>
<td>1</td>
<td>Guggenheim Aeronautical Laboratory California Institute of Technology ATTN: Tech Lib Pasadena, CA 91104</td>
<td></td>
<td>Southwest Research Institute</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN: Mr. Peter S. Westine P.O. Drawer 28510 8500 Culebra Road San Antonio, TX 78228</td>
</tr>
<tr>
<td>2</td>
<td>Franklin Institute ATTN: Dr. Carfagno Dr. Wachtell Race &amp; 20th Streets Philadelphia, PA 19103</td>
<td></td>
<td>Aberdeen Proving Ground</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Marine Corps Ln Ofc Dir, USAMSAA Cdr, USAEA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN: A. Flatau, SAREA-DE-W Bldg E3516</td>
</tr>
<tr>
<td>1</td>
<td>Director Applied Physics Laboratory The Johns Hopkins University Johns Hopkins Road Laurel, MD 20810</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

75